Collection Ressources Informatiques

LINUX

Maîtrisez l'administration du système

Seconde Edition

Sébastien ROHAUT

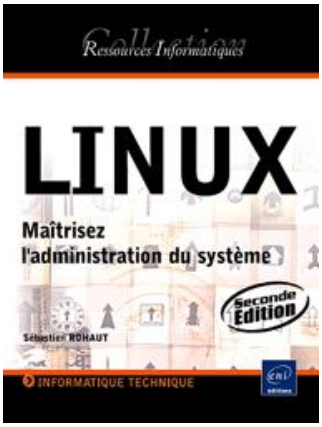
INFORMATIQUE TECHNIQUE

eni éditions

LINUX

Maîtrisez l'administration du système [2ième édition]

Sébastien ROHAUT



Résumé

Ce livre sur **l'administration du système Linux** s'adresse à tout informaticien appelé à gérer ce système d'exploitation et désireux d'apprendre ou de consolider des bases acquises sur le terrain.
Quelle que soit la distribution Linux utilisée (que ce soit en entreprise ou à la maison), toutes les méthodes et commandes d'administration de Linux sont abordées et détaillées.
Le livre fait le tour des connaissances nécessaires à **l'installation** d'une distribution, la gestion des **paquetages** logiciels **RPM** et **APT**, la **compilation** depuis les sources, les **bibliothèques** partagées, les principales commandes **Gnu** et les **scripts shell**, la gestion des **disques** et **systèmes de fichiers**, la mise en place de **volumes RAID** et **LVM**, le **démarrage** et **l'arrêt** du système, **l'impression** et les tâches d'administration communes dont la gestion des **utilisateurs** et **l'automatisation des tâches**, la configuration du **réseau** et des services associés, le **noyau** et sa **compilation**, les bases de la **sécurité**, la configuration de **l'environnement graphique X11**.
Tous les points traités sont agrémentés d'exemples et leur maîtrise fera de vous un administrateur système Linux compétent.

Présentation

Ce livre sur l'administration du système Linux s'adresse à tout informaticien appelé à gérer ce système d'exploitation et désireux d'apprendre ou de consolider des bases acquises sur le terrain.
Quelle que soit la distribution Linux utilisée (que ce soit en entreprise ou à la maison), toutes les méthodes et commandes d'administration de Linux sont abordées et détaillées.

Le livre fait le tour des connaissances nécessaires à l'installation d'une distribution, la gestion des paquetages logiciels RPM et APT, la compilation depuis les sources, les bibliothèques partagées, les principales commandes Gnu et les scripts shell, la gestion des disques et systèmes de fichiers, la mise en place de volumes RAID et LVM, le démarrage et l'arrêt du système, l'impression et les tâches d'administration communes dont la gestion des utilisateurs et l'automatisation des tâches, la configuration du réseau et des services associés, le noyau et sa compilation, les bases de la sécurité, la configuration de l'environnement graphique X11.

Chapitre 1 : Présentation de Linux

Chapitre 2 : Installation de Linux et des logiciels

Chapitre 3 : Le shell et les commandes GNU

Chapitre 4 : Les disques et le système de fichiers

Chapitre 5 : Démarrage de Linux, services, noyau et périphériques

Chapitre 6 : Les tâches administratives

Chapitre 7 : Le réseau

Chapitre 8 : La sécurité

Chapitre 9 : X Window

Chapitre 10 : Partitionnement avancé : RAID et LVM

Introduction

Installer Linux est très simple. Les tâches d'administration communes le deviennent aussi. La complexité du système est masquée par de nombreux outils, graphiques notamment, qui tendent à simplifier le travail des utilisateurs et des administrateurs. Cette simplicité apparente cache pourtant une réalité différente.

Chaque distribution est livrée avec une interface qui lui est propre. Les centres de contrôle de Redhat, Mandriva, openSUSE, Ubuntu, etc. sont tous différents. Il ne s'agit pas de se spécialiser dans l'une ou l'autre des interfaces. Ce serait une erreur : toutes ces interfaces s'appuient sur les mêmes outils : ce sont des front-ends. Ils modifient les mêmes fichiers de configuration. Ces commandes et fichiers de configuration sont communs à l'ensemble des distributions. Plutôt que d'utiliser une interface qui risque d'être désuète à la prochaine version, apprenez directement à maîtriser les arcanes de votre système. Ainsi vous ne serez pas bloqué par votre dépendance à un outil spécifique.

Unix est avant tout un système d'exploitation qui est installé sur des serveurs, mais aussi grâce à Linux, à la maison. Que vous soyez simple utilisateur ou ingénieur système, vous comprendrez bien mieux les réactions de votre système d'exploitation si vous savez comment il fonctionne et comment l'administrer. Ce livre se propose de vous apprendre à administrer une machine fonctionnant sous Linux. Tous les points essentiels sont abordés, de l'installation de base à l'administration avancée, tant système que réseau. C'est à la fois un guide pratique mais aussi un ouvrage de référence auquel vous pourrez vous rapporter en cas de besoin. Il n'est pas spécifique : son contenu est valable pour toutes les distributions, les différences entre les deux grands courants (Redhat et Debian) étant abordées.

À l'issue de la lecture de ce livre et de la mise en pratique des connaissances que vous aurez acquises, vous serez apte à gérer l'administration système d'un poste de travail ou d'un serveur sous Linux. Vous serez, réellement, un administrateur système Linux.

Bonne lecture !

Bienvenue dans le monde Unix

1. Un nouveau monde

Linux n'est plus un simple effet de mode et d'annonce. Depuis ses tous premiers développements en 1991 et jusqu'à aujourd'hui Linux ne cesse d'évoluer, de changer. Le monde de l'informatique est vivant. S'il n'évolue pas, il végète. Avec Linux, des millions de personnes ont trouvé enfin ce qu'elles cherchaient.

Linux n'est pas plus compliqué à utiliser que n'importe quel autre système. Le frein au développement de Linux auprès du plus grand nombre n'est pas lié à un quelconque niveau de difficulté. L'expérience acquise auprès de nombreux utilisateurs débutants ou confirmés, des groupes d'utilisateurs Linux et des acteurs professionnels montre qu'il s'agit surtout d'un problème lié aux habitudes des gens, accoutumés des années durant à un système d'exploitation unique. En effet, ces habitudes doivent parfois être quelque peu modifiées pour s'adapter à un environnement Linux, tout comme conduire une voiture familiale ne fait pas de vous un as de la conduite sportive en Ferrari.

2. Histoire des ordinateurs

a. Complexité des ordinateurs

Un ordinateur est une machine électronique extrêmement complexe. Si le principe même de l'ordinateur tel que nous le connaissons n'a pas changé depuis l'époque de Alan Turing ou de Conrad Suze et date du début des années 1940 et même d'avant (machine de Charles Babbage), les évolutions technologiques et la miniaturisation ont permis de créer des machines de plus en plus puissantes tout en étant de plus en plus petites. Des premiers ordinateurs électromécaniques composés de milliers de lampes à vide et dont la programmation se faisait en branchant des câbles, à l'ordinateur moderne d'aujourd'hui, la complexité du matériel a été croissante. Entre l'époque où un ordinateur occupait tout un étage et où le circuit d'eau permettant de le refroidir (certains étaient même refroidis avec des pains de glace) chauffait tout un immeuble, et aujourd'hui, où il suffit d'aller faire les courses au supermarché du quartier pour acheter un ordinateur, on pourrait logiquement croire que la simplicité d'utilisation a suivi la même courbe de croissance, l'ordinateur devenant aussi simple à utiliser que votre lecteur DVD de salon.

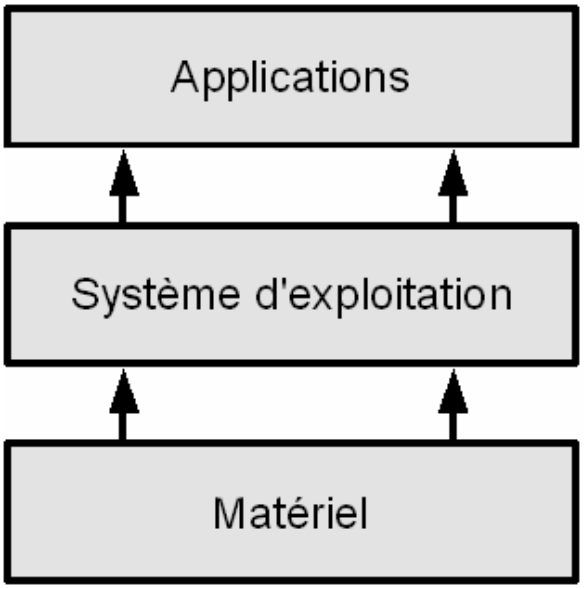
Ce n'est malheureusement pas le cas. Alors qu'un ordinateur est tellement polyvalent doit-on lui demander d'être aussi simple à utiliser qu'une machine basique ? Doit-on forcément connaître les méandres techniques de son ordinateur pour pouvoir l'utiliser ? La réponse est généralement non. Cependant il y a quelques notions et bases élémentaires à retenir et à respecter.

b. L'intelligence

Un ordinateur ne sait rien faire tout seul. Il n'y a rien de plus stupide qu'un ordinateur, il faut toujours lui dire ce qu'il doit faire. L'ordinateur est moins intelligent que le moindre insecte. Ce qui le rend « intelligent » au sens puissance de calcul c'est vous et les programmes que vous lui faites exécuter. Bien qu'inventé par des humains et pour des humains, l'ordinateur ne comprend pas votre langage. Il parle le binaire, assemblage de zéro (0) et de un (1). Ces valeurs assemblées les unes aux autres forment des mots et des données pour l'ordinateur. Le binaire devient un langage appelé le langage machine. Les microprocesseurs utilisent un langage appelé l'assembleur où chaque instruction dispose d'un équivalent en binaire. À l'aide de ce langage assembleur, les informaticiens vont créer divers produits dont un appelé le compilateur, un traducteur de langage dit de haut niveau et compréhensible cette fois par un plus grand nombre d'informaticiens.

3. Le système d'exploitation

Entre le moment où vous appuyez sur le bouton d'allumage de votre ordinateur et celui où vous pouvez enfin travailler et utiliser vos logiciels il se passe un certain temps durant lequel des programmes sont chargés dans la mémoire de votre ordinateur. Le but de ces programmes est de vous simplifier la vie en rendant les choses plus simples et pas seulement pour l'utilisateur mais aussi pour l'informaticien. Ces programmes forment un ensemble appelé le système d'exploitation. Comme son nom l'indique, le rôle du système d'exploitation est d'exploiter l'ordinateur le plus souvent à votre place, ou plutôt le système d'exploitation vous fournit toute la base nécessaire pour exploiter du mieux possible les ressources de votre ordinateur.



Principe du système d'exploitation

Un système d'exploitation est un programme ou un ensemble de programmes assurant la gestion de l'ordinateur et des périphériques. Il sert d'interface entre le matériel (hardware) et le logiciel (software). C'est un ensemble de programmes très complexes dont le but est de rendre plus simples les programmes et l'utilisation de l'ordinateur.

Le système d'exploitation propose aux programmeurs une interface de programmation d'applications appelée **API**, *Application Programming Interface*. Tous les programmeurs utilisent les mêmes fonctions dans leurs programmes ce qui simplifie fortement le travail. Ils peuvent se concentrer sur le but de leur programme (créer un traitement de texte par exemple) sans avoir sans arrêt à écrire des morceaux de programmes pour gérer le disque dur, l'imprimante ou comment accéder au clavier. C'est le rôle du système d'exploitation de gérer :

- la mémoire,
- les accès aux périphériques,
- les données sur les disques,
- les programmes,
- la sécurité,
- la collecte des informations.

Il manque l'interface graphique. Dans un produit comme Microsoft Windows l'interface graphique est incluse au sein même du système d'exploitation. Il est d'ailleurs impossible de travailler sans, le moindre réglage se fait depuis une boîte de dialogue. Les utilisateurs ont de ce fait tendance à intégrer l'interface graphique comme composant de tout système d'exploitation. Historiquement l'interface graphique ne fait pas partie du système d'exploitation. Elle vient en complément. Vous ne trouverez aucun livre sur la théorie des systèmes d'exploitation traitant des interfaces graphiques. Quel est l'intérêt, sauf à ajouter de la lourdeur et occuper de précieuses ressources de la machine, d'avoir une interface graphique pour faire fonctionner un serveur Internet ? Linux propose des interfaces, mais ce sont des programmes comme les autres.

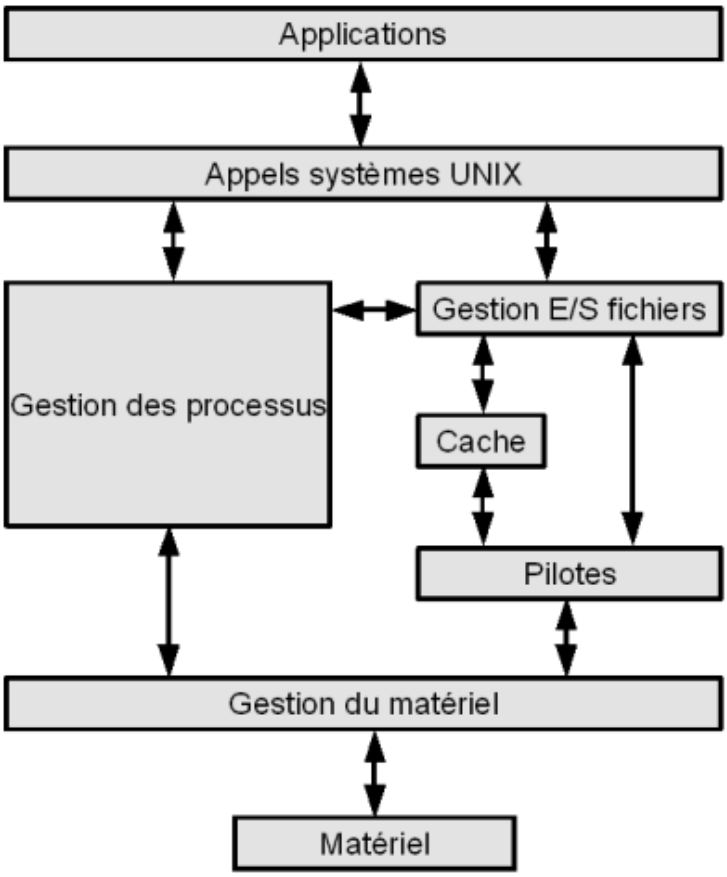
L'interface graphique n'est pas un composant du système d'exploitation Linux qui n'en a pas besoin pour fonctionner correctement. C'est un ensemble de plusieurs programmes classiques exécutés au-dessus du système d'exploitation, qu'il utilise.

Linux est un système d'exploitation de type Unix. Il existe des dizaines de systèmes d'exploitation dans cette famille. Unix est un système d'exploitation de la famille des systèmes **multitâches** et **multi-utilisateurs** :

- **Multitâche** : le système gère l'exécution simultanée de plusieurs programmes appelés des processus (Note :

un vrai multitâche nécessite d'avoir plusieurs microprocesseurs ou équivalents - Hyper Threading par exemple).

- **Multi-utilisateurs** : le système permet l'existence de plusieurs utilisateurs différents sur une même machine, connectés ou non (un utilisateur peut faire tourner un programme sans être connecté, comme par exemple un serveur Internet).



Architecture logique d'un système UNIX

Le schéma précédent est une synthèse simplifiée de la structure interne d'un système d'exploitation Unix. En bas se trouve votre matériel, en haut les programmes que vous faites fonctionner sur votre machine. Entre les deux les divers composants du système assurent son bon fonctionnement :

- Les **appels systèmes** sont utilisés par les programmes pour communiquer avec le système d'exploitation Unix.
- La **gestion des processus** s'occupe de la commutation des tâches et de leur priorité. Ce composant s'occupe donc du multitâche.
- La **gestion des entrées et des sorties fichiers** s'occupe aussi bien de la lecture et de l'écriture des données sur vos disques durs mais aussi sur vos périphériques (carte son, imprimante, etc.)
- Certaines informations peuvent être placées dans une zone mémoire tampon appelée **cache**. Plutôt que d'écrire des données directement sur le disque dur (ce qui est lent), Unix va les écrire dans une zone mémoire puis ensuite les écrire sur le disque après quelques secondes. Ainsi, la relecture de ces données est plus rapide car elles sont déjà en mémoire et le logiciel ne perd pas de temps à attendre la fin de l'écriture des données.
- Les **pilotes** ont pour rôle de gérer au plus bas niveau le matériel ou les structures logiques du matériel (par exemple les données d'une partition).

Une application bien programmée sur un système d'exploitation bien programmé ne peut pas court-circuiter ce schéma : elle ne « discute » jamais avec le matériel et passe obligatoirement par les API fournies.

---

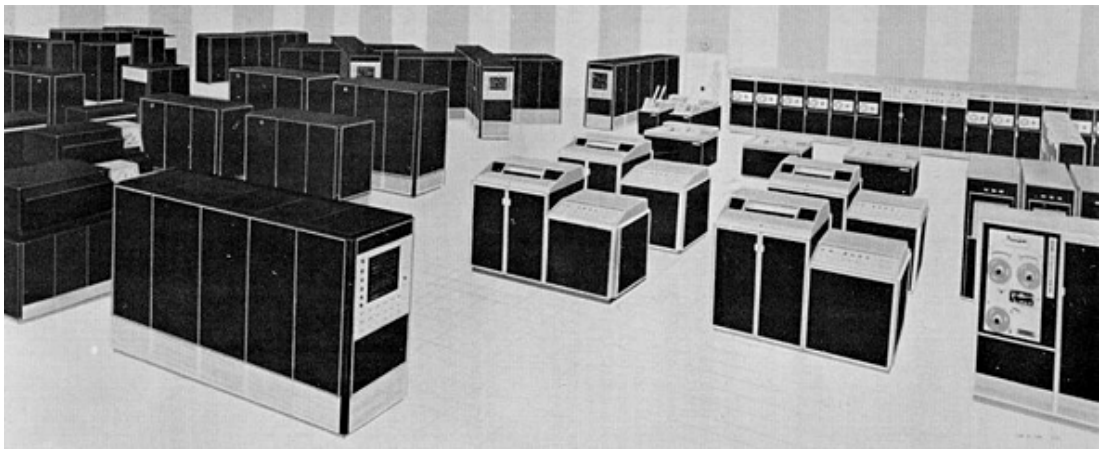
4. Le système Unix, une brève histoire

a. De MULTICS à UNIX

L'histoire d'Unix débute en **1964** quand le MIT, le laboratoire Bell Labs de AT&T et la General Electric commencent à développer le projet expérimental **MULTICS** (*Multiplexed Information and Computing Service*). Le projet Multics répond à de nouveaux besoins :

- pouvoir être utilisé par plusieurs personnes à la fois,
- pouvoir lancer des traitements en tâche de fond,
- une gestion accrue de la sécurité.

Multics était développé sur un gros système GE-645 de General Electric, équipé de deux processeurs sachant traiter chacun 435 000 instructions par seconde, trois unités de mémoire de 1 Mo chacune et 136 Mo de stockage. Il a fonctionné au MIT jusqu'en 1988, 82 sites en ont disposé et un maximum de 200 utilisateurs a pu y travailler simultanément chez General Motors. La dernière installation Multics à avoir été désactivée est celle de la Défense Canadienne le 30 octobre 2000.



Le GE-645

Dès le début pourtant, si Multics a vite atteint le degré de stabilité suffisant pour passer en production, il s'est révélé avoir des performances moindres que celles attendues. En **1969** Bell Labs se retire du projet pour se tourner vers le développement d'un autre système appelé GECOS.

Ken Thompson, développeur chez Bell, continue cependant à travailler sur le GE-645 et écrit un jeu appelé Space Travel. Tournant avec Multics il se révèle très lent et coûteux à faire tourner en temps partagé (avant le multitâche, le temps de la machine était découpé en tranches et chaque tranche d'utilisation était décomptée et facturée).

Ken réécrit alors le jeu en assembleur pour le mini ordinateur DEC PDP-7. Il est aidé de **Dennis Ritchie** qui lui aussi vient de Bell Labs. Cette expérience combinée avec celle de la conception de Multics pousse les deux hommes et leur équipe à créer un nouveau système d'exploitation pour le PDP-7. **Rudd Canaday**, encore de Bell Labs, était justement en train de développer un nouveau système de fichier qu'il voit comme un système d'exploitation. De là vient le fait que Unix est un système orienté fichier, où tout (ou presque) est fichier. Ils y rajoutèrent un interpréteur de commandes et quelques utilitaires. Ils nommèrent le système **UNICS** (*Uniplexed Information and Computing System*), selon une idée de **Brian Kernighan**. Le projet pouvait déjà gérer, dès le début, deux utilisateurs en même temps en vrai multitâche.

Le DEC PDP-7

L'origine du mot UNICS est le sujet de nombreuses légendes, qui ont probablement toutes leur part de vérité. UNICS est une dérision de MULTICS dont l'architecture était sujette à de nombreuses critiques à l'époque : « MULTICS (multiple) faisait la même chose de plusieurs façons alors que UNICS (unique) faisait chaque

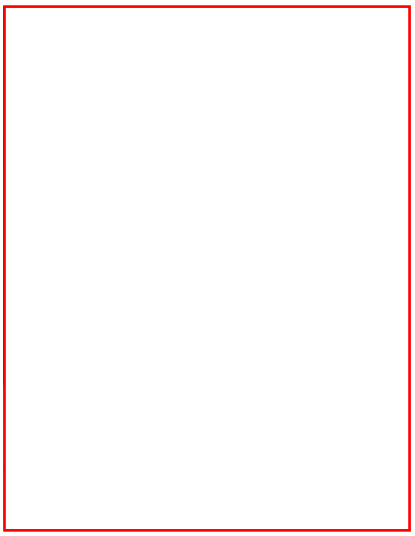
chose de la même façon ». L'autre fait est qu'en anglais UNICS se prononce comme « eunuchs » c'est-à-dire eunuque, un système Multics « castré ».

UNICS reprend les concepts essentiels développés pour MULTICS en les améliorant. Notamment ses concepteurs proposent un tout nouveau système de communication entre les programmes où un premier programme peut renvoyer ses données à un autre programme. Rapidement le *CS* est remplacé par un *X*, une lettre de moins pour la même présentation. La légende **UNIX** venait de naître.

Tout aurait pu s'arrêter là car les équipes travaillaient sans aucun financement, Bell Labs s'étant en principe totalement désengagé de Multics et de ses successeurs. Pour continuer les travaux, Thompson et Ritchie proposent à Bell Labs l'ajout d'un traitement de texte à Unix pour le PDP-11/20. Bell acceptant, la machine est mise à disposition et l'équipe obtient un financement et un support officiels. L'outil *runoff* (qui deviendra roff puis troff) et l'éditeur *ed* sont développés et pour la première fois en **1970** la dénomination **Unix Operating System** est utilisée. Bell utilise alors Unix comme un système de traitement de texte pour la rédaction de ses brevets. Le premier manuel de programmation Unix date du 3 novembre 1971.

b. Le langage C

Un nouveau problème apparaît rapidement. Développé en assembleur et donc en langage machine, Unix doit être en partie réécrit pour chaque nouveau modèle d'ordinateur DEC. Or le langage machine est un art difficile. La question de la portabilité se pose alors. Dès 1970, Thompson se penche sur le problème. Il pense tout d'abord à développer Unix en langage TMG puis Fortran. Trouvant le langage incomplet il s'associe avec Dennis Ritchie pour créer le **langage B**, issu du langage **BCPL**. Là encore ça ne convient pas (problème avec le typage des variables et les nombres réels). Ritchie part du langage B et développe le **New B** qu'il appelle logiquement le **langage C**. Le langage C est transformé en langage machine une fois passé par une étape de compilation. L'écriture des programmes est plus rapide.



Dennis Ritchie

Unix est réécrit en langage C à partir de **1973**. Pour passer Unix d'une machine à une autre il suffit qu'un compilateur C soit disponible sur la nouvelle machine. Il est beaucoup plus simple et rapide d'écrire un compilateur C (lui-même écrit en grande partie en C) que de réécrire tout un système d'exploitation en assembleur. Seules les parties très proches de l'architecture matérielle de la machine sont écrites en langage machine. Unix devient portable et son développement s'accélère.

c. Les licences et l'avènement de BSD et System V

Un premier événement majeur va alors contribuer à la large diffusion d'Unix (le mot large prend une signification particulière lorsqu'on parle de quelques dizaines de copies). AT&T, dont dépend Bell Labs, a fait l'objet en 1956 d'un décret antitrust lui interdisant de commercialiser d'autres produits que ceux situés au cœur de son métier : les télécommunications. Il ne peut pas vendre Unix. AT&T (qui n'en voit même pas l'avenir commercial) décide en **1974** de diffuser le système UNIX complet à des fins éducatives auprès des universités et des entreprises sous une licence peu restrictive. Seul le code source (*le programme sous forme de texte compréhensible et pas encore compilé*) du noyau en assembleur n'est pas officiellement diffusé ou via des moyens détournés. Les versions les plus diffusées sont la sixième en **1975** et la septième en **1978**. Unix v7 est la première version à avoir été spécifiquement retravaillée afin d'être portée sur d'autres machines que les PDP, notamment sur le VAX 11/780. La v7 est considérée comme la dernière version entièrement commune à tous les Unix suivants.

Le second événement majeur se produit à ce moment. Alors que Unix va fêter ses dix ans et que les universités

américaines contribuent fortement à sa diffusion et son amélioration, AT&T rend la licence d'Unix plus restrictive. La branche commerciale d'Unix est en effet autorisée à vendre des licences du code source. Les tarifs prohibitifs forcent les universités à continuer pour le meilleur et pour le pire leurs développements à partir des développements antérieurs à cette nouvelle licence. L'une de ces universités est celle de Californie, appelée **Berkeley**. Berkeley est le plus gros contributeur à Unix sur lequel elle a commencé à travailler dès 1974. La version 1 de **BSD** (*Berkeley Software Distribution*) est basée sur Unix v6 en **1977** et est appelée **1BSD**. La version **2BSD** basée sur Unix v7 date de 1978.

À partir de là deux écoles vont s'affronter. La première, en théorie officielle, est celle de AT&T qui va continuer à développer les versions 8, 9 et 10 durant les années 1980 dans des buts de recherche. Dans le même temps, elle développe un Unix entièrement commercial appelé **Unix System III** et le vend dès **1982**. En **1983** AT&T développe et vend les premières versions Unix System V. La dernière version, **Unix System V release 4.2** date de **1993**. On note cette version d'Unix par l'abréviation SVR4. Son code source est disponible sous licence. Un organisme peut en acheter une et développer sa propre version commerciale.

Durant ce temps, l'université de Berkeley ne chôme pas et continue le développement de BSD comme alternative sous licence Open Source de Unix System III et V dont elle n'a plus le droit d'utiliser les sources. C'est dans BSD que va être implémenté pour la première fois le protocole **TCP/IP**, base de l'Internet moderne, grâce au financement du ministère américain de la défense. La dernière version officielle de BSD est **4.4BSD** en juin **1994**.

d. La guerre des Unix

La période allant du milieu des années 1980 à 1994 n'est pas de tout repos. Les effets de la séparation de Unix en deux branches ont été désastreux et ont failli causer sa perte. Les deux camps (AT&T avec son System V et Berkeley avec son BSD) ne s'entendent pas sur un standard commun. L'effet, outre les multiples procès (jusqu'en 1993) sur l'utilisation du nom et des outils dérivés d'Unix, est que de multiples versions d'Unix commerciales et surtout incompatibles entre elles ont poussé comme de la mauvaise herbe. C'est de cette époque que datent les grands noms des clones Unix dont **Solaris**, **AIX**, **OSF1 / Digital Unix / True64**, **Xenix** , **HP-UX**, **IRIX**, **Ultrix**, **Unixware**, **A/UX**, tous souvent incompatibles avec le voisin mais clamant haut et fort leur appartenance à Unix. Cette guerre des Unix est réellement connue comme la période sombre des « *Unix wars* ». Personne n'arrivant à se mettre d'accord sur une base et un standard communs. L'effet direct de cette guerre a été la création d'une niche dans le marché des systèmes d'exploitation dans laquelle la société Microsoft s'est largement engouffrée avec son système d'exploitation **Windows NT** (qui, peu de monde le sait, est aussi dérivé d'Unix).

En **1984** un groupe d'éditeurs d'Unix commerciaux tente une première standardisation en créant X/Open Standards afin de diffuser un document appelé **X/Open Portability Guide** décrivant un standard ouvert (accessible à tous) pour Unix. Ce comité aboutit en **1987** quand Sun Microsystems et AT&T décident de travailler sur un Unix unifié, fusion de BSD et de System V. Le résultat est en fait System V Release 4.

La jalousie est un vilain défaut. La concurrence accuse Sun de vouloir devenir le maître du jeu et fonde *Open Software Foundation* soit **OSF** en **1988**. OSF se veut lui aussi LE standard ouvert Unix, sauf qu'il se base ouvertement sur BSD. Ses spécifications sont connues en **1990**.

En réponse, AT&T et un nouveau groupe créent *Unix International* en **1989** dans une énième tentative d'unification. Sans plus de succès. Devant cet imbroglio AT&T décide de se débarrasser d'Unix dont elle est toujours officiellement propriétaire et crée pour cela une société appelée *Unix System Laboratories* en **1992**. Tous les droits d'Unix sont transférés à USL.

e. La standardisation

Alors que la situation semble bloquée, un nouvel acteur apparaît et va réussir là où les autres ont échoué. La société **Novell** rachète USL l'année de sa création et devient propriétaire de SVR4.2. En **1993** Novell cède la marque Unix à X/Open. Unix International disparaît en **1994** et OSF est restructurée. Enfin, en **1995** Novell cède la licence d'exploitation du code source d'Unix à la société **SCO** *Santa Cruz Operations* (qui deviendra Caldera puis de nouveau SCO). La même année X/Open et OSF fusionnent définitivement et deviennent **The Open Group**.

Il n'existe qu'un seul organisme de standard Unix. Unix est de ce fait un standard ouvert : ses spécifications sont connues et chaque éditeur de système Unix commercial ou gratuit désirant assurer une compatibilité avec l'ensemble des Unix doit implémenter ce standard. Chaque éditeur est cependant libre de programmer ce standard comme il le souhaite, une même fonction pouvant être écrite de plusieurs manières. Les dégâts des *Unix Wars* ont été nombreux permettant l'émergence de nouveaux systèmes d'exploitation comme Windows NT de Microsoft.

f. Unix est un standard

Pour s'assurer que tous les Unix suivent les mêmes recommandations, The Open Group diffuse des normes (*Single Unix Specification*, *Unix95*, *Unix98*, *Linux Standard Base*, etc.) et peut faire passer des certifications. Ces normes s'appuient aussi en partie sur celles définies depuis **1988** par l'**IEEE** *Institute of Electrical and Electronics Engineers* (que l'on prononce I3E) et notamment **IEEE 1003** aussi appelée **POSIX** (*Portable Operating System Interface*). Le X est un héritage et une reconnaissance du travail effectué sur Unix. IEEE 1003 est composée de 15 documents regroupant par thèmes tout ce qu'un Unix doit contenir (commandes de base, interpréteur de commandes, interfaces utilisateur, fonctions de programmation, etc.) pour être conforme au standard POSIX. POSIX n'est pas limité à Unix.

Windows NT est conforme à POSIX pour certains de ses composants. POSIX n'est pas un standard ouvert. Les spécifications de *The Open Group* sont ouvertes et accessibles à tous et les éditeurs préfèrent s'y référer.

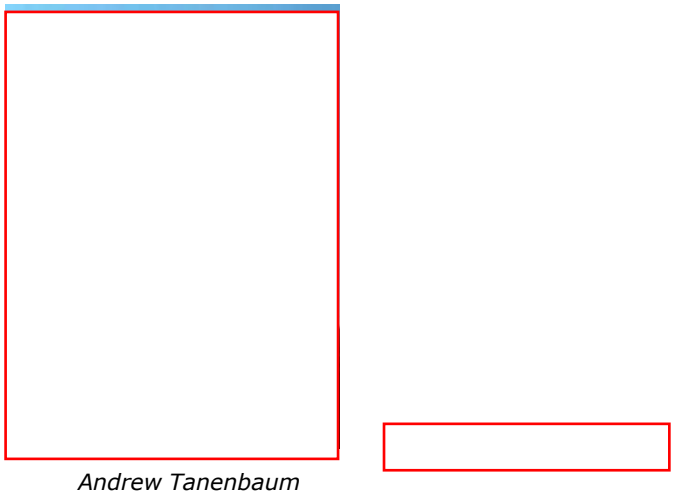
Pour être utilisé dans certaines administrations américaines, un système d'exploitation Unix doit être conforme au standard POSIX. De ce fait quand Linux a dû être utilisé, le gouvernement de Bill Clinton a entièrement fait financer la certification **PCTS** (*Posix Conformance Test Suite*) par le Trésor américain.

Les dernières versions officielles des versions BSD et System V datent de 1994. Les Unix conçus à partir de 1995 implémentent les recommandations de The Open Group. Cependant historiquement quelques versions continuent à être « orientées » plutôt BSD, ou plutôt System V concernant leur configuration ou parfois les deux, comme Linux (et selon la distribution).

---

g. Unix sur les ordinateurs personnels

Le premier Unix pour ordinateur personnel, au sens ordinateur de type IBM PC est **Xenix**. Il est issu de Unix v7 et est sorti en **1983** sur PC (des versions ont été disponibles plus tôt sur d'autres architectures matérielles). C'est la société Microsoft qui a effectué le portage de Xenix, au prix de nombreuses modifications. La version 2 de Xenix date de 1985 et est basée sur Unix System V. Lorsque IBM démarre le développement de OS/2 en association avec Microsoft, ce dernier transfère les droits de Xenix à SCO en 1987. La version 2.3.1 de cette même année supporte le 386, le SCSI et TCP/IP. Xenix devient SCO Unix en 1989 puis disparaît au profit de SVR4.



Andrew Tanenbaum

L'américain **Andrew Stuart Tanenbaum** (surnommé Andy) est chercheur et enseignant en informatique et actuellement à la tête de l'Université libre d'Amsterdam. Il est aussi l'auteur d'ouvrages de références en informatique sur la théorie des systèmes d'exploitation. En **1987**, dans un but pédagogique, il conçoit et écrit le système d'exploitation Minix. Il utilise 20 Mo d'espace disque et ne nécessite que peu de ressources étant parfaitement à l'aise avec 2 Mo de mémoire vive. Minix aura une grande importance pour Linux. Minix existe toujours et la version 3 est sortie en octobre 2005.

De nombreux dérivés de BSD ont été portés sur PC. Le premier est **386BSD** en octobre 1989 et dérive de 4BSD. S'il existe encore c'est son successeur **NetBSD** qui est le plus connu, dérivant lui-même de 4.3BSD et de 4.4BSD. NetBSD est le système d'exploitation qui a été le plus porté sur d'autres architectures matérielles. **OpenBSD** est dérivé de 4.4BSD et très orienté sécurité. En huit ans une seule faille de sécurité a pu être exploitée. **FreeBSD** est aussi dérivé de 4.4BSD et est issu directement de l'époque des procès entre BSDI et AT&T. De ce fait, FreeBSD est entièrement libre et ouvert.

Solaris, l'Unix de Sun Microsystems est disponible depuis plusieurs années sur PC et la version OpenSolaris (version 10) est Open Source.

Enfin, **Linux** est probablement l'Unix libre le plus connu et répandu sur le PC. L'histoire de sa création mérite bien quelques détails supplémentaires.

Le logiciel libre

1. Les origines du logiciel libre

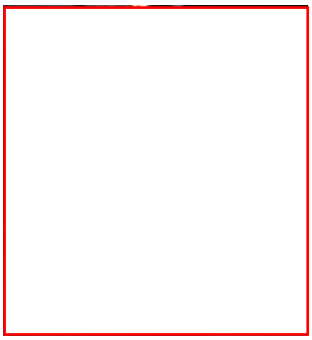
Unix est le parfait exemple du travail qui peut être effectué quand toutes les énergies sont canalisées à la recherche d'un idéal technologique. Quand AT&T diffuse presque librement en 1974 le code source du système d'exploitation auprès des universités parce que, parmi ses raisons, il ne voit pas d'avenir économique pour son produit, il ne semble pas se douter de l'engouement des étudiants, des professeurs et des chercheurs en informatique. Cette première communauté va passer beaucoup de temps à modifier et à améliorer le produit, remontant toutes les nouveautés à AT&T pour une intégration dans le produit officiel. Au contraire lors du changement de licence de 1978 l'énergie de la communauté a été canalisée vers le projet universitaire BSD délaissant l'Unix commercial de AT&T. Notez que les plus grandes avancées eurent lieu avec l'Unix de Berkeley.

Les premiers ordinateurs étaient essentiellement des outils de recherche pour les universitaires (et aussi des monstres de calcul pour des besoins militaires). Dans les laboratoires de recherche, les logiciels circulaient comme les idées : librement. Il n'y avait rien de plus banal qu'un logiciel développé par une équipe de programmeurs ou de chercheurs soit diffusé à d'autres équipes d'autres universités et partout où il y en avait besoin. Il n'y avait rien de plus normal que ce logiciel soit modifié par une autre équipe, et ainsi de suite. Aujourd'hui encore quand un illustre mathématicien démontre un théorème difficile il diffuse le résultat de ses travaux dans des ouvrages spécialisés dans le but de faire avancer la science. Tout le monde y a accès.

Mais l'informatique n'a pas suivi le même chemin. Bien que science, le fruit des recherches en informatique ne s'est pas restreint au monde des universitaires. Rapidement les entreprises ont pu voir l'immense intérêt d'automatiser certaines de leurs tâches comme la comptabilité, la paie, etc. Avec l'achat des premiers gros ordinateurs de gestion il fallait des programmes. Ces programmes ont commencé à être protégés comme des secrets industriels et une nouvelle branche commerciale est née : l'édition de logiciels. Une fois arrivée dans le monde des affaires, l'informatique est devenue très rapidement beaucoup moins libre. On s'est mis à parler de licences, de taxes et de redevances, de droit d'auteur (qui n'empêche pas d'autoriser selon le cas la copie), de limitation des droits, d'interdiction de copier, etc.

2. Le projet GNU et la FSF

Richard Stallman n'a probablement pas été le premier à déplorer ce fait mais a décidé de réagir. Informaticien au laboratoire d'intelligence artificielle au MIT à la fin des années 1970, il utilise une imprimante qui tombe souvent en panne. Comme ses collègues et lui disposent du code source du pilote (programme de gestion) de l'imprimante ils l'ont modifié pour qu'un signal leur soit envoyé à chaque panne. Quand le laboratoire achète un nouveau modèle de Xerox plus fiable, le pilote pour leur système d'exploitation n'est pas livré. Désirant l'adapter à ses besoins, Richard Stallman fait appel à un autre laboratoire qui dispose du code source mais qui refuse de le lui fournir : Xerox l'interdit. Ainsi l'imprimante ne marchera jamais, et Stallman est tellement choqué de cette réaction qu'il décide d'œuvrer dans la défense et la diffusion du logiciel libre en réaction au monde fermé du logiciel propriétaire.



Richard Stallman

Stallman décide en **1983** d'écrire un nouveau système d'exploitation entièrement libre d'accès, d'utilisation, de modification et de redistribution. Basé sur Unix il le nomme **GNU** (*Gnu's Not Unix*). Les acronymes récursifs sont très à la mode chez les informaticiens. On trouve l'annonce du projet et des motivations de Stallman sur http://www.gnu.org/gnu/initial-announcement.html. Pour son système il a besoin d'un noyau (le cœur du système d'exploitation) et d'outils (pour gérer les fichiers par exemples). Ce n'est pas un coup d'essai pour Stallman qui a déjà écrit un grand éditeur de texte appelé **Emacs**. Les premiers développements vont très vite et les outils sont très nombreux et souvent de meilleure qualité que ceux du commerce. Par contre la conception d'un noyau Unix est beaucoup plus complexe et nécessite une phase théorique importante. Le projet **HURD** (*Hird of Unix Replacing Daemons*) est lancé. Il n'a toujours pas abouti.

La bataille n'est pas que technique, elle est aussi politique, philosophique, commerciale et juridique. Pour défendre le logiciel libre Stallman crée la **FSF** (*Free Software Foundation*) en 1985 qui diffuse les idées du logiciel libre. Parmi ses

premiers travaux figure la rédaction (avec l'aide d'avocats) d'une licence spéciale pour ces logiciels appelée la **GPL** (*General Public License*). Un logiciel libre garantit quatre libertés :

- **Liberté 0** : la liberté d'utiliser un logiciel quel que soit l'usage que vous en faites.
- **Liberté 1** : la liberté d'étudier le fonctionnement du programme et de l'adapter à votre besoin.
- **Liberté 2** : la liberté de redistribuer des copies afin d'aider votre voisin (au sens large du terme).
- **Liberté 3** : la liberté d'améliorer le programme et de diffuser les améliorations au public à fin d'en faire bénéficier l'ensemble de la communauté.

Les libertés 1 et 3 nécessitent d'avoir obligatoirement accès au code source du programme. La liberté 3 définit la notion de communauté autour du logiciel libre.

Remarquez que le mot « gratuit » n'est indiqué nulle part. En anglais « free » signifie tant libre que gratuit. Le logiciel libre est à prendre dans le sens de « liberté » et pas gratuit (Free as a speech et non pas Free as a beer comme disent les anglais). Il est tout à fait possible et même parfois conseillé de commercer avec le logiciel libre. Mais comme les libertés 2 et 3 autorisent la diffusion du logiciel, il est toujours possible d'en récupérer une copie gratuitement et ce tout à fait légalement. La gratuité est un effet de la liberté telle que définie pour le logiciel libre.

Les travaux de HURD avancent peu ou mal. Ses développeurs ont pris le pari de développer un micro-noyau : les composants de base du système d'exploitation sont « éclatés » en plusieurs sous-unités indépendantes mais devant communiquer ensemble. Le choix théorique est excellent mais l'implémentation technique est très difficile. GNU ne dispose pas de noyau. C'est Linux qui va faire aboutir le projet en **1992** quand il passe sous licence GPL.

3. L'Open Source

Et l'**Open Source** ? L'expression est apparue en 1998 quand Netscape Communicator est devenu un logiciel libre. L'expression *Open Source* (source ouverte) était utilisée dans les slogans pour associer libre et diffusion du code source et faire comprendre et admettre les logiciels libres auprès des entreprises. Le but était de faire abstraction des apports fondamentaux du libre pour se concentrer uniquement sur les avantages techniques et économiques de ce nouveau modèle. Avec le temps, l'expression a été reprise dans tous les sens par les médias et les entreprises, et sa définition a été largement entachée. On a parlé de « Open Source limité » en proposant l'accès aux sources mais sans droit de modification ou de redistribution. Or, le logiciel libre ne souffre d'aucun aménagement. Il est libre ou n'est pas.

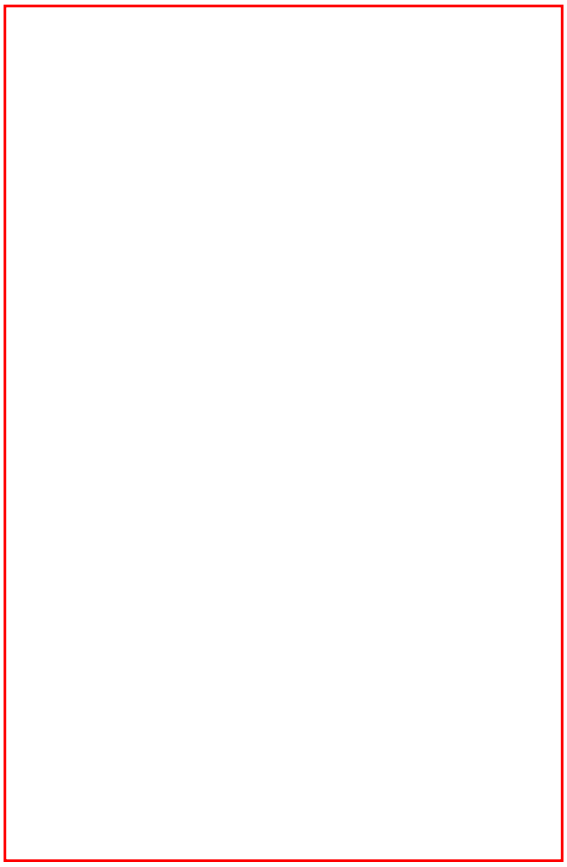
Si vous voulez être certain que le programme que vous utilisez est libre, vérifiez le nom de la licence et rendez-vous sur le site de **OSI** « *Open Source Initiative* » http://www.opensource.org qui en recense la majorité des plus connues. C'est une initiative de **Eric S. Raymond** (ESR) grand hacker (spécialiste de très haut niveau) et l'un des grands noms de l'Open Source. Parfois en conflit avec Richard Stallman, leurs deux visions (techniques pour ESR, philosophiques pour Stallman) sont pourtant complémentaires.

---

4. GNU/Linux

a. Linus Torvalds

L'histoire de Linux commence quand Linus Torvalds, jeune étudiant finlandais à l'université de Helsinki âgé de 21 ans, acquiert en 1991 un ordinateur à base de 386 pour remplacer son Sinclair QL qui commence à montrer ses limites. Le 386 est un microprocesseur 32 bits génial qui gère, entre autres, la mémoire virtuelle et la commutation des tâches. Mais le gros problème est qu'un PC est livré avec MS-DOS, un système d'exploitation loin d'être optimal et surtout n'exploitant aucune possibilité de ce processeur. Linus eut alors l'idée d'installer un autre système appelé Minix, un petit Unix simple et gratuit développé par le célèbre Andrew Tanenbaum, qui permet d'exploiter son beau PC tout neuf acheté à crédit. Linus se met à travailler et à développer dessus. Son but est d'apprendre le fonctionnement du 386, notamment la commutation des tâches en langage assembleur. Il commence à travailler sur un projet assez simple : un émulateur de terminal, entièrement en assembleur, pour se connecter au serveur de son université.



Linus Torvalds

b. L'accident

Oui mais voilà qu'un jour, suite à une mauvaise manipulation, il efface par accident les premiers secteurs d'amorce de la partition de son disque dur contenant Minix, effaçant ainsi son principal outil de développement. Il ne reste que deux solutions : soit tout réinstaller, soit partir de son existant et l'étoffer de manière à le rendre autonome. Bien entendu, l'environnement de développement est réinstallé, mais Linus décide d'améliorer son projet, en lui rajoutant le nécessaire : code de base, pilote rudimentaire de disque dur, passage au langage C, etc. Le 25 août 1991, la version 0.01 est prête et diffusée dans la plus grande indifférence ou presque. Pour les outils, rien de plus simple, le projet GNU initié par Richard Stallman dispose déjà de tout le nécessaire. Linux sera le noyau qui manque au système d'exploitation GNU.

c. La première version officielle

Le but de Linux est de faire quelque chose qui dépasserait Minix. Par la première version diffusée, il faut que le shell (interpréteur de commandes) et gcc (compilateur C) soient utilisables. C'est le cas pour la version 0.02 annoncée le 5 octobre 1991 sur le groupe comp.os.minix :

« Vous regrettez les beaux jours de Minix-1.1, époque bénie où les hommes étaient dignes de ce nom et écrivaient leurs propres pilotes de périphériques ? Vous cherchez à vous investir dans un projet original et vous vous languissez d'un système modifiable à votre convenance ? Vous êtes frustré que tout fonctionne sous Minix ? Vous regrettez les nuits blanches passées à tenter d'implanter un programme récalcitrant ? Si tel est le cas, lisez ce qui suit : Comme signalé il y a un mois, je travaille actuellement sur une version libre d'un système analogue à Minix pour ordinateur AT-386. Ce système est à présent utilisable (mais peut-être ne vous conviendra-t-il pas, tout dépend de ce que vous recherchez) et je compte en diffuser les sources. Il s'agit pour l'instant de la version 0.02, capable néanmoins d'exécuter bash, gcc, gnu-make, gnu-sed, compress, etc. »

d. Le succès communautaire

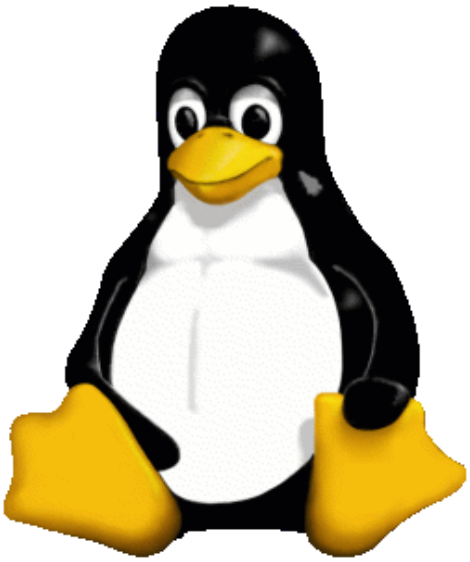
À partir de ce moment le succès, grâce à la diffusion par Internet, est au rendez-vous et les contributions commencent à arriver. Une communauté Linux se forme. La version 0.03 arrive, puis la 0.10. En 1992, Linux peut enfin faire fonctionner l'interface graphique X11. De ce fait, Linux pense qu'il est temps d'accélérer le mouvement et nomme la version suivante 0.99, pensant toucher au but. Ce fut, rétrospectivement, une erreur. En effet, il a fallu attendre 18 mois que la version 0.99pl114 soit finalisée et qu'enfin la version 1.0 sorte en janvier 1994. Entre les

premiers balbutiements et la version 1.0 il y a eu quelques rebondissements, notamment le conflit « technique » entre la conception monolithique de Linux et la vision micro-noyau de Andrew Tanenbaum, ce dernier engageant les hostilités avec la fameuse phrase : « Linux est obsolète ».

e. Les années 1994-1997

Ces années voient l'apparition des grandes distributions Linux que vous connaissez encore aujourd'hui : Red Hat, Debian, Suse, Slackware. Mandrake est arrivé un peu plus tard. Durant ces années, Linux ne cesse de s'améliorer, avec l'arrivée notable de la modularité et de la version 2.0. C'est surtout durant ces années que Linux sort du petit monde des hackers et se fait connaître en entreprise. Les projets foisonnent, et déjà l'idée d'améliorer le système et de l'ouvrir au monde du bureau (desktop) fait son bout de chemin avec le début du développement de produits comme Gnome ou KDE.

La mascotte de Linux appelée **Tux** date de 1996 et a été créé par Larry Ewing à l'aide du logiciel libre GIMP. Tux (apocope de *Tuxedo et Torvalds UniX*) n'est pas un pingouin mais un *manchot pygmée*. Le fait est que le mot anglais penguin désigne dans cette langue aussi bien le véritable pingouin (*razorbill*) que le manchot, d'où une certaine confusion.



Tux, la mascotte de Linux

f. À partir de 1998 : l'explosion

On ne sait pas si c'est dû à un ras le bol général des utilisateurs, mais l'année 1998 est celle d'annonces spectaculaires. Le monde de l'informatique réalise enfin que Linux n'est pas qu'un joujou pour étudiant bidouilleur. En janvier 1998, Netscape annonce que son produit passe en Open Source. Il en sortira Mozilla, Firefox et Thunderbird. Les instituts de formation ajoutent Linux à leur catalogue. En juillet 1998, Oracle et Informix sont portés. En septembre, IBM porte DB2 et Sybase fait de même. Linus Torvalds fait la une de « *Forbes* ». KDE et Gnome arrivent en version 1.0. En bourse, les cours montent, les sociétés Linux voient le jour. C'est le succès.

Janvier 1999, c'est l'arrivée de Linux 2.2 et la continuité du succès, qui commence à faire réagir Microsoft. C'est David contre Goliath. C'est toujours le cas. On aurait pu croire que l'explosion de la bulle Internet en bourse en 2000 allait tout faire capoter. Vous constatez que non. Linux n'est pas un colosse aux pieds d'argile. Ses pieds, c'est la communauté, inébranlable. Le noyau 2.4 sort le 4 janvier 2001. Le noyau 2.6 sort le 18 décembre 2003.

g. Aujourd'hui et demain

Aujourd'hui Linux est reconnu comme un système d'exploitation stable, robuste et performant. Il est utilisé dans plus du tiers des serveurs dans le monde et dans la moitié des serveurs web. Il a conquis le monde de l'entreprise, le monde universitaire. Il a surtout su conserver son indépendance, garantie par la communauté et le nombre de contributeurs, face aux géants de l'informatique. La prochaine grosse cible de Linux, c'est le poste de travail, et pourquoi pas, l'usage familial en remplacement de Windows. Il reste encore un peu de chemin, mais nombreux sont ceux qui ont déjà franchi le pas.

Quel matériel pour Linux ?

1. L'architecture

Linux existe pour au moins trois architectures matérielles courantes :

- **x86** pour les ordinateurs dont les processeurs sont du type Intel (du 386 au Pentium 4) ou AMD (Athlon, Duron, Sempron) 32 bits. Cette version fonctionne aussi sur les machines à base de processeurs 64 bits.
- **x86_64** pour les ordinateurs dont les processeurs sont du type Intel (Pentium 4 à partir des séries 600, Xeon, Dual Core/Quad Core) ou AMD (Athlon 64, Sempron 64, Opteron) 64 bits. Cette version ne marche pas sur les processeurs 32 bits.
- **ppc** pour les ordinateurs dont les processeurs sont de type PowerPC c'est-à-dire les anciens ordinateurs de marque Apple. Cette version ne s'installera pas sur les dernières machines Apple basées sur un processeur de marque Intel.

Certains pilotes matériels ou applications sont encore peu ou mal adaptés à la version 64 bits. Si vous constatez des dysfonctionnements gênants, pensez à installer la version 32 bits qui devrait résoudre vos problèmes. N'oubliez pas que Linux est le premier système d'exploitation offrant le support complet des processeurs 64 bits et que ces problèmes sont le reflet de la jeunesse de cette version.

Configuration matérielle de base

Linux supporte théoriquement tous les types de processeurs depuis la version 386, et peut fonctionner avec seulement quelques Mo de mémoire. La distribution Polux Linux fonctionne sur un 386 avec 4 Mo de mémoire. La distribution Damn Small Linux fonctionne avec un 486, 16 Mo de mémoire et utilise 50 Mo d'espace disque. On trouve même des distributions sur une ou deux disquettes démarrant avec 2 Mo de mémoire.

N'espérez cependant pas travailler correctement avec une version moderne de Linux et son environnement bureautique graphique dans ces conditions pseudo-préhistoriques. Les pré-requis suivants doivent être respectés :

- **Un processeur** (ou plus) de type Intel Pentium et supérieur ou un équivalent de marque AMD.
- **Au moins** 128 Mo de mémoire, mais 256 Mo ou plus apportent une réel confort d'utilisation. Pensez plutôt à disposer de 512 Mo voire 1 Go pour une utilisation optimale. Au prix de la mémoire ce n'est pas un luxe. Dans le cadre d'une installation minimale en mode texte, 64 Mo suffisent.
- 500 Mo d'espace disque pour une installation minimale (sans interface graphique et seulement les outils de base), mais 2,5 Go pour une installation standard, auquel il faut rajouter l'espace pour les données de l'utilisateur et la partition d'échange.
- Une carte graphique même ancienne compatible avec la norme Vesa, acceptant de préférence le 1024x768 en 65 356 couleurs pour l'environnement graphique, et sans aucune importance en mode texte.

Ce sont des pré-requis de base. Si la fréquence d'horloge de votre processeur joue principalement pour la vitesse d'exécution de vos applications, elle peut être fortement bridée par le manque de mémoire ou un disque dur trop lent. La quantité de mémoire est un facteur important de confort. Plus il y en a mieux c'est : plusieurs programmes pourront fonctionner en même temps, la partition d'échange ne sera pas sollicitée et le système pourra utiliser plus de mémoire tampon pour accélérer les accès aux disques et périphériques. Si vous disposez de 256 Mo ou moins, envisagez de passer à 512 Mo. La différence est flagrante.

Les performances globales restent acceptables sur un Pentium II 300 avec 256 Mo pour une utilisation bureautique ou Internet simples. Les performances s'écroulent lors du lancement simultané de plusieurs programmes. Sur un simple AMD Duron 800 avec 512 Mo, les performances sont excellentes pour la plupart des usages classiques.

2. Compatibilité du matériel

Avant d'installer Linux vérifiez si votre matériel est correctement pris en charge par Linux. Établissez une liste des composants de votre ordinateur et de vos différents périphériques. Le support du matériel est régulièrement mis en avant par les débutants lorsque l'installation échoue. Telle carte graphique, telle imprimante, tel scanner ne fonctionnent pas correctement ou pas du tout. Bien que Linux supporte la plupart des composants des ordinateurs récents, l'achat d'un ordinateur dernier cri n'est pas une garantie de bon fonctionnement.

Dans la liste, le plus important n'est pas la marque et le nom du modèle commercial mais le composant, la puce principale, appelée « *chipset* », du produit. Dans le cas du Wi-Fi, peu importe que la carte soit une Palmnet BZ46G. Mais si vous savez qu'elle est construite autour d'une puce Centrino (Intel 2200 par exemple), alors vous trouverez vite qu'elle fonctionne avec Linux. Les produits de certains constructeurs doivent être évités car leurs matériels ne disposent pas de pilotes permettant de les utiliser. La quasi-totalité du matériel d'impression proposé par Hewlett Packard fonctionne parfaitement avec Linux alors qu'il faut fuir les imprimantes à jet d'encre Lexmark (attention : ce n'est pas la qualité du produit qui est mise en cause, mais son support sous Linux).

Sauf à disposer d'une machine très ancienne, toutes les cartes graphiques fonctionneront. Dans tous les cas Linux propose un pilote générique appelé « vesa » qui, s'il n'offre pas les meilleures performances, permet d'utiliser toutes les cartes compatibles avec ce standard vieux de plus de dix ans. Certains constructeurs proposent des pilotes très performants. Les dernières cartes des constructeurs Nvidia et ATI sont supportées avec des pilotes 3D offrant les mêmes performances qu'avec les autres systèmes d'exploitation. Le système graphique de Linux supporte par défaut un grand nombre de carte, y compris avec l'accélération 3D. Les meilleures cartes graphiques pour Linux restent les cartes à base de composants NVIDIA et Intel.

Les cartes son intégrées aux cartes mère respectent un standard de facto (AC97) qui est supporté par Linux. Les cartes son intégrées sur les cartes mère sont rarement des composants haut de gamme. Une simple carte Live coûtant moins de 30 euros est bien plus performante. Certains modèles spécifiques de cartes son peuvent poser des problèmes.

Le Wi-Fi devrait fonctionner soit avec un pilote natif pour votre matériel, soit à l'aide d'un outil particulier appelé Ndiswrapper qui permet d'utiliser les pilotes de Windows pour Linux. Selon votre choix de distribution de petits composants appelés firmwares et nécessaires à la carte Wi-Fi ne sont pas fournis par défaut et doivent être récupérés à part, soit depuis le système de mise à jour, soit sur un support (le fameux add-on) supplémentaire, soit chez le constructeur de la carte. Le Bluetooth est parfaitement reconnu et supporté.

Différents sites disposent de bases de données de matériels compatibles pour vous renseigner rapidement. Les moteurs de recherche restent votre meilleure source. À titre indicatif, voici une liste de sites qui vous aideront dans vos recherches :

- Liste de compatibilité Novell : http://cdb.novell.com/index.php?LANG=en_UK
- Liste de compatibilité openSUSE : http://en.opensuse.org/HCL
- Imprimantes : http://www.linuxprinting.org
- Scanners : http://sane-project.org/
- Périphériques USB en général : http://www.qbik.ch/usb/devices/
- Cartes son : http://www.alsa-project.org/
- Les cartes Wi-Fi : http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/
- Modems internes ou externes de type Windomem : http://linmodems.org/
- Webcams : http://www.linux.com/howtos/Webcam-HOWTO/hardware.shtml

---

Malgré toutes les bonnes volontés du monde il arrive parfois qu'un matériel ne fonctionne absolument pas avec Linux. À qui la faute ? Les pilotes de périphériques sont très souvent écrits par des développeurs n'ayant même pas accès aux spécifications du matériel et qui font tout par ingénierie inverse, c'est-à-dire en tentant de reproduire le fonctionnement du périphérique depuis son résultat. C'est très long. Certains constructeurs jouent le jeu. À défaut de fournir un vrai pilote ils diffusent auprès des développeurs une documentation technique. D'autres fournissent eux-mêmes un pilote au code fermé pour une distribution donnée ou que vous devez adapter vous-même à chaque nouvelle version de Linux. Dans ce cas l'avenir du pilote n'est plus garanti (ce qui s'est par exemple passé avec les cartes à base de chipset graphique Kyro II). Aussi avant de critiquez Linux si votre matériel ne marche pas critiquez en premier le constructeur de celui-ci.

Choisir une distribution

1. Debian

Le projet Debian a été fondé en 1993 par Ian Murdock à une époque où l'idée même de distribution Linux en était encore à ses balbutiements. Le nom Debian provient de Debra (la femme de Murdock) et Ian. Debian a longtemps été la seule distribution entièrement et uniquement composée de logiciels libres et Open Source ce qui lui vaut toujours le nom officiel de Debian GNU/Linux. Debian a aussi été supporté quelques temps officiellement par la FSF comme distribution Linux de référence. Les avantages de Debian sont nombreux :

- un nombre gigantesque de packages qui se chiffre en milliers,
- un logiciel d'installation appelé APT très pratique et performant,
- une distribution 100% open source,
- une stabilité à toute épreuve pour un environnement de production.

Ces avantages entraînent aussi quelques inconvénients :

- des packages souvent anciens,
- des mises à jour de la distribution irrégulières et trop espacées,
- des risques liés à la multiplication des paquets et des dépendances,
- une installation et une configuration compliquées.

Tous ces inconvénients ne sont pas forcément des défauts. Faut-il préférer une version ancienne mais totalement exempte de bugs ou la toute dernière version d'un produit dont la fiabilité n'a pas été pleinement éprouvée ?

Tous ces éléments font de Debian une distribution idéale pour les informaticiens, les ingénieurs et administrateurs système et réseau, les environnements de production en entreprise, les puristes du libre, les amateurs éclairés qui n'ont pas peur de mettre les mains dans le cambouis. Quant aux débutants ils passeront probablement leur chemin au début sauf à vouloir apprendre sur le tas.

a. Ubuntu

Le milliardaire sud-africain Mark Shuttleworth, principalement connu du monde entier pour avoir été l'un des premiers touristes de l'espace, mais aussi des informaticiens pour avoir fait fortune en revendant sa société Thawte spécialisée dans la sécurité à Verisign, est un vrai informaticien qui a contribué au projet Debian. Devant les

quelques inconvénients de la distribution il crée la distribution Ubuntu Linux en 2005 avec un budget initial de 10 millions de dollars pour rémunérer les développeurs. Le mot Ubuntu est un mot du langage africain bantou signifiant « humanité aux autres » ou encore « je suis ce que je suis grâce à ce que nous sommes tous ». Cette définition reflète ce qu'est la distribution : un dérivé de Debian dont le but est de fournir des logiciels plus récents et très fortement axés sur la convivialité et l'ergonomie à l'aide du support du plus grand nombre :

- une distribution issue de Debian,
- une compatibilité avec les packages de Debian,
- un système d'installation très simple,
- une sortie tous les 6 à 8 mois,
- un environnement graphique agréable.

Cette distribution est idéale pour les étudiants, cependant la tentation est très forte de revenir au fonctionnement d'une distribution Debian, les deux étant compatibles.

b. Red Hat et Fedora

Logo Red Hat

S'il y a bien une société commerciale dans le monde Linux qui a marqué et qui continue à marquer son époque, c'est bien la société Red Hat. Fondée en 1995 par Robert Young et Marc Ewing, elle édite la célèbre distribution éponyme dont la première version officielle date de 1994 (la société a été fondée après la sortie de la distribution). Le système de package RPM est apparu avec la version 2.0. Les distributions Red Hat ont très fortement marqué les esprits car elles sont restées la référence pendant presque dix ans. Chaque version était innovante tant dans l'intégration des logiciels que dans son installateur (appelé anaconda) et ses outils de configuration.

Cependant en 2003 la version 9.0 est la dernière destinée officiellement au grand public. Les versions suivantes ont été confiées au projet communautaire **Fedora** qui continue tous les six mois à sortir une nouvelle version. Red Hat se concentre maintenant sur le monde de l'entreprise avec des distributions commerciales appelées **RHEL** (*Red Hat Enterprise Linux*) :

- des versions professionnelles destinées aux entreprises,
- des solutions du poste de travail au plus gros serveur,
- des architectures matérielles nombreuses,
- un support commercial,
- des mises à jour assurées pendant sept ans,
- 100% libre.

Vous vous doutez bien que même si l'installation d'une version RHEL AS (*Advanced Server*) est possible sur un PC de bureau elle n'a pas forcément d'intérêt pour un poste de travail ou un débutant. Bien que libre (ses sources sont intégralement disponibles librement) son coût avec le support est très élevé. Cependant si l'installation ne vous fait pas peur la distribution **CentOS** (*Community Enterprise Operating System*) est une copie exacte et téléchargeable de

RHEL dont toute trace des noms et visuels Red Hat a été supprimée.

Logo Fedora

Quant au projet Fedora, il suit un cycle de développement rapide et reste destiné au grand public. Son installation est simple. Cependant l'ensemble manque un peu de cohérence (par exemple l'outil de partitionnement des disques n'est accessible que durant l'installation) ce qui en fait une distribution idéale pour tous ceux qui, amateurs éclairés, souhaitent rentrer un peu plus dans le détail.

c. Mandriva (ex-Mandrake)

Mandriva Linux (ex-Mandrake) est une distribution dérivée et longtemps entièrement compatible avec la distribution Red Hat. Elle a été créée par Gaël Duval afin d'intégrer à la distribution l'environnement de bureau graphique KDE contrairement à Red Hat qui intégrait l'environnement GNOME. Pendant plusieurs années Mandrake a été la distribution phare en forte compétition avec Red Hat. Mandrake était en effet (et est toujours) plus conviviale. Son processus d'installation est un modèle du genre et son utilisation des plus simples. Renommée Mandriva suite au rachat de la société Connectiva, la distribution est pourtant en perte d'audience depuis quelques temps. Les raisons sont multiples mais fortement liées aux aléas de la société Mandriva. Une gestion difficile suite à une mauvaise orientation dans les années 2000-2001 (le e-Learning et l'expérience américaine des Start-up) a failli une première fois conduire à sa perte et a provoqué un redressement judiciaire dont la société a réussi à sortir avec brio, pour rencontrer de nouveau quelques temps plus tard des problèmes. L'introduction sur le marché boursier n'a pas donné les résultats espérés. Souffrant d'une image trop grand public, les solutions professionnelles n'arrivent pas à s'imposer. Enfin la distribution grand public si elle reste toujours au top techniquement souffre parfois de quelques problèmes d'instabilité.

Mandriva continue cependant d'innover fortement, notamment dans le poste de travail nomade avec des distributions clé en main bootables depuis des clés USB, et c'est généralement plus par habitude et ouïe-dire qu'elle est bien souvent automatiquement conseillée aux débutants.

d. openSUSE

Se prononçant *sousse*, **openSUSE** est une distribution d'origine allemande datant de 1992. Le nom de l'entreprise lui-même était un hommage au célèbre **Konrad Zuse** l'inventeur des ordinateurs modernes.

La distribution est originellement basée sur la distribution Slackware. En 1996 SuSE se rapproche d'une distribution française appelée **Jurix** créée par Florian La Roche qui est utilisée comme base à la place de Slackware. Cette même année le développement de l'outil YaST est démarré et la version 4.2, en fait totalement nouvelle, sort. Au même moment SuSE utilise le nouveau gestionnaire de packages de Red Hat appelé RPM.

Début 1997 SuSE tente l'aventure américaine en installant de nouveaux bureaux à Oakland. Entre 1997 et 2003 la distribution SuSE ne cesse d'être améliorée pour devenir une référence en matière de simplicité d'installation, d'administration et d'utilisation.

Si l'avenir de la distribution était garanti, la société Novell rachète tout d'abord la société Ximian spécialisée dans le développement Open Source d'outils pour Linux dont un bureau Gnome, une messagerie appelée Evolution et une suite de configuration appelée Red Carpet. Novell annonce le rachat de la société SuSE en janvier 2004. Le développement est désormais communautaire avec le projet **openSUSE**. Le monde entier s'il le souhaite peut contribuer à l'amélioration du produit. En réponse, Novell s'engage à fournir à la communauté tous les six à huit mois une version stable, libre et gratuite.

Geeko, mascotte openSUSE

e. Les autres

Il est impossible de nommer toutes les distributions tant elles sont nombreuses. Outre les grandes distributions que vous venez de rencontrer quelques autres noms sont à retenir. La distribution **Slackware** est l'une des plus anciennes distributions. Elle était même livrée sur disquette. Durant les toutes premières années la Slackware était la distribution de référence pour apprendre à utiliser Linux. Elle est extrêmement dépouillée. Son installateur est réduit à la plus simple expression et la plupart de la configuration doit être effectuée à la main. Son système de package est inexistant (il s'agit de simples archives de fichiers compressés). C'est donc l'idéal pour les bidouilleurs et les fondus de Unix. Cependant, ce n'est pas l'idéal pour les débutants.

La distribution **Gentoo** est très particulière. Plutôt que de vous livrer tous les logiciels déjà prêts à être utilisés, son installateur va avec votre aide déterminer exactement la configuration de votre machine et notamment votre modèle de microprocesseur en fonction de quoi il compilera (il transformera le programme source sous forme de langage compréhensible en langage machine) chaque composant logiciel que vous aurez sélectionné avec toutes les optimisations prévues pour votre matériel. C'est ce qu'on appelle une distribution source. Le résultat peut être intéressant : les performances de vos programmes peuvent être améliorées, étant en moyenne de 10% à 20% plus rapide. Mais à quel prix ! L'installation n'est pas forcément aisée pour les débutants et surtout elle est très longue : plusieurs heures (voire dizaines d'heures) selon vos choix de logiciels et la puissance de votre machine.

Une autre distribution surprenante est la **LFS** (*Linux From Scratch*). Ce n'est pas précisément une distribution mais plutôt un guide vous donnant une méthode pour construire votre propre configuration. Pas à pas, c'est à vous de choisir vos divers composants et la configuration de votre système. Ainsi vous êtes certain d'obtenir exactement la distribution que vous voulez, ni plus ni moins. Mais là encore les débutants, et même d'ailleurs les amateurs éclairés, passent leur chemin.

A côté de toutes ces distributions on trouve de nombreux dérivés. **Aurox Linux** dérive de Red Hat. **PCLinuxonline** dérive de Mandriva. **Kunbuntu** dérive de Ubuntu (ou plutôt est une distribution Ubuntu pleinement supportée mais intégrant l'environnement bureautique KDE) qui dérive de Debian. **CentOS** dérive de RHEL, et ainsi de suite. Encore à côté il y a les mini-distributions qui tiennent sur un mini cd ou une clé USB et c'est idéal pour dépanner un ordinateur.

2. Les LiveCD

Le LiveCD est une catégorie surprenante. Vous êtes certainement très nombreux à vouloir essayer Linux pour voir à quoi ça ressemble ou pour vérifier s'il fonctionne correctement avec votre matériel. Plutôt que de l'installer sur votre disque dur (si cette étape vous fait peur le chapitre Le shell et les commandes GNU de cet ouvrage vous propose un guide pas à pas pour installer votre Linux) pensez d'abord à tester Linux sans l'installer. Le LiveCD sert principalement à ça : c'est une installation complète de Linux qui est fortement compressée et qui tient sur un seul cd ou dvd (dans ce cas on parle de liveDVD).

Pour utiliser un liveCD c'est très simple : insérez le CD ou le DVD dans votre lecteur et redémarrez votre ordinateur en ayant bien vérifié dans la configuration de votre machine (le setup du BIOS, voire le mode d'emploi de votre ordinateur) que votre lecteur CD ou DVD est le premier à démarrer. Après quelques secondes (ou minutes parfois) de chargement, voici que le bureau apparaît et tous les programmes les plus connus sont accessibles alors que strictement rien n'est installé sur votre disque dur. Le LiveCD le plus connu actuellement est **Knoppix**. Il est basé sur une distribution Debian, et qui plus est, s'il vous plaît un installateur est prévu pour le copier sur votre disque dur. Chaque nouvelle version de SUSE Linux arrive avec un LiveCD pour tester les dernières nouveautés sans l'installer.

Obtenir de l'aide

1. L'aide propre aux commandes

Il n'est pas possible de connaître par cœur tous les paramètres et arguments d'une commande. Linux propose heureusement au moins deux mécanismes pour connaître ceux qui sont supportés par une commande. La plupart du temps, le paramètre `--help` affiche l'aide incluse directement au sein du programme appelé. Parfois l'aide apportée est suffisante pour trouver ce que vous cherchez. C'est le cas avec la commande **date** dont la sortie est volontairement tronquée ici car elle prendrait deux pages.

```
$ date --help
Usage: date [OPTION]... [+FORMAT]
   ou: date [-u|--utc|--universal] [MMJJhhmm[[CC]AA][.ss]]
Afficher la date courante selon le FORMAT spécifié ou
initialiser la date du système.

  -d, --date=CHAÎNE        afficher la date selon la description donnée
par la CHAÎNE,
                           excluant le mot réservé « now »
  -f, --file=FICHIER       identique à --date pour chaque ligne du
                           FICHIER de dates
  -r, --reference=FILE      display the last modification time of FILE
  -R, --rfc-2822            output date and time in RFC 2822 format.
...
```

Il peut cependant arriver que l'aide soit trop concise ou manque d'explications, ou bien soit totalement absente. Dans ce cas `--help` est considéré comme un paramètre invalide et vous risquez d'obtenir un message d'erreur et/ou une ligne d'informations :

```
$ cal --help
cal: option invalide -- -
usage: cal [-13smjyV] [[mois] année]
```

La dernière ligne n'explique pas la syntaxe des paramètres.

2. L'aide interne au shell

Les commandes internes n'acceptent pas de paramètre `--help`, mais pour ces commandes l'interpréteur de commandes propose une commande **help**. Utilisée seule elle vous fournit la liste des commandes internes. Si vous lui passez comme paramètre le nom d'une commande interne, l'aide de celle-ci est affichée. C'est ainsi que vous pouvez apprendre que pwd admet deux paramètres optionnels.

```
$ help pwd
pwd: pwd [-LP]
    Print the current working directory.  With the -P option, pwd
prints the physical directory, without any symbolic links; the -L
option makes pwd follow symbolic links.
```

3. Le manuel en ligne

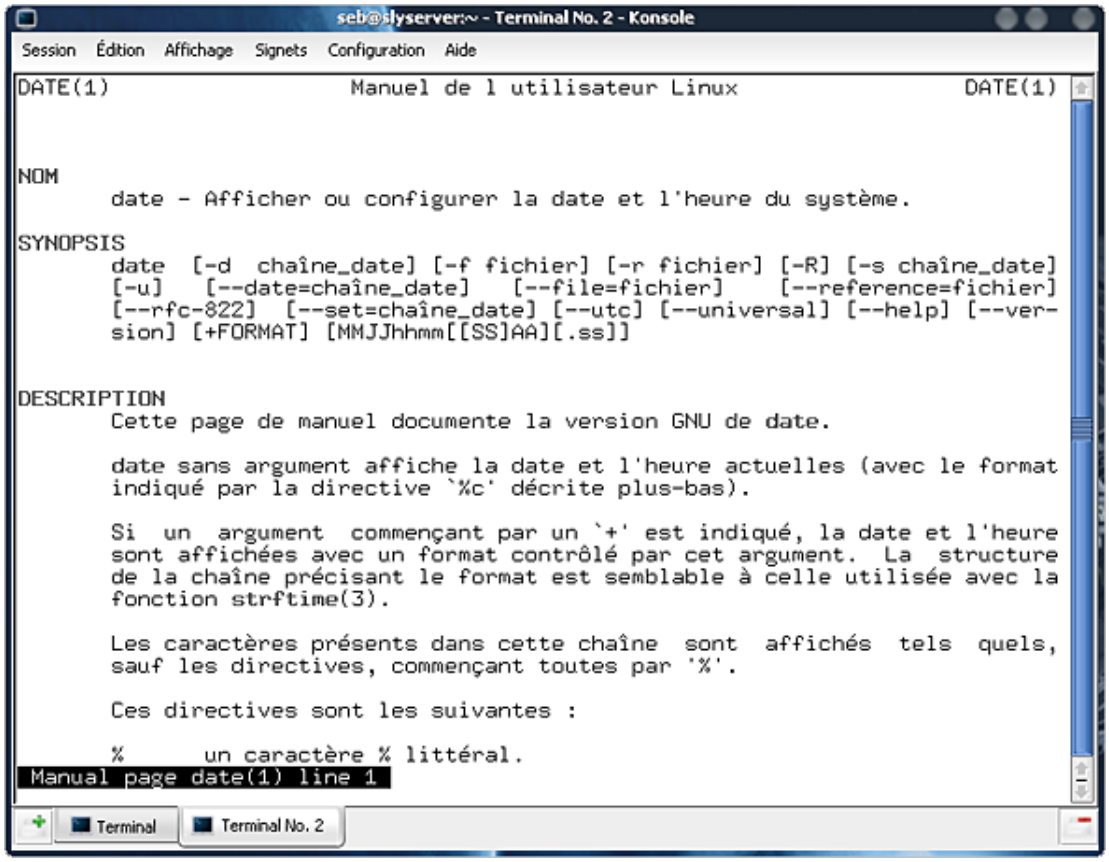
a. Accès

Quand les deux mécanismes d'aide précédents se révèlent être insuffisants, il est très probable que l'aide recherchée se situe au sein du manuel Unix. Ce manuel est standard sur tous les Unix dont Linux, et quel que soit le shell puisqu'il s'agit d'une commande externe.

Le manuel est accessible depuis la commande **man**. Vous pouvez faire un essai simple avec la commande **date** : `$ man date`

Le mode d'emploi de la commande en paramètre de man est affiché.

b. Structure d'une page



Le manuel en ligne

Une page du manuel est composée de plusieurs sections dont celles-ci, sachant qu'elles ne sont pas forcément toutes présentes :

- **Nom** : nom et rôle de la commande.
- **Synopsis** : syntaxe générale, paramètres et arguments acceptés.
- **Description** : mode d'emploi détaillé du fonctionnement de la commande et des arguments principaux.
- **Options** : description détaillée de chaque paramètre possible, généralement sous forme de liste.
- **Exemples** : le manuel peut fournir des exemples concrets d'utilisation de la commande.
- **Environnement** : le fonctionnement de la commande peut réagir différemment si des variables du shell sont positionnées à certaines valeurs.
- **Conformité** : la commande est conforme à des recommandations ou normes (par exemple POSIX).
- **Bogues** : la commande peut parfois rencontrer des dysfonctionnements dans des cas ponctuels qui peuvent être énumérés à cet endroit.
- **Diagnostics/retour** : la commande, selon son résultat, peut retourner des codes d'erreurs significatifs dont la valeur permet de déterminer le type de problème (fichier en argument absent, etc.).
- **Voir aussi** : liste des commandes liées au programme qui peuvent intéresser l'utilisateur.

c. Navigation

Vous naviguez dans l'aide très simplement :

- La barre d'espace défile une page complète.
- La touche [Entrée] défile ligne par ligne.
- Les touches [Haut] et [Bas] défilent d'une ligne vers le haut ou vers le bas.
- Les touches [Pageup] et [Pagedown] défilent d'une demi-page vers le haut ou vers le bas.
- Les touches [Début] et [Fin] font exactement ce qu'on attend d'elles.
- La touche / permet une recherche. /toto recherche toto. Dans ce cas la touche n cherche l'occurrence suivante, tandis que [Shift] **n** (N) recherche la précédente.
- La touche Q quitte l'aide et revient au shell.

d. Les sections

Le manuel Linux ne fait pas que référencer les commandes classiques. C'est un manuel bien plus complet que ça. Les commandes simples, celles d'administration, les fichiers de configuration, les périphériques, les appels systèmes, les fonctions de programmation de divers langages, et bien d'autres choses encore, peuvent y être référencés. C'est pourquoi le manuel est composé de plusieurs sections distinctes.

Section	Contenu
1	Instructions exécutables ou commandes du shell
2	Appels système (API du noyau...)
3	Appels des bibliothèques (fonctions C...)
4	Fichiers spéciaux (contenu de /dev comme sd, hd, pts, etc.)
5	Format des fichiers (/etc/passwd, /etc/hosts, etc.)
6	Les jeux, économiseurs d'écran, gadgets, etc.
7	Divers, commandes non standard, ne trouvant pas place ailleurs
8	Commandes d'administration du système Linux
9	Sous-programmes du noyau (souvent vide)

Il arrive parfois que l'appel au manuel pour une commande ne retourne pas la page du manuel concernée. C'est que man recherche par défaut la première occurrence dans l'ordre des sections. Si vous recherchez de l'aide sur le format du fichier des mots de passe, vous tomberez tout d'abord sur l'aide de la commande passwd. Regardez l'en-tête de la page. Le numéro de la section est indiqué juste après le nom de la commande entre parenthèses. La commande **man** a trouvé une occurrence de passwd dans la section 1 et affiche la page du manuel associée.

```
$ man passwd
PASSWD(1)               Manuel de l'utilisateur Linux
NOM
       passwd  -  mettre à jour les marques d'authentification d'un
utilisateur.
...
```

Vous pouvez demander à man de rechercher le manuel concerné dans une section spécifique en indiquant son numéro juste avant le nom de la commande. Pour accéder au manuel du fichier passwd, faites comme ceci.

```
$ man 5 passwd
PASSWD(5)             Manuel de l administrateur Linux
NOM
       passwd - Fichier des mots de passe.
...
```

e. Rechercher par correspondance

Si vous avez un doute sur la commande à utiliser, ou que vous ayez perdu son nom, ou encore que vous vouliez connaître toutes les commandes liées à un mot, alors utilisez le paramètre `-k` de man :

```
$ man -k passwd
/etc/rpasswd.conf (5) [rpasswd.conf] - configuration file for remote
password update client
chpasswd (8)         - change user passwords in batch
Crypt::SmbHash (3pm) - Perl-only implementation of lanman and nt md4
hash functions, for use in Samba style smbpasswd entries
fgetpwent_r (3)      - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
gpasswd (1)          - change group password
ldappasswd (1)       - change the password of an LDAP entry
lppasswd (1)         - add, change, or delete digest passwords.
makepasswd (1)       - generate and/or encrypt passwords
mkpasswd (1)         - Overfeatured front end to crypt(3)
pam_localuser (8)    - require users to be listed in /etc/passwd
pam_rpasswd (8)      - PAM module to change remote password
passwd (1)           - change user password
passwd (1ssl)        - compute password hashes
passwd (5)           - password file
passwd2des (3)       - RFS password encryption
rpasswd (1)          - change user password on remote server
rpasswd.conf (5)     - configuration file for remote password update
client
rpasswdd (8)         - remote password update daemon
saslpasswd2 (8)      - set a user’s sasl password
smbpasswd (5)        - The Samba encrypted password file
smbpasswd (8)        - change a user’s SMB password
vncpasswd (1)        - set passwords for VNC server
yppasswd (1)         - change your password in the NIS database
```

4. Rechercher de l’aide sur Internet

Comme indiqué au début du chapitre, une communauté existe autour de Linux et du logiciel libre, et les éditeurs de distributions fournissent de la documentation et du support. De ce fait, vous disposez de beaucoup de moyens pour obtenir de l’aide notamment sur Internet :

- la documentation de l’éditeur,
- les sites communautaires (FAQ, forum),
- les newsgroups,
- le projet de documentation libre (HOWTOs).
- etc.

Pensez tout d’abord à la documentation des éditeurs :

- Red Hat : http://www.redhat.com/support

- Debian : http://www.debian.org/doc/
- openSUSE : http://en.opensuse.org/Documentation
- Ubuntu : https://help.ubuntu.com/
- Mandriva : http://club.mandriva.com/xwiki/bin/view/KB/OfficialDocumentation
- Fedora : http://docs.fedoraproject.org

Sur chacun de ces sites, vous trouverez aussi très probablement :

- une base de connaissance,
- un Wiki,
- un forum,
- des rapports de bugs.

Il est impossible de lister tous les sites communautaires, mais en voici quelques-uns :

- LinuxFr : http://linuxfr.org
- Freshmeat : http://freshmeat.net
- Slashdot : http://slashdot.org
- Planet Libre : http://www.planet-libre.org
- Forum Fedora : http://forums.fedora-fr.org/
- Forum Mandriva : http://forum.mandriva.com/index.php?op=Fr
- Forum Debian : http://forum.debian-fr.org/
- Forum Ubuntu : http://forum.ubuntu-fr.org
- Forum openSUSE Alionet : http://alionet.org

Parmi les sites de documentation :

- Lea Linux : http://lea-linux.org
- The Linux Documentation Project : http://tldp.org
- LinuxDocs : http://linuxdocs.org

Installer une Debian

1. Support d'installation

Voici comment, étape par étape, installer une distribution Debian. Il s'agit de la dernière version stable, soit la 4.0r3 Etch, à l'écriture de ce livre. L'installation est effectuée en mode texte, ce qui est la meilleure méthode pour Debian (il existe un installateur graphique peu utilisé). Si vous souhaitez effectuer la même installation, vous pouvez récupérer l'image ISO correspondant à une installation via le réseau (Internet), appelée netinst et accessible via le site de Debian :

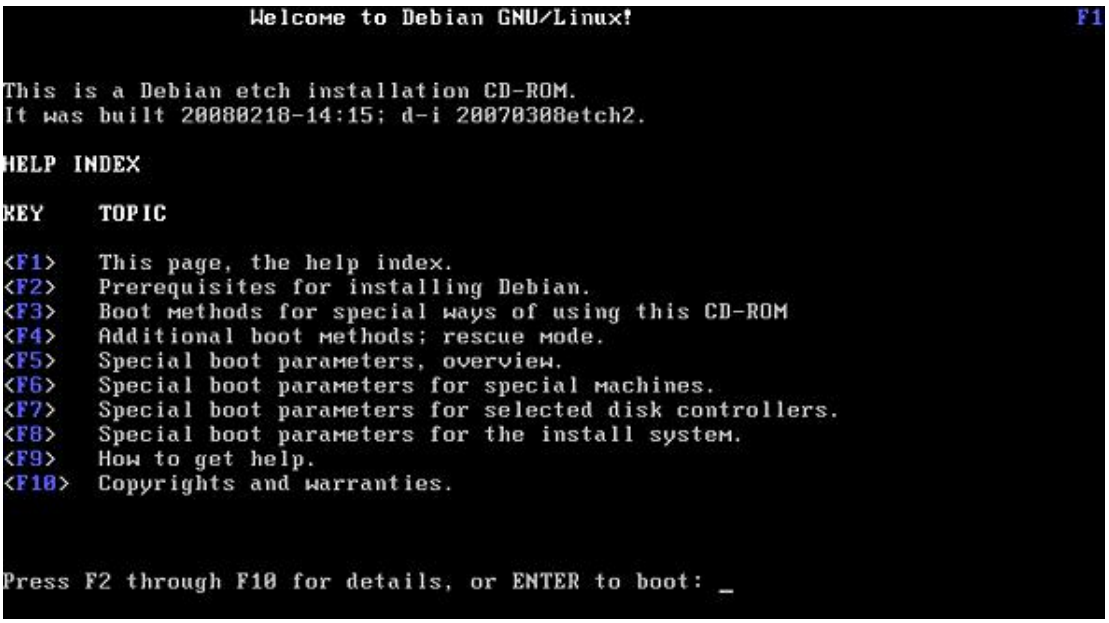
http://cdimage.debian.org/debian-cd/4.0_r3/i386/iso-cd/debian-40r3-i386-netinst.iso

Gravez cette image comme CD ou clé USB. Pour les besoins de ce livre, Debian a été installée dans une machine virtuelle VMWare. VMWare Server est un produit qui n'est pas libre mais qui est gratuit.

2. Boot sur le support

Boot du support Debian

Configurez votre ordinateur pour qu'il démarre sur le support d'installation. Au moment du boot, vous avez accès à une ligne de commande permettant de lancer l'installation en appuyant sur [Entrée], ou les touches de [F1] à [F10] pour accéder à des écrans d'aide supplémentaires. Vous pouvez en effet passer des options en ligne de commande selon votre machine (de [F5] à [F10]) car dans certains cas ponctuels il peut être nécessaire de modifier des valeurs passées au noyau Linux pour un bon fonctionnement.



Options spéciales au boot

Appuyez sur la touche [Entrée] pour lancer l'installation. L'installateur lui-même fonctionne sous Linux, vous voyez défiler toutes les lignes du démarrage du noyau Linux. Enfin le premier écran de l'installateur est affiché.

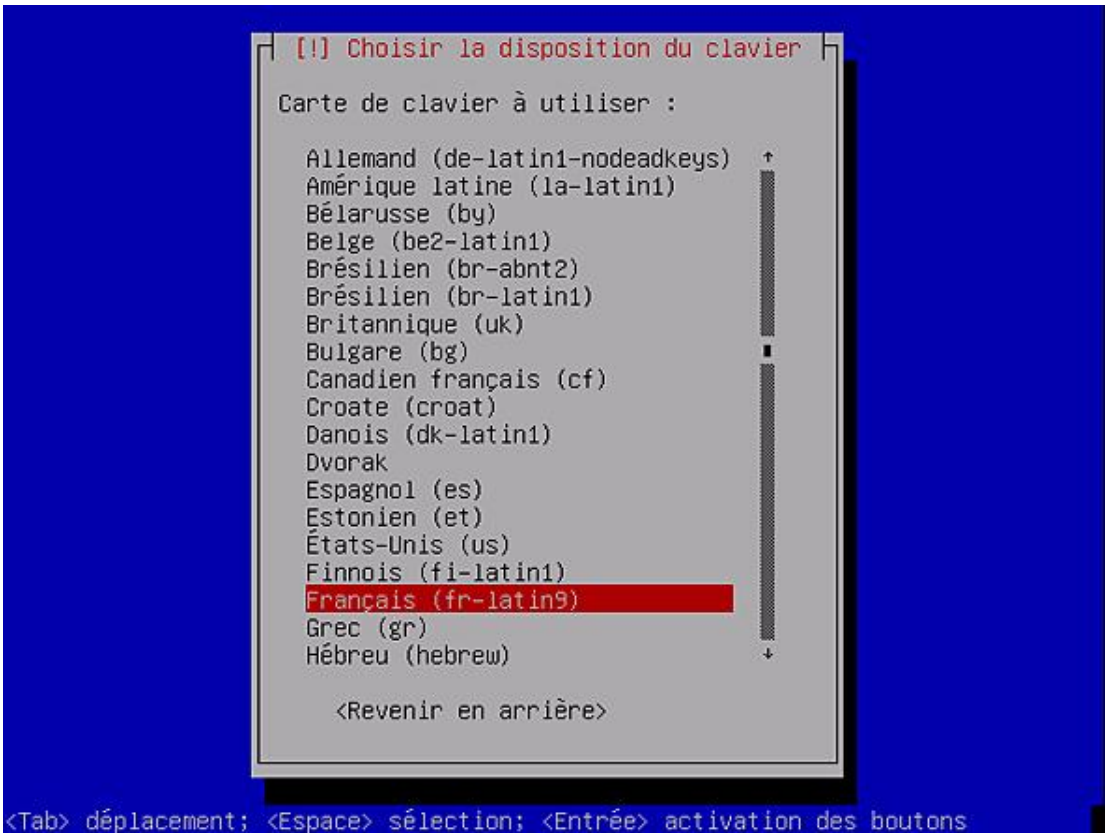
3. Choix des langues et pays

Trois écrans vous permettent de choisir :

- La langue utilisée par le processus d'installation. Naviguez avec les flèches, appuyez sur [Entrée] pour continuer. Dans la suite, c'est le français qui est utilisé.

- Selon la langue initiale choisie, Debian vous demande ensuite dans quel pays vous vous situez. C'est utile car c'est ainsi que sont positionnées les variables locales : format de date, d'heure, encodage des caractères, formats numériques et monétaires, etc.

- Enfin, choisissez votre type de clavier. Pour la France, c'est fr-latin9.



4. Paramètres du réseau

Les trois étapes suivantes concernent les informations réseaux de base. Si l'installateur n'a pas réussi à configurer la carte réseau par DHCP, il vous demandera de saisir les informations de base :

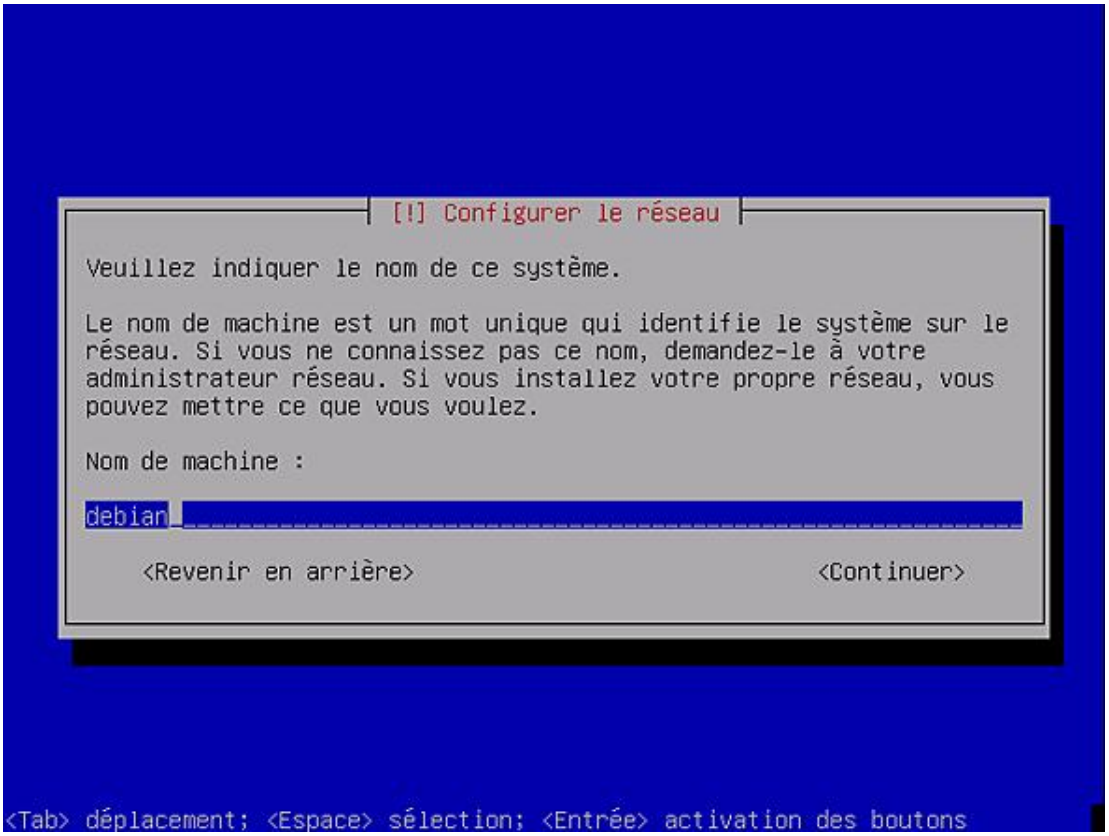
- Adresse IP ;

- masque de sous réseau ;

- passerelle par défaut ;

- DNS.

Puis vous devez saisir un nom d'hôte (le nom de la machine sur le réseau) et le nom du domaine. Si votre machine n'appartient à aucun domaine, laissez le champ vide.



Nom d'hôte de la machine

5. Partitionner les disques

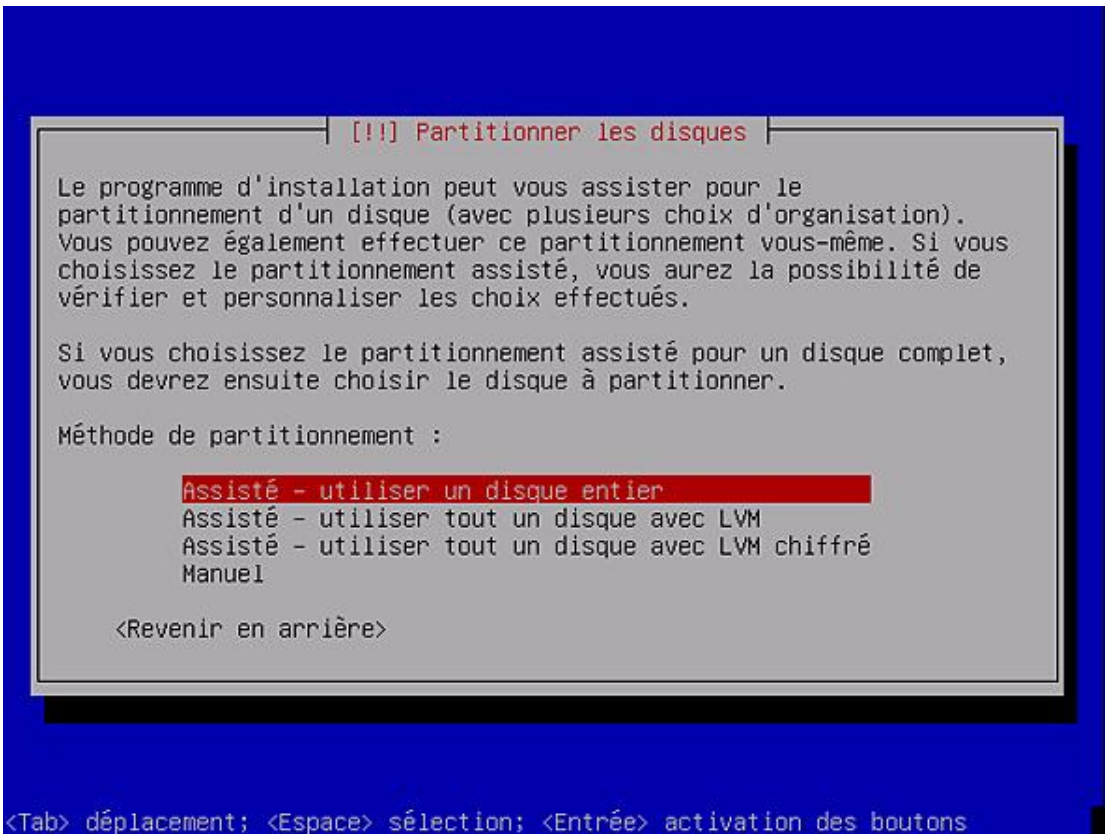
De manière simpliste, vous avez le choix entre trois principales méthodes pour partitionner vos disques :

- Une méthode assistée (voire automatique) en utilisant le partitionnement classique (voir à ce propos le chapitre Les disques et le système de fichiers).
- Une méthode assistée proposant le LVM (*Logical Volume Manager*).
- Une méthode manuelle.

La méthode assistée classique dans le cas d'une nouvelle installation donne des bons résultats. Si vous réinstallez une machine, ou que vous installez Debian sur une machine disposant déjà de partitions contenant les dossiers personnels par exemple, passez par un partitionnement personnalisé.

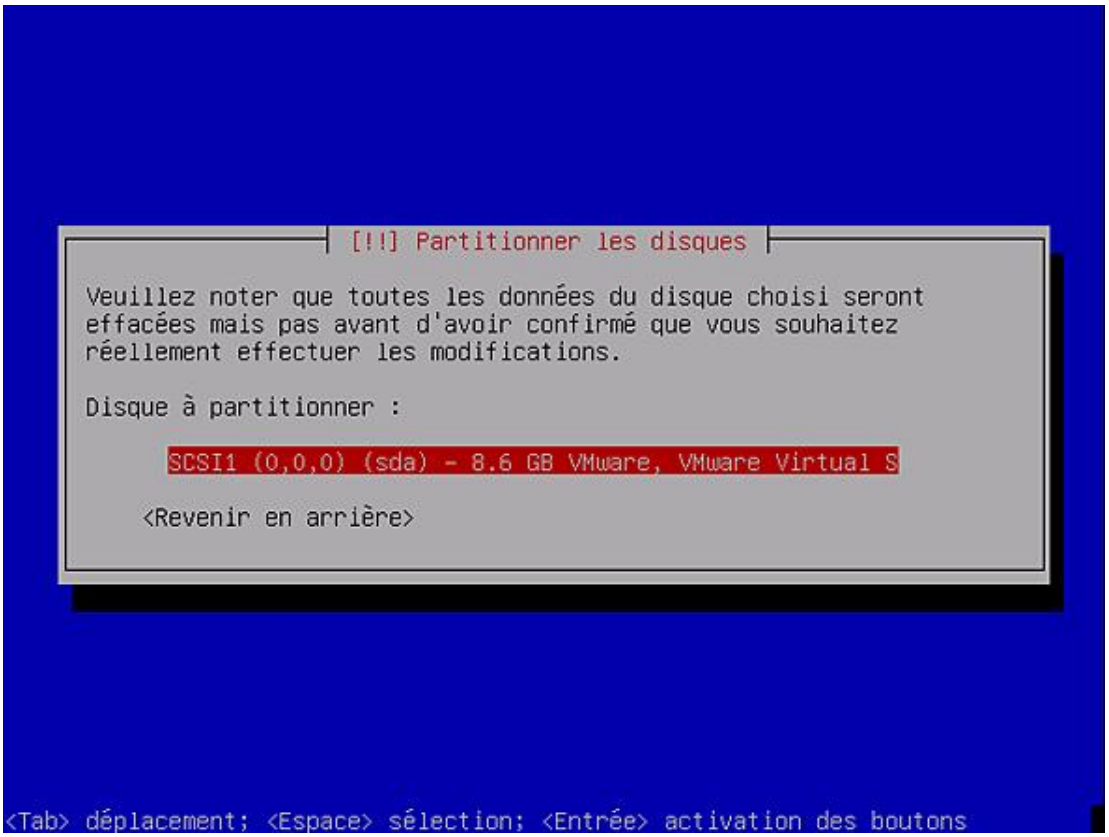
Le LVM consiste à regrouper des disques physiques ou partitions (appelés volumes physiques) en un seul grand espace (appelé groupe de volumes) dans lequel vous pouvez découper des espaces logiques à volonté (appelés volumes logiques), les agrandir, les réduire, etc.

Cependant vous devriez envisager la solution LVM dans le cadre d'un serveur d'entreprise ou si vous pensez rajouter à terme des disques dans votre machine pour rajouter de l'espace de stockage. Le LVM apporte une très grande souplesse.



Étape de partitionnement

Si vous avez choisi la première méthode, vous accédez à l'écran suivant. Pour les besoins du livre, un espace de 8 Go (environ) a été créé sous VMWare comme premier disque SCSI. C'est celui-ci qui va servir pour l'installation.



Choix du disque

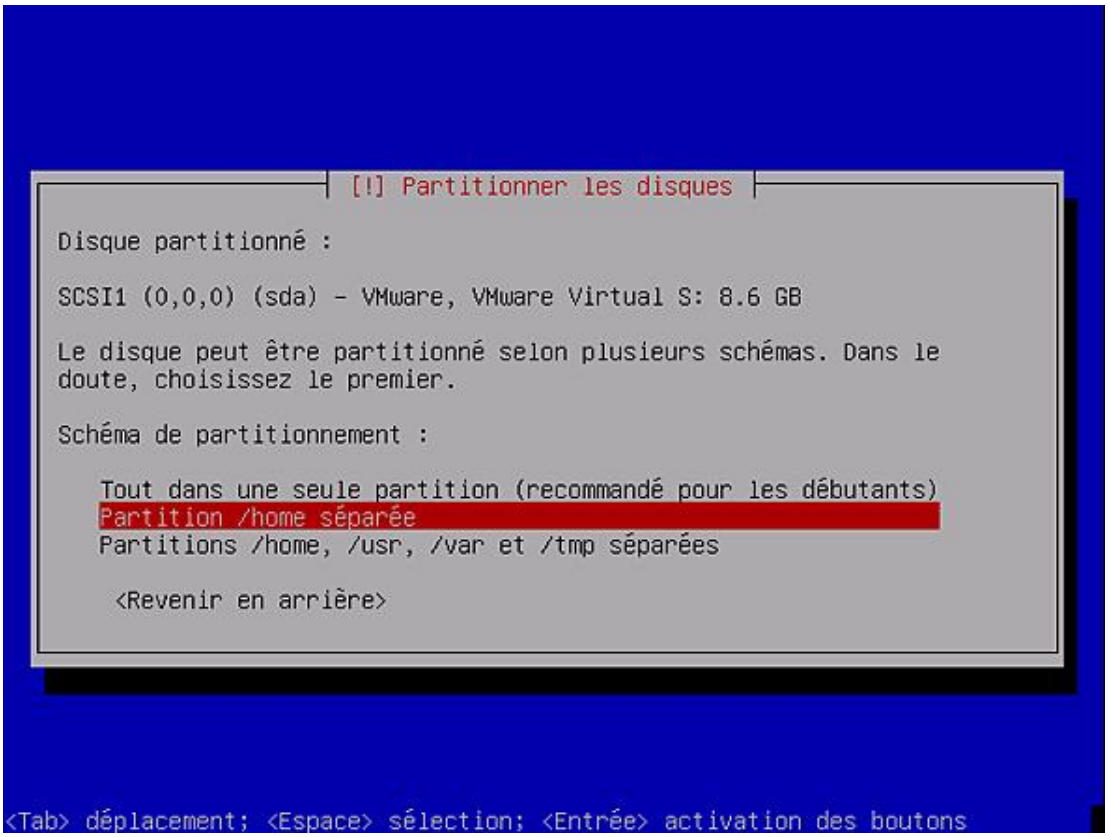
L'étape suivante consiste à choisir le schéma de partitionnement :

- Soit une seule grosse partition dans laquelle vous mettez tout (système, programmes, données). Tout est mis dans la partition racine /.

- Soit deux partitions : une partition racine qui contiendra le système et tous ses composants (programmes, paramètres systèmes, etc.), et une partition qui va contenir les données des utilisateurs. Pour un poste de travail ou un PC personnel (à la maison), c'est la méthode la plus pertinente : elle permet de réinstaller facilement un autre système (mise à jour ou réinstallation complète) sans casser les données personnelles : la nouvelle distribution ainsi installée pourra réutiliser la partition montée sur /home et ainsi récupérer les données.

- La troisième méthode propose de créer cinq partitions différentes : la racine /, les données personnelles /home, les composants utilisateurs (programmes, bibliothèques, données partagées associées, etc.) /usr, le contenu variable /var et les fichiers temporaires /tmp. Ce choix est tout à fait pertinent sur un serveur. Les mails, informations DHCP, sites web, etc., sont souvent stockés dans /var. Les mises à jour des divers programmes (services) sont dans /usr, etc. Ce schéma de partitionnement est quasi-parfait : tout est indépendant. Il devient bien plus simple ensuite de changer de disque, de migrer les données, d'étendre la taille des volumes, etc., sans casser le reste.

Le meilleur choix pour l'installation de test est le second, la suite des opérations se base sur celui-ci.



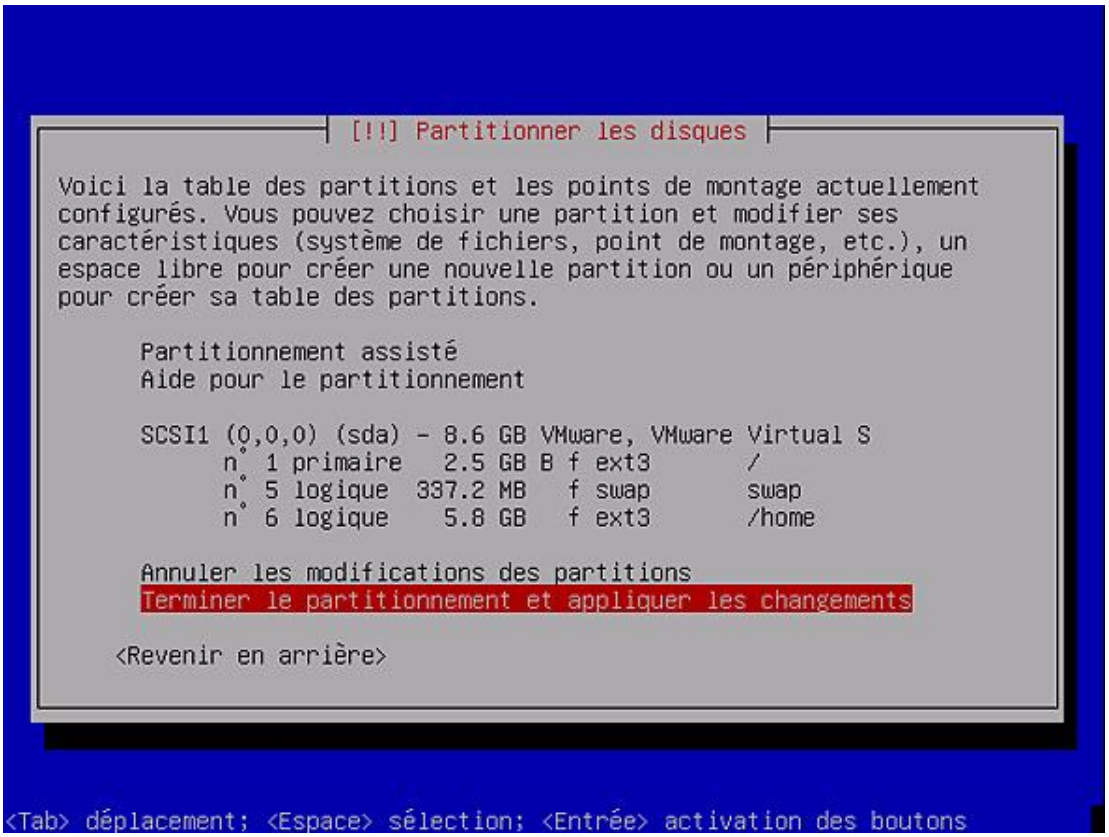
Choix du schéma de partitionnement

S'agissant d'une méthode semi-automatisée, Debian vous indique ses choix pour chacune des partitions. Vous constatez la présence de trois partitions au lieu des deux proposées. Debian a analysé la taille mémoire de la machine et propose la création d'une zone de swap correspondant au meilleur choix possible.

Ne soyez pas non plus surpris par les numéros de partitions. Très souvent la seule partition primaire est la racine / tandis que toutes les autres sont des partitions logiques au sein d'une partition étendue, et Linux numérote les primaires de 1 à 4 (une partition étendue est une partition primaire) tandis que la numérotation des partitions logiques débute à 5. Pour une meilleure compréhension, veuillez vous reporter au chapitre Les disques et le système de fichiers.

Si ce schéma de partitionnement vous convient, validez. L'écran suivant vous donne un récapitulatif que vous devez de nouveau valider.

Attention ! Le partitionnement est suivi de l'écriture des systèmes de fichiers sur les partitions concernées. Cette opération est identique au formatage sous Windows. Cette opération est destructive.

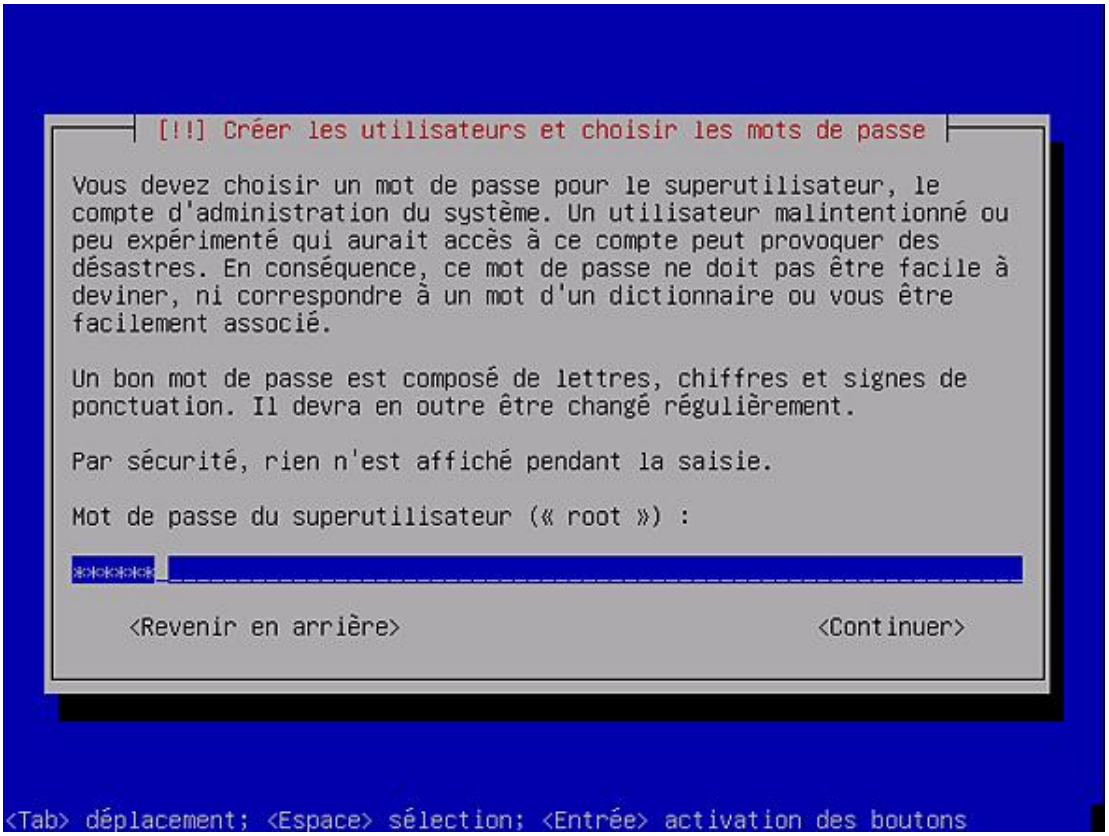


Une fois les changements validés, une barre de progression vous informe de l'état du partitionnement et de l'écriture des nouveaux systèmes de fichiers. Debian va ensuite monter ces systèmes pour l'installation.

6. Comptes root et utilisateurs

Vous devez maintenant saisir le mot de passe de l'administrateur root de la machine. Celui-ci vous sera demandé deux fois pour confirmation. Ne le perdez pas ! Bien qu'il existe quelques méthodes pour réinitialiser le mot de passe, sauf à utiliser un outil spécialisé de "cracking" de mot de passe, il n'y a aucun moyen de retrouver le mot de passe d'origine. Debian vous prévient et peut même refuser un mot de passe s'il est trop simple.

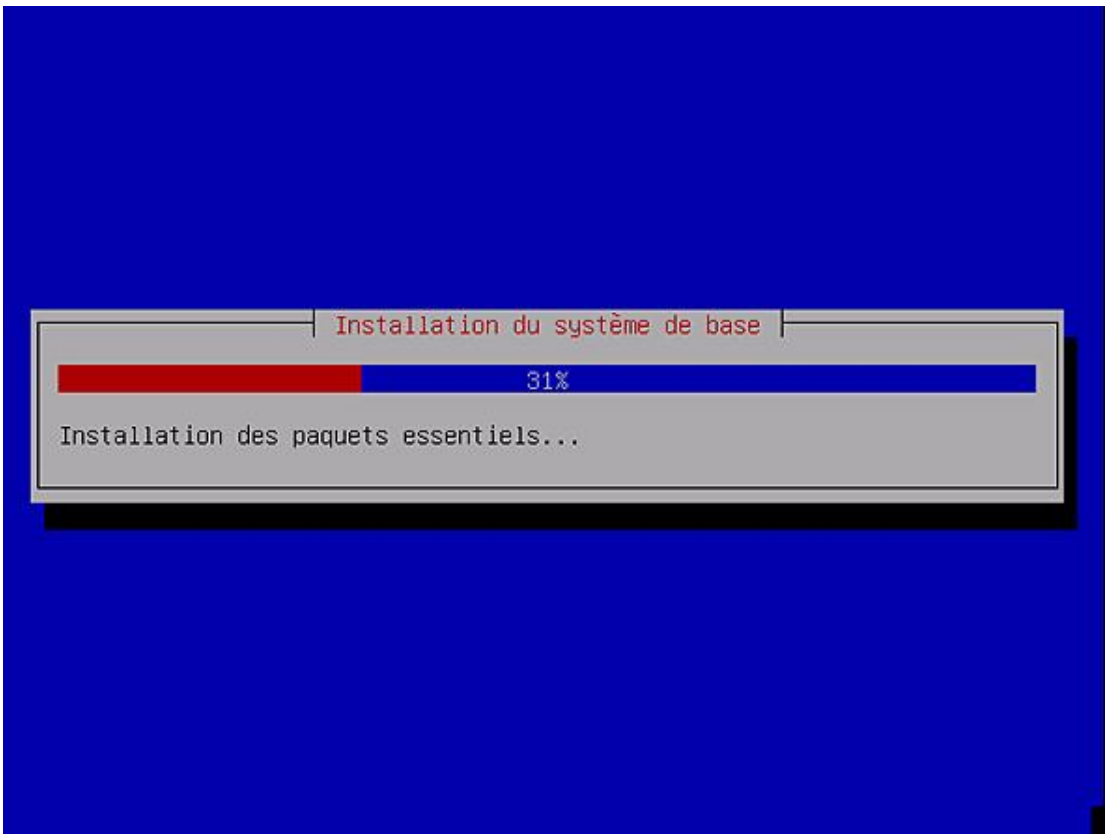
Vous devez ensuite créer au moins un utilisateur simple. Vous devrez saisir le nom complet de la personne et Debian vous propose un login. Saisissez ensuite les mots de passe associés. Vous pouvez créer plusieurs utilisateurs mais rien ne vous empêche de faire ceci après l'installation.



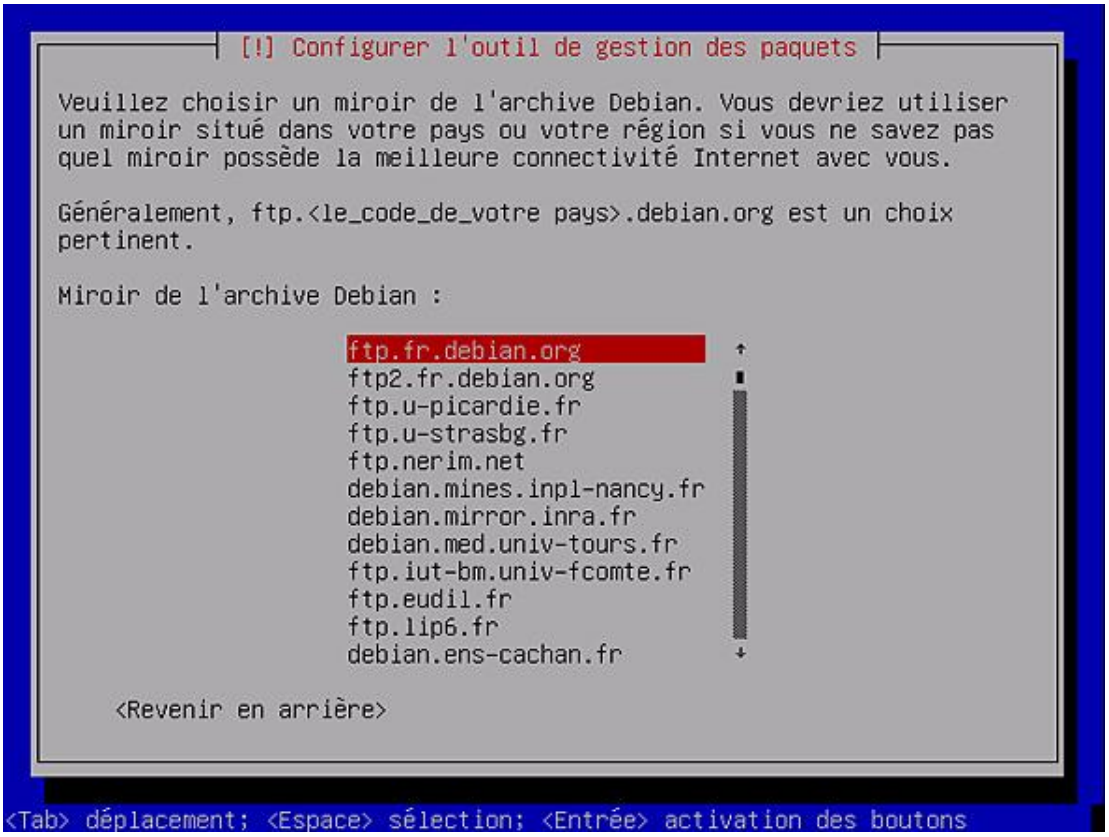
7. Installation

L'installation se déroule en plusieurs étapes :

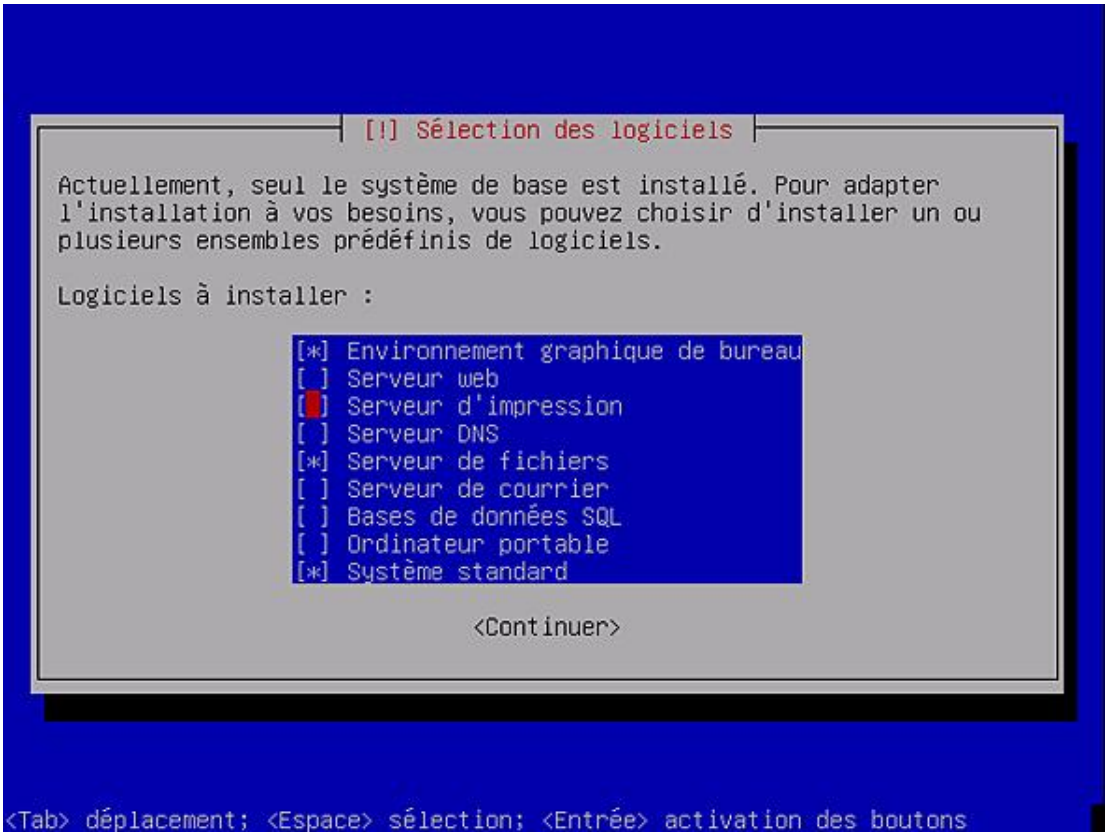
- Installation des éléments de base. Soit téléchargés ou contenus sur le support d'installation, les éléments de base sont recopiés sur votre disque. Ils se composent des packages essentiels.



- Vous choisissez ensuite un miroir : c'est le lieu (sur Internet) où sont présents les dépôts de logiciels Debian. Si vous n'utilisez pas de dépôts, Debian se base uniquement sur le contenu du support d'installation. S'il s'agit d'un DVD il n'y a pas de problèmes, mais si vous utilisez comme ici un CD d'installation réseau, seule la base est présente et rien d'autre. Les miroirs contiennent aussi les mises à jour, notamment de sécurité, parues entre le moment où votre support est sorti et le moment présent. Il est donc fortement conseillé, même si vous disposez de l'ensemble des supports, de configurer un dépôt.



- Vous choisissez d'installer un ou plusieurs ensemble de logiciels prédéfinis. Ces ensembles sont regroupés par thèmes : environnements de bureau (kde, gnome, xfce, etc.), les différents serveurs (fichiers, impression, dns, courrier, SQL, etc.).

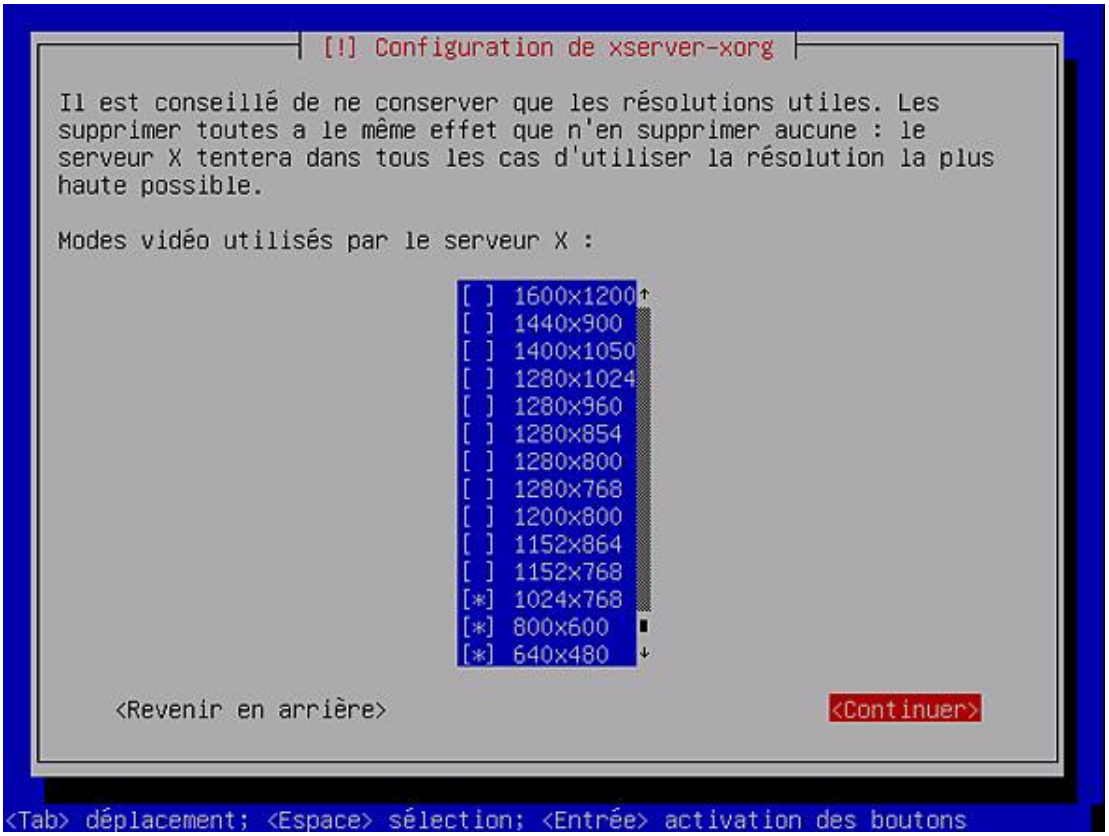


Choix des éléments prédéfinis à installer

Ces étapes passées, l'installation commence.

8. Configuration des packages

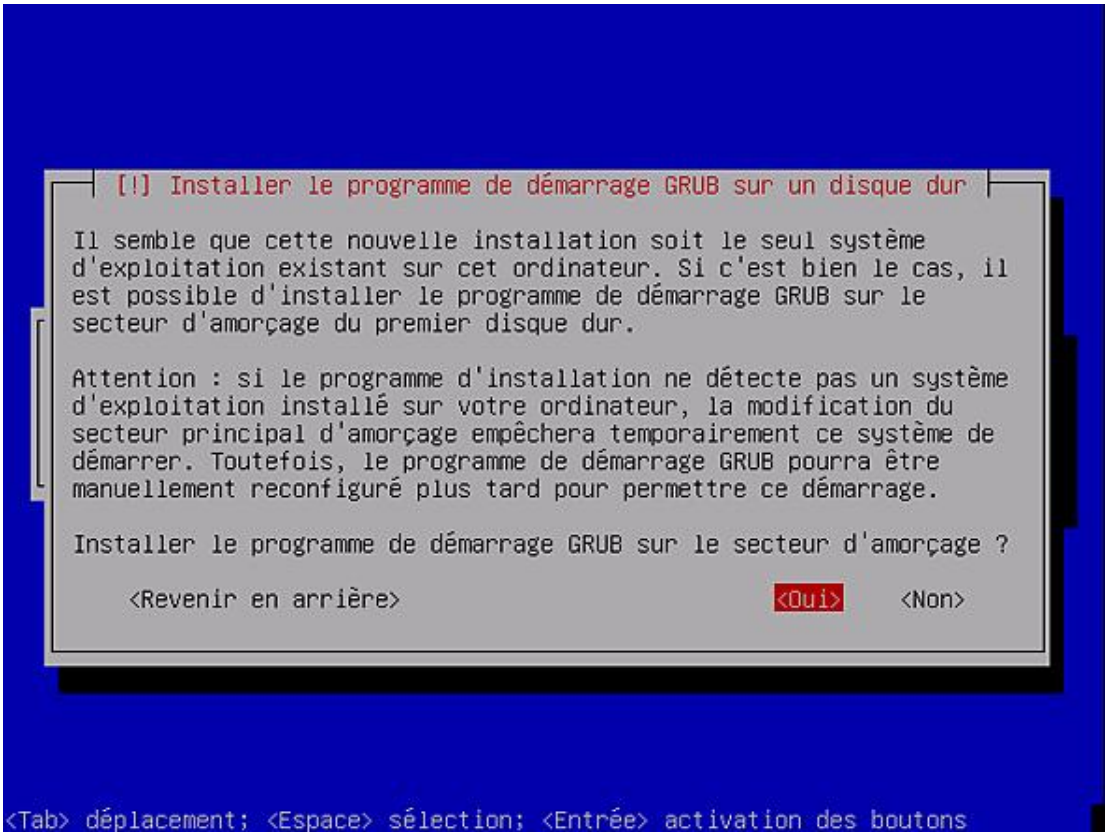
Au cours de l'installation, il se peut que certains packages doivent être configurés. C'est le cas par exemple si vous installez le serveur Samba (partages pour Windows) où le type de serveur, le domaine, etc. vous seront demandés. C'est aussi le cas pour la configuration du serveur graphique X Window. Il n'est pas possible de vous donner une liste des étapes de configuration de ces packages. Cependant il est possible en cas d'erreurs dans la configuration de reconfigurer le paquet concerné (son nom est indiqué en haut, dans l'exemple c'est xserver-xorg) à l'aide de la commande **dpkg-reconfigure** (ex : dpkg-reconfigure xserver-xorg).



Configuration du serveur X

9. Fin d'installation et redémarrage

La dernière étape avant la fin de l'installation est l'écriture du chargeur de démarrage (bootloader). Il s'agit de l'installation de **GRUB**. C'est très simple ici car c'est le seul système installé. Sachez que le chargeur de démarrage écrase celui précédemment installé, mais que vous pouvez parfaitement utiliser grub pour démarrer n'importe quel système y compris Windows.



Installation de GRUB

Il n'y a plus qu'à rebooter. Grub vous propose par défaut deux entrées :

- le mode normal ;
- le mode single-user, ou mode de secours.

Démarrez sur le premier. Si tout va bien et que la carte graphique a été reconnue, vous devez arriver sur le gestionnaire de sessions.

Tout a fonctionné. Le gestionnaire d’affichage attend une connexion.

Installation de openSUSE

1. Support d'installation

Voici maintenant selon le même modèle l'installation pas à pas d'une distribution basée sur des packages au format RPM avec un installateur en mode graphique. Moins sobre que la Debian, à la fois plus simple mais plus étendue, l'installation d'une distribution openSUSE est très appréciée des débutants.

L'installation a été effectuée dans une machine virtuelle VMWare paramétrée exactement de la même manière que pour Debian. Si vous souhaitez installer la même distribution, rendez-vous sur le site suivant : http://www.opensuse.org

Puis accédez à la section de téléchargement. L'installation a été effectuée depuis un DVD de la version 10.3 fournie dans un magazine spécialisé Linux, mais il vous suffit de télécharger l'image ISO correspondante (une image de DVD) et de graver ensuite le DVD correspondant. Une installation via le réseau est aussi possible, tout comme avec Debian.

2. Boot sur le support

Configurez votre ordinateur pour qu'il démarre sur le support et bootez dessus.

Écran d'accueil du support d'installation openSUSE

L'écran d'accueil du support d'installation offre de multiples options et est bien moins sobre que pour certaines distributions concurrentes. Notamment si vous oubliez le support dans le lecteur, le choix par défaut est de démarrer sur le disque dur, ce qui permet de conserver le support dans le lecteur sans repasser par le bios durant toute l'installation. Notez aussi la présence des entrées « live ». Vous pouvez ainsi tester le système avant de l'installer, un principe repris par Ubuntu ou Mandriva.

Une autre entrée sympathique est le test mémoire. Beaucoup de dysfonctionnements d'un ordinateur sont souvent à tort associés au système d'exploitation : écrans bleus de Windows ou kernel panic sous Linux. Il se peut même que Linux soit instable sur quelques PC. Or bien souvent la défaillance du système est due à un problème matériel dont le plus courant est un problème sur l'une (ou plus) des barrettes mémoire. L'outil memtest+ permet d'effectuer un diagnostic de celles-ci via des tests poussés (mais longs). L'idéal en cas de crashes à répétition est de lancer ce test en conservant uniquement une seule barrette, et de tester les barrettes les unes après les autres.

En informatique comme ailleurs, la qualité a un coût généralement élevé. Fuyez les composants « noname » ou génériques et privilégiez la marque pour tous vos composants, surtout pour la mémoire (Corsair, Gskill, Kingston, PQI, etc.), les cartes mères (Gigabyte, Asus, DFI, etc.) et les alimentations (Enermax, Fortron, etc.). Assurez-vous auprès des revendeurs et documentations techniques de la compatibilité entre vos composants. L'auteur a vu des barrettes de grande qualité refuser de fonctionner sur certaines cartes mères, et une webcam pourtant inactive faire planter la 3D d'une carte graphique...

Le mode rescue (secours) est un grand classique permettant de démarrer sur le support et de disposer ensuite, en ligne de commande, des outils pour réparer votre système.

Avant même l'installation, passez la langue en français via la touche [F2], et éventuellement modifiez la résolution du mode vidéo avec la touche [F3] en fonction de votre écran (par exemple 1280x1024 sur de nombreux LCD 4/3). Sélectionnez ensuite **Installer** et appuyez sur la touche [Entrée].

Notez que vous pouvez commuter l'écran en mode texte en appuyant sur la touche [Echap]. Dans ce cas vous arrivez sur un écran classique de type Grub très proche de celui de l'installateur Debian. Vous pouvez aussi installer openSUSE en mode texte en sélectionnant un mode de ce type avec [F3].

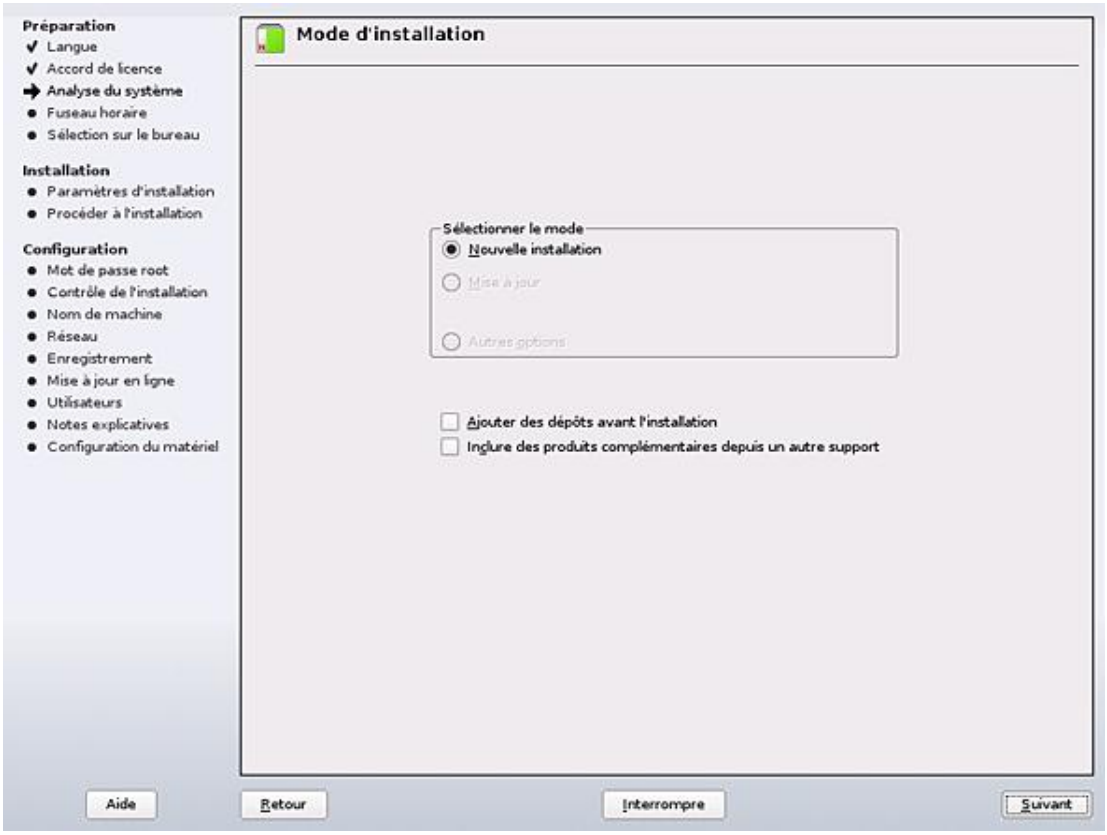
---

3. Choix de base

openSUSE est composé de produits libres ou non. Vous devez accepter la licence avant de continuer. Vous remarquerez quelque chose d'important tout au long de l'installation et après :

- l'installateur et le centre de configuration, YaST, ont la même interface ;
- une aide est systématiquement proposée via le bouton **Aide** en bas à gauche ;
- Il y a deux modes de fonctionnement : simple et expert.

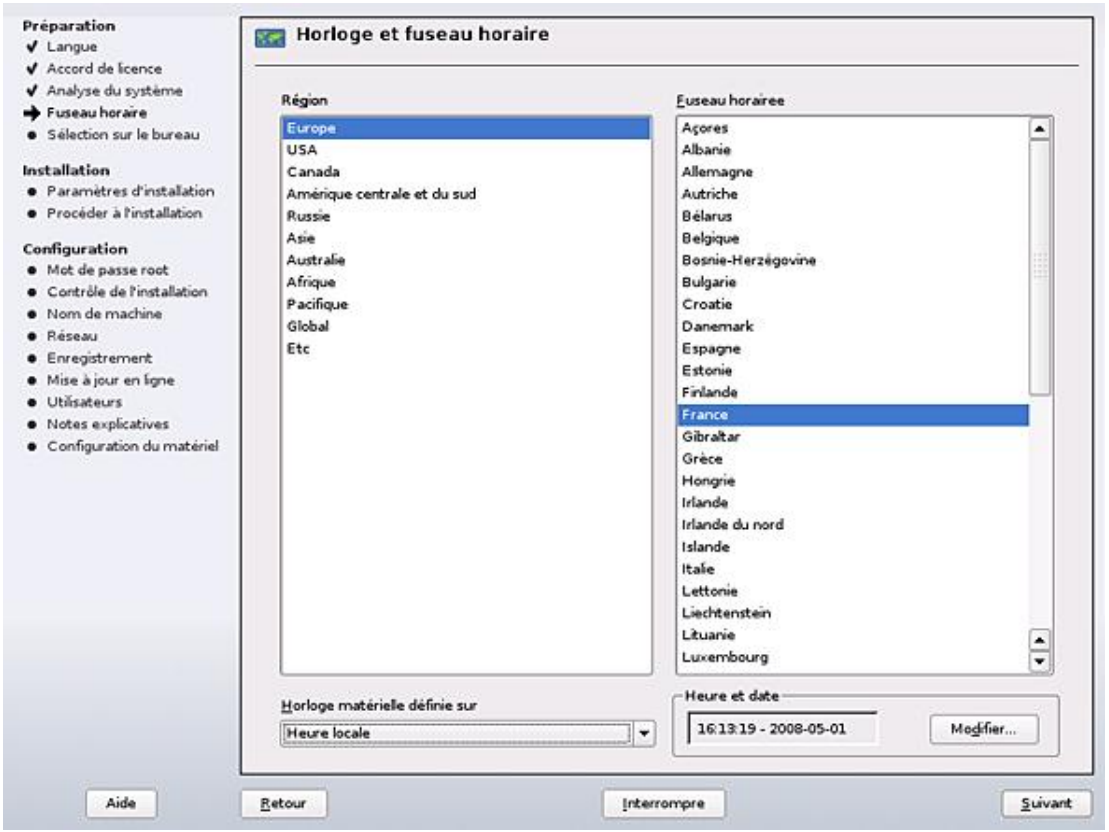
L'écran suivant vous demande le type d'installation souhaité.



YaST, mode d'installation

Vous pouvez dès cette étape proposer d'autres dépôts de packages logiciels pour openSUSE, mais pour éviter de surcharger inutilement l'installation, il est préférable de le faire une fois celle-ci terminée. Si vous avez déjà une version d'openSUSE installée l'installateur vous propose d'effectuer une mise à jour ou, via le bouton **Autres options**, de démarrer ou réparer le système. Sélectionnez **Nouvelle installation** et cliquez sur **Suivant**.

L'étape suivante est classique : il s'agit de régler le fuseau horaire et l'heure. Faites cependant attention à bien choisir sur quoi l'horloge matérielle est réglée. De même faites attention, en cas de double boot Windows/Linux, aux conflits : notamment aux passages entre l'heure d'été et l'heure d'hiver, vous risquez d'obtenir des décalages.



Cliquez sur **Suivant**.

4. Type de bureau

openSUSE n'est pas une distribution axée sur un type d'environnement bureautique par défaut (comme Debian et Ubuntu sous Gnome, Mandriva sous KDE, etc.) et supporte les deux environnements (voire plus avec XFCE) sur un pied d'égalité. Vous pouvez installer les deux. Cependant sur cette étape vous devez choisir l'environnement de bureau principal, celui qui sera utilisé par défaut lors des connexions.

Ce choix n'est pas définitif : dans les étapes suivantes vous pourrez choisir un autre environnement, et sur l'écran du gestionnaire de sessions vous pourrez aussi changer d'environnement (s'il est installé) à la volée.

Les deux environnements proposés par défaut sont Gnome et KDE :

- Gnome est l'environnement de bureau originel du projet GNU basé sur les bibliothèques GTK et très évolué. Ses dernières versions ont largement compensé leur retard sur d'autres projets. Il est très simple, beaucoup d'options complexes étant masquées, laissant seuls paraître les choix évidents, permettant ainsi au débutant de ne pas se perdre. Il dispose d'outils puissants : le gestionnaire de fichiers Nautilus, la messagerie Evolution, le navigateur Firefox sont basés sur les mêmes widgets et donc parfaitement intégrés (tout au moins visuellement) à Gnome.

- KDE est historiquement le premier vrai environnement bureautique fonctionnel sous Linux. Il a évolué très vite et ses versions actuelles sont très intuitives et puissantes, proposant des fonctionnalités qu'on ne retrouve pas sur d'autres environnements, y compris sur d'autres OS propriétaires. Si vous venez de Windows et que vous êtes habitué à modifier votre environnement, KDE est pour vous.

Vous pouvez faire fonctionner des applications de n'importe quel environnement dans n'importe quel autre environnement.

Gnome ou KDE ?

Sélectionnez votre environnement, par exemple **KDE**, et cliquez sur **Suivant**.

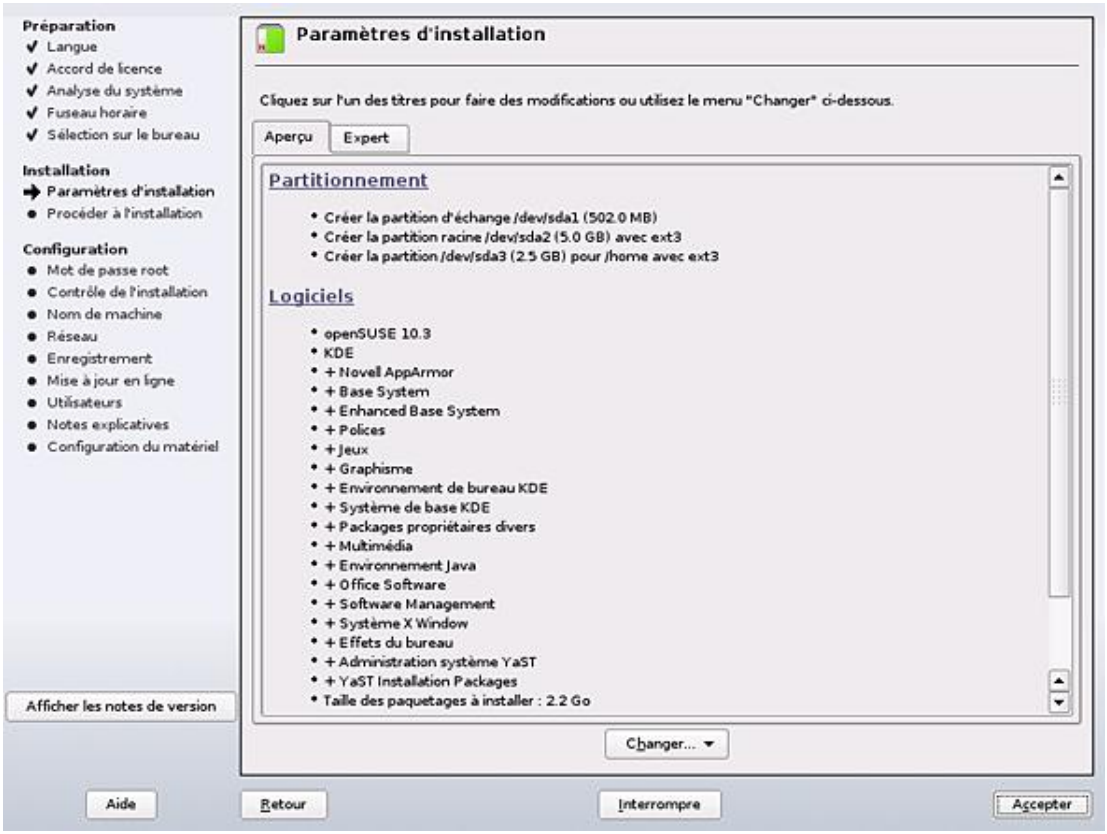
5. Paramètres d'installation

L'écran des paramètres d'installation est le plus important. C'est là notamment que vous avez accès à l'onglet du mode **Expert**, qui n'a d'expert que le nom puisqu'il étend simplement le nombre d'options proposées (bootloader, etc.). Restez sur l'onglet **Aperçu**.

Dans ce mode, vous disposez d'informations sur les deux plus importants choix d'installation qui sont le partitionnement des disques et le choix des packages logiciels. Le partitionnement a déjà été présenté lors de l'installation Debian. La logique de openSUSE est la même : il tente de déterminer les meilleurs choix possibles pour vous et celui-ci est généralement la création de trois partitions : la racine « / », les données personnelles « /home » et un espace de swap. C'est exactement la même chose que l'un des choix proposés par Debian et si vous comparez seule la taille varie, chacune des distributions disposant de ses propres méthodes de calcul.

Le choix des logiciels dépend en partie du choix du bureau. Comme vous avez sélectionné KDE, vous remarquez que les choix par défaut sont prédéfinis : environnement de bureau KDE, système de base KDE, etc.

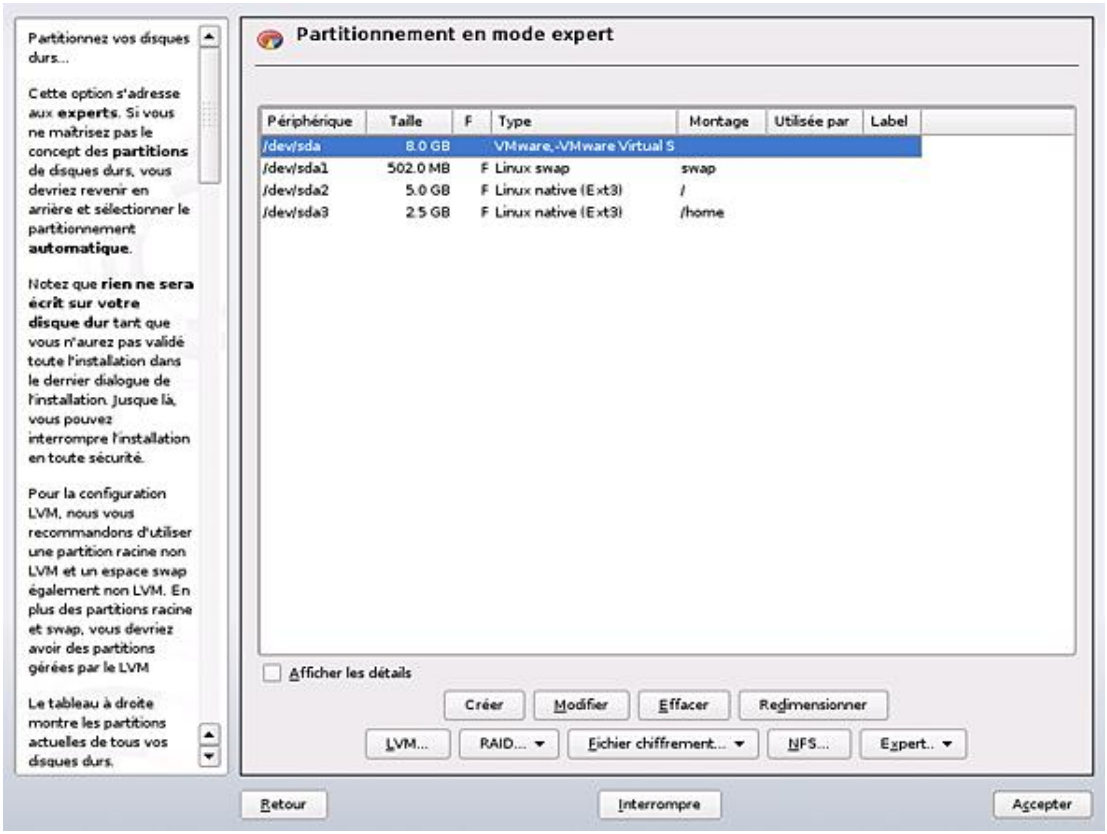
Vous pouvez modifier ces choix, bien évidemment. En cliquant sur les titres des sections, vous accédez aux outils associés : partitionnement et gestion des logiciels.



6. Partitionnement

L'outil de partitionnement de openSUSE basé sur YaST est d'une très grande simplicité, pourtant appelé partitionnement en mode expert. Le cadre central vous donne la liste des disques (/dev/hdX, /dev/sdX) puis en dessous la liste des partitions associées (/dev/hdxy, /dev/sdxy), la taille, si elle sera formatée (F), le type, le point de montage, etc. Vous pouvez créer, modifier, supprimer et redimensionner les partitions. Le même outil permet de créer des volumes RAID, LVM ou même les deux en même temps, de créer des partitions chiffrées ou des points de montage NFS.

Il est possible de redimensionner tout type de partition, y compris les partitions NTFS. Dans ce cas, il est préférable de vous rendre au préalable sous Windows et de faire une défragmentation et une analyse complète du disque. Pour tout redimensionnement, cela implique qu'il reste de la place sur la partition pour déplacer les blocs correspondants.



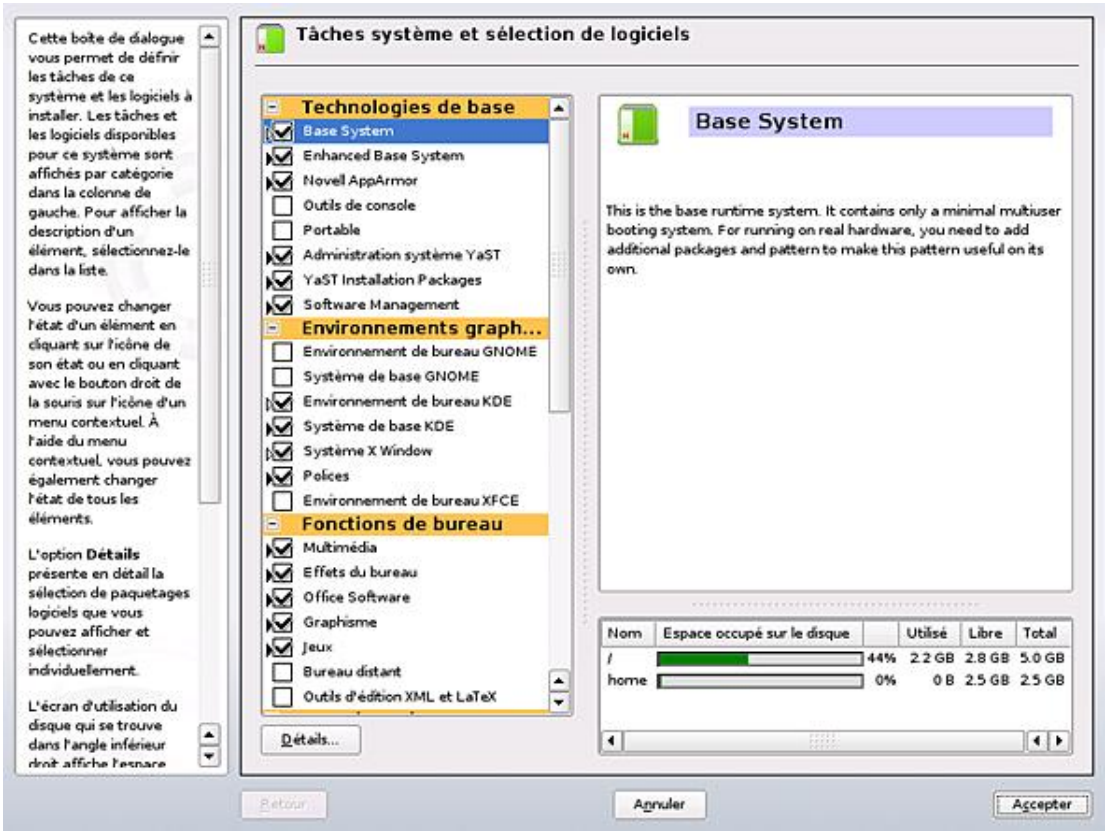
7. Installation des logiciels

Le composant d'installation de logiciels est comme toujours sous openSUSE et YaST le même avant et après l'installation. Les dernières versions ont apporté la gestion des meta-packages c'est-à-dire le regroupement par thèmes des paquets logiciels. Dans l'écran par défaut, vous disposez d'une arborescence basée sur ce principe, et vous ne voyez pas le détail des packages associés.

Si vous souhaitez rester sous openSUSE pour la suite de ce livre, pensez à installer (en cochant) les packages de développement du noyau, ainsi que quelques outils serveurs comme les partages Windows, NFS, et quelques outils serveurs comme DNS ou Apache.

Si vous cliquez sur le bouton **Détails** vous retournez à un affichage classique pour ceux qui ont connu les anciennes versions, où la liste de gauche fournit toujours une arborescence par type de packages (jeux, services, développement, etc.) tandis que la liste de droite fournit la liste des packages associés. Notez qu'un package peut être présent dans plusieurs meta-packages du fait de la gestion des dépendances.

Validez votre choix par le bouton **Accepter**. Faites de même dans l'écran des paramètres d'installation.



Choix de logiciels à installer

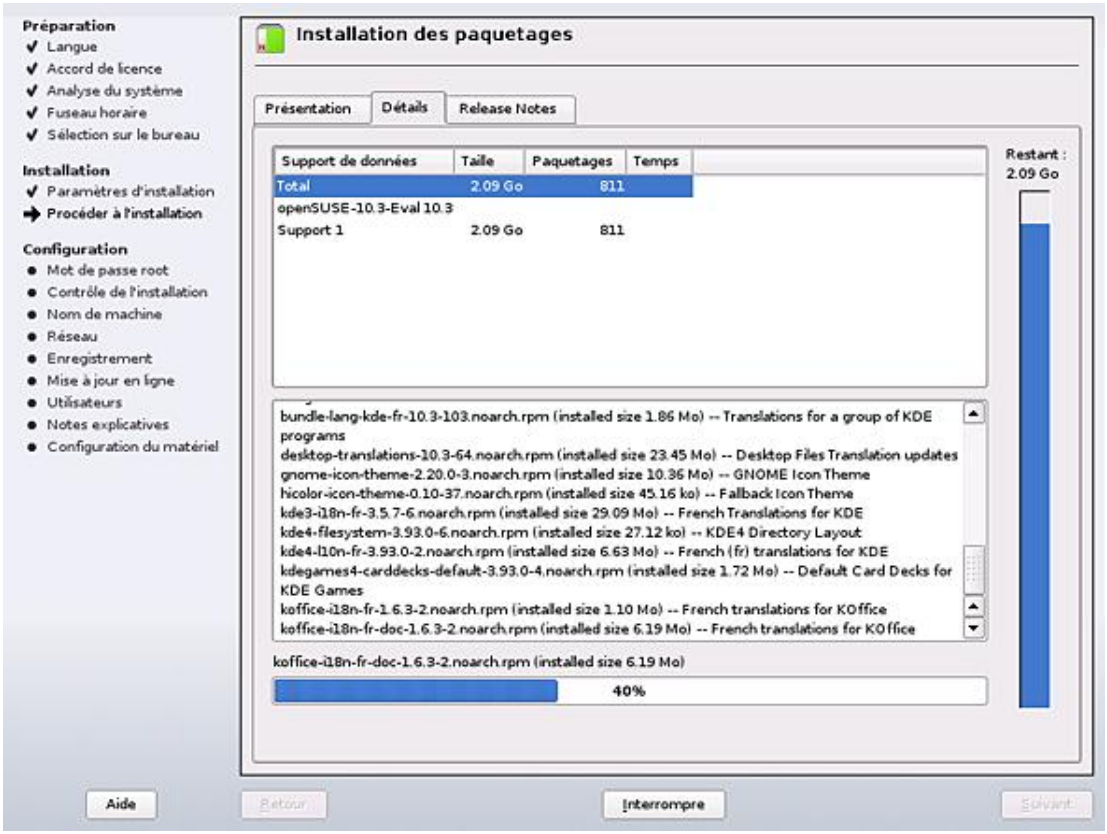
8. Installation en cours

Après avoir validé les paramètres d'installation, YaST installe la distribution sur vos partitions. Les partitions sont tout d'abord créées, puis formatées (ce terme est incorrect, car c'est le système de fichiers qui est écrit sur la partition).

Les différents packages sont ensuite récupérés (ici sur le support de type DVD) puis décompressés et installés. La barre de progression verticale à droite vous donne une estimation de la taille et du temps restant avant la fin de l'installation. La barre horizontale inférieure indique l'état d'installation du package en cours.

Trois onglets vous permettent de passer le temps en assistant à une petite présentation sous forme de diapositives (onglet **Présentation**), de lire les notes de version (onglet **Release Notes**) ou de voir le détail de l'installation (onglet **Détails**). Sur ce dernier point, vous voyez en temps réel les packages installés et les différents supports associés.

Ne négligez pas la lecture des notes de version. Celles-ci contiennent souvent des informations de très haute importance, comme le passage à la libata pour le support des disques, quelques points importants sur le support du matériel, etc.



Une fois l'intégralité des packages installés, vous devez cliquer sur le bouton **Suivant** en bas à droite. Vous devez ensuite choisir un mot de passe administrateur root, selon le même principe que lors de l'installation Debian, avec les mêmes contraintes.

9. Configuration du réseau

L'installation du réseau est une étape importante, notamment pour l'installation des mises à jour. Elle se fait en trois étapes :

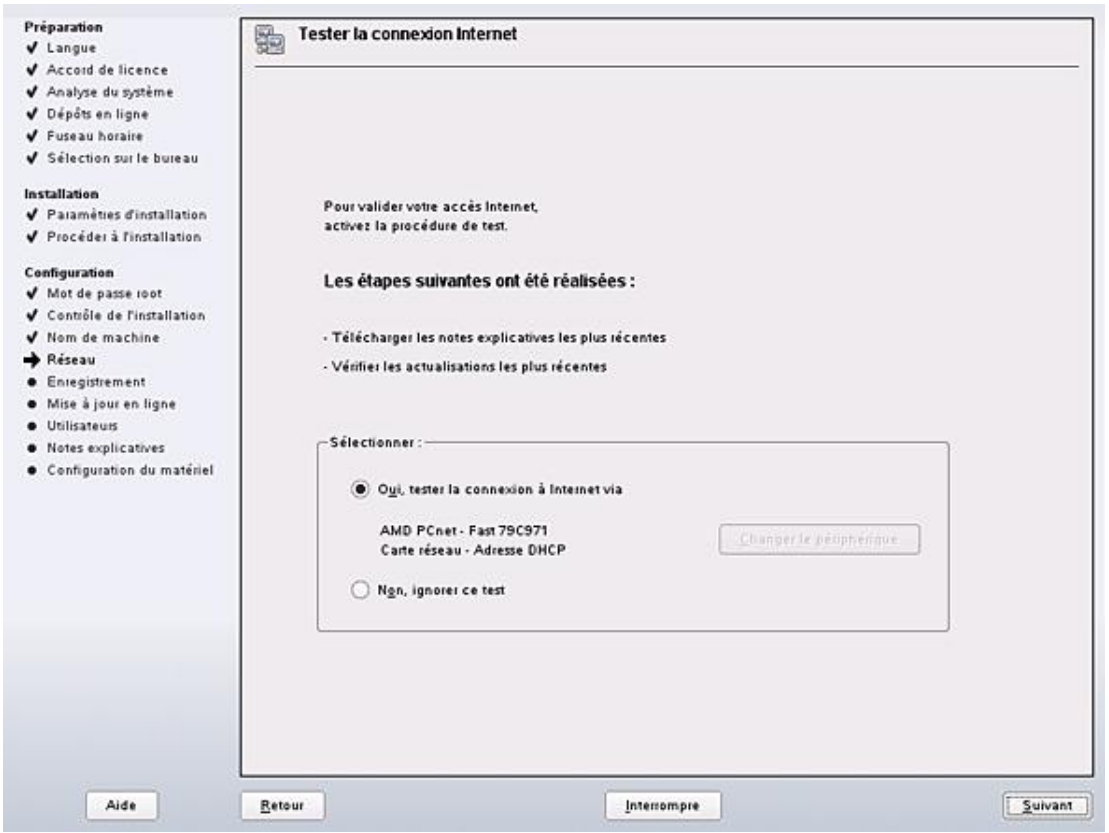
- Configuration du nom d'hôte et du domaine.
- Configuration globale du réseau : cartes et adaptateurs (Ethernet, Wi-Fi, Modems RTC et ADSL), adressage, firewall, proxy, etc.
- Test de la configuration.

L'installation de test dispose d'une unique carte qui est configurée via le protocole DHCP. Il n'y a donc quasiment rien à modifier. Si vous êtes déjà situé derrière un pare-feu, vous pouvez le désactiver.

Évitez de désactiver le support IPv6 durant l'installation, ce qui semble être la cause de certains problèmes par la suite. Vous pouvez procéder à cette étape après l'installation, car elle nécessitera probablement un redémarrage.

Cliquez sur **Suivant** quand vos interfaces sont configurées. Le réseau est démarré. L'étape suivante consiste à tester votre connexion à Internet, étape presque indispensable notamment pour pouvoir accéder aux mises à jour. Si vous avez configuré plusieurs interfaces réseaux, vous pouvez sélectionner celles vous permettant d'accéder à Internet. Vous pouvez aussi ignorer le test.

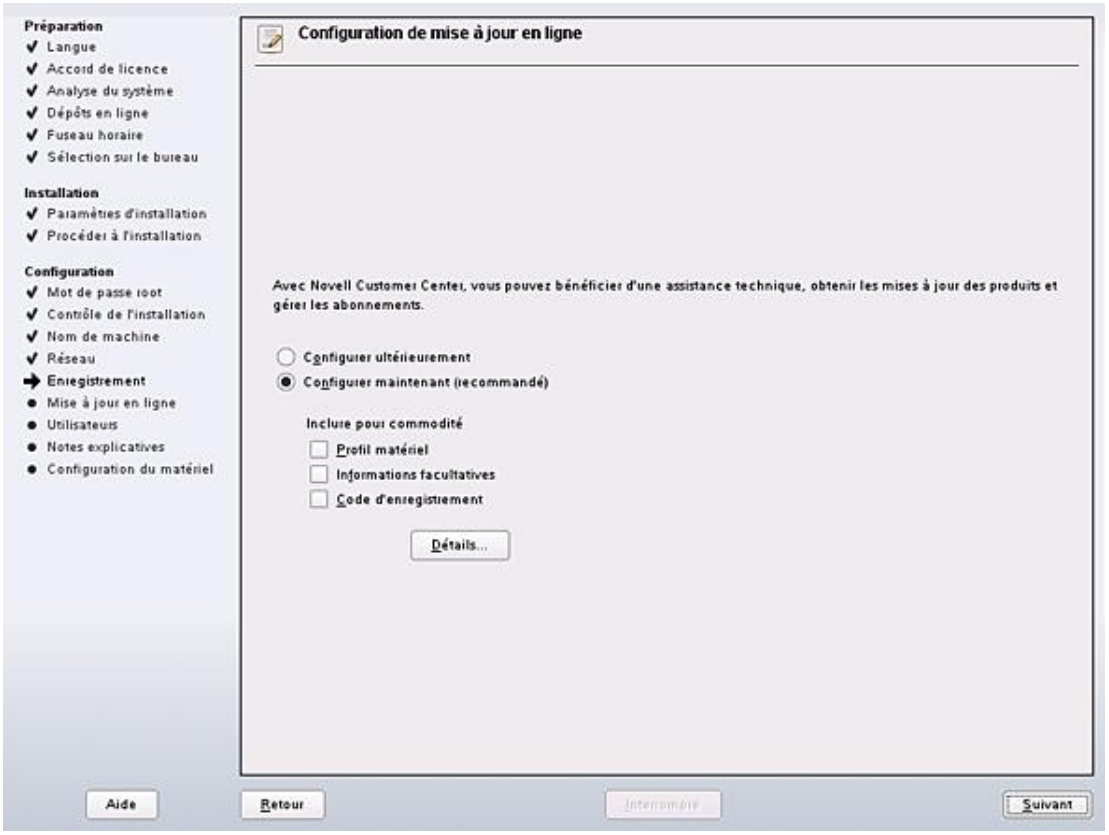
Si vous ignorez le test, ou qu'aucune connexion à Internet n'est disponible, l'étape suivante de mise à jour du système ne sera pas accessible. Vous devrez passer par le module d'installation de dépôt de mise à jour de YaST après l'installation pour accéder aux mises à jour.



10. Mise à jour du système

Si votre connexion Internet est correctement configurée, vous pouvez maintenant configurer votre source de mise à jour. Cliquez sur **Configurer maintenant**. YaST recherche un miroir sur Internet pour configurer une source.

Vous pouvez en profiter pour fournir auprès du support openSUSE des informations sur votre installation et sur votre machine et un éventuel code d'enregistrement pour le support (rarement utilisé avec les versions openSUSE, le support étant plutôt communautaire). Aucune information personnelle ne circule, et tout se fait sur votre autorisation.

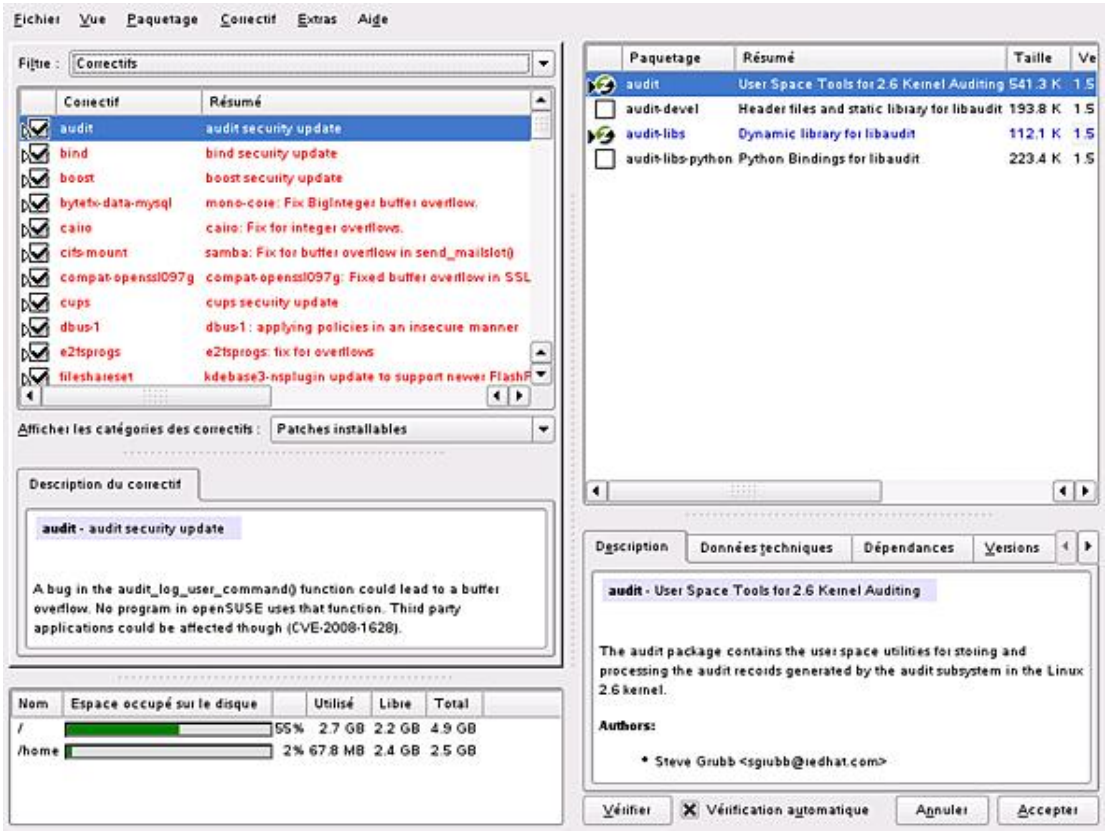


Si cette étape s'est déroulée correctement, vous pouvez demander à ce que la mise à jour de votre installation s'effectue dès à présent. Cette étape va télécharger les packages récents (mises à jour de sécurité et correction de bug notamment) et les installer.

Il se peut que le processus de mise à jour redémarre plusieurs fois. Il n'est pas étonnant de voir que les composants gérant les mises à jour (le backend zypp sous openSUSE) sont eux-mêmes mis à jour. Dans ce cas le processus de mise à jour redémarre pour réutiliser les nouveaux composants.

En principe une installation de openSUSE n'a pas besoin d'un redémarrage de l'ordinateur et s'effectue d'un trait : aucun reboot entre le démarrage sur le support et le premier login. Ceci s'appuie sur le fait que le noyau (et donc les pilotes) installé est le même que sur le support d'installation. Or lors d'une mise à jour, il se peut qu'un nouveau noyau, composant très critique, soit installé. Dans ce cas, le processus d'installation doit redémarrer l'ordinateur pour recharger et gérer les nouveaux modules du noyau (dont les pilotes) afin de pouvoir configurer le matériel par la suite.

L'installation des mises à jour utilise la même interface que l'installation de tout autre package. Les correctifs majeurs sont en rouge : ils corrigent un bug important présentant un risque de sécurité. Si vous descendez dans la liste, vous trouverez aussi des mises à jour qui ne sont pas critiques, ou qui n'ont pas forcément besoin d'être installées pour garantir le bon fonctionnement de Linux. C'est le cas par exemple de l'installation des polices (fontes) TrueType fournies par Microsoft. Elles apportent cependant un confort supplémentaire à l'utilisation.



Choix et installation des mises à jour

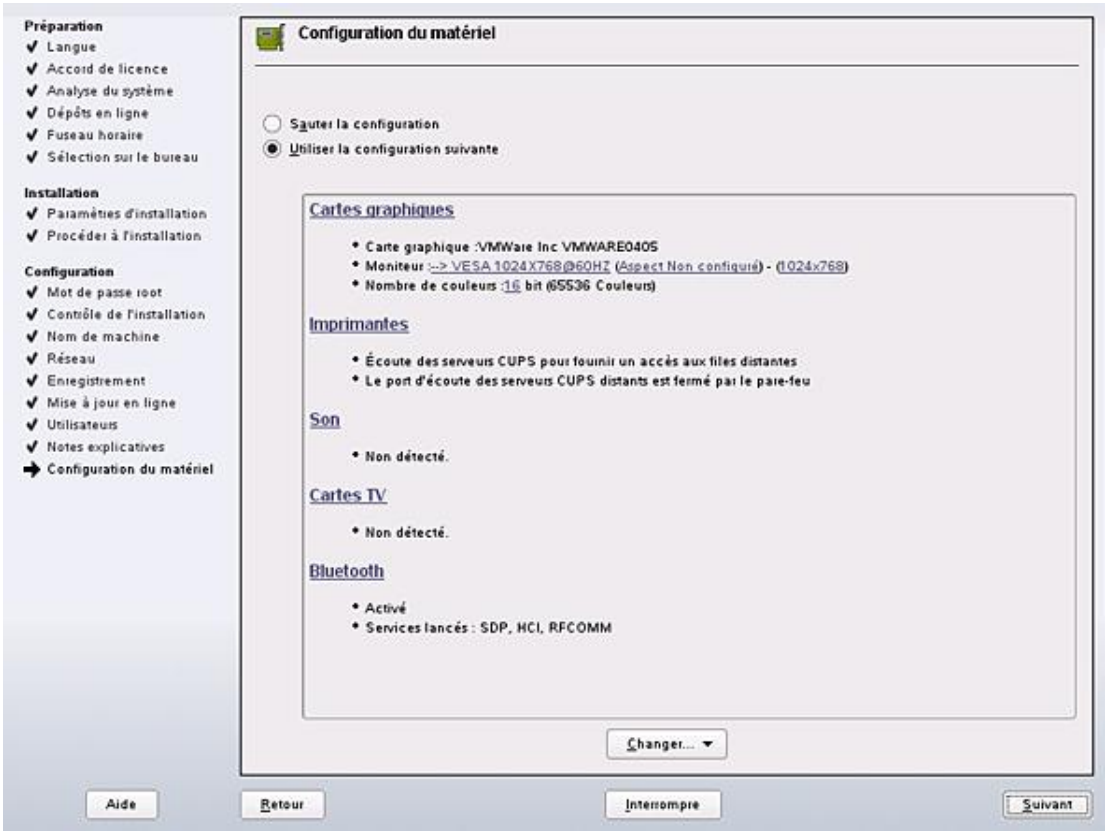
11. Gestion des utilisateurs

L'étape suivante consiste à rajouter des utilisateurs. Outre les utilisateurs locaux, vous pouvez vous raccorder à une base LDAP, un serveur NIS et même un domaine Windows pour l'authentification.

Vous pouvez créer un seul ou plusieurs utilisateurs en cliquant sur le bouton **Gestion utilisateurs**. Vous pouvez aussi faire en sorte d'activer l'autoconnexion de l'utilisateur afin d'éviter la saisie des logins et mots de passe. N'utilisez pas cette possibilité si plusieurs utilisateurs existent. De même ce n'est pas forcément une bonne idée de router les messages à destination de l'administrateur vers un compte utilisateur, sauf si celui-ci a comme vocation d'être utilisé comme passerelle vers les tâches d'administration.

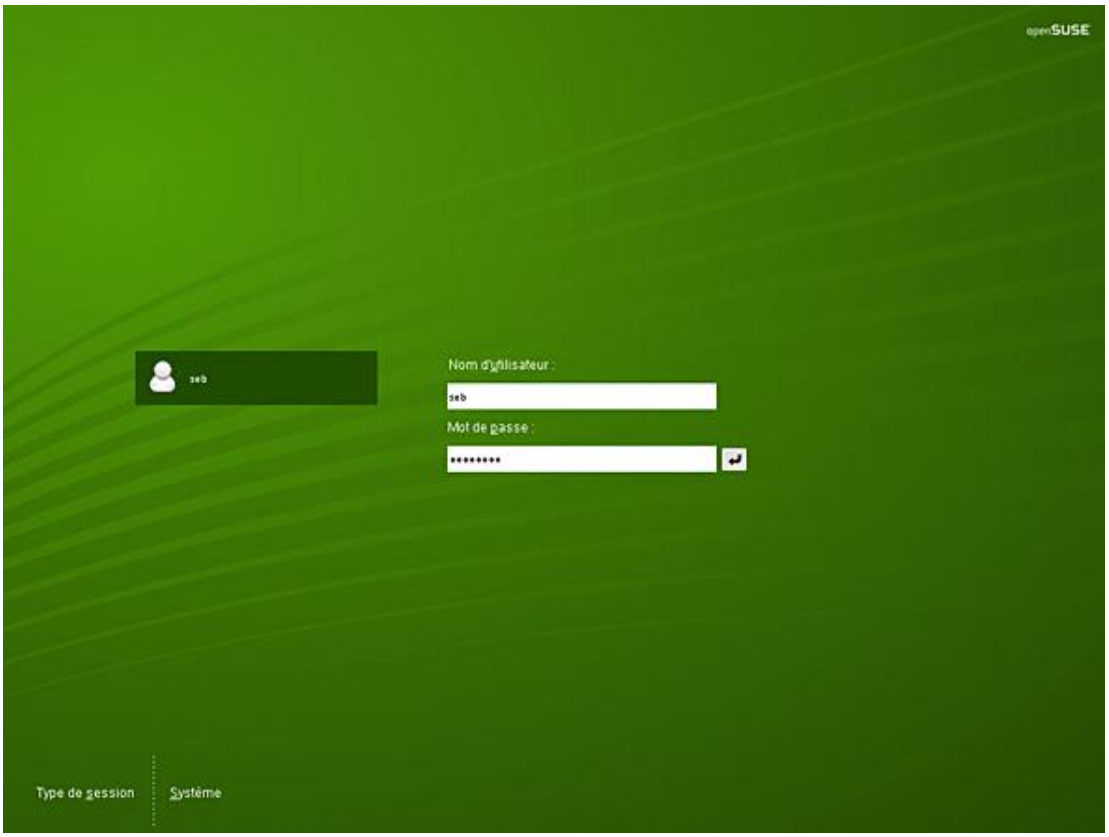
12. Configuration du matériel

La configuration du matériel est la dernière étape de l'installation de openSUSE. Cette distribution est très réputée pour la détection et le support du matériel qu'elle propose, dans la limite des pilotes disponibles. Il est fort probable que vos uniques actions consistent à modifier le moniteur ou plutôt la résolution par défaut, et peut-être (si elles sont reliées par réseau ou port parallèle) les imprimantes. Les scanners devront être configurés après la fin de l'installation.



Cliquez sur **Suivant**. La configuration du matériel est enregistrée. Un écran vous informe enfin de la fin de l'installation. Il n'y a pas de redémarrage : l'installateur se termine et il est possible de commencer directement à travailler.

La dernière capture montre le gestionnaire de sessions KDM dont le thème a été modifié aux couleurs de la distribution.



L'installation a réussi, KDM attend votre connexion.

Red Hat Package Manager

1. Notion de package

Contrairement à d'autres systèmes d'exploitation, il n'est pas courant sur Linux et Unix en général de disposer de logiciels fournis avec un programme l'installation interactif (pas de install.exe). Certains éditeurs proposent des scripts d'installation et bien souvent ceux-ci se contentent de décompresser et de désarchiver quelques fichiers.

Avec Linux, il est très classique de disposer des divers produits, outils, mises à jour, etc. sous forme de paquetages (packages). Un package est un fichier (parfois gros) qui contient le produit à installer et des règles. Ces règles peuvent être multiples :

- Gestion des dépendances : le produit ne pourra être installé que si les produits qu'il utilise lui-même sont déjà présents.
- Pré-installation : des actions sont à prévoir avant de pouvoir installer le produit (changer des droits, créer des répertoires, etc.).
- Post-installation : des actions sont à prévoir après l'installation du produit (paramétrage d'un fichier de configuration, compilation annexe, etc.).

Sur Red Hat, Fedora, SuSE, Mandriva et quelques autres distributions le format de package par défaut est le **RPM** (*Red Hat Package Manager*). Sous Debian, Knoppix, Kaella, Ubuntu, c'est le format **DPKG** (*Debian Package*). Outre le format, ce sont surtout les outils qui les différencient.

Le fait de disposer des informations de dépendances permet d'obtenir des outils performants qui peuvent seuls les résoudre en cascade. En installant un package, l'outil pourra installer toutes les dépendances nécessaires. On peut parfois spécifier plusieurs emplacements (repositories) pour ces packages, soit locaux (disque dur, CD-Rom, DVD, etc.) soit distants (http, ftp, etc.).

Il faut toujours utiliser un package prévu pour sa distribution quand il existe. Si ce n'est pas le cas, il est parfois possible d'utiliser un package d'un produit concurrent ou de recompiler le produit soi-même.

Les mises à jour d'un système Linux utilisant un système de packaging sont très simplifiées. Pour passer d'une version d'un produit à un autre, il suffit de récupérer le package du produit en version supérieure et de l'installer. Toutes les mises à jour sont sous cette forme. Depuis peu, il existe un format de **delta-rpm** qui ne fournit dans le package que les différences d'une version à une autre. Mais il est toujours possible d'utiliser un package complet.

2. Le gestionnaire RPM

RPM est un gestionnaire de packages inventé par Red Hat puis utilisé massivement par de nombreuses autres distributions. Il simplifie fortement la distribution, l'installation, la mise à jour et la suppression des logiciels. Il se base sur des commandes (ex : **rpm**), une base de données locale et des packages au format rpm (extension rpm).

La base de données est située dans **`/var/lib/rpm`**. Toutes les informations concernant les logiciels installés, leurs versions, leurs fichiers et droits, et leurs dépendances y sont précisées. Sauf gros problème, il ne faut JAMAIS modifier cette base à la main. Il faut utiliser les outils RPM.

Chaque logiciel est fourni sous forme de package au format RPM. Le rpm répond à une nomenclature précise.

```
nom-version-edition.architecture.rpm
```

par exemple :

```
php-4.1.2-2.1.8.i586.rpm
```

L'édition est un identifiant de version du package RPM propre à l'éditeur. Ici c'est la version 2.1.8 du package PHP version 4.1.2. L'architecture est i586 (Intel Pentium). On peut aussi trouver i386, i686, x86_64 (64 bits), ppc64, s390x ou noarch. Un package noarch ne contient pas de programmes ou bibliothèques binaires mais du code indépendant comme des scripts, de la documentation, des images, du son, de la vidéo, etc.

3. Installation, mise à jour et suppression

Vous installez un package rpm avec le paramètre `-i`.

```
rpm -i php-4.1.2-2.1.8.i586.rpm
```

Comme il est possible d'utiliser des caractères de substitution (**`rpm -i *.rpm`**), vous pouvez afficher le nom du package en cours d'installation avec le paramètre `-v`. Le paramètre `-h` affiche des caractères # pour indiquer la progression de l'installation. L'installation ne fonctionnera pas si les dépendances ne sont pas résolues.

La mise à jour d'un produit vers une version supérieure depuis un package se fait avec le paramètre `-U`. Dans ce cas tous les fichiers sont mis à jour par ceux de la nouvelle version : les anciens sont supprimés et remplacés par les nouveaux. Les anciens fichiers de configuration sont sauvés avec l'extension **`.rpmsave`**. Si le package n'était pas installé, la mise à jour joue le rôle d'installation. Attention avec ce paramètre : il installe le package même si une version précédente n'était pas installée.

```
rpm -Uvh php-4.1.3-1.i586.rpm
```

La mise à jour est aussi possible avec `-F`. Mais si le package n'était pas installé, il ne le sera pas non plus lors de la mise à jour contrairement à `-U`. Ainsi, si vous disposez de tous les packages de mise à jour du système et que vous ne souhaitez mettre à jour que ceux qui sont réellement installés, alors vous pouvez taper :

```
rpm -Fvh *.rpm
```

La suppression s'effectue avec le paramètre `-e`. Attention cependant, c'est le nom du package installé qui doit être passé en paramètre et pas le nom du fichier de package.

```
rpm -e php
```

Plusieurs options supplémentaires sont possibles :

- **`--force`** : en cas de conflit avec un autre package (le cas le plus courant est celui où deux packages proposent le même fichier au même endroit), cette option force tout de même l'installation.
- **`--nodeps`** : si le package refuse de s'installer à cause d'un problème de dépendances, cette option forcera l'installation. Il arrive parfois que cette erreur se produise quand la dépendance en question a été installée autrement que depuis un package rpm (ex : compilation, binaire copié à la main).

Dans la mesure du possible, évitez d'utiliser ces options qui peuvent casser certaines dépendances, notamment si vous utilisez un système de meta-package (yum, apt, zypper, urpmi, etc.).

4. Cas du noyau

L'installation ou la mise à jour d'un noyau est un cas particulier. En effet, la mise à jour supprime l'ancienne version. Le noyau est un composant très critique du système. S'il devait être avéré que le système ne fonctionne plus (ou mal) avec le noyau mis à jour, il faudrait donc réinstaller un ancien noyau depuis le support d'installation. Aussi la procédure est la suivante :

- Installation du nouveau noyau avec le paramètre `-i`, il sera rajouté au système.
- Redémarrage et test de vos logiciels et périphériques avec le nouveau noyau.
- S'il fonctionne correctement, suppression éventuelle de l'ancien noyau avec `-e`.
- Édition de **`/boot/grub/grub.conf`** et modification de la ligne **Default** pour démarrer par défaut sur le nouveau noyau.

5. Requêtes RPM

La base de données RPM peut être interrogée facilement avec le paramètre `-q` suivi de plusieurs options.

-a : liste de tous les packages installés.

-i : informations générales (le résumé) du package.

-l : liste des fichiers installés.

-f nom : trouve le package qui contient le fichier donné.

-p nom : la recherche s'effectue dans le fichier de package donné.

--requires : dépendances du package.

--provides : ce que fournit le package.

--scripts : scripts exécutés à l'installation et la suppression.

--changelog : l'historique du package.

```
$ rpm -qilp libjpeg-6.2.0-738.i586.rpm
Name        : libjpeg                      Relocations: (not relocateable)
Version     : 6.2.0                             Vendor: SUSE LINUX Products
GmbH, Nuernberg, Germany
Release     : 738                          Build Date: Sat Mar 19 20:07:55
2005
Install date: (not installed)              Build Host: d121.suse.de
Group       : System/Libraries             Source RPM: jpeg-6b-738.src.rpm
Size        : 125804                          License: BSD, Other License(s),
see package
Signature   : DSA/SHA1, Sat Mar 19 20:12:06 2005, Key ID a84edae89c800aca
Packager    : http://www.suse.de/feedback
URL         : http://www.ijg.org/
Summary     : JPEG libraries
Description :
The libraries (static and dynamic) for the jpeg-graphics format. The
sources are contained in the jpeg source package.



Authors:
--------
    Rob Hooft <hooft@EMBL-Heidelberg.DE>
    Michael Mauldin <mlm@cs.cmu.edu>
/usr/lib/libjpeg.so.62
/usr/lib/libjpeg.so.62.0.0

$ rpm -qp libjpeg-6.2.0-738.i586.rpm -requires
/sbin/ldconfig
/sbin/ldconfig
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(CompressedFileNames) <= 3.0.4-1
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1.3)
rpmlib(PayloadIsBzip2) <= 3.0.5-1

$ rpm -qi php
Name        : php                          Relocations: (not relocateable)
Version     : 4.1.2                             Vendor: Red Hat, Inc.
Release     : 2.1.8                         Build Date: mer 14 jui 2004
11:24:16 CEST
Install date: lun 27 jun 2005 19:36:32 CEST      Build Host: porky.build.
redhat.com
Group       : Development/Languages        Source RPM: php-4.1.2-2.1.8.src.
rpm
Size        : 3843552                         License: The PHP License,
version 2.02
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://www.php.net/
Summary     : The PHP HTML-embedded scripting language. (PHP: Hypertext
Preprocessor)
Description :
```

```
PHP is an HTML-embedded scripting language. PHP attempts to make it
easy for developers to write dynamically generated webpages. PHP also
offers built-in database integration for several commercial and
non-commercial database management systems, so writing a
database-enabled webpage with PHP is fairly simple. The most common
use of PHP coding is probably as a replacement for CGI scripts. The
mod_php module enables the Apache Web server to understand and process
the embedded PHP language in Web pages.

$ rpm -qa | grep php
php-ldap-4.1.2-2.1.8
php-imap-4.1.2-2.1.8
asp2php-0.75.17-1
php-4.1.2-2.1.8

$ rpm -qf /usr/bin/passwd
passwd-0.68-1.2.1
```

6. Vérification des packages

Il est possible qu'après l'installation d'un package, un ou plusieurs des fichiers installés aient été altérés (changement de droit, de propriétaire, édition, suppression, etc.). Comme la base RPM contient toutes les informations nécessaires, on peut demander une vérification avec le paramètre `-V`.

```
$ rpm -V php
S.5....T c /etc/php.ini
```

Un point signifie qu'une étape de vérification est OK. Sinon :

- **S** : la taille du fichier a été modifiée.
- **5** : la somme MD5 ne correspond plus.
- **T** : la date de modification n'est plus la même.
- **U** : le propriétaire a été modifié.
- **G** : le groupe a été modifié.
- **L** : le lien symbolique a été modifié.
- **M** : les permissions ou le type du fichier ont été modifiés.
- **D** : le périphérique a été modifié (major/minor).

Le **c** indique qu'il s'agit d'un fichier de configuration.

Les fichiers de packages RPM sont très souvent signés par l'éditeur de la distribution de manière à en garantir l'intégrité. On peut vérifier l'intégrité d'un package avec une clé publique GPG mais il faut par avance avoir déjà chargé cette clé publique sur le système.

```
gpg --import RPM-GPG-KEY
rpm --import RPM-GPG-KEY
rpm --checksig libjpeg-6.2.0-738.i586.rpm
```

7. Les dépendances

Si vous utilisez les outils graphiques fournis par votre distribution, ceux-ci tenteront de résoudre les dépendances à votre place. La commande **rpm** seule ne le fait pas par défaut. Des outils complémentaires « frontend » comme **yast**, **apt** ou **yum** le font à sa place. La distribution Red Hat fournissait jusqu'aux versions 4 (RHEL) un outil appelé **rpmdb-**

redhat pour installer automatiquement les dépendances via rpm. Cela implique notamment le fait que tous les packages de la distribution doivent se trouver au même endroit (dans le même répertoire) et le système ne fonctionne qu'avec les packages officiels de Red Hat. On emploie le paramètre `--aid`.

```
$ rpm -ivh --aid libjpeg-6.2.0-738.i586.rpm
```

8. Mises à jour automatisées

Chaque distribution fournit maintenant un outil de mise à jour interactif ou automatisé. La openSUSE propose **YOU** (*Yast Online Update*), la Red Hat propose **up2date**. La version RHEL de Red Hat étant payante, l'accès aux mises à jour dépend d'un numéro de licence et d'une inscription à **RHN** (*Red Hat Network*). Les versions dérivées comme CentOS ne peuvent pas se mettre à jour via RHN mais proposent leur propre site distant de mise à jour.

YUM

YUM est aux fichiers rpm ce que APT est aux fichiers dpkg : un logiciel de gestion de packages. Il récupère les packages au sein de dépôts et gère les dépendances à votre place. YUM signifie *Yellow dog Updater Modified*. Il est principalement utilisé sur les distributions Redhat (les version Entreprise) et Fedora, mais peut être utilisé sur n'importe quelle distribution de type RPM, si les dépôts associés le supportent.

Les commandes et exemples suivants se basent sur un serveur Redhat Enterprise Linux 5. Le fichier de configuration est **`/etc/yum.conf`**.

1. Configuration des dépôts

Les dépôts sont placés soit dans le fichier de configuration principal, soit dans le répertoire **`/etc/yum.repos.d`**. Le format est le suivant :

```
[rhel5]
name=ES5
baseurl=ftp:/ftp.server.com/redhat/x86/ES5u3/Server/
gpgcheck=1
enabled=1
gpgkey=ftp://ftp.server.com/x86/ES5u3/RPM-GPG-KEY-redhat-release
```

Le dépôt se nomme (nom court) `rhel5`.

- **`name`** : le nom long du dépôt, détaillé.
- **`baseurl`** : l'URL du dépôt.
- **`gpgcheck`** : demande une vérification de la signature GPG du dépôt.
- **`enabled`** : si absent ou à 1, le dépôt est actif.
- **`gpgkey`** : chemin de la clé publique GPG.

Les URL des dépôts peuvent être locales (file://) ou distantes (http:// ou ftp://). Elles doivent pointer sur un répertoire contenant les informations de dépôts qui sont dans le dossier `repodata`.

En tenant compte des valeurs par défaut, un simple dépôt peut être déclaré ainsi :

```
[updates-rhel5]
name=UPDATES-RHEL5
baseurl=ftp://ftp.server.com/RPMS.rhel5_updates_x86
```

Attention cependant car la configuration de YUM peut modifier les valeurs par défaut. La section `[main]` de **`/etc/yum.conf`** peut ainsi contenir la ligne :

```
gpgcheck=1
```

Dans ce cas, vous devrez modifier la valeur `gpgcheck` à 0 dans les dépôts ne nécessitant pas de signature.

2. Utilisation des dépôts

a. Rafraîchir le cache

À chaque commande, YUM tente de rafraîchir ses données si le délai d'expiration a été dépassé. Ce délai peut être réduit ou étendu en modifiant la ligne `metadata_expire` du fichier de configuration.

```
metadata_expire=1h
```

Vous pouvez forcer la mise à jour du cache avec le paramètre makecache :

```
root@slyserver ~]# yum makecache
Updates-rhel5        |  951 B     00:00
other.xml.gz         |  94 kB     00:00
rhel5                | 1.3 kB     00:00
other.xml.gz         | 5.9 MB     00:00
rhel5-VT             | 1.3 kB     00:00
other.xml.gz         |  32 kB     00:00
updates-rhel5                                14/14
rhel5                                      2255/2255
rhel5-VT                                     35/35
Metadata Cache Created
```

L'autre possibilité est de forcer la suppression du cache afin qu'à la prochaine commande YUM celui-ci soit automatiquement reconstruit :

```
root@slyserver ~]# yum clean all
Cleaning up Everything
```

b. Lister les packages

Le paramètre `list` permet de lister les packages. Tous sont listés par défaut. Vous pouvez préciser une liste de packages, ou fournir des caractères jokers. Plusieurs options sont disponibles :

- **all** : c'est le cas par défaut : les packages installés sont listés en premier, puis les packages disponibles pour installation.
- **available** : les packages disponibles pour installation.
- **updates** : les packages pouvant être mis à jour.
- **installed** : les packages mis à jour.
- **obsoletes** : les packages du systèmes rendus obsolètes par des versions supérieures disponibles.
- **recent** : les derniers packages ajoutés dans les dépôts.

La commande suivante liste les noyaux disponibles :

```
[root@slyserver etc]# yum list available kernel\*

Available Packages
kernel-PAE.i686               2.6.18-128.el5            rhel5
kernel-PAE-devel.i686         2.6.18-128.el5            rhel5
kernel-debug.i686             2.6.18-128.el5            rhel5
kernel-debug-devel.i686       2.6.18-128.el5            rhel5
kernel-devel.i686             2.6.18-128.el5            rhel5
kernel-doc.noarch             2.6.18-128.el5            rhel5
kernel-xen.i686               2.6.18-128.el5            rhel5
kernel-xen-devel.i686         2.6.18-128.el5            rhel5
```

Le paramètre `info` retourne les informations détaillées d'un package. C'est l'équivalent du paramètre `-i` de la commande `rpm`. Ainsi, pour le package `mc` vous obtiendrez quelque chose de ce type :

```
[root@slyserver ~]# yum info mc
Installed Packages
Name       : mc
Arch       : i386
Epoch      : 1
Version    : 4.6.1a
Release    : 35.el5
Size       : 5.2 M
Repo       : installed
```

```
Summary    : Shell visuel et gestionnaire de fichiers en console
texte ergonomique
URL        : http://www.ibiblio.org/mc/
License    : GPL
Description: Midnight Commander est un shell visuel comparable à un
gestionnaire de fichiers, si ce n'est qu'il offre beaucoup plus de
fonctions. Il s'agit d'une application en mode caractère, mais
elle intègre aussi un support souris. Les fonctions les plus
agréables de Midnight Commander sont ses capacités à effectuer des
transferts FTP, afficher des fichiers tar et zip, et rechercher dans
les RPM des fichiers spécifiques.
```

c. Installer des packages

Passez le paramètre `install`, suivi des noms des packages à installer.

Voici un exemple d'installation du package `mc` :

```
[root@slyserver /etc]# yum install mc
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package mc.i386 1:4.6.1a-35.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================
 Package   Arch   Version          Repository  Size
================================================
Installing:
 mc        i386   1:4.6.1a-35.el5  rhel5       2.1 M

Transaction Summary
================================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 2.1 M
Is this ok [y/N]: y
Downloading Packages:
mc-4.6.1a-35.el5.i386.rpm                    2.1 MB     00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : mc                         [1/1]

Installed: mc.i386 1:4.6.1a-35.el5
Complete!
```

d. Mises à jour

Vérifiez la présence de mises à jour avec le paramètre `check-update` :

```
[root@slyserver /etc]# yum check-update
```

Si rien n'est retourné, c'est qu'aucune mise à jour n'est disponible.

Vous avez deux possibilités pour installer les mises à jour :

- **update** : mise à jour d'un package ou de tous si aucun package n'est précisé.

- **upgrade** : mise à niveau complète de la distribution : les packages vus comme obsolètes sont remplacés par ceux de la dernière version disponible.

Dans certains cas, pour le noyau par exemple, vous devrez éviter, lors d'une mise à jour, d'installer automatiquement certains packages. Dans ce cas, utilisez le paramètre `--exclude` :

```
[root@slyserver etc]# yum list -exclude=kernel\* update
```

Pour rendre permanente cette exclusion, mettez-la en dur dans le fichier de configuration en ajoutant une ligne comme ceci :

```
exclude=php* kernel*
```

e. Rechercher un package

Utilisez le paramètre `search`, suivi du ou des packages à rechercher dans les dépôts. Les caractères jockers sont autorisés.

```
root@slyserver etc]# yum search tomcat
============== Matched: tomcat ==========================
jakarta-commons-collections-tomcat5.i386 : Jakarta Commons
Collection dependency for Tomcat5
struts-webapps-tomcat5.i386 : Exemples d'applications Web struts
pour tomcat5
tomcat5.i386 : Moteur Servlet/JSP Apache, RI pour Servlet 2.4/JSP
2.0 API
tomcat5-admin-webapps.i386 : Applications Web d'administration pour
tomcat
tomcat5-common-lib.i386 : Bibliothèque nécessaire à l'éxecution du
containeur Tomcat Web
tomcat5-jasper.i386 : Compilateur JARs et scripts associés pour tomcat5
tomcat5-jasper-javadoc.i386 : Documentation Javadoc pour tomcat5-jasper
tomcat5-jsp-2.0-api.i386 : Implémentations de classes Jakarta
Tomcat Servlet et JSP
tomcat5-jsp-2.0-api-javadoc.i386 : Documentation Javadoc générée
pour tomcat5-jsp-2.0-api
tomcat5-server-lib.i386 : Bibliothèque nécessaire à l'éxecution du
containeur Tomcat Web
tomcat5-servlet-2.4-api.i386 : Implantation des classes Servlets
pour Jakarta Tomcat
tomcat5-servlet-2.4-api-javadoc.i386 : Documentation générée Javadoc
pour tomcat5-servlet-2.4-api
tomcat5-webapps.i386 : Applications Web pour Jakarta Tomcat
```

f. Supprimer un package

Pour supprimer un package, utilisez le paramètre `remove` :

```
[root@slyserver~]# yum remove mc
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package mc.i386 1:4.6.1a-35.el5 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package    Arch    Version          Repository        Size
================================================================
Removing:
 mc         i386    1:4.6.1a-35.el5  installed         5.2 M

Transaction Summary
================================================================
```

```
Install      0 Package(s)
Update       0 Package(s)
Remove       1 Package(s)

Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing        : mc                          [1/1]

Removed: mc.i386 1:4.6.1a-35.el5
Complete!
```

Debian Package

1. dpkg : le gestionnaire de paquets Debian

La commande **dpkg** est le pendant de rpm pour les distributions Debian et dérivées, dont Ubuntu. Elle fait la même chose, ou presque, que rpm. Les packages Debian portent une extension .deb pour les reconnaître et disposent des mêmes informations et moyens qu'un package rpm. La commande **dpkg** est chargée de l'installation, la création, la suppression et la gestion des paquets Debian.

La base de données dpkg est généralement placée dans **/var/lib/dpkg**. Les fichiers qui y sont présents sont au format texte. Cependant n'éditez pas les fichiers à la main. Le fichier **/var/lib/dpkg/status** contient l'intégralité des packages connus par dpkg avec leur état.

```
# grep ^Package: /var/lib/dpkg/status | grep glibc
Package: devhelp-book-glibc
Package: libg++2.8.1.3-glibc2.2
Package: glibc-doc
Package: libstdc++2.10-glibc2.2
```

Dpkg dispose d'une interface graphique, GDebi, qui permet d'éviter l'utilisation de la ligne de commande.

2. Installation, mise à jour et suppression

L'option **-i**, ou **-install**, installe le ou les packages passés comme argument.

```
# dpkg -i monpaquet.deb
```

Notez que comme rpm, dpkg ne gère pas seul les dépendances. S'il manque des dépendances, la commande vous en informera. Dans ce cas, vous devez installer les dépendances de la même manière avant d'installer votre package.

Vous pouvez demander l'installation de tous les packages présents au sein d'une arborescence avec le paramètre `-R`, pour récursif. Dans ce cas, indiquez comme argument un nom de répertoires : tous les packages présents dans le répertoire et ses sous-répertoires seront installés.

```
# dpkg -R folder
```

La mise à jour s'effectue de la même manière que l'installation, avec le **-i**. Si vous installez un package déjà présent, dpkg en effectue une mise à jour. Ainsi, une installation ou une mise à jour respectent la méthodologie suivante :

- Extraction des fichiers de contrôle du nouveau paquet.
- Quand une ancienne version du même paquet est déjà installée, exécution du script pré-supression de l'ancien paquet.
- Lancement du script de préinstallation s'il est fourni par le paquet.
- Dépaquetage des nouveaux fichiers et sauvegarde des anciens pour pouvoir les restaurer en cas de problème.
- Si une ancienne version du paquet est déjà installée, exécution du script de post-suppression de l'ancien paquet.
- Configuration du paquet.

Il n'existe pas d'équivalence au mode freshen (-F) de rpm. Si vous souhaitez mettre à jour un package uniquement s'il est déjà installé, vous devez tout d'abord vérifier s'il est installé. Si un package est installé il commence par ii dans la liste :

```
# (dpkg -l zip | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
PRESENT
```

```
# (dpkg -l slapd | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
ABSENT
```

Pour mettre à jour un paquet seulement s'il est présent utilisez ce genre de ligne de commande, ou un équivalent de votre cru :

```
# (dpkg -l zip | grep ^ii >/dev/null) && dpkg -l zip.deb
```

Notez que le plus simple pour mettre à jour vos paquets déjà présents est de créer un dépôt APT contenant vos mises à jour et d'exécuter un apt-get upgrade depuis le client.

La suppression d'un package s'effectue avec le paramètre `-r` (en minuscule). Là encore, c'est à vous de gérer les dépendances.

La suppression d'un paquet effectue les étapes suivantes :

- Exécution du script de pré-suppression.
- Suppression des fichiers installés.
- Exécution du script de post-suppression.

```
# dpkg -r zip
```

Tout est supprimé sauf les fichiers de configuration et ce, afin d'éviter une reconfiguration de l'outil si vous le réinstallez. Pour tout supprimer, y compris ces fichiers, précisez le paramètre `-P` (purge).

```
# dpkg -P apache
```

Attention : ne confondez pas les paramètres `-r` et `-R` au risque d'un drame !

Si vous remplacez le nom du package par les paramètres `-a` ou `--pending` les packages non installés (non dépaquetés) mais présents dans les informations de la base pour être purgés ou supprimés, sont effacés.

L'utilisation des options `--force-all` et `--purge` permet de forcer la désinstallation du paquet et de supprimer les fichiers de configuration associés.

```
# dpkg --force-all --purge nom_du_paquet
```

3. Requêtes dpkg

a. Lister les paquets

Vous pouvez lister tous les packages Debian connus du système avec le paramètre `-l` :

```
# dpkg -l
...
ii  adduser                        3.102
Add and remove users and groups
ii  alien                          8.64
install non-native packages with dpkg
rc  amavisd-new                    2.4.2-6.1
Interface between MTA and virus scanner/cont
ii  antlr                          2.7.6-7
language tool for constructing recognizers,
rc  apache                         1.3.34-4.1
versatile, high-performance HTTP server
ii  apache-common                  1.3.34-4.1+etch1
support files for all Apache webservers
ii  apache2                        2.2.3-4+etch4
Next generation, scalable, extendable web se
```

```
rc  apache2-common                  2.0.54-5sarge2
next generation, scalable, extendable web se
ii  apache2-mpm-prefork             2.2.3-4+etch4
Traditional model for Apache HTTPD 2.1
ii  apache2-utils                   2.2.3-4+etch4
utility programs for webservers
ii  apache2.2-common                2.2.3-4+etch4
Next generation, scalable, extendable web se
...
```

Vous pouvez indiquer un motif particulier :

```
# dpkg -l "apt*" |grep ^ii
ii  apt                 0.6.46.4-0.1    Advanced front-end for dpkg
ii  apt-file            2.0.3-7         APT package searching utility -
command-lin
ii  apt-listchanges     2.72.5etch2     Display change history from .deb
archives

ii  apt-rpm-repository 0.5.15lorg3.2-1 tools to create an APT RPM repository
ii  apt-utils           0.6.46.4-0.1    APT utility programs
ii  aptitude            0.4.4-4         terminal-based apt frontend
```

Astuce : si votre console est trop petite pour afficher les noms des packages (seconde colonne) vous pouvez ruser comme ceci :

```
# COLUMNS=160 dpkg -l "kernel*" | grep ^ii | awk '{print $2}'
kernel-image-2.6.7-1-686
kernel-image-2.6.8-1-686-smp
kernel-image-2.6.8-2-686
kernel-image-2.6.8-2-686-smp
```

Une autre méthode consiste à employer l'option **`--get-selections`** :

```
# dpkg --get-selections | grep kernel
fai-kernels                                     install
kernel-image-2.6.7-1-686                        install
kernel-image-2.6.8-1-686-smp                    install
kernel-image-2.6.8-2-686                        install
kernel-image-2.6.8-2-686-smp                    install
linux-kernel-headers                            install
```

b. Trouver un paquet contenant un fichier

Le paramètre `-S` suivi du nom d'un fichier (son chemin) permet de retrouver le paquet d'origine.

```
# dpkg -S /usr/bin/basename
coreutils: /usr/bin/basename
```

c. Lister le contenu d'un paquet

Le paramètre `-L` liste le contenu du ou des paquets indiqués :

```
# dpkg -L coreutils | grep bin
/bin
/bin/mkdir
/bin/mv
/bin/true
/bin/mknod
/bin/sleep
/bin/touch
/bin/chgrp
/bin/uname
```

```
/bin/echo
/bin/sync
/bin/ln
/bin/date
/bin/dir
/bin/readlink
...
```

4. Convertir des packages

L'outil **alien** permet de convertir des packages RPM en DPKG et vice versa. Certains packages ne sont fournis que pour l'un ou l'autre des systèmes. C'est embêtant lorsqu'un produit n'est fourni que sous une forme et qu'il faut tout de même l'installer sur une autre plate-forme Linux.

Voici l'exemple d'un package, le client Networker, uniquement fourni pour Red Hat. Avec Alien, il est possible de le convertir au format dpkg.

Le paramètre par défaut `-d` convertit du rpm au dpkg :

```
# alien -d lgtoclnt-7.4-1.i686.rpm
Warning: Skipping conversion of scripts in package lgtoclnt: postinst postrm
preinst prerm
Warning: Use the --scripts parameter to include the scripts.
lgtoclnt_7.4-2_i386.deb generated
```

Comme indiqué, la conversion par défaut va vérifier les dépendances, mais ne va pas inclure les scripts de pré-installation et de post-installation. Vous devez alors préciser le paramètre `--scripts`.

```
# alien --scripts -d lgtoclnt-7.4-1.i686.rpm
lgtoclnt_7.4-2_i386.deb generated
# ls -l *.deb
-rw-r--r-- 1 root root 29471546 2008-05-09 14:45 lgtoclnt_7.4-2_i386.deb
```

Le résultat est le suivant :

```
# dpkg -I lgtoclnt_7.4-2_i386.deb
 nouveau paquet Debian, version 2.0.
 taille 29471546 octets: archive de controle = 4498 octets.
     923 octets,    18 lignes      control
    3142 octets,    57 lignes      md5sums
    4014 octets,   148 lignes   *  postinst             #!/bin/sh
    1362 octets,    35 lignes   *  postrm               #!/bin/sh
     317 octets,    11 lignes   *  preinst              #!/bin/sh
    1828 octets,    52 lignes   *  prerm                #!/bin/sh
      61 octets,     3 lignes      shlibs
 Package: lgtoclnt
 Version: 7.4-2
 Section: alien
 Priority: extra
 Architecture: i386
 Depends: libc6 (>= 2.3.6-6), libgl1-mesa-glx | libgl1, libice6 (>=
1:1.0.0), libncurses5 (>= 5.4-5), libsm6, libx11-6, libxext6,
libxmu6, libxp6, libxrender1, libxt6
 Installed-Size: 71632
 Maintainer: root <root@s64p17bib76.dsit.sncf.fr>
 Description: NetWorker Client
  EMC NetWorker protects the critical business data of more than 10,000
...
  and the smallest satellite branch offices.
  .
  (Converted from a rpm package by alien version 8.64.)
```

5. L'outil dselect

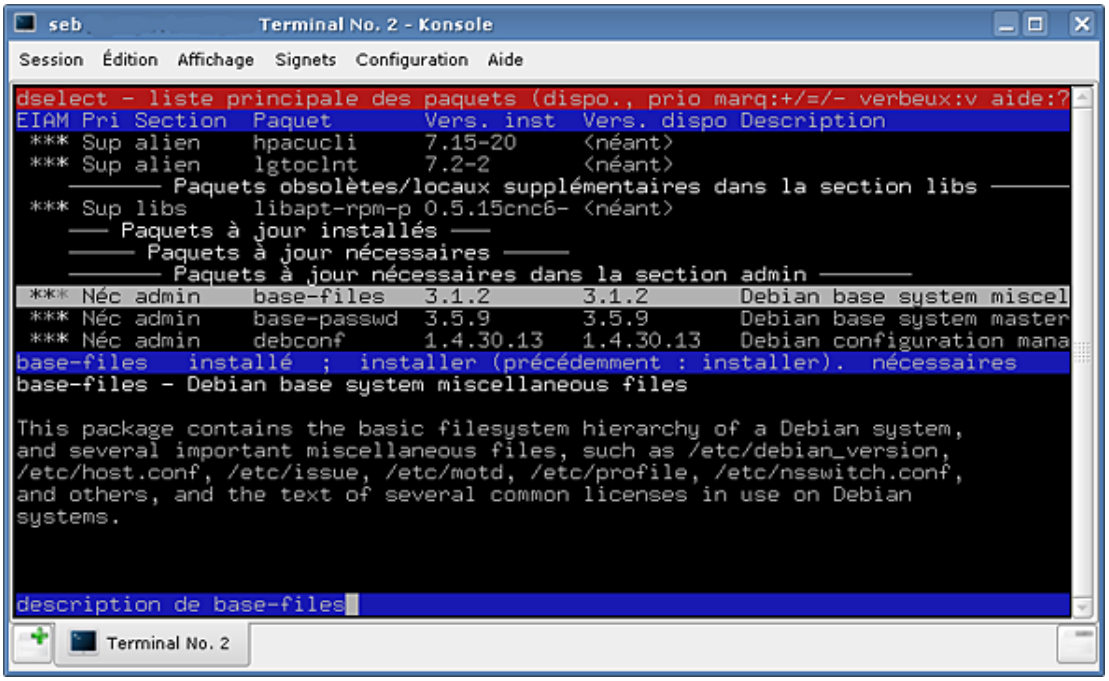
L'outil **dselect** est un frontend (comme APT) pour dpkg, qui gère les dépendances et les conflits. Historiquement, il est

le premier. Cependant son remplaçant APT dispose d'une bien meilleure qualité.

Le manuel de dselect indique clairement que l'outil est aujourd'hui tombé en désuétude.

- L'interface est confuse.
- Il y a peu de maintenance.
- Le manuel est incomplet.
- Il n'y a pas de filtre.
- Les accès ne sont plus standards.

Pour toutes ces raisons, il est préférable d'utiliser APT.



dselect ne devrait plus être utilisé.

Gestionnaire APT

1. Principe

Que ce soit avec rpm ou dpkg le problème est le même : ces deux outils contrôlent les dépendances des packages pour autoriser ou non leur installation, mais ne les gèrent pas. Autrement dit, si une dépendance sur un package est absente, il ne sera pas installé, sauf si la dépendance est résolue :

- soit en installant auparavant les packages manquants,
- soit en indiquant sur la même ligne le chemin de ces mêmes packages.

De même lors d'une mise à niveau il se pose un problème avec les fichiers de configuration. Que faut-il en faire ?

APT permet de résoudre ces problèmes en gérant les dépendances à votre place. **APT** signifie *Advanced Packaging Tool*. Au lieu de spécifier un paquet (local ou distant), il prend en charge des dépôts de packages situés sur un CD, un DVD, dans un répertoire local, sur une source distante sur Internet (ftp, http), etc.

Un dépôt contient un ensemble de packages qui dépendent soit les uns des autres, soit d'autres packages en provenance d'autres dépôts. APT peut gérer plusieurs dépôts, à divers endroits. Il se débrouille seul : lorsque vous installez un package, il installe aussi ses dépendances (s'il les trouve).

2. Les dépôts

a. Configuration

Les dépôts sont indiqués dans le fichier **`/etc/apt/sources.list`**. Le fichier suivant provient d'une installation Debian Etch où les dépôts contrib et non-free ont été rajoutés.

```
$ cat /etc/apt/sources.list
## etch
deb http://ftp.fr.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ etch main contrib non-free

# security
deb http://security.debian.org/ etch/updates main contrib non-free
deb-src http://security.debian.org/ etch/updates main contrib non-free
```

Les dépôts sont préparés côté serveur dans une arborescence de répertoires. La commande **genbasedir** permet de créer un dépôt. La syntaxe d'une ligne du fichier sources.list est la suivante :

```
deb uri distribution composant1 composant2 ...
```

- `uri` est le chemin vers la racine du ou des dépôts. Ce peut être une URL de type http ou ftp, mais aussi un chemin local (file), un CD-Rom ou DVD-Rom (CDrom), un chemin ssh, etc.
- La distribution est, comme son nom l'indique, le nom de la distribution Debian. Ici c'est **etch**, mais il est possible de spécifier d'autres versions de la distribution (testing, sarge, etc.) pour pouvoir récupérer des packages d'autres dépôts, plus récents par exemple. L'architecture peut être précisée. Si elle ne l'est pas, APT se débrouille seul pour rajouter le suffixe nécessaire.
- Les composants sont les noms des dépôts pour la distribution donnée.

En pratique, l'uri, la distribution et les composants permettent de reconstituer l'url complète d'accès au dépôt.

- Rendez-vous sur http://ftp.fr.debian.org/debian/.
- Cliquez sur le dossier appelé **dists**. Il contient la liste des distributions Debian.
- Dans dists, cliquez sur **etch**, le nom de la distribution actuelle.

- Dans etch, vous trouvez des dossiers **contrib**, **main** et **non-free**.

La ligne `deb http://ftp.fr.debian.org/debian/ etch main contrib non-free` correspond donc aux URLs :

- http://ftp.fr.debian.org/debian/dists/etch/main
- http://ftp.fr.debian.org/debian/dists/etch/contrib
- http://ftp.fr.debian.org/debian/dists/etch/non-free

Si vous continuez, par exemple en rentrant dans **main**, vous trouverez une série de dossiers suffixés en fonction de l'architecture de votre installation Linux. La machine VMWare de test est de type i386. Les packages binaires seront donc cherchés dans **binary-i386**.

Ne soyez pas surpris de ne pas trouver de packages dans ce dernier répertoire, mais des fichiers :

- **`Release`** : description du dépôt.
- **`Packages.gz`** : index des packages du dépôt, au format gzip.
- **`Packages.bz2`** : la même chose au format bzip2.

Où sont réellement les packages ? La réponse est dans les fichiers Packages.*. Voici le début de l'un d'eux :

```
Package: 3270-common
Priority: optional
Section: net
Installed-Size: 96
Maintainer: Bastian Blank <waldi@debian.org>
Architecture: i386
Source: ibm-3270
Version: 3.3.4p6-3.3
Depends: libc6 (>= 2.3.6-6)
Recommends: x3270 (= 3.3.4p6-3.3) | c3270 (= 3.3.4p6-3.3), pr3287 (=
3.3.4p6-3.3)
Filename: pool/main/i/ibm-3270/3270-common_3.3.4p6-3.3_i386.deb
Size: 21910
MD5sum: 209bb0595c53421c433f4524147d6335
SHA1: c89e5ef06fa0978b5a0935c90273b5c5997b2142
SHA256: 881cf62382b9e1945155bdd366645d9660c1848aaab3a58e73d2bdfaa49301ae
Description: Common files for IBM 3270 emulators and pr3287
 3270-common contains files referenced in other 3270 packages
```

La ligne **Filename** vous indique que le fichier est dans **`pool/main/i/ibm-3270/`** depuis l'uri, donc accessible depuis l'URL http://ftp.fr.debian.org/debian/pool/main/i/ibm-3270/. Notez aussi dans la description des packages, les lignes Depends et Recommends, qui permettent à APT de résoudre les dépendances. Ce fichier ressemble fortement, et pour cause, au fichier **`status`** de la base dpkg locale.

b. Mise à jour de la base

Une fois vos dépôts configurés, vous devez mettre à jour la base de données locale de APT avec la commande **apt-get** et l'option **update**.

```
# apt-get update
Réception de : 1 http://ftp.fr.debian.org etch Release.gpg [378B]
Atteint http://ftp.fr.debian.org etch Release
Ign http://ftp.fr.debian.org etch/main Packages/DiffIndex
Réception de : 2 http://ftp.fr.debian.org etch/contrib Packages
[59,2kB]
Réception de : 3 http://security.debian.org etch/updates Release.gpg
[189B]
Réception de : 4 http://security.debian.org etch/updates Release
[37,6kB]
```

```
Réception de : 5 http://ftp.fr.debian.org etch/non-free Packages
[83,8kB]
Ign http://ftp.fr.debian.org etch/main Sources/DiffIndex
Réception de : 6 http://ftp.fr.debian.org etch/contrib Sources [18,3kB]
Réception de : 7 http://ftp.fr.debian.org etch/non-free Sources [28,2kB]
Atteint http://ftp.fr.debian.org etch/main Packages
Atteint http://ftp.fr.debian.org etch/main Sources
Ign http://security.debian.org etch/updates/main Packages/DiffIndex
Ign http://security.debian.org etch/updates/contrib Packages/DiffIndex
Réception de : 8 http://security.debian.org etch/updates/non-free
Packages
[3614B]
...
541ko réceptionnés en 4s (115ko/s)
Lecture des listes de paquets... Fait
```

3. Mise à jour de la distribution

Une fois les dépôts à jour, vous pouvez mettre à jour en une seule commande tous les packages installés sur votre distribution : APT vérifie si des packages plus récents sont disponibles dans les dépôts. Il se base pour cela sur la base de données locale. Si elle n'est pas à jour il est possible que certains des packages trop anciens ne soient plus présents.

Exécutez la commande **apt-get** avec l'option **upgrade**. APT vous informe que huit packages peuvent être mis à jour. Vous pouvez accepter ou refuser. Si vous acceptez, APT télécharge ces packages et leurs éventuelles dépendances, et les installe. Le processus peut être plus ou moins long selon le nombre de mises à jour et le type de support.

```
# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Les paquets suivants seront mis à jour :
  cpio libgnutls13 libspeex1 libssl0.9.8 linux-image-2.6.18-6-686
  openssh-client openssl rdesktop
8 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 21,3Mo dans les archives.
Après dépaquetage, 1876ko d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ? O
Réception de : 1 http://security.debian.org etch/updates/main linux-
image-2.6.18
 -6-686 2.6.18.dfsg.1-18etch4 [16,3MB]
Réception de : 2 http://security.debian.org etch/updates/main cpio
2.6-18.1+etch
1 [132kB]
Réception de : 3 http://security.debian.org etch/updates/main lib-
gnutls13
1.4.4-
3+etch1 [282kB]
Réception de : 4 http://security.debian.org etch/updates/main libs-
sl0.9.8
0.9.8c
-4etch3 [2717kB]
Réception de : 5 http://security.debian.org etch/updates/main opens-
sh-client
1:4
.3p2-9etch2 [660kB]
Réception de : 6 http://security.debian.org etch/updates/main libs-
peex1
1.1.12-3
etch1 [76,4kB]
Réception de : 7 http://security.debian.org etch/updates/main openssl
0.9.8c-4et
ch3 [1001kB]
Réception de : 8 http://security.debian.org etch/updates/main rdesktop
1.5.0-1et
ch2 [124kB]
21,3Mo réceptionnés en 2m51s (125ko/s)
Préconfiguration des paquets...
(Lecture de la base de données... 80122 fichiers et répertoires déjà
```

```
installés.)
...
Préparation du remplacement de cpio 2.6-18 (en utilisant
.../cpio_2.6-18.1+etch1
_i386.deb) ...
Dépaquetage de la mise à jour de cpio ...
Préparation du remplacement de libgnutls13 1.4.4-3 (en utilisant
.../libgnutls13
_1.4.4-3+etch1_i386.deb) ...
Dépaquetage de la mise à jour de libgnutls13 ...
Préparation du remplacement de libssl0.9.8 0.9.8c-4etch1 (en utilisant
.../libss
l0.9.8_0.9.8c-4etch3_i386.deb) ...
Dépaquetage de la mise à jour de libssl0.9.8 ...
Préparation du remplacement de openssh-client 1:4.3p2-9 (en utilisant
.../openss
h-client_1%3a4.3p2-9etch2_i386.deb) ...
Dépaquetage de la mise à jour de openssh-client ...
Préparation du remplacement de libspeex1 1.1.12-3 (en utilisant
.../libspeex1_1.
1.12-3etch1_i386.deb) ...
Dépaquetage de la mise à jour de libspeex1 ...
Préparation du remplacement de openssl 0.9.8c-4etch1 (en utilisant
.../openssl_0
.9.8c-4etch3_i386.deb) ...
Dépaquetage de la mise à jour de openssl ...
Préparation du remplacement de rdesktop 1.5.0-1etch1 (en utilisant
.../rdesktop_
1.5.0-1etch2_i386.deb) ...
Dépaquetage de la mise à jour de rdesktop ...
...
Paramétrage de cpio (2.6-18.1+etch1) ...

Paramétrage de libgnutls13 (1.4.4-3+etch1) ...

Paramétrage de libssl0.9.8 (0.9.8c-4etch3) ...
Checking for services that may need to be restarted...done.
Checking init scripts...

Restarting services possibly affected by the upgrade:
  exim4: stopping...starting...done.

Services restarted successfully.
...
```

Une autre possibilité est de faire une mise à jour profonde (appelée mise à jour distante). APT garde une certaine cohérence dans les packages lors de la mise à jour, notamment concernant la version de la distribution. Vous pouvez spécifier plusieurs distributions Debian dans vos dépôts. Mais même si une distribution est plus récente, un simple upgrade ne va pas transformer la vôtre en la toute dernière. Vous pouvez demander à APT de forcer la mise à jour vers la nouvelle distribution avec un **dist-upgrade**.

Pour les besoins de cet ouvrage, les dépôts de la version de test Debian appelée lenny ont été ajoutés :

```
## lenny
deb http://ftp.fr.debian.org/debian/ lenny main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ lenny main contrib non-free
# security lenny
deb http://security.debian.org/ lenny/updates main contrib non-free
deb-src http://security.debian.org/ lenny/updates main contrib non-free
```

Effectuez une mise à jour de la base. Notez qu'il a fallu agrandir le cache de APT pour ceci et nettoyer la base :

```
# apt-get update
echo 'APT::Cache-Limit "141943904";' > /etc/apt/apt.conf.d/00Cache
# apt-get clean
# apt-get update
```

puis

```
$ apt-get dist-upgrade
```

```
Lecture des listes de paquets...
Construction de l'arbre des dépendances...
...
731 mis à jour, 228 nouvellement installés, 23 à enlever et 0
non mis à jour.
Il est nécessaire de prendre 858Mo dans les archives.
Après dépaquetage, 623Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ? O
...
```

Bon courage !

4. Rechercher et installer un package individuel

La commande **apt-cache** permet de rechercher un package, par son nom ou son commentaire, au sein de la base de données locale APT.

```
# apt-cache search torrent
bittornado - bittorrent client with enhanced curses interface
bittornado-gui - bittorrent client with enhanced GUI interface
bittorrent - Scatter-gather network file transfer
bittorrent-gui - Scatter-gather network file transfer (GUI files)
cfv - versatile file checksum creator and verifier
qtorrent - BitTorrent client for QT 3.x
```

La commande **apt-get install** xxx installe le package xxx :

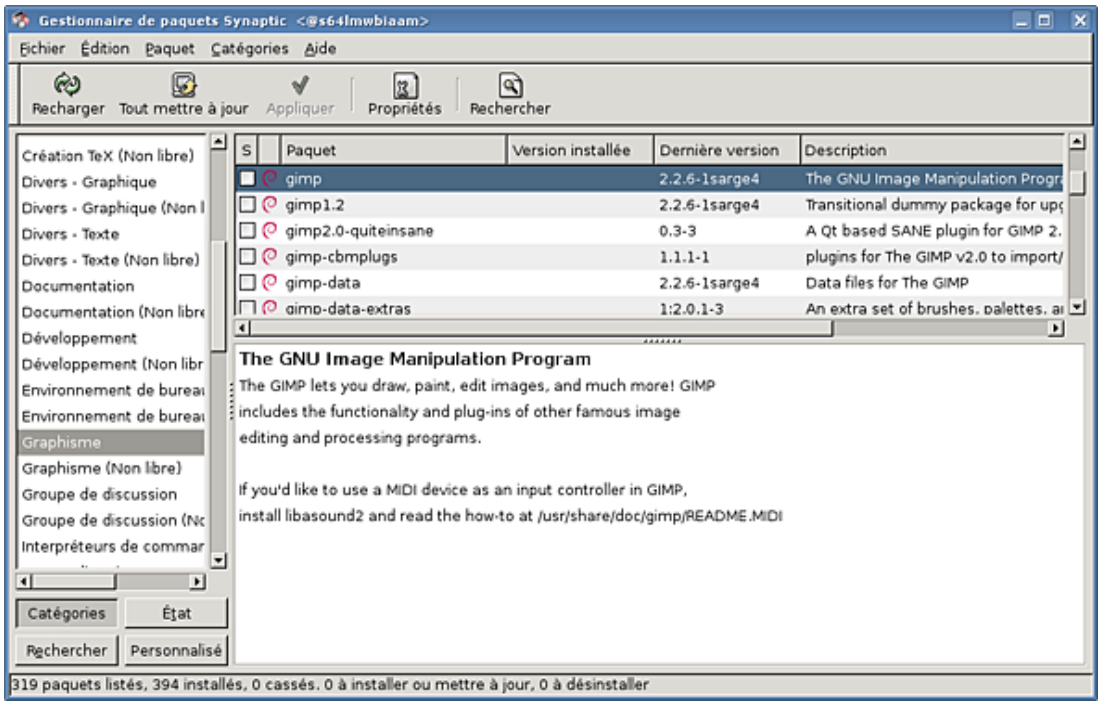
```
# apt-get install vim-gtk
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Les paquets supplémentaires suivants seront installés :
  vim vim-common
Paquets suggérés :
  ctags vim-doc vim-scripts cscope
Les NOUVEAUX paquets suivants seront installés :
  vim-gtk
Les paquets suivants seront mis à jour :
  vim vim-common
2 mis à jour, 1 nouvellement installés, 0 à enle-
ver et 58 non mis à jour.
Inst vim [1:6.3-071+1sarge1] (1:6.3-071+1sarge3 Debian:3.1r8/old-
stable) []
Inst vim-common [1:6.3-071+1sarge1] (1:6.3-071+1sarge3 Debian:3.1r8/
oldstable)
Inst vim-gtk (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim-common (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim-gtk (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
```

Deux options méritent d'être retenues :

- le **-s** pour la simulation : APT indique ce qu'il devrait faire, mais ne le fait pas.
- le **-f** pour « fix-broken » : APT tente de réparer les problèmes de dépendances comme il le peut (ajout de packages).

5. Client graphique

L'outil synaptic est un front-end : une interface graphique qui fait appel aux fonctions de APT. Il permet toutes les opérations proposées par APT tout en étant très convivial.



Synaptic est un front-end à APT.

Installer depuis les sources

1. Obtenir les sources

Il n'est parfois pas possible d'obtenir un logiciel ou une bibliothèque depuis un package pour sa distribution. Dans ce cas, il reste la solution de compiler et d'installer soi-même le produit depuis les sources.

Cela est possible pour une majorité de produits sous Linux, grâce aux avantages des logiciels libres et de la licence GPL telle que définie au premier chapitre. Tout logiciel libre est fourni avec ses sources. Il est donc possible de reconstruire soi-même le logiciel en le recompilant.

Une archive source est souvent récupérée sur divers sites Internet comme par exemple SourceForge. C'est une archive bien souvent compressée au format tgz (archive tar compressée avec gzip) ou tar.bz2 (archive tar compressée au format bzip2). Elle contient :

- le code source sous forme de fichiers **`.c`**, **`.h`**, **`.cpp`**, etc., selon le langage ;
- parfois un fichier **`Makefile`** permettant d'automatiser la compilation du produit ;
- souvent un fichier **`.configure`** permettant de générer le fichier Makefile en fonction de votre installation et de diverses options.

2. Pré-requis et dépendances

Pour compiler votre produit vous devez respecter quelques pré-requis :

- présence de l'outil make ;
- présence du ou des compilateurs nécessaires, notamment gcc ;
- présence des dépendances : bibliothèques, interpréteurs, etc.

Ce dernier point est très important. S'il manque une dépendance vous risquez divers problèmes :

- vous n'arriverez pas préparer les sources pour la compilation ;
- la compilation générera des erreurs ;
- le produit sera compilé mais avec des possibilités moindres ;
- le binaire résultant ne se lancera pas.

La commande **./configure** vous fournira les dépendances manquantes et leur version si c'est possible. Dans ce cas vous pouvez soit les installer depuis les packages de votre distribution, soit les installer depuis les sources.

Quand vous compilez depuis les sources sans passer par votre gestionnaire de packages, vous perdez une partie de la gestion des dépendances. Si vous installez des packages qui dépendent d'une version de l'outil installée depuis les sources il est possible, si les dépendances se basent sur l'existence d'un package et non d'un fichier, que le gestionnaire vous empêche d'installer votre package. Cherchez bien parmi les dépôts officiels ou non si le logiciel existe sous forme de package avant de le recompiler, ou si un package source existe.

Dans tous les cas, il n'y a pas besoin d'être root pour compiler votre logiciel. Cependant, selon la destination vous devrez passer root pour finaliser l'installation.

3. Exemple d'installation

Vous allez compiler et installer le produit PDFedit qui permet d'éditer et de créer des fichiers PDF.

- Téléchargez-le depuis le lien suivant. La version testée est la version 0.4.1 :

  http://sourceforge.net/project/showfiles.php?group_id=177354

```
$ ls -l pdfedit-0.4.1.tar.bz2
-rw-r--r-- 1 seb users 2958137 mai 20 14:26
```

- Décompressez le fichier :

```
$ tar xvjf pdfedit-0.4.1.tar.bz2
pdfedit-0.4.1/COPYING
pdfedit-0.4.1/Changelog
pdfedit-0.4.1/Makefile.flags.in
pdfedit-0.4.1/Makefile.in
pdfedit-0.4.1/Makefile.rules
pdfedit-0.4.1/README
pdfedit-0.4.1/config/
pdfedit-0.4.1/config/freetype2.m4
pdfedit-0.4.1/config/macro.m4
pdfedit-0.4.1/config/xpdf.m4
pdfedit-0.4.1/config/boost_iostreams.m4
pdfedit-0.4.1/config/boost_base.m4
pdfedit-0.4.1/config/env.m4
... (1558 fichiers en tout)
pdfedit-0.4.1/tools/
pdfedit-0.4.1/tools/mass_patch.sh
pdfedit-0.4.1/tools/cygwinbuild
pdfedit-0.4.1/tools/cygwin_build.bat
pdfedit-0.4.1/tools/generate_online_help.sh
pdfedit-0.4.1/tools/commit_patch/
pdfedit-0.4.1/tools/commit_patch/Makefile
pdfedit-0.4.1/tools/commit_patch/commit-patch
pdfedit-0.4.1/tools/commit_patch/README
pdfedit-0.4.1/tools/commit_patch/COPYING
pdfedit-0.4.1/tools/commit_patch/commit-patch.1
pdfedit-0.4.1/tools/commit_patch/commit-patch-buffer.el
pdfedit-0.4.1/tools/headergen
pdfedit-0.4.1/tools/make_release
pdfedit-0.4.1/tools/headergen.txt
```

- Déplacez-vous dans le dossier **`pdfedit-0.4.1`** créé par la décompression :

```
$ cd pdfedit-0.4.1/
$ ls -l
total 436
-rw-r--r--  1 seb users   5893 fév 24 19:58 Changelog
drwxr-xr-x  2 seb users   4096 fév 24 19:58 config
-rwxr-xr-x  1 seb users 303300 fév 24 19:59 configure
-rw-r--r--  1 seb users  11593 fév 20 18:45 configure.in
-rw-r--r--  1 seb users    969 fév 21 18:02 COPYING
drwxr-xr-x  5 seb users   4096 fév 24 19:58 doc
-rwxr-xr-x  1 seb users    148 jun 18  2006 getversion
-rw-r--r--  1 seb users   5880 fév 20 18:45 Makefile.flags.in
-rw-r--r--  1 seb users   2267 jan 16 14:50 Makefile.in
-rw-r--r--  1 seb users    835 fév 20 18:45 Makefile.rules
drwxr-xr-x  5 seb users   4096 fév 24 19:58 projects
-rw-r--r--  1 seb users  13689 fév 20 18:45 README
drwxr-xr-x 11 seb users   4096 fév 24 19:59 src
drwxr-xr-x  3 seb users   4096 fév 24 19:59 tools
```

- Remarquez la présence du fichier configure qui est exécutable.

```
$ ./configure --help
```

```
`configure' configures PDFedit 0.4.1 to adapt to many kinds of
systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify
them as
VAR=VALUE.  See below for descriptions of some of the useful
variables.
...
--with-t1-library=PATH  use t1 library (Type 1 font rasterizer)
  --with-t1-includes=DIR  set directory for t1 headers
  --with-qmake=BIN        Use specific qmake binary with the
absolute path.
                          (only for platforms where qmake is
not installed
                          under QTDIR tree).
  --with-cppunit-prefix=PFX   Prefix where CppUnit is installed
(optional)
  --with-cppunit-exec-prefix=PFX  Exec prefix where CppUnit is
installed (optional)
  --with-doxygen=BIN      Use specific doxygen binary with the
absolute path.
  --with-xsltproc=BIN     Use specific xsltproc binary with the
absolute path.
  --with-root-dir=DIR     Use different installation root (path
to prepend
                          before the prefix, empty by default).
  --with-x                use the X Window System
...
```

Une option importante de configure est **--prefix**. Elle définit l'emplacement de l'installation une fois le produit compilé. Par défaut le logiciel s'installe dans **/usr/local/**.

- Exécutez **./configure** seul. Il vous informera des dépendances manquantes le cas échéant.

```
$ ./configure
checking for g++... g++
checking for C++ compiler default output file name... a.out
checking whether the C++ compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for gcc... gcc
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking for ranlib... ranlib
checking whether ln -s works... yes
checking how to run the C++ preprocessor... g++ -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
...
checking for freetype-config... /usr/bin/freetype-config
checking for FreeType -- version >= 7.0.1... yes
checking whether to use t1 library... maybe
checking where to find the t1 header files...
checking t1lib.h usability... no
checking t1lib.h presence... no
checking for t1lib.h... no
not using t1 library
```

```
...
config.status: creating Makefile
config.status: creating Makefile.flags
config.status: creating src/xpdf/Makefile
config.status: creating src/xpdf/goo/Makefile
config.status: creating src/xpdf/fofi/Makefile
config.status: creating src/xpdf/splash/Makefile
config.status: creating src/xpdf/xpdf/Makefile
config.status: creating src/utils/aconf.h
config.status: creating src/xpdf/aconf.h
```

- Notez le bloc en gras : une bibliothèque est manquante. Vous avez deux possibilités :

  - installer la bibliothèque manquante (et son package de développement) soit depuis les sources, soit depuis le gestionnaire de packages de votre distribution ;

  - ne pas l'installer : elle n'est pas vitale. Cependant il n'y aura pas de support des polices Type1, ce qui est embêtant pour les PDF.

- Après avoir résolu, si vous le voulez, les dépendances, vous devez relancer la commande **./configure**. La commande configure crée le fichier **`Makefile`** correspondant. Ce fichier contient l'ensemble des règles, chemins et options pour compiler le logiciel. La commande **make** y fait appel.

- Lancez la compilation avec la commande **make**. Il se peut que des avertissements apparaissent (lignes warning). Cela ne signifie pas forcément que le programme ne compilera pas ou ne marchera pas par la suite. De toute façon si la compilation produit des erreurs, elle s'arrêtera toute seule avec un message d'erreur du compilateur pouvant parfois (mais pas toujours) vous mettre sur la voie d'une solution.

  La compilation peut être plus ou moins longue selon le produit compilé. Pour PDFEdit, une machine à 1800MHz a mis environ 20 minutes.

```
$ make
cd /home/seb/pdfedit-0.4.1/src && make
make[1]: entrant dans le répertoire « /home/seb/pdfedit-0.4.1/src »
cd /home/seb/pdfedit-0.4.1/src/xpdf && make libxpdf
make[2]: entrant dans le répertoire « /home/seb/pdfedit-
0.4.1/src/xpdf »
cd goo && make
make[3]: entrant dans le répertoire « /home/seb/pdfedit-
0.4.1/src/xpdf/goo »
g++ -c  -O2   -fmessage-length=0 -D_FORTIFY_SOURCE=2 -fno-strict-
aliasing -fexceptions  -pipe  -I. -I/home/seb/pdfedit-0.4.1/src -
I/home/seb/pdfedit-0.4.1/src/xpdf/  -I/usr/include -
I/usr/include/freetype2  -o GHash.o Ghash.cc
g++ -c  -O2   -fmessage-length=0 -D_FORTIFY_SOURCE=2 -fno-strict-
aliasing -fexceptions  -pipe  -I. -I/home/seb/pdfedit-0.4.1/src -
I/home/seb/pdfedit-0.4.1/src/xpdf/  -I/usr/include -
I/usr/include/freetype2  -o GList.o Glist.cc
g++ -c  -O2   -fmessage-length=0 -D_FORTIFY_SOURCE=2 -fno-strict-
aliasing -fexceptions  -pipe  -I. -I/home/seb/pdfedit-0.4.1/src -
I/home/seb/pdfedit-0.4.1/src/xpdf/  -I/usr/include -
I/usr/include/freetype2  -o GString.o Gstring.cc
GString.cc:72: warning: deprecated conversion from string constant
to 'char*'
GString.cc:72: warning: deprecated conversion from string constant
to 'char*'
GString.cc:72: warning: deprecated conversion from string constant
to 'char*'
... (plusieurs milliers de lignes)
g++  -o pdfedit .obj/additemdialog.o .obj/aboutwindow.o
.obj/option.o .obj/optionwindow.o .obj/dialog.o .obj/imagewidget.o
.obj/stringoption.o .obj/realoption.o .obj/intoption.o
.obj/booloption.o .obj/combooption.o .obj/dialogoption.o
.obj/fileoption.o .obj/fontoption.o
...
0.4.1/src/utils -lxpdf -L/home/seb/pdfedit-0.4.1/src/xpdf/xpdf -
```

```
lfofi -L/home/seb/pdfedit-0.4.1/src/xpdf/fofi -lGoo -
L/home/seb/pdfedit-0.4.1/src/xpdf/goo -lsplash -L/home/seb/pdfedit-
0.4.1/src/xpdf/splash -lutils -L/home/seb/pdfedit-0.4.1/src/utils  -
lfreetype -lz -lt1 -lqt-mt -lXext -lX11 -lm
make[2]: quittant le répertoire « /home/seb/pdfedit-0.4.1/src/gui »
make[1]: quittant le répertoire « /home/seb/pdfedit-0.4.1/src »
```

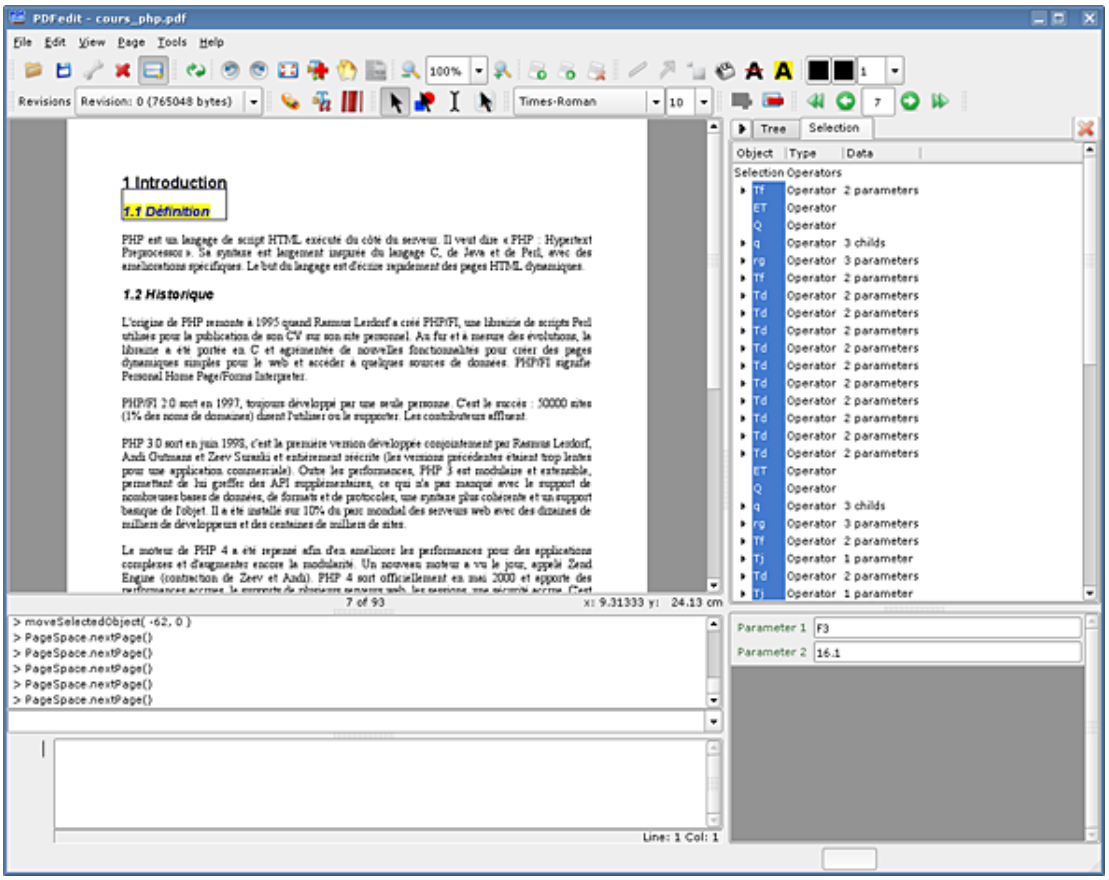
- La compilation s'étant terminée sans erreur, finissez en installant le produit avec **make install**. Attention, le produit va s'installer dans /usr/local/ ce qui nécessite les droits de l'utilisateur root.

```
$ su -c "make install"
password :
...
cp -f "../../doc/user/gui/menuAndToolbarsFun/images/toolbars_
text.png"
"/usr/local/share/doc/pdfedit/gui/menuAndToolbarsFun/images/"
cp -f "../../doc/user/gui/menuAndToolbarsFun/images/toolbars_
treeview.png"
"/usr/local/share/doc/pdfedit/gui/menuAndToolbarsFun/images/"
cp -f "pdfedit" "/usr/local/bin/pdfedit"
make[1]: quittant le répertoire « /home/seb/pdfedit-0.4.1/src/gui »
cd /home/seb/pdfedit-0.4.1/doc && make doc_dist
make[1]: entrant dans le répertoire « /home/seb/pdfedit-0.4.1/doc »
cd user && make pdfedit.1
make[2]: entrant dans le répertoire « /home/seb/pdfedit-
0.4.1/doc/user »
cat cmdline/pdfedit.head cmdline/description.xml cmdli-
ne/localization.xml cmdline/pdfedit.tail >pdfedit.xml
../tools/docbook2man.pl pdfedit.xml >pdfedit.1
```

- Lancez le produit :

```
$ pdfedit
```

Félicitations, la compilation et l'installation ont parfaitement fonctionné.



4. Désinstallation

La plupart des **Makefile**, en tout cas ceux générés par configure, permettent la désinstallation. Elle s'effectue par la commande **make uninstall**.

```
$ su -c "make uninstall"
password :
...
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_commandline.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_editor.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_execute.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_lookandfeel.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_objecttree.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_paths.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_toolbars.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/pagespace.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/pdfedit.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_all.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_catalog.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_edit.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_edit_add.png"
...
```

5. Les bases du Makefile

a. Bases

Un fichier Makefile est utilisé par le programme **make** pour exécuter un ensemble d'actions comme la compilation d'un projet mais il n'est pas limité à cela : il s'agit d'une sorte de script par niveaux.

Soit le projet suivant chargé d'afficher "Bonjour". Beaucoup de choses pour un résultat limité mais c'est un exemple.

```
$ cat bonjour.h
#ifndef H_BONJOUR
#define H_BONJOUR
void Bonjour(void);
#endif

$ cat bonjour.c
#include <stdio.h>
#include <stdlib.h>

void Bonjour(void)
{
        printf("Bonjour\n");
}

$ cat main.c
#include <stdio.h>
#include <stdlib.h>
#include "bonjour.h"

int main(void)
{
        Bonjour();
        return 0;
}
```

Pour compiler ce projet à la main vous devez exécuter les étapes suivantes :

```
$ gcc -o bonjour.o -c bonjour.c
```

```
$ gcc -o bonjour bonjour.o main.o
$ gcc -o main.o -c main.c
$ ./bonjour
Bonjour
```

Le Makefile est composé de règles qui ont la structure suivante :

```
cible: dependance
commandes
```

Un premier Makefile pourrait donc être :

```
$ cat Makefile
bonjour: bonjour.o main.o
        gcc -o bonjour bonjour.o main.o

bonjour.o: bonjour.c
        gcc -o bonjour.o -c bonjour.c

main.o: main.c bonjour.h
        gcc -o main.o -c main.c
```

Première règle : pour exécuter la règle bonjour il faut disposer des fichiers **`bonjour.o`** et **`main.o`**. Si on les a, il faut exécuter la commande **gcc -o bonjour bonjour.o main.o**.

Deuxième règle : pour exécuter la règle `bonjour.o` il faut disposer du fichier **`bonjour.c`**. S'il est présent alors la commande **gcc -o bonjour.o -c bonjour.c** est exécutée.

Troisième règle : pour exécuter la règle `main.o` il faut disposer des fichiers **`main.c`** et **bonjour.h**. S'ils sont présents alors la commande **gcc -o main.o -c main.c** est exécutée.

Les deux dernières règles permettent de résoudre la première. Si vous lancez la commande **make**, elle va déterminer quelles sont les règles applicables, dans quel ordre, et les appliquer, dans l'ordre des dépendances. Si les fichiers sont à jour, make ne les reconstruit pas sauf s'ils ont été modifiés.

```
$ rm -f *.o
$ make
gcc -o bonjour.o -c bonjour.c
gcc -o main.o -c main.c
gcc -o bonjour bonjour.o main.o
$ ./bonjour
Bonjour
```

b. Makefile intermédiaire

Le Makefile précédent fonctionne mais n'est pas optimal :

- Il ne permet pas de compiler plusieurs binaires.
- Il ne permet pas de nettoyer les fichiers temporaires (.o) après la compilation.
- Il ne permet pas de forcer la recompilation du projet.

L'ajout de nouvelles règles permet de pallier ces problèmes :

- all : génère n règles ;
- clean : nettoie les .o ;
- mrproper : appelle clean et supprime les binaires.

```
$ cat Makefile
all: bonjour
```

```
bonjour: bonjour.o main.o
        gcc -o bonjour bonjour.o main.o

bonjour.o: bonjour.c
        gcc -o bonjour.o -c bonjour.c

main.o: main.c bonjour.h
        gcc -o main.o -c main.c

clean:
        rm -rf *.o

mrproper: clean
        rm -rf bonjour

$ make clean
rm -rf *.o
$ make mrproper
rm -rf *.o
rm -rf bonjour
$ make all
gcc -o bonjour.o -c bonjour.c
gcc -o main.o -c main.c
gcc -o bonjour bonjour.o main.o
```

c. Un peu plus complexe

Variables utilisateur

Pour finir cette petite présentation, vous pouvez définir des variables dans votre fichier, et utiliser des variables internes prédéfinies :

Le Makefile devient :

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
        gcc -o bonjour bonjour.o main.o $(LDFLAGS)

bonjour.o: bonjour.c
        gcc -o bonjour.o -c bonjour.c $(CFLAGS)

main.o: main.c bonjour.h
        gcc -o main.o -c main.c $(CFLAGS)

clean:
        rm -rf *.o

mrproper: clean
        rm -rf $(EXEC)
```

Variables internes

Parmi les variables internes :

- $@ : nom de la cible.
- $< : nom de la première dépendance.
- $^ : liste des dépendances.

- $? : dépendances plus récentes que la cible.
- $* : nom du fichier sans le suffixe.

Le Makefile devient :

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
        gcc -o $@ $^ $(LDFLAGS)

bonjour.o: bonjour.c
        gcc -o $@ -c $< $(CFLAGS)

main.o: main.c bonjour.h
        gcc -o $@ -c $< $(CFLAGS)

clean:
        rm -rf *.o

mrproper: clean
        rm -rf $(EXEC)
```

Règles d'inférence

Il existe des règles prédéfinies, à base de raccourcis, qui permettent de générer des cibles en fonction du nom du fichier C et objet : **%.o : %.c**. La tentation est grande de créer une règle unique pour main.o et bonjour.o :

```
%.o: %.c
        gcc -o $@ -c $< $(CFLAGS)
```

Cette règle est correcte mais il manque la dépendance du header bonjour.h : si celui-ci est modifié le projet n'est plus compilé. Il faut rajouter une règle spécifique :

```
main.o : bonjour.h
```

Le Makefile devient :

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
        gcc -o $@ $^ $(LDFLAGS)

%.o: %.c
        gcc -o $@ -c $< $(CFLAGS)

main.o: bonjour.h

clean:
        rm -rf *.o

mrproper: clean
        rm -rf $(EXEC)
```

Gérer les bibliothèques partagées

1. Principe

Une bibliothèque partagée est un fichier particulier qui contient une liste de fonctions, ou API, accessible à tout programme en ayant besoin sans avoir à les réécrire. À l'opposé de la bibliothèque statique, le programme accède dynamiquement aux fonctions qui sont placées dans un fichier à part. N programmes différents peuvent accéder aux fonctions proposées par la bibliothèque. Les bibliothèques regroupent des fonctions propres à un domaine ou un ensemble de domaines cohérents : traitement d'images, du son, de l'accès à une base de données, etc.

Un ensemble de fonctions proposées par une ou plusieurs bibliothèques partagées forme une **API**, *Application Programming Interface*, et sont parfois regroupées au sein d'un framework offrant une solution complète pour un domaine donné.

Un lien est établi entre le programme et une bibliothèque partagée lors de l'étape de l'édition des liens par l'éditeur de liens **ld**, lui-même appelé par le compilateur **gcc** avec l'option **-l<lib>**.

Une autre possibilité pour un programme est d'utiliser la fonction C **dlopen** qui ouvre une bibliothèque dynamique comme un fichier et qui accède aux fonctions qui y sont contenues avec des pointeurs de fonctions.

Si un programme dépend d'une bibliothèque partagée et que celle-ci est absente, le programme ne pourra plus fonctionner.

Sous Linux (et Unix en général) les bibliothèques partagées sont appelées des **Shared Objects** (so) dans le sens où il s'agit de fichiers objets sans bloc d'instruction **main**. Ils portent le suffixe **.so**.

Une bibliothèque peut disposer de plusieurs versions, pouvant être ou non compatibles, et la version peut être précisée lors de l'édition des liens, avec une version par défaut possible.

2. Lieu de stockage

Les bibliothèques partagées sont par convention placées dans des répertoires appelés lib :

- **/lib** : bibliothèques systèmes de base, vitales ;
- **/usr/lib** : bibliothèques utilisateur de base, non nécessaires au boot ;
- **/usr/local/lib** : bibliothèques locales aux produits pour la machine ;
- **/usr/X11R6/lib** : bibliothèques de l'environnement X Window ;
- **/opt/kde3/lib** : bibliothèques de KDE ...

```
$ ls -l /lib
total 6024
...
-rwxr-xr-x  1 root root  114636 oct 23  2007 ld-2.6.1.so
lrwxrwxrwx  1 root root      11 oct  5  2007 ld-linux.so.2 -> ld-2.6.1.so
lrwxrwxrwx  1 root root      13 oct  5  2007 ld-lsb.so.2 -> ld-linux.so.2
lrwxrwxrwx  1 root root      13 oct  5  2007 ld-lsb.so.3 -> ld-linux.so.2
lrwxrwxrwx  1 root root      15 oct  5  2007 libacl.so.1 -> libacl.so.1.1.0
-rwxr-xr-x  1 root root   27864 sep 22  2007 libacl.so.1.1.0
lrwxrwxrwx  1 root root      15 oct  5  2007 libaio.so.1 -> libaio.so.1.0.1
-rwxr-xr-x  1 root root    5248 sep 21  2007 libaio.so.1.0.1
-rwxr-xr-x  1 root root   10256 oct 23  2007 libanl-2.6.1.so
lrwxrwxrwx  1 root root      15 oct  5  2007 libanl.so.1 -> libanl-2.6.1.so
lrwxrwxrwx  1 root root      20 oct  5  2007 libapparmor.so.1 ->
libapparmor.so.1.0.2
-rwxr-xr-x  1 root root   30404 sep 22  2007 libapparmor.so.1.0.2
lrwxrwxrwx  1 root root      16 oct  5  2007 libattr.so.1 ->
libattr.so.1.1.0
-rw-r--r--  1 root root   18272 sep 21  2007 libattr.so.1.1.0
...
```

La bibliothèque la plus importante du système est la bibliothèque C. Tous les programmes compilés sont liés à libc. Il suffit de supprimer ce fichier (une erreur de débutant) pour faire tomber tout le système.

```
$ ls -l libc.so.6
lrwxrwxrwx 1 root root 13 oct  5  2007 libc.so.6 -> libc-2.6.1.so
```

Les répertoires des bibliothèques contiennent beaucoup de liens symboliques. Ces liens sont là, entre autres, pour gérer les versions et la compatibilité entre les versions. Par exemple quand deux versions cohabitent :

```
$ cd /usr/lib
$ ls -l libXm.*
lrwxrwxrwx 1 root root      14 oct 17  2007 libXm.so.3 -> libXm.so.3.0.3
-rwxr-xr-x 1 root root 2371164 oct 12  2007 libXm.so.3.0.3
lrwxrwxrwx 1 root root      14 oct  5  2007 libXm.so.4 -> libXm.so.4.0.0
-rwxr-xr-x 1 root root 2496528 sep 22  2007 libXm.so.4.0.0
```

3. Quelles bibliothèques liées ?

La commande **ldd** permet de déterminer quelles sont les bibliothèques liées à un programme, et aussi si celles-ci sont présentes ou non.

```
$ ldd pdfedit
        linux-gate.so.1 =>  (0xffffe000)
        libfreetype.so.6 => /usr/lib/libfreetype.so.6 (0xb7ea5000)
        libz.so.1 => /lib/libz.so.1 (0xb7e92000)
        libt1.so.5 => /usr/lib/libt1.so.5 (0xb7e3d000)
        libqt-mt.so.3 => /usr/lib/libqt-mt.so.3 (0xb7739000)
        libXext.so.6 => /usr/lib/libXext.so.6 (0xb772a000)
        libX11.so.6 => /usr/lib/libX11.so.6 (0xb760f000)
        libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0xb7520000)
        libm.so.6 => /lib/libm.so.6 (0xb74fb000)
        libgcc_s.so.1 => /lib/libgcc_s.so.1 (0xb74ef000)
        libc.so.6 => /lib/libc.so.6 (0xb73bc000)
        libpthread.so.0 => /lib/libpthread.so.0 (0xb73a5000)
        libpng12.so.0 => /usr/lib/libpng12.so.0 (0xb7380000)
        libXi.so.6 => /usr/lib/libXi.so.6 (0xb7376000)
        libXrender.so.1 => /usr/lib/libXrender.so.1 (0xb736d000)
        libXrandr.so.2 => /usr/lib/libXrandr.so.2 (0xb7366000)
        libXcursor.so.1 => /usr/lib/libXcursor.so.1 (0xb735c000)
        libXinerama.so.1 => /usr/lib/libXinerama.so.1 (0xb7358000)
        libXft.so.2 => /usr/lib/libXft.so.2 (0xb7345000)
        libfontconfig.so.1 => /usr/lib/libfontconfig.so.1 (0xb7318000)
        libSM.so.6 => /usr/lib/libSM.so.6 (0xb730f000)
        libICE.so.6 => /usr/lib/libICE.so.6 (0xb72f6000)
        libdl.so.2 => /lib/libdl.so.2 (0xb72f2000)
        libXau.so.6 => /usr/lib/libXau.so.6 (0xb72ee000)
        libxcb-xlib.so.0 => /usr/lib/libxcb-xlib.so.0 (0xb72ea000)
        libxcb.so.1 => /usr/lib/libxcb.so.1 (0xb72d1000)
        /lib/ld-linux.so.2 (0xb7f3c000)
        libXfixes.so.3 => /usr/lib/libXfixes.so.3 (0xb72cb000)
        libexpat.so.1 => /lib/libexpat.so.1 (0xb72aa000)
```

Prenez maintenant un cas où une bibliothèque est manquante (elle a été volontairement déplacée pour les besoins de la démonstration) :

```
$ ldd /usr/bin/esd
        linux-gate.so.1 =>  (0xffffe000)
        libwrap.so.0 => /lib/libwrap.so.0 (0xb7f6d000)
        libesd.so.0 => not found
        libasound.so.2 => /usr/lib/libasound.so.2 (0xb7eb1000)
        libaudiofile.so.0 => /usr/lib/libaudiofile.so.0 (0xb7e8e000)
        libm.so.6 => /lib/libm.so.6 (0xb7e69000)
        libc.so.6 => /lib/libc.so.6 (0xb7d36000)
        libdl.so.2 => /lib/libdl.so.2 (0xb7d31000)
        libpthread.so.0 => /lib/libpthread.so.0 (0xb7d1a000)
        /lib/ld-linux.so.2 (0xb7f9f000)
```

La bibliothèque libesd.so.0 est manquante. Il est impossible de lancer le programme :

```
$ ./esd
./esd: error while loading shared libraries: libesd.so.0: cannot
open shared object file: No such file or directory
```

4. Configurer le cache de l'éditeur de liens

L'édition des liens avec une bibliothèque partagée est dynamique et se fait au moment de l'exécution du programme par le système à l'aide de la bibliothèque ld.so. Le binaire fournit le nom des bibliothèques à lier à l'exécution, mais pas le chemin. Les fonctions de **ld.so** déterminent en fonction de son nom la bibliothèque à utiliser parmi les chemins qu'elles connaissent.

Tout programme est lié à la bibliothèque ld.so ou plutôt **ld-linux.so** (ld-linux.so.2).

Le chargeur de liens ld.so recherche les bibliothèques dans plusieurs endroits dont, et dans cet ordre :

- les chemins précisés dans la variable d'environnement **LD_LIBRARY_PATH**. Les chemins sont séparés, comme pour PATH, par des ":" ;
- le contenu du fichier **/etc/ld.so.cache** qui contient une liste compilée (format binaire) des bibliothèques trouvées dans les chemins prédéfinis ;
- les répertoires **/lib** et **/usr/lib**.

La recherche dans **/lib** et **/usr/lib** est implicite. De même, le fait de remplir la variable **LD_LIBRARY_PATH** n'empêche en rien la recherche des bibliothèques aux autres endroits si elle n'est pas dans un des chemins de la liste.

Pour éviter la mise en place d'une variable dont le contenu peut être difficile à manipuler, ld.so propose un cache que vous pouvez modifier vous-même. Le cache est construit depuis le contenu du fichier **/etc/ld.so.conf** et de la commande **ldconfig**.

Ce fichier contient la liste des répertoires contenant les bibliothèques partagées :

```
# cat /etc/ld.so.conf
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/lib/Xaw3d
/usr/i386-suse-linux/lib
/usr/local/lib
/opt/kde3/lib
include /etc/ld.so.conf.d/*.conf
```

Plutôt que de modifier ce fichier, un package ou vous-même pouvez décider de rajouter un fichier dans **/etc/ld.so.conf.d** contenant le ou les chemins de vos nouvelles bibliothèques.

Il ne suffit pas de rajouter le chemin : vous devez regénérer le cache avec la commande **ldconfig**.

```
# ldconfig
```

La commande **ldconfig** :

- met à jour le cache pour les chemins définis dans /etc/ld.so.conf et associés, ainsi que pour /usr/lib et /lib ;
- met à jour les liens symboliques sur les bibliothèques ;
- permet aussi de lister les bibliothèques connues dans le cache.

Les options suivantes sont acceptées :

Option	Rôle
-v	Mode bavard : indique ce que ldconfig effectue

-N	Ne reconstruit pas le cache
-X	Ne met pas à jour les liens
-p	Liste le contenu du cache

Pour lister les bibliothèques connues de l'éditeur de liens :

```
# ldconfig -p
1940 libs trouvé dans la cache « /etc/ld.so.cache »
        libzypp.so.324 (libc6) => /usr/lib/libzypp.so.324
        libzvbi.so.0 (libc6) => /usr/lib/libzvbi.so.0
        libzvbi-chains.so.0 (libc6) => /usr/lib/libzvbi-chains.so.0
        libzip.so.1 (libc6) => /usr/lib/libzip.so.1
        libzio.so.0 (libc6) => /usr/lib/libzio.so.0
        libz.so.1 (libc6) => /lib/libz.so.1
        libz.so (libc6) => /usr/lib/libz.so
        liby2util.so.3 (libc6) => /usr/lib/liby2util.so.3
        liby2storage.so.2 (libc6) => /usr/lib/liby2storage.so.2
        liby2.so.2 (libc6) => /usr/lib/liby2.so.2
        libycpvalues.so.3 (libc6) => /usr/lib/libycpvalues.so.3
        libycp.so.3 (libc6) => /usr/lib/libycp.so.3
        libx264gtk.so.54 (libc6) => /usr/lib/libx264gtk.so.54
        libx264gtk.so (libc6) => /usr/lib/libx264gtk.so
...
```

Pour voir ce que ferait ldconfig mais sans rien mettre à jour :

```
# ldconfig -N -X -v
/usr/X11R6/lib:
        libfglrx_pp.so.1.0 -> libfglrx_pp.so.1.0
        libfglrx_gamma.so.1 -> libfglrx_gamma.so.1.0
        libfglrx_tvout.so.1 -> libfglrx_tvout.so.1.0
        libGL.so.1 -> libGL.so.1.2
/usr/local/lib:
/opt/kde3/lib:
        libkdeinit_ksmserver.so -> libkdeinit_ksmserver.so
        libkdeinit_klipper.so -> libkdeinit_klipper.so
        libkdeinit_kcminit.so -> libkdeinit_kcminit.so
        libkdetvvideo.so.0 -> libkdetvvideo.so.0.0.0
        libkmailprivate.so -> libkmailprivate.so
...
```

Enfin pour mettre à jour et voir le résultat :

```
# ldconfig -v
```

(la sortie est la même que la commande précédente).

Le shell bash

1. Rôle

Si les récentes distributions de Linux permettent de faire abstraction de la saisie d'instructions texte en offrant des environnements graphiques attrayants, il est inenvisageable pour un professionnel de Linux de ne pas connaître le fonctionnement de l'interpréteur de commandes et des principales commandes qui lui sont associées.

L'interpréteur de commandes, ou interprète, permet d'exécuter des instructions que vous saisissez au clavier ou au sein d'un script et vous en retourne les résultats. Cet interpréteur est un programme appelé shell. C'est à rapprocher du mot kernel vu précédemment : le kernel, signifiant noyau, est souvent entouré d'une coquille dure (pensez à un noyau d'abricot ou de pêche). shell signifiant coquille, c'est donc ce qui « entoure » le **noyau** Linux : le moyen de l'utiliser à l'aide de commandes. C'est donc une interface fonctionnant en mode texte entre le noyau Linux et les utilisateurs (avancés), voire les applications.

Il existe plusieurs shells, chacun disposant de spécificités propres. Le Bourne Shell (sh) est le shell le plus connu et le plus courant sur les Unix. Le C-Shell (csh) reprend la structure du langage C. Le Korn Shell (ksh) est une évolution du Bourne Shell. Le Z-Shell (zsh) est lui-même une évolution du Korn Shell. Le shell de référence sous Linux se nomme le Bourne Again Shell (bash). Voici une liste non exhaustive d'interpréteurs de commandes que vous pouvez rencontrer sous Linux :

- sh : Thompson Shell (n'existe plus) ;
- sh : Bourne Shell (a remplacé le précédent) ;
- bash : Bourne Again Shell ;
- ksh : Korn Shell ;
- csh : C Shell ;
- zsh : Z Shell ;
- tcsh : Tenex C Shell ;
- ash : A Shell ;
- dash : Debian Almquist Shell.

La liste des shells actuellement présents sur votre installation Linux est présente dans le fichier /etc/shells.

2. Bash : le shell par défaut

a. Un shell puissant et libre

Le bash est un dérivé du Bourne Shell. Bourne est le nom du principal programmeur de ce shell. L'expression Bourne Again est à la fois un clin d'oeil aux origines du bash (Bourne), et un jeu de mots sur « I born again » ce qui signifie « né de nouveau » ou « réincarné ». Le bash reprend sh mais aussi des fonctionnalités de ksh ou csh.

Le bash n'est pas présent que sous Linux. Étant un logiciel libre, il peut être compilé et exécuté sur de nombreuses plates-formes. C'est le shell de référence sur les systèmes MacOS et il existe aussi pour Windows.

Le shell fonctionne au sein d'un terminal. Un terminal est originellement une véritable machine ne disposant que du nécessaire pour saisir des instructions (le clavier) et visualiser les résultats (un écran, voire il y a très longtemps une simple imprimante à papier listing). S'il existe encore de vrais terminaux physiques comme ceux-ci, ils ont été remplacés par des programmes qui émulent des terminaux. On en distingue deux genres sous Linux :

- les consoles virtuelles texte, le mode par défaut de Linux lorsqu'il démarre ou fonctionne sans

environnement graphique ;

- les consoles ou terminaux graphiques, comme xterm, eterm ou konsole, qui sont des émulateurs de terminaux au sein de fenêtres graphiques.

Le shell fonctionne au sein d'un terminal. Il attend des saisies au clavier dans la console ou la fenêtre, et affiche ses résultats au même endroit. Tout utilisateur avancé de Linux et d'Unix en général a au moins un terminal ouvert en quasi-permanence. L'ouverture d'un terminal (ou console, dans ce cas ces mots sont synonymes) lance automatiquement le shell par défaut.

b. L'invite de commande

Le shell attend des entrées au clavier sur une ligne appelée l'invite de commande ou prompt. Un curseur, qui peut être représenté par un rectangle fixe, clignotant ou un caractère souligné, indique la position actuelle de votre saisie.

L'invite (prompt) fournit des informations sur le terminal et votre position dans le système de fichiers.

```
seb@slyserver:/home/public>
```

Dans cette invite tout à fait classique, vous trouvez quatre informations :

- seb : c'est le nom de connexion, ou login de l'utilisateur, actuellement connecté au terminal ;
- slyserver : c'est le nom d'hôte (hostname), le nom logique de la machine raccordée au terminal ;
- /home/public : c'est la position actuelle du shell dans le système de fichiers ;
- > : c'est la terminaison standard du bash pour un utilisateur sans pouvoirs.

Cette invite vous informe que c'est l'utilisateur sans pouvoirs d'administration seb qui utilise le terminal (est connecté) sur la machine slyserver et qu'il est actuellement positionné dans /home/public.

Le caractère de terminaison peut avoir d'autres significations :

- un $ indique que l'utilisateur n'a pas de pouvoirs particuliers, comme le >.
- un # indique que l'utilisateur est l'administrateur root qui a tous les pouvoirs.

Le chemin peut varier :

seb@slyserver:~> : le caractère tilde ~ indique que vous êtes dans votre répertoire personnel.

seb@slyserver:~/test> : le ~ se rapportant à votre répertoire personnel, vous êtes dans le répertoire test au sein de celui-ci.

Dans la suite, l'invite de commande sera généralement remplacé par un simple dollar, $, ceci afin de gagner de la place sur la ligne de commande.

3. Utiliser le shell

a. La saisie

Dans le terminal, le clavier s'utilise comme d'habitude. Vous pouvez vous déplacer sur la ligne avec les flèches de droite et de gauche du clavier et effacer des caractères avec les touches [Retour arrière] et [Suppr]. Vous lancez l'exécution de la commande que vous avez saisi en appuyant sur la touche [Entrée].

Il est temps de tester quelques commandes. La commande **date** indique la date et l'heure actuelles. Vous n'obtiendrez évidemment pas le même résultat, et pas toujours dans la même langue, selon votre installation Linux.

```
$ date
ven fév 22 21:28:12 CET 2008
```

Une commande pratique, **pwd**, permet de savoir à quel endroit vous vous situez dans les répertoires.

```
$ pwd
/home/seb
```

Le shell indique qu'il est actuellement positionné dans le répertoire /home/seb.

b. Syntaxe générale des commandes

Les commandes ou instructions (les deux mots sont synonymes dans ce cas) GNU ont très souvent une syntaxe reprenant la même structure :

```
Commande [paramètres] [arguments]
```

Une commande peut avoir ni paramètres, ni arguments. Dans ce cas elle exécute l'action par défaut pour laquelle elle est programmée, ou affiche un message d'erreur si ceux-ci sont nécessaires.

Un paramètre est une option de la commande. Les deux mots sont ici synonymes. C'est souvent une simple lettre ou un simple chiffre précédé d'un tiret : `-l`, `-p`, `-s`, etc. Si la commande accepte plusieurs paramètres, vous les saisissez les uns après les autres en les séparant par des espaces : `-l -r -t`, ou en écrivant qu'un seul tiret puis tous les paramètres : `-lrt`. Les deux syntaxes sont acceptées et produisent le même résultat. La seconde est seulement plus courte.

Dans certains cas un paramètre nécessite un argument, par exemple un nom de fichier. Dans ce cas il est préférable de séparer ce paramètre des autres : `-lrt -f monfichier`.

Les arguments sont les entités sur lesquelles la commande doit exécuter son action. Leur type dépend de la commande. Ce peut être un fichier, du texte, des nombres, etc.

c. Premier exemple concret avec cal

Prenez l'exemple de la commande **cal**. Celle-ci admet plusieurs paramètres et arguments. Appelée seule, elle affiche le calendrier du mois en cours et surligne le jour actuel.

```
$ cal
    février 2008
di lu ma me je ve sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

La commande **cal** admet deux arguments optionnels. Si un seul est précisé, il s'agit de l'année, et l'intégralité du calendrier de cette année est affichée. Si deux arguments sont précisés, le premier est le mois, le second l'année.

```
$ cal 12 1975
   décembre 1975
di lu ma me je ve sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

La commande admet aussi quelques paramètres. Vous remarquez que par défaut l'affichage est prévu pour les anglo-saxons : la première colonne est un dimanche, représentant le premier jour de la semaine. En France c'est le lundi. Le paramètre `-m` permet de le préciser :

```
$ cal -m 12 1975
   décembre 1975
lu ma me je ve sa di
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

```
29 30 31
```

Un second paramètre `-3` permet d'afficher les mois précédant et suivant le mois précisé (ou le mois en cours).

```
seb@slyserver:~> cal -m -3 12 1975
   novembre 1975         décembre 1975          janvier 1976
lu ma me je ve sa di  lu ma me je ve sa di  lu ma me je ve sa di
                1  2   1  2  3  4  5  6  7            1  2  3  4
 3  4  5  6  7  8  9   8  9 10 11 12 13 14   5  6  7  8  9 10 11
10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
24 25 26 27 28 29 30  29 30 31              26 27 28 29 30 31
```

Et comme vous pouvez grouper les paramètres la commande suivante produit le même résultat.

```
$ cal -m3 12 1975
```

d. Chaîner les commandes

Vous pouvez exécuter plusieurs commandes sur une seule ligne, les unes après les autres. Pour cela il suffit de les séparer avec un point-virgule.

```
$ date;pwd;cal
lun fév 25 22:29:09 CET 2008
/usr/share/man/man9
    février 2008
di lu ma me je ve sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

e. Afficher du texte avec echo

Il n'y a rien de plus simple que d'afficher du texte. La commande **echo** est là pour ça. Comme presque toutes les commandes, elle accepte, outre les arguments sous forme de texte, des paramètres. Pour afficher un texte simple :

```
$ echo Bonjour les amis
Bonjour les amis
```

Vous pouvez placer le texte entre guillemets ou simples apostrophes (touche 4) pour clarifier et regrouper le texte à afficher. Vous verrez plus loin que ceci a une signification particulière.

Remarquez que par défaut votre texte s'affiche et echo effectue seul un retour chariot pour passer à la ligne. Vous pouvez modifier votre texte pour y rajouter des séquences de caractères ayant une action particulière. Si vous connaissez le langage C, ce sont les mêmes. Seules les plus utilisés sont listés ici :

Séquence	Action
\n	Passage à la ligne
\t	Tabulation horizontale
\c	Supprimer le saut de ligne final
\b	Retour d'un caractère en arrière
\\	Afficher l'antislash (barre oblique inverse)
\nnn	Afficher le caractère spécifié en octal

Pour utiliser ces séquences rajoutez l'argument `-e` :

```
$ echo -e "Salut.\tJe m'appelle Seb\b\b\bPersonne\n"
Salut. Je m'appelle Personne
```

f. Commandes internes et externes

Il existe deux types de commandes :

- Les commandes externes sont des programmes binaires présents en tant que fichiers sur votre disque dur (ou tout autre support de données). Quand vous exécutez la commande, ce fichier est chargé en mémoire et lancé en tant que processus (cette notion sera expliquée dans ce même chapitre).

- Les commandes internes sont internes au shell et exécutées au sein de celui-ci. Ces commandes font partie du programme shell, le bash. Les commandes **cd** ou **pwd** sont deux exemples de commandes internes. Quand vous les exécutez, le shell exécute les fonctions définies en son sein correspondant à celles-ci.

Vous pouvez distinguer une commande interne d'une commande externe à l'aide de la commande interne **type**. C'est ainsi que **date** est une commande externe, vous constatez que c'est un fichier présent dans **/bin**, tandis que **pwd** est interne au shell.

```
$ type date
date is hashed (/bin/date)
$ type pwd
pwd is a shell builtin
```

Vous pouvez tomber sur certains autres types comme les alias de commandes qui sont des raccourcis de commandes propres au shell. Ainsi le shell bash de certaines distributions Linux proposent des alias comme ll qui correspond en fait à ls -l.

```
$ type ll
ll is aliased to `ls -l'
```

g. Quelques raccourcis utiles

Quelques séquences de raccourcis de commandes sont à connaître :

- **[Ctrl] C** : interruption du programme : il se termine.

- **[Ctrl] Z** : stoppe le programme (voir les processus).

- **[Ctrl] D** : interrompt une saisie sur un prompt >.

4. Rappel de l'historique

Vous trouverez très utile de pouvoir rappeler une commande que vous avez déjà exécutée en naviguant dans l'historique des commandes avec les touches [Flèche en haut] et [Flèche en bas]. La flèche du haut remonte dans l'historique. Si vous avez saisi les deux précédentes commandes (**date** puis **pwd**), le premier appui sur la flèche du haut affiche la ligne de commande **pwd**, un second la commande **date**. La flèche du bas navigue dans l'autre sens, jusqu'à l'invite d'origine. Si vous appuyez sur la touche [Entrée] vous lancez de nouveau la commande.

Plus vous tapez des commandes, plus l'historique s'agrandit. Le shell conserve ainsi un grand nombre d'entrées dans l'historique (le nombre de lignes conservées peut être modifié). Cet historique est conservé dans un fichier caché de votre répertoire personnel appelé .bash_history. Vous pouvez voir le contenu de l'historique avec la commande **history**. Le résultat suivant est volontairement tronqué, la liste étant trop longue.

```
$ history
...
 1000  date
 1001  pwd
 1002  uname -a
```

```
 1003  ls
 1004  fc -l -5
 1005  history
```

La commande **fc** effectue presque la même chose lorsqu'on utilise le paramètre `-l`. Par défaut elle se limite aux quinze dernières commandes. Aussi vous pouvez lui passer le nombre des dernières commandes, comme ceci pour les dix dernières :

```
$ fc -l -10
995      ssh -X seb@192.168.1.130
996      fc -l
997      fc -l -20
998      ls
999      pwd
1000     cd
1001     uname -a
1002     fc -l
1003     cat /etc/passwd
1004     ls -lrtR
```

Vous pouvez rappeler une commande avec fc et le paramètre `-s` suivi du numéro de la commande. Elle sera alors automatiquement lancée.

```
$ fc -s 1001
uname -a
Linux slyserver 2.6.22.17-0.1-default #1 SMP 2008/02/10 20:01:04 UTC
x86_64 x86_64 x86_64 GNU/Linux
```

Enfin, vous pouvez remplacer un élément de la commande par un autre avant de la lancer. Par exemple, vous voulez remplacer `fc` par `ls` dans l'entrée 1002 de l'historique :

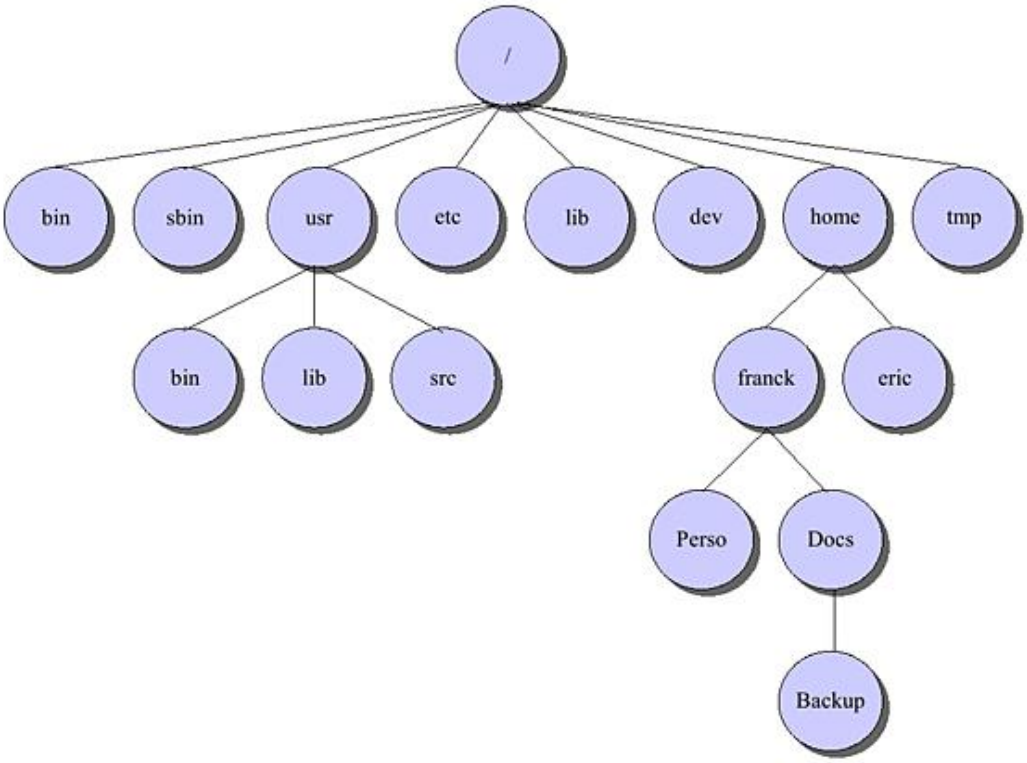
```
$ fc -s fc=ls 1002
ls -l
total 775944
-rw-r--r--  1 seb users     15391 mai 14  2007 AR-1179161176460.pdf
...
```

La gestion des fichiers

1. Le système de fichiers

Un système de fichiers, appelé communément File System ou FS, définit l'organisation des données sur un support de stockage, donc comment sont gérés et organisés les fichiers par le système d'exploitation.

Linux est, comme tout Unix, un système d'exploitation entièrement orienté fichier. Tout (ou presque) est représenté par un fichier, tant les données (fichiers de données de tout type comme une image ou un programme), que les périphériques (terminaux, souris, clavier, carte son, etc.) ou encore les moyens de communication (sockets, tubes nommés, etc.). On peut dire que le système de fichier est le coeur de tout système Unix.



Exemple d'arborescence Linux

Le système de fichiers de Linux est hiérarchique. Il décrit une arborescence de répertoires et de sous-répertoires, en partant d'un élément de base appelé la racine ou root directory.

2. Les divers types de fichiers

On distingue trois types de fichiers : ordinaires, catalogue, spéciaux.

a. Les fichiers ordinaires ou réguliers

Les fichiers ordinaires sont aussi appelés fichiers réguliers, ordinary files ou regular files. Ce sont des fichiers tout à fait classiques qui contiennent des données. Par données, comprenez n'importe quel contenu :

- texte ;
- image ;
- audio ;

- programme binaire compilé ;
- script ;
- base de données ;
- bibliothèque de programmation, etc.

Par défaut, rien ne permet de différencier les uns des autres, sauf à utiliser quelques options de certaines commandes (ls -F par exemple) ou la commande **file**.

```
$ file nom_fic
nom fic : 32 Bits ELF Executable Binary (stripped)
```

La notion d'extension de fichier comme composante interne de la structure du système de fichier est inconnue de Linux. Autrement dit une extension n'a aucun rôle au niveau du système de fichier et est simplement considérée comme une partie du nom du fichier. Elle sert simplement à distinguer visuellement et rapidement l'éventuel contenu d'un fichier par rapport à un autre.

Comme les extensions ne sont pas gérées par Linux, le nom d'un programme ne finit quasiment jamais par un « .exe », il faudra trouver autre chose pour le distinguer.

b. Les catalogues

Les fichiers catalogues sont les répertoires, dossiers ou directory. Les répertoires permettent d'organiser le disque dur en créant une hiérarchie. Un répertoire peut contenir des fichiers normaux, des fichiers spéciaux et d'autres répertoires, de manière récursive.

Un répertoire n'est rien d'autre qu'un fichier particulier contenant la liste des fichiers eux-mêmes présents dans ce répertoire. Cette notion se révélera très utile lorsque la question des droits sera abordée.

c. Les fichiers spéciaux

Le troisième type de fichier est le fichier spécial. Il existe plusieurs genres de fichiers spéciaux. Ils se trouvent principalement dans le répertoire /dev s'ils représentent des périphériques.

Ce sont principalement des fichiers servant d'interface pour les divers périphériques. Ils peuvent s'utiliser, suivant le cas, comme des fichiers normaux. Un accès en lecture ou écriture sur ces fichiers est directement redirigé vers le périphérique (en passant par le pilote associé s'il existe). Par exemple si vous redirigez un fichier d'onde sonore (wave) vers le fichier représentant la sortie de la carte son, il y a de fortes chances que ce son soit audible par vos haut-parleurs.

3. Nomenclature des fichiers

On ne peut pas donner n'importe quel nom à un fichier, il faut pour cela suivre quelques règles simples. Ces règles sont valables pour tous les types de fichiers.

Sur les anciens systèmes Unix un nom de fichier ne pouvait pas dépasser 14 caractères. Sur les systèmes actuels, dont Linux, on peut aller jusqu'à 255 caractères. L'éventuelle extension est comprise dans la longueur du nom du fichier.

Un point extrêmement important : Linux fait la distinction entre les noms de fichiers en minuscules et en majuscules. Toto, TOTO, ToTo et toto sont des noms de fichiers différents, avec un contenu différent.

La plupart des caractères (les chiffres, les lettres, les majuscules, les minuscules, certains signes, les caractères accentués) sont acceptés, y compris l'espace. Cependant quelques caractères sont à éviter car ils ont une signification particulière au sein du shell : & ; ( ) ~ <espace> \ / | ` ? - (en début de nom).

Les noms suivants sont valides :

- Fichier1
- Paie.txt

- 123traitement.sh
- Paie_juin_2002.xls
- 8

Ces noms peuvent poser des problèmes :

- Fichier*
- Paie(decembre)
- Ben&Nuts
- Paie juin 2002.xls
- -f

4. Les chemins

a. Structure et nom de chemin

Les chemins permettent de définir un emplacement au sein du système de fichiers. C'est la liste des répertoires et sous-répertoires empruntés pour accéder à un endroit donné de l'arborescence jusqu'à la position souhaitée (répertoire, fichier). Un nom de fichier est ainsi généralement complété par son chemin d'accès. C'est ce qui fait que le fichier toto du répertoire rep1 est différent du fichier toto du répertoire rep2. Le FS d'Unix étant hiérarchique, il décrit une arborescence.

Le schéma présenté dans la section La gestion des fichiers - Le système de fichiers de ce chapitre représente une arborescence d'un système de fichier Linux. Le / situé tout en haut s'appelle la racine ou root directory (à ne pas confondre avec le répertoire de l'administrateur root). Le nom de chemin ou path name d'un fichier est la concaténation, depuis la racine, de tous les répertoires qu'il est nécessaire de traverser pour y accéder, chacun étant séparé par le caractère /. C'est un chemin absolu comme celui-ci.

```
/home/toto/Docs/Backup/fic.bak
```

Un chemin absolu ou complet :

- démarre de la racine, donc commence par un /,
- décrit tous les répertoires à traverser pour accéder à l'endroit voulu,
- ne contient pas de . ni de ..

b. Répertoire personnel

Lors de la création d'un utilisateur, l'administrateur lui alloue un répertoire personnel appelé home directory. Lorsqu'il se connecte, l'utilisateur arrive directement dans ce répertoire, qui est son répertoire personnel. C'est dans ce répertoire que l'utilisateur pourra créer ses propres fichiers et répertoires.

```
Login : seb
Password : xxxxxxxxxx
$ pwd
/home/seb
```

c. Chemin relatif

Un nom de chemin peut aussi être relatif à sa position courante dans le répertoire. Le système (ou le shell) mémorise la position actuelle d'un utilisateur dans le système de fichier, le répertoire actif. Vous pouvez accéder à un autre

répertoire de l'arborescence depuis l'emplacement actuel sans taper le chemin complet uniquement en précisant le chemin le plus court relativement à votre position actuelle au sein de l'arborescence.

Il faut pour cela souvent utiliser deux entrées particulières de répertoires :

- Le point . représente le répertoire courant, actif. Il est généralement implicite.
- Les doubles points .. représentent le répertoire de niveau inférieur.

Un chemin relatif :

- décrit un chemin relatif à une position donnée dans l'arborescence, généralement (mais pas toujours) depuis la position courante ;
- décrit en principe le plus court chemin pour aller d'un point à un autre ;
- peut contenir des points ou des doubles points ;

Ces trois affirmations ne sont pas des obligations :

- `/usr/local/bin` est un chemin complet ou absolu ;
- `Documents/Photos` est un chemin relatif : le répertoire Documents est considéré comme existant dans le répertoire courant ;
- `./Documents/Photos` est un chemin relatif parfaitement identique au précédent, sauf que le répertoire actif (courant) est explicitement indiqué par le point. « ./Documents » indique explicitement le répertoire Documents dans le répertoire actif ;
- `/usr/local/../bin` est un chemin relatif : les .. sont relatifs à /usr/local et descendent d'un niveau vers /usr. Le chemin final est donc /usr/bin.

d. Le tilde

Le bash interprète le caractère tilde ~ comme un alias du répertoire personnel. Les chemins peuvent être relatifs au tilde, mais le tilde ne doit être précédé d'aucun caractère. Pour vous déplacer dans le répertoire tmp de votre dossier personnel d'où que vous soyez :

```
$ cd ~/tmp
```

Si vous entrez ceci, vous obtenez une erreur :

```
$ cd /~
```

e. cd

Pour vous déplacer dans les répertoires, vous utilisez la commande **cd** (*change directory*). La commande **pwd** (*print working directory*) que vous avez déjà rencontrée affiche le chemin complet du répertoire courant.

Si vous saisissez cd ., vous ne bougez pas. Le point sera très utile lorsque vous devrez spécifier des chemins explicites à des commandes situées dans le répertoire où vous êtes positionné.

Le cd .. remonte d'un niveau. Si vous étiez dans /home/seb, vous vous retrouvez dans home.

La commande cd sans argument permet de retourner directement dans son répertoire utilisateur.

Voici un petit exemple. L'utilisateur seb démarre de son répertoire personnel. Il se déplace via un chemin relatif vers /home/public. Le .. remonte vers /home, donc ../public se déplace dans /home/public. De là, via un chemin complet, il se dirige vers /usr/local/bin, puis décide à l'aide d'un chemin relatif de se rendre dans /usr/lib : le premier .. descend vers /usr/local, le second vers /usr, puis remonte vers /usr/lib. Enfin seb retourne dans son répertoire personnel avec cd sans argument. L'invite est ici donnée complète pour une meilleure compréhension.

```
seb@slyserver:~> pwd
/home/seb
```

```
seb@slyserver:~> cd ../public
seb@slyserver:/home/public> cd /usr/local/bin
seb@slyserver:/usr/local/bin> cd ../../lib
seb@slyserver:/usr/lib> cd
seb@slyserver:~>
```

5. Les commandes de base

a. Lister les fichiers et les répertoires

La commande **ls** permet de lister le contenu d'un répertoire (catalogue) en lignes ou colonnes. Elle supporte plusieurs paramètres dont voici les plus pertinents.

Paramètre	Signification
-l	Pour chaque fichier ou dossier, fournit des informations détaillées.
-a	Les fichiers cachés sont affichés (ils commencent par un point).
-d	Sur un répertoire, précise le répertoire lui-même et non son contenu.
-F	Rajoute un caractère à la fin du nom pour spécifier le type : / pour un répertoire, * pour un exécutable, @ pour un lien symbolique, etc.
-R	Si la commande rencontre des répertoires, elle rentre dans les sous-répertoires, sous-sous-répertoires, etc., de manière récursive.
-t	La sortie est triée par date de modification du plus récent au plus ancien. Cette date est affichée.
-c	Affiche / tri (avec -t) par date de changement d'état du fichier.
-u	Affiche / tri (avec -t) par date d'accès du fichier.
-r	L'ordre de sortie est inversé.
-i	Affiche l'inode du fichier.
-C	L'affichage est sur plusieurs colonnes (par défaut).
-1	L'affichage est sur une seule colonne.

Le paramètre qui vous fournit le plus d'informations est le `-l` : il donne un certain nombre de détails sur les fichiers.

```
$ ls -l
total 4568
-rw-r--r-- 1 seb users   69120 sep  3  2006 3i_rattrapage_2006.doc
-rw-r--r-- 1 seb users    9632 sep  3  2006 3i_rattrapage_2006.odt
-rw-r--r-- 1 seb users    6849 nov 17  2003 controle_1I2_mardi.sxw
...
```

La ligne total indique la taille totale en kilo-octets du contenu du répertoire. Cette taille est celle de l'ensemble des fichiers ordinaires du répertoire et ne prend pas en compte les éventuels sous-répertoires et leur contenu (pour ceci, il faudra utiliser la commande du).

Vient ensuite la liste détaillée de tout le contenu.

-rw-r--r--	1	seb	users	69120	sep 3 2006	3i_rattrapage_2006.doc
1	2	3	4	5	6	7

- 1 : Le premier caractère représente le type de fichier (- : ordinaire, d : répertoire, l : lien symbolique...) ; les autres, par blocs de trois, les droits pour l'utilisateur (rw-), le groupe (r--) et tous (r--). Les droits sont expliqués au chapitre Les disques et le système de fichiers.
- 2 : Un compteur de liens (chapitre Les disques et le système de fichiers).
- 3 : Le propriétaire du fichier, généralement celui qui l'a créé.
- 4 : Le groupe auquel appartient le fichier.
- 5 : La taille du fichier en octets.
- 6 : La date de dernière modification (parfois avec l'heure), suivant le paramètre (t, c, u).
- 7 : Le nom du fichier.

Vous pouvez trouver très utile de pouvoir lister vos fichiers de manière à ce que ceux modifiés le plus récemment soient affichés en fin de liste. Ainsi en cas de présence d'un très grand nombre de fichiers, cela vous évite de remonter tout en haut de la console. Le tri par date de modification se fait avec `-t` et dans l'ordre inverse avec `-r`. Rajoutez-y les détails avec `-l`.

```
$ ls -lrt
-rw-r--r-- 1 seb users   66107 jan  9 17:24 Partiel_1_1I_2008.pdf
-rw-r--r-- 1 seb users   13777 jan 10 17:58 partiel_3I_ppa_2007.odt
-rw-r--r-- 1 seb users   64095 jan 10 17:58 partiel_3I_ppa_2007.pdf
-rw-r--r-- 1 seb users  100092 fév 22 22:21 cours_shell_unix.odt
```

`ls -l -r -t` est strictement identique à `ls -lrt` comme indiqué dans la syntaxe générale des commandes.

Un moyen mnémotechnique de se rappeler cette séquence d'arguments est de l'utiliser sous sa forme `-rtl` (l'ordre des arguments n'a pas d'importance ici) et de penser ainsi à la célèbre radio.

b. Gérer les fichiers et les répertoires

Créer des fichiers vides

Pour vos tests ou durant vos actions vous pouvez avoir besoin de créer des fichiers vides. Une commande pratique pour cela est **touch**. Utilisée avec uniquement le nom d'un fichier en argument, elle crée un fichier avec une taille nulle.

```
$ touch fictest
$ ls -l fictest
-rw-r--r-- 1 seb users 0 fév 29 15:13 fictest
```

La création de fichiers vides n'est pas à l'origine le principal usage de touch. Si vous relancez la même commande sur le fichier, vous remarquez que la date de modification a changé. Le manuel de touch vous informera qu'il est ainsi possible de modifier complètement l'horodatage d'un fichier. Ceci peut être utile pour forcer les sauvegardes incrémentales sur des fichiers.

Créer des répertoires

La commande **mkdir** (*make directory*) permet de créer un ou plusieurs répertoires, ou une arborescence complète. Par défaut la commande ne crée pas d'arborescence. Si vous passez comme arguments rep1/rep2 et que rep1 n'existe pas, la commande retourne une erreur. Dans ce cas, utilisez le paramètre `-p`.

```
mkdir [-p] rep1 [rep2] ... [repn]
```

```
$ mkdir Documents
$ mkdir Documents/Photos
$ mkdir -p Archives/vieilleries
$ ls -R
.:
```

```
Archives  Documents  fictest

./Archives:
vieilleries

./Archives/vieilleries:

./Documents:
Photos
```

Supprimer des répertoires

La commande **rmdir** (*remove directory*) supprime un ou plusieurs répertoires. Elle ne peut pas supprimer une arborescence. Si des fichiers sont encore présents dans le répertoire, la commande retourne une erreur. Le répertoire ne doit donc contenir ni fichiers ni répertoires et ceci même si les sous-répertoires sont eux-mêmes vides.

```
rmdir rep1 [rep2] ... [repn]
```

Il n'y a pas de paramètre `-r` (pour récursif) à la commande **rmdir**. Pour supprimer une arborescence vous devrez utiliser la commande **rm**.

```
$ rmdir Documents/
rmdir: Documents/: Le répertoire n'est pas vide.
$ rmdir Documents/Photos
$
```

Copier des fichiers

La commande **cp** (copy) copie un ou plusieurs fichiers vers un autre fichier ou vers un répertoire.

```
cp fic1 [fic2 ... ficn] Destination
```

Dans le premier cas, fic1 est recopié en Destination. Si Destination existe, il est écrasé sans avertissement selon le paramètre passé et selon les droits. Dans le second cas, fic1, fic2 et ainsi de suite sont recopiés dans le répertoire Destination. Les chemins peuvent êtres absolus ou relatifs. La commande peut prendre, entre autres, les options suivantes :

Paramètre	Signification
-i	Demande de confirmation de copie pour chaque fichier.
-r	Récursif : copie un répertoire et tout son contenu.
-p	Les permissions et dates sont préservées.
-f	Forcer la copie.

Votre attention doit être attirée sur le fonctionnement de cp avec les copies de répertoires. Le fonctionnement est différent selon la présence du répertoire de destination ou non. Dans le premier cas, rep2 n'existe pas. Le répertoire rep1 est copié en rep2. À la fin rep2 est une copie exacte de rep1.

```
$ ls -d rep2
ls: ne peut accéder rep2: Aucun fichier ou répertoire de ce type
$ cp -r rep1 rep2
$ ls
rep1  rep2
```

Maintenant que rep2 existe, exécutez de nouveau la commande cp. Cette fois, comme rep2 existe, il n'est pas écrasé comme vous pourriez le penser. La commande détermine que la destination étant le répertoire rep2, rep1 doit être copiée dans la destination : rep1 est copié dans rep2.

```
$ cp -r rep1 rep2
$ ls rep2
rep1
```

Déplacer et renommer un fichier

La commande **mv** (*move*) permet de déplacer, de renommer un fichier, ou les deux en même temps. Elle fonctionne comme la commande cp. Les paramètres `-f` et `-i` ont le même effet. Avec les trois commandes mv successives suivantes :

- txt1 est renommé en txt1.old ;
- txt2 est déplacé dans rep1 ;
- txt3 est déplacé dans rep1 et renommé en txt3.old.

```
$ touch txt1 txt2 txt3
$ mv txt1 txt1.old
$ mv txt2 rep1/txt2
$ mv txt3 rep1/txt3.old
```

Notez l'existence du paramètre `-u` : si le fichier de destination existe avec une date plus récente, cela vous évite de l'écraser.

Supprimer un fichier ou une arborescence

La commande **rm** (*remove*) supprime un ou plusieurs fichiers, et éventuellement une arborescence complète, suivant les options. La suppression est définitive.

```
rm [Options] fic1 [fic2...]
```

Les options sont classiques mais vu la particularité et la dangerosité de la commande il est bon de faire un rappel.

Paramètre	Signification
-i	La commande demandera une confirmation pour chacun des fichiers à supprimer. Suivant la version d'Unix, le message change et la réponse aussi : y, Y, O, o, N, n, parfois toutes.
-r	Le paramètre suivant attendu est un répertoire. Dans ce cas, la suppression est récursive : tous les niveaux inférieurs sont supprimés, les répertoires comme les fichiers.
-f	Force la suppression.

Dans l'ordre, les commandes suivantes suppriment un simple fichier, suppriment un répertoire, et une arborescence de manière forcée :

```
$ rm fic1
$ rm -r rep1
$ rm -rf /home/public/depots
```

L'utilisation combinée des paramètres `-r` et `-f` bien que très utile et pratique est très dangereuse, notamment en tant que root. Aucune confirmation ne vous est demandée. À moins d'utiliser des outils de récupération de données spécifiques, chers et peu performants, vos données sont irrémédiablement perdues.

Voici une astuce. Vous pouvez créer des fichiers qui commencent par un tiret. Mais avez-vous essayé de les supprimer avec rm ?

```
$ >-i # voir les redirection
$ rm -i
rm: opérande manquante
Pour en savoir davantage, faites: « rm --help ».
```

Il est impossible de supprimer le fichier « -i » de cette manière car rm l'interprète comme un paramètre et non comme un argument. Aussi faut-il ruser. Il y a deux solutions :

- Utiliser l'option `GNU --` signifiant la fin des paramètres et le début des arguments.

- Rajouter un chemin, relatif ou complet, avant le tiret.

Cette dernière solution a l'avantage d'être standard. Les deux lignes sont équivalentes :

```
$ rm -- -i
$ rm ./-i
```

Les liens symboliques

Vous pouvez créer des liens, qui sont un peu comme des raccourcis. Un lien est un fichier spécial contenant comme information un chemin vers un autre fichier. C'est une sortie d'alias. Il existe deux types de liens : le lien dur (hard link) que vous verrez plus loin, lors de l'étude des systèmes de fichiers, et le lien symbolique (soft link) qui correspond à la définition donnée.

Il est possible de créer des liens symboliques vers n'importe quel type de fichier, quel qu'il soit et où qu'il soit. La commande de création des liens symboliques ne vérifie pas si le fichier pointé existe. Il est même possible de créer des liens sur des fichiers qui n'existent pas.

```
ln -s fichier lien
```

Le cas échéant le lien se comportera à l'identique du fichier pointé avec les mêmes permissions et les mêmes propriétés :

- si le fichier pointé est un programme, lancer le lien lance le programme ;
- si le fichier pointé est un répertoire, un cd sur le lien rentre dans ce répertoire ;
- si le fichier pointé est un fichier spécial (périphérique), le lien est vu comme périphérique ;
- etc.

Le seul cas où le lien symbolique se détache du fichier pointé est la suppression. La suppression d'un lien symbolique n'entraîne que la suppression de ce lien, pas du fichier pointé. La suppression du fichier pointé n'entraîne pas la suppression des liens symboliques associés. Dans ce cas le lien pointe dans le vide.

```
$ touch fic1
$ ln -s fic1 lienfic1
$ ls -l
-rw-r--r-- 1 seb users    0 mar  4 19:16 fic1
lrwxrwxrwx 1 seb users    4 mar  4 19:17 lienfic1 -> fic1
$ ls -F
fic1  lienfic1@
$ echo titi>fic1
$ cat lienfic1
titi
```

Cet exemple montre bien qu'un lien symbolique est en fait un fichier spécial de type « l » pointant vers un autre fichier. Notez dans la liste détaillée la présence d'une flèche indiquant sur quel fichier pointe le lien. On distingue le caractère @ indiquant qu'il s'agit d'un lien symbolique lors de l'utilisation du paramètre -F. Si vous disposez d'un terminal en couleur, il est possible que le lien symbolique apparaisse, par convention sous Linux, en bleu ciel. S'il apparaît en rouge, c'est qu'il pointe dans le vide.

Ce n'est pas parce que un lien pointe dans le vide qu'il est forcément mauvais. C'est peut-être fait exprès car il est possible de créer des liens vers des clés USB, des CD-Roms, entre divers systèmes de fichiers, qui peuvent être amovibles. Dans ce cas, le lien redevient actif quand le support est inséré et/ou que la cible est de nouveau présente.

La commande **echo** et le signe **>** seront expliqués plus loin. L'effet est ici l'écriture dans le fichier fic1 de « titi ». La commande **cat** affiche le contenu d'un fichier. Le lien représentant fic1, la sortie est bien celle attendue.

Attention, les droits indiqués sont ceux du fichier spécial et n'ont pas de signification autre : ils ne veulent pas dire que tout le monde à tous les droits sur le fichier pointé. Lors de son utilisation, ce sont les droits du fichier ou du dossier pointés qui prennent le dessus.

---

c. Wildcards : caractères de substitution

Lors de l'utilisation de commandes en rapport avec le système de fichier, il peut devenir intéressant de filtrer la sortie de noms de fichiers à l'aide de certains critères, par exemple avec la commande **ls**. Au lieu d'afficher toute la liste des fichiers, on peut filtrer l'affichage à l'aide de divers critères et caractères spéciaux.

Caractère(s)	Rôle
*	Remplace une chaîne de longueur variable, même vide.
?	Remplace un caractère unique quelconque.
[...]	Une série ou une plage de caractères.
[a-b]	Un caractère parmi la plage indiquée (de a à b inclus).
[!...]	Inversion de la recherche.
[^...]	Idem.

- Soit le contenu suivant :

```
$ ls
afic   afic2  bfic   bfic2  cfic   cfic2  dfic   dfic2
afic1  afic3  bfic1  bfic3  cfic1  cfic3  dfic1  dfic3
```

- Vous obtenez tous les fichiers commençant par a :

```
$ ls a*
afic1  afic2  afic3
```

- Tous les fichiers de quatre caractères commençant par a :

```
$ ls a???
afic
```

- Tous les fichiers d'au moins trois caractères et commençant par b :

```
$ ls b??*
bfic  bfic1  bfic2  bfic3
```

- Tous les fichiers finissant par 1 ou 2 :

```
$ ls *[12]
afic1  afic2  bfic1  bfic2  cfic1  cfic2  dfic1  dfic2
```

- Tous les fichiers commençant par les lettres de a à c, possédant au moins un second caractère avant la terminaison 1 ou 2 :

```
$ ls [a-c]?*[12]
afic1  afic2  bfic1  bfic2  cfic1  cfic2
```

- Tous les fichiers ne finissant pas par 3 :

```
$ ls *[ !3]
```

```
afic  afic1  afic2  bfic  bfic1  bfic2  cfic  cfic1  cfic2  dfic
dfic1  dfic2
```

Interprétation par le shell

C'est le shell qui est chargé d'effectuer la substitution de ces caractères avant le passage des paramètres à une commande. Ainsi lors d'un `$ cp * Documents`, cp ne reçoit pas le caractère * mais la liste de tous les fichiers et répertoires du répertoire actif.

Les wildcards sont utilisables au sein de tous les arguments représentant des fichiers ou des chemins. Ainsi la commande suivante va recopier tous les fichiers README de tous les sous-répertoires de Documents à la position actuelle :

```
$ cp Documents/*/README .
```

d. Verrouillage de caractères

Certains caractères spéciaux doivent être verrouillés, par exemple en cas de caractères peu courants dans un nom de fichier.

- L'**antislash** \ permet de verrouiller un caractère unique. `ls paie\ *.xls` va lister tous les fichiers contenant un espace après paie.
- Les **guillemets** "..." permettent l'interprétation des caractères spéciaux, des variables, au sein d'une chaîne.
- Les **apostrophes** '...' verrouillent tous les caractères spéciaux dans une chaîne ou un fichier.

Rechercher des fichiers

1. Considérations générales

La commande **find** permet de rechercher des fichiers au sein de l'arborescence du système de fichiers à l'aide de critères et donne la possibilité d'agir sur les résultats retournés.

```
find chemin critères options
```

La commande **find** étant récursive, il suffit d'indiquer un répertoire de base pour que toute l'arborescence depuis ce répertoire soit développée. L'option de base est `-print` (souvent implicite sur la plupart des Unix) qui permet d'afficher sur écran les résultats.

```
seb@slyserver:~/Documents/slyunix> find
.
./logos-carre.tif
./logos-carre.eps
./Page 5.pdf
./logos-carre-grand.jpg
./LOGOS
./site_2.jpg
./pub_planete.pdf
./index_logon_inc.php
./logo-iceberg.eps
./flyer
./flyer/sly4.jpg
./flyer/flyerx4.sxd
./flyer/sly1.jpg
./flyer/sly2.jpg
./flyer/flyer.sxd
./flyer/sly3.jpg
./flyer/flyer.jpg
...
```

Le chemin précisé étant relatif, l'affichage est relatif. Si le chemin précisé était absolu, l'affichage aurait été absolu.

2. Critères de recherche

Les paramètres permettent de définir les critères de recherche. Ces critères, s'ils sont plusieurs, sont combinés entre eux par un ET (critère1 ET critère2).

a. -name

`-name` permet une sélection par noms de fichiers. Il est possible d'utiliser les wildcards déjà vus. Le critère est idéalement placé entre guillemets. Ici la liste de tous les fichiers depuis l'emplacement courant et commençant par « fic » est affichée.

```
$ find . name "fic*" -print
./fic1
./fic2
./fic3
./fic4
```

b. -type

`-type` permet une sélection par type de fichier. Vous savez déjà que outre les liens, les répertoires et les fichiers simples, étaient présents d'autres types de fichiers.

Code	Type de fichier

B	Fichier spécial en mode bloc
C	Fichier spécial en mode caractère
D	Répertoire (directory)
F	Fichier ordinaire
L	Lien symbolique
P	Tube nommé (pipe)
S	Socket (Connexion réseau)

Tous les répertoires dont le nom commence par « re » sont affichés.

```
$ find . -name "re*" -type d -print
./rep1
./rep2
```

c. -user et -group

`-user` et `-group` permettent une recherche sur le propriétaire et le groupe d'appartenance des fichiers. Il est possible de préciser le nom (utilisateur, groupe) ou l'ID (UID, GID). L'exemple suivant recherche tous les fichiers ordinaires appartenant à seb et au groupe users.

```
$  find . -type f -user seb -group users -print
./fic1
./fic3
```

d. -size

`-size` permet de préciser la taille des fichiers recherchés. Sa syntaxe est particulière car elle travaille par défaut en blocs si vous ne précisez rien. C'est parfois surprenant d'autant plus que le bloc, qui a ici une taille de 512 octets, est une unité un peu virtuelle (avec certaines commandes un bloc peut faire 1 Ko ou plus).

La valeur située après le critère peut être suivie des caractères b, c, w ou k.

Caractère	Signification
B	Par défaut si non précisé, c'est un bloc de 512 octets.
C	C'est un caractère, au sens ASCII, donc 1 octet.
W	C'est un mot (au sens ancien) de 2 octets.
K	1 Ko (1024 octets).

La valeur peut être précédée d'un + ou d'un - signifiant "plus de" ou "moins de". Sans cette indication, la taille recherchée doit correspondre EXACTEMENT.

- -size 5 : recherche les fichiers d'une taille de 5 blocs (512 octets par bloc, soit ici 2560 octets).
- -size 152c : recherche les fichiers d'une taille de 152 caractères (octets).
- -size 10k : recherche les fichiers d'une taille de 10 Ko (10*1024 octets = 10240 octets).
- -size +5000k : les fichiers de plus de 5000 Ko.

- -size -100k : les fichiers de moins de 100 Ko.

```
seb@slyserver:/var/log> find -size +100k
./zypper.log-20080227.bz2
./lastlog
./zypper.log-20080302.bz2
./wtmp
./zypper.log-20080226.bz2
./zypper.log
./messages
```

Le critère de recherche **-empty** peut être utilisé en remplacement de -size 0.

e. -atime, -mtime et -ctime

- `-atime` : recherche sur la date du dernier accès (access time). Un accès peut être la lecture du fichier, mais aussi le simple fait de le lister spécifiquement.
- `-mtime` : recherche sur la date de dernière modification (modification time). C'est de la modification du contenu qu'il s'agit.
- `-ctime` : recherche sur la date de changement (change time, en fait la date de dernière modification du numéro d'inode).

La date de changement du fichier correspond à la date où les informations liées à l'inode (voir chapitre Les disques et le système de fichiers) ont été modifiées pour la dernière fois : modification du nom, déplacement, changement des droits, de la taille, etc.).

Ces trois critères ne travaillent qu'avec des jours (périodes de 24 heures). 0 est le jour même, 1 hier, 2 avant-hier, etc. La valeur n située après le critère correspond donc à n*24 heures. Cette plage n'est pas fixe car « hier » signifie il y a entre 24 et 48 heures...

Les signes + ou - permettent de préciser les termes « de plus » et « de moins » :

- -mtime 1 : fichiers modifiés hier (entre 24 et 48 heures).
- -mtime -3 : fichiers modifiés il y a moins de trois jours (72 heures).
- -atime +4 : fichiers modifiés il y a plus de 4 jours (plus de 96 heures).

```
seb@slyserver:/var/log> find . -mtime -1
./kdm.log
./vmware
./vmware/vmware-serverd-0.log
./vmware/vmware-serverd.log
./mail.info
./Xorg.0.log
./lastlog
./Xorg.0.log.old
./warn
...
```

Vous pouvez jeter un oeil aux critères **-newer**, **-anewer** et **-cnewer** qui prennent comme paramètre un fichier. Dans ce cas find recherche les fichiers plus récents que celui précisé.

f. -perm

-perm permet d'effectuer des recherches sur les autorisations d'accès (droits, SUID, SGID, Sticky). Les droits doivent être précisés en base 8 (valeur octale) et complets. Le caractère - placé devant la valeur octale signifie que les fichiers recherchés doivent au moins avoir les droits désirés. Le + indique que le fichier doit avoir au moins l'un des droits spécifiés, d'où une nuance. Dans l'exemple suivant sont recherchés les répertoires où tout le monde (user, group, others) a le droit de pénétrer (droit x, soit 1).

```
seb@slyserver:/var/log> find -type d -perm -111
.
./vmware
./vmware/vmsd-xaction
./cups
```

g. -links et -inum

Bien que ces critères fassent appel à des notions plus avancées du système de fichier, il est bon de les présenter dès maintenant. Vous pourrez y revenir dès que le chapitre Les disques et le système de fichiers vous aura présenté le fonctionnement interne d'un système de fichiers.

L'option -links permet une recherche par nombre de hard links. Vous pouvez préciser les signes + ou - (plus de n liens et moins de n liens). Un fichier normal seul possède 1 lien. Un répertoire 2 liens (l'entrée dans le catalogue dont il fait partie et dans le point). Pour une recherche de liens symboliques il faudra utiliser l'option -type l.

```
$ find . -type f -links +2 -print
./fic2
./hardlink3_fic2
./hardlink_fic2
./hardlink2_fic2
```

-inum permet une recherche par numéro d'inode. Elle est utile dans le cas d'une recherche de tous les liens portant un même numéro d'inode. Le numéro d'inode est visible par l'option -i de la commande **ls**.

```
seb@slyserver:/var/log> ls -i
491891 acpid                491793 mail.info   491860 Xorg.0.log
491791 boot.log             491794 mail.warn   490686 Xorg.0.log.old
491729 boot.msg             492046 mcelog      492060 Xorg.1.log
seb@slyserver:/var/log> find . -inum 491791 -print
./boot.log
```

3. Commandes

Outre l'option -print on trouve d'autres options permettant d'effectuer une action sur les fichiers trouvés.

a. -ls

Le critère affiche des informations détaillées sur les fichiers trouvés correspondant au critère au lieu du simple nom de fichier. La sortie correspond à une commande **ls** avec les paramètres d, i, l et s (taille en blocs de 1 Ko).

```
seb@slyserver:~> find -size +500000k -ls
2342935 584388 -rw-r--r--   1 seb      users    597817344 fév 24
11:52 ./eeexubuntu-7.10.3-desktop-i386.iso
```

b. -exec

Le critère -exec va exécuter la commande située juste après pour chaque occurrence trouvée. Quelques remarques s'imposent :

- -exec doit obligatoirement être la dernière option de la commande **find**.
- La commande exécutée par -exec doit se terminer par un « ; ». Ce caractère spécial doit s'écrire \; pour ne pas être interprété par le shell.

- Pour passer comme paramètre pour la commande le fichier trouvé par find, il faut écrire {} (substitution du fichier).

Exemple pour effacer tous les fichiers finissant par « .mp3 » :

```
$ find . -type f -name "*.mp3" -exec rm -f {} \;
```

La commande **find** n'attend pas d'avoir trouvé tous les fichiers avant d'exécuter la commande précisée. Elle la lance dès qu'un fichier est trouvé. Aussi si la commande précédente vous a affiché n fichiers avant que vous ne pensiez à l'interrompre, alors ces n fichiers sont déjà perdus.

c. -ok

Le critère -ok est identique à l'option -exec mais, pour chaque occurrence, une confirmation est demandée à l'utilisateur.

```
$ find . -inum 95 -ok rm -f {} \;
< rm ... ./fic1 > (yes)?   n
< rm ... ./lien_fic1 > (yes)?   y
```

4. Critères AND / OR / NOT

Il est possible de combiner les options de critère de sélection. Sans aucune précision c'est le ET logique qui est implicite.

Critère	Action
-a, -and	AND, ET logique, par défaut
-o, -or	OR, OU logique
!	Négation du critère

Exemple avec tous les fichiers ne contenant pas fic dans leur nom, et tous les fichiers n'étant ni normaux ni des répertoires.

```
$ find . ! -name "*fic*" -print
.
./rep1
./liste
./mypass
./users
./liste2
./ls.txt
./toto.tar.gz
./nohup.out
./liste_ls
./rep2
./seb1
./seb2
$ find . ! \( -type f -o type d \) -ls
     409    0 lrwxrwxrwx  1 oracle   system          4 Aug 14 15:21
./lien_fic1 -> fic1
     634    0 lrwxrwxrwx  1 oracle   system          4 Aug 14 15:21
./lien_fic2 -> fic2
```

5. Retrouver des exécutables

a. whereis

La commande **whereis** recherche dans les chemins de fichiers binaires, du manuel et des sources les fichiers correspondant aux critères fournis.

```
$ whereis date
date: /bin/date /usr/share/man/man1/date.1.gz
/usr/share/man/man1p/date.1p.gz
```

Vous pouvez préciser quelques paramètres :

- `-b` uniquement pour les binaires,
- `-m` uniquement pour les manuels,
- `-s` uniquement pour les sources.

Les fichiers sont recherchés par défaut dans :

```
/{bin,sbin,etc}
/usr/{lib,bin,old,new,local,games,include,etc,src,man,sbin,X386,TeX,
g++-include}
/usr/local/{X386,TeX,X11,include,lib,man,etc,bin,games,emacs}
```

Donc ne soyez pas étonné d'obtenir ceci :

```
$ whereis -b passwd
passwd: /usr/bin/passwd /etc/passwd /etc/passwd.old
/etc/passwd.YaST2save /etc/passwd.vipwKSnTgH /usr/bin/X11/passwd
```

b. which

La commande **which** recherche une commande dans le PATH (chemin des exécutables) et vous fournit la première qu'elle trouve :

```
$ which date
/bin/date
```

Il arrive que des commandes de même nom existent dans plusieurs chemins, vous pouvez dès lors préciser le paramètre `-a` pour que which continue sa recherche. Sachez cependant que c'est la première qui sera exécutée par défaut si vous la lancez.

```
$ which -a passwd
/usr/bin/passwd
/usr/bin/X11/passwd
```

6. locate

La commande **locate** recherche un fichier selon le modèle donné dans une base de données de fichiers construite par la commande **updatedb**. La commande **updatedb** prend une série de chemins dans laquelle elle va effectuer un **find** et stocker tous les résultats dans une base indexée. Cela évite donc pour les recherches classiques d'effectuer de nouveau un find. Dans la pratique, il suffit de préciser à updatedb la liste des chemins ou ne pas mettre en base les fichiers. **updatedb** est généralement lancé par la crontab de manière quotidienne. Les paramètres de la commande sont parfois placés dans un fichier **`/etc/sysconfig/locate`**.

```
$ cat /etc/sysconfig/locate | grep -Ev "^(#|$)"
RUN_UPDATEDB=yes
RUN_UPDATEDB_AS=nobody
UPDATEDB_NETPATHS=""
UPDATEDB_PRUNEPATHS="/mnt /cdrom /tmp /usr/tmp /var/tmp /var/spool
/proc /media /sys"
UPDATEDB_NETUSER=""
UPDATEDB_PRUNEFS=""
```

La commande lancée dans ce cas est la suivante :

```
# updatedb --localuser=nobody --prunepaths=/mnt /cdrom /tmp /usr/tmp
/var/tmp /var/spool /proc /media /sys
```

Si vous lancez ainsi la commande, elle risque de consommer toutes les ressources du processeur de votre machine, aussi en réalité **updatedb** est lancé avec une priorité basse. La base est placée dans **/var/lib/locatedb**.

```
$ locate toto
/opt/kde3/share/apps/ksgmltools2/docbook/xsl/html/autotoc.xsl
/opt/kde3/share/apps/ksgmltools2/docbook/xsl/params/autotoc.label.se
parator.xml
/usr/share/gnome/help/gnome-doc-xslt/C/db2html-autotoc.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/fo/autotoc.xsl
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/html/autotoc.xsl
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/params/autotoc.label
.in.hyperlink.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/params/autotoc.label
.separator.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/xhtml/autotoc.xsl
/usr/share/xml/gnome/xslt/docbook/html/db2html-autotoc.xsl
/usr/src/linux-2.6.22.17-0.1/drivers/mtd/maps/omap-toto-flash.c
/usr/src/linux-2.6.22.17-0.1/drivers/mtd/nand/toto.c
/usr/src/linux-2.6.24.4/drivers/mtd/maps/omap-toto-flash.c
/usr/src/linux-2.6.24.4/drivers/mtd/nand/toto.c
```

L'éditeur vi

1. Présentation

L'éditeur Unix par défaut se nomme **vi** (*visual editor*). S'il n'est pas des plus ergonomiques par rapport à des éditeurs en mode graphique, il a l'avantage d'être disponible et d'utiliser la même syntaxe de base sur tous les Unix. Chaque Unix propose généralement une syntaxe étendue au-delà de la syntaxe de base. L'éditeur **vi** sous Linux se nomme vim. Vim respecte toute la syntaxe de vi, la réciproque n'étant pas vraie. Vi est petit : il occupe peu d'espace disque, consomme peu de mémoire.

```
vi [options] Fichier [Fichier2 ...]
```

Vi n'a pas de menus, pas d'interface graphique et n'est pas intuitif. Cela nécessite de connaître par cœur un certain nombre de raccourcis-claviers pour pouvoir l'utiliser. Si l'apprentissage est un peu difficile, une fois maîtrisé vi se révèle rapide et pratique, au point qu'on va plus vite qu'avec des éditeurs de texte graphiques.

Le débat opposant les partisans de emacs (ou d'autres éditeurs) et ceux de vi n'a pas lieu d'être. Tout système Linux (et Unix) dispose quoi qu'il arrive de l'éditeur vi, le rendant incontournable. Si vous en avez la possibilité, vous pouvez par la suite installer l'éditeur qui vous plaira. Ce ne sera pas toujours possible, notamment en entreprise, sur des serveurs, etc.

---

2. Fonctionnement

Il y a trois modes de fonctionnement :

- mode **commande** : les saisies représentent des commandes. On y accède en appuyant sur [Echap]. Chaque touche ou combinaison de touches déclenche une action (suppression de lignes, insertions, déplacement, copier, coller, etc.).
- mode **saisie** : c'est la saisie de texte classique.
- mode **ligne de commande** : une ligne en bas d'écran permet de saisir des commandes spéciales, validée avec la touche [Entrée]. On y accède en appuyant, en mode commande, sur la touche « : ».

Quand vous lancez vi, il est par défaut en mode commande. Pour commencer à taper du texte, ce qui est vite énervant quand on ne connaît pas vi, il faut taper une commande d'ajout ou d'insertion : a ou i. Donc la commande [Echap] **i** permet de saisir du texte. Pour quitter, vous pouvez passer par le mode ligne de commande. Vous tapez [Echap] **:** et enfin q et [Entrée]. C'est simple !

Si la commande saisie en mode commande n'est pas une commande de saisie, vous ne quittez pas ce mode, et vous n'êtes pas obligé de taper [Echap] avant de saisir une nouvelle commande. Par exemple le « x » supprime un caractère. Si vous êtes déjà en mode commande, vous pouvez appuyer dix fois sur « x » sans appuyer sur [Echap] entre chaque appui.

Il y a bien entendu une pointe d'ironie. Il n'est pas évident de connaître tous les raccourcis, l'auteur de ces lignes lui-même ne les retient pas tous. Aussi, outre l'existence de guides en accordéon dans la presse spécialisée, voici la liste des commandes les plus utilisées sous forme de tableaux. Quand la commande est précédée du « : », c'est une commande à saisir en ligne de commande.

Si vous êtes perdu, dans tous les cas et quoi qu'il arrive, un appui sur la touche [Echap] revient toujours en mode commande.

3. Les commandes

a. La saisie

Les actions suivantes sont à effectuer en mode commande. Elles doivent être précédées d'un appui sur [Echap] :

[Echap] a, [Echap] i, etc.

Commande	Action
a	Ajout après le caractère actuel.
A	Ajout de texte en fin de ligne.
i	Insertion devant le caractère actuel, comme dans un traitement de texte.
I	Insertion de texte en début de ligne.
o	Ajout d'une ligne sous la ligne actuelle.
O	Insertion d'une ligne au-dessus de la ligne actuelle.

b. Quitter et sauver

Pour mémoire, les « : » signifient que la commande se tape en ligne de commande : [Echap] :, saisie de la commande, puis [Entrée].

Commande	Action
ZZ	Sauve le fichier et quitte.
:q!	Quitte sans sauver.
:q	Quitte si le fichier n'a pas été modifié (apparition d'un message d'erreur sinon).
:w	Sauve le fichier. Vous pouvez préciser un nom à la suite.
:wq ou **:x**	Sauve et quitte.
1,10w fic	Sauve les lignes de 1 à 10 dans fic.

c. Déplacement

Vous allez probablement sourire, mais il existe encore des claviers sans touches directionnelles, et des logiciels et des interpréteurs de commande ne sachant pas les interpréter (exemple du shell ksh mal configuré sur certains Unix). Sur ces machines, c'est le mode de saisie de vi qui est activé par défaut, et même sous vi il est possible de se passer des touches fléchées, en mode commande.

Commande	Action
h	Aller vers la gauche.
l (petit L)	Aller vers la droite.
k	Aller vers le haut.
j	Aller vers le bas.
0 (zéro)	Début de ligne.
:0	Début de fichier (première ligne).
$	Fin de ligne.

:$	Fin de fichier (dernière ligne).
w	Aller au mot suivant.
b	Aller au mot précédent.
f<c>	Sauter au caractère <c> suivant.
Control+f	Avance d'un écran (Forward).
Control+b	Recule d'un écran (Backward).
G	Dernière ligne du fichier.
<n>G	Saute à la ligne « n » (ex : 10G va à la 10ème ligne).
:<n>	Idem (:10 va à la 10ème ligne).

d. La correction

Commande	Action
x	Efface le caractère sous le curseur.
X	Efface le caractère devant le curseur.
r<c>	Remplace le caractère sous le curseur par le caractère <c>.
dw	Efface depuis le curseur jusqu'à la fin du mot.
d$ ou **D**	Efface depuis le curseur jusqu'à la fin de la ligne.
dO	Efface depuis le début de la ligne jusqu'au curseur.
df<c>	Efface tout jusqu'au caractère <c>.
dG	Efface tout jusqu'à la dernière ligne, y compris la ligne actuelle.
d1G	Efface tout jusqu'à la première ligne, y compris la ligne actuelle.
dd	Efface la ligne actuelle.
u	Undo. Annule la dernière action.

Ces commandes peuvent être répétées. 5dd supprime 5 lignes, 4dw 4 mots, 5x 5 caractères, etc.

e. Recherche dans le texte

Contrairement à un éditeur de texte classique, vi peut rechercher autre chose que des mots simples et fonctionne à l'aide de caractères spéciaux et de critères. La commande de recherche est le caractère /. La recherche démarre du caractère courant jusqu'à la fin du fichier. Le caractère ? effectue la recherche en sens inverse. On indique ensuite le critère, puis [Entrée].

```
/echo
```

recherche la chaîne 'echo' dans la suite du fichier. Quand la chaîne est trouvée, le curseur s'arrête sur le premier caractère de cette chaîne.

La commande n permet de continuer la recherche dans le sens indiqué au début. La commande N effectue la recherche en sens inverse.

Quelques critères

- /[FfBb]oule : Foule, foule, Boule, boule.
- /[A-Z]e : tout ce qui commence par une majuscule avec un e en deuxième position.
- /[A-Za-Z0-9] : tout ce qui commence par une majuscule, une minuscule ou un chiffre.
- /[^a-z] : plage négative : tout ce qui ne commence pas par une minuscule.
- /vé.o : le point remplace un caractère, vélo, véto, véro, ...
- /Au*o : l'étoile est un caractère de répétition, de 0 à n caractères, Auo, Auto, Automoto, ...
- /.* : l'étoile devant le point, une chaîne quelconque de taille variable, le "." représentant un caractère.
- /[A-Z][A-Z]* : répétition du motif entre [] de 0 à n fois, recherche d'un mot comportant au moins une majuscule (en début de mot).
- /^Auto : le ^ indique que la chaîne recherchée devra être en début de ligne.
- /Auto$: le $ indique que la chaîne recherchée devra être en fin de ligne.

f. Commandes de remplacement

Pour remplacer du texte, il faut se placer au début de la chaîne à modifier, puis taper l'une des commandes suivantes. Ensuite tapez simplement votre texte. Le principe est simple, il est identique à celui de la suppression : le morceau précisé est supprimé et vi passe en mode d'insertion pour la saisie.

Commande	Action
cw	Remplace le mot courant.
c$	Remplace jusqu'à la fin de la ligne.
c0 (zéro)	Remplace jusqu'au début de la ligne.
cf<x>	Remplace jusqu'au prochain caractère <x>.
c/<rech>	Remplace jusqu'à la prochaine occurrence de la chaîne <rech>.

g. Copier-Coller

La commande **v** permet une sélection visuelle. Le texte est surligné et vous pouvez déplacer le curseur pour sélectionner le texte. Utilisez ensuite l'une des commandes suivantes :

- pour couper (déplacer), c'est la commande « d » ;
- le c fait presque la même chose, mais vi reste en mode d'édition ;
- pour coller le texte à l'endroit choisi, c'est la commande p (derrière le caractère) ou P (devant le caractère). Si c'est une ligne complète qui a été copiée, elle sera placée en dessous de la ligne active.

Les actions suivantes sont possibles en mode commande :

- Pour copier une ligne : yy.
- Pour copier cinq lignes : 5yy.
- Pour placer les lignes copiées à un endroit donné : p.
- L'éditeur vi dispose de 26 tampons pour y stocker les données que l'on peut nommer comme on le souhaite. On utilise pour cela le ".
- Pour copier cinq mots dans la mémoire m1 : "m1y5w.
- Pour coller le contenu de la mémoire m1 à un endroit donnée : "m1p.

h. Substitution

La substitution permet de remplacer automatiquement plusieurs occurrences par une autre chaîne.

```
:[1ere ligne, dernière ligne]s/Modèle/Remplacement/[gc]
```

Les numéros de lignes sont optionnels. Dans ce cas la substitution ne se fait que sur la ligne courante. En remplacement des numéros de lignes, **.** détermine la ligne courante, **1** la première ligne, **$** la dernière ligne.

Le modèle est un critère de recherche présenté dans ces dernières pages. Remplacement est une chaîne quelconque qui remplacera le modèle.

Par défaut seule la première occurrence est remplacée. La lettre g indique qu'il faut remplacer toutes les occurrences. Avec c, vi demande une confirmation pour chacune des occurrences. L'exemple remplace toutes les occurrences de Unix ou Unix en UNIX.

```
:1,$s/[Uu]nix/UNIX/g
```

i. Autres

Édition avancée

Voici quelques commandes pratiques.

:r fic	Insère le contenu de fic à partir de l'endroit actuel.
:! cmd	Exécute la commande. Appuyez sur [Entrée] pour revenir sous vi.
:r! cmd	Le résultat de la commande est inséré à l'endroit actuel.
:e fic	Charge le fichier fic pour édition.
:e#	Commute entre les divers fichiers ouverts.

Commande set

La commande **set** permet de configurer l'éditeur et d'accéder à ses options :

- set all : affiche l'ensemble des options possibles.
- set number (ou nu) / nonumber (ou nonu) : affiche/supprime les numéros de lignes.
- set autoindent / noautoindent : l'indentation est conservée lors d'un retour à la ligne.
- set showmatch / noshowmatch : lors de la saisie d'une accolade ou d'une parenthèse de fermeture, celle d'ouverture est affichée un très court instant, puis l'éditeur revient au caractère courant.

- set showmode / noshowmode : vi affichera une ligne d'état (INPUT MODE).
- set tabstop=x : définit le nombre de caractères pour une tabulation.

Vim propose bien souvent la coloration syntaxique. Il détecte le type de fichier chargé et colore les lignes, les mots clés, etc. Vous pouvez désactiver la coloration :

- syntax off pour désactiver ;
- syntax on pour activer.

Redirections

1. Principe

Les redirections sont l'une des plus importantes possibilités offertes par le shell. Par redirection, on entend la possibilité de rediriger l'affichage de l'écran vers un fichier, une imprimante ou tout autre périphérique, les messages d'erreur vers un autre fichier, de remplacer la saisie clavier par le contenu d'un fichier.

Tout flux de données en entrée ou en sortie de commande passe par un canal. Comme pour l'eau, il est possible de dévier le cours des données vers une autre destination ou depuis une autre source.

Linux utilise des canaux d'entrées/sorties pour lire et écrire ses données. Par défaut le canal d'entrée est le clavier, et le canal de sortie, l'écran. Un troisième canal, le canal d'erreur, est aussi redirigé vers l'écran par défaut.

Il est possible de rediriger ces canaux vers des fichiers, ou du flux texte de manière transparente pour les commandes Linux.

2. En sortie

On se sert du caractère **>** pour rediriger la sortie standard (celle qui va normalement sur l'écran). On indique ensuite le nom du fichier où seront placés les résultats de sortie.

```
$ ls -l > resultat.txt
$ cat resultat.txt
total 1
-rw-r--r--    1 Administ ssh_user        0 Jul  4 12:04 TOTO
-rw-r--r--    1 Administ ssh_user        0 Jul 25 15:13 resultat.txt
-rw-r--r--    1 Administ ssh_user      171 Jul 25 15:13 test.txt
```

Si le fichier n'existe pas, il sera créé. S'il existe, son contenu sera écrasé, même si la commande tapée est incorrecte. **Le shell commence d'abord par créer le fichier puis exécute ensuite la commande**.

C'est un aspect important des redirections : les redirections sont interprétées de la droite vers la gauche, et les redirections sont mises en place AVANT l'exécution des commandes : il faut bien créer le fichier avant de pouvoir y écrire. D'où le fait que même si la commande est fausse, le fichier est créé ou écrasé...

Pour rajouter des données à la suite du fichier, donc sans l'écraser, on utilise la double redirection **>>**. Le résultat de la commande est ajouté à la fin du fichier.

```
$ ls -l > resultat.txt
$ date >> resultat.txt
$ cat resultat.txt
total 1
-rw-r--r--    1 Administ ssh_user        0 Jul  4 12:04 TOTO
-rw-r--r--    1 Administ ssh_user        0 Jul 25 15:13 resultat.txt
-rw-r--r--    1 Administ ssh_user      171 Jul 25 15:13 test.txt
Thu Jul 25 15:20:12  2002
```

3. En entrée

Les commandes qui attendent des données ou des paramètres depuis le clavier peuvent aussi en recevoir depuis un fichier, à l'aide du caractère <. Un exemple avec la commande **wc** (*word count*) qui permet de compter le nombre de lignes, de mots et de caractères d'un fichier.

```
$ wc  < resultat.txt
      4      29     203
```

4. Document en ligne

La redirection << est particulière. Elle permet l'utilisation des documents en ligne. Vous trouverez parfois le terme Herescript ou Here Document. Cela permet la saisie d'un texte jusqu'à un point donné et l'envoi de son résultat à une commande ou un filtre. Les redirections classiques sont aussi autorisées. Après le << vous indiquez une chaîne définissant la fin de saisie, par exemple ici 'end'.

```
$ tr "[a-z]" "[A-Z]" << end
> bonjour les amis
> ceci est un exemple
> de herescript
> end
BONJOUR LES AMIS
CECI EST UN EXEMPLE
DE HERESCRIPT
```

5. Les canaux standards

On peut considérer un canal comme un fichier, qui possède son propre descripteur par défaut, et dans lequel on peut ou lire ou écrire.

- Le canal d'entrée standard se nomme **stdin** et porte le descripteur 0.
- Le canal de sortie standard se nomme **stdout** et porte le descripteur 1.
- Le canal d'erreur standard se nomme **stderr** et porte le descripteur 2. On peut rediriger le canal d'erreur vers un autre fichier.

```
$ rmdir dossier2
rmdir: `dossier2': No such file or directory
$ rmdir dossier2 2>error.log
$ cat error.log
rmdir: `dossier2': No such file or directory
```

Vous pouvez rediriger les deux canaux de sortie dans un seul et même fichier, en les liant. On utilise pour cela le caractère **>&**. Il est aussi important de savoir dans quel sens le shell interprète les redirections. Les redirections étant en principe en fin de commande, le shell recherche d'abord les caractères **<**, **>**, **>>** en fin de ligne. Ainsi si vous voulez grouper les deux canaux de sortie et d'erreur dans un même fichier, il faut procéder comme suit.

```
$ ls -l > resultat.txt 2>&1
```

La sortie 2 est redirigée vers la sortie 1, donc les messages d'erreurs passeront par la sortie standard. Puis le résultat de la sortie standard de la commande **ls** est redirigé vers le fichier resultat.txt. Ce fichier contiendra donc à la fois la sortie standard et la sortie d'erreur.

Vous pouvez utiliser les deux types de redirection à la fois :

```
$ wc < resultat.txt > compte.txt
$ cat compte.txt
      4      29     203
```

6. Ouverture de canaux

Les canaux standards sont au nombre de trois et numérotés de 0 à 2. Ainsi 0< équivaut à < et 1> à >. La commande **exec** permet d'ouvrir sept autres canaux numérotés de 3 à 9. On a donc en tout dix canaux.

Vous pouvez, et même devez, envisager dans le cadre de traitements de sortir certains résultats par le canal 3, d'autres par le 4, et ainsi de suite. Les canaux ouverts le sont en entrée et en sortie.

```
$ exec 3>dump.log
$ ls -l >&3
$ cat dump.log
total 3952
-rw-r--r--  1 seb users  167212 oct  9 09:27 battlestar_1280.jpg
drwxr-xr-x  2 seb users    4096 mar  4 08:51 bin
```

```
drwxr-xr-x  8 seb users    4096 mar  4 08:45 cxoffice
drwx------  2 seb users    4096 mar 10 12:29 Desktop
drwx------ 13 seb users    4096 mar  6 11:49 Documents
-rw-r--r--  1 seb users       0 mar 11 11:34 dump.log
-rw-r--r--  1 seb users 3785296 déc 12 15:15 e3555_EeePC4G.pdf
drwxr-xr-x  3 seb users    4096 mar 10 11:16 Games
drwxr-xr-x  5 seb users    4096 mar 10 11:16 karchiver-3.4.2.b4
-rw-r--r--  1 seb users     358 mar 11 08:51 liste
-rw-r--r--  1 seb users     608 mar 11 09:14 tmpgrp
-rw-r--r--  1 seb users    1555 mar 11 09:15 tmppwd
```

Tous ce qui sera écrit dans le canal 3 sera écrit dans le fichier dump.log. On peut ensuite fermer le canal en le réunissant avec un pseudo-canal (canal de fermeture -).

```
$ exec 3>&-
```

7. Filtre : définition

Un **filtre** (ou une commande filtre) est un programme sachant écrire et lire des données par les canaux standards d'entrée et de sortie. Il en modifie ou traite éventuellement le contenu. wc est un filtre. En voici quelques-uns : **more** (affiche les données page par page), **sort** (tri des données), **grep** (critères de recherche).

8. Pipelines / tubes

Les redirections d'entrée/sortie telles que vous venez de les voir permettent de rediriger les résultats vers un fichier. Ce fichier peut ensuite être réinjecté dans un filtre pour en extraire d'autres résultats. Cela oblige à taper deux lignes : une pour la redirection vers un fichier, l'autre pour rediriger ce fichier vers le filtre. Les **tubes** ou **pipes** permettent de rediriger directement le canal de sortie d'une commande vers le canal d'entrée d'une autre. Le caractère permettant cela est **|** accessible depuis la combinaison [AltGr] **6** des claviers français.

```
$ ls -l > resultat.txt
$ wc < resultat.txt
```

devient

```
$ ls -l | wc
```

Il est possible de placer plusieurs | sur une même ligne.

```
$ ls -l | wc | wc
      1       3      24
```

La première commande n'est pas forcément un filtre. C'est même rarement le cas. L'essentiel est qu'un résultat soit délivré. Idem pour la dernière commande qui peut par exemple être une commande d'édition ou d'impression. Enfin, la dernière commande peut elle-même faire l'objet d'une redirection en sortie.

```
$ ls -l | wc > resultat.txt
```

Les filtres et utilitaires

Un **filtre** (ou une commande filtre) est un programme sachant écrire et lire des données par les canaux standards d'entrée et de sortie. Il en modifie ou traite éventuellement le contenu. **wc** est un filtre. Les utilitaires sans être obligatoirement des filtres permettent un certain nombre d'actions sur des fichiers ou leur contenu comme le formatage ou l'impression.

1. Extraction des noms et chemins

La commande **basename** permet d'extraire le nom du fichier dans un chemin.

```
$ basename /tmp/seb/liste
liste
```

La commande **dirname** effectue l'inverse, elle extrait le chemin.

```
$ dirname /tmp/seb/liste
/tmp/seb
```

2. Recherche de lignes

Il s'agit d'extraire des lignes d'un fichier selon divers critères. Pour cela vous disposez de trois commandes **grep**, **egrep** et **fgrep** qui lisent les données soit depuis un fichier d'entrée, soit depuis le canal d'entrée standard.

a. grep

La syntaxe de la commande **grep** est `grep [Options] modèle [Fichier1...]`.

Le modèle se compose de critères de recherche ressemblant beaucoup aux critères déjà exposés pour vi par exemple. Il ne faut pas oublier que ces critères doivent être interprétés par la commande **grep** et pas par le shell. Il faut donc verrouiller tous les caractères.

```
$ cat fic4
Cochon
Veau
Boeuf
rat
Rat
boeuf
$ grep "^[bB]" fic4
Boeuf
boeuf
```

La commande **grep** peut aussi prendre quelques options intéressantes.

- **-v** effectue la recherche inverse : toutes les lignes ne correspondant pas aux critères sont affichées.
- **-c** ne retourne que le nombre de lignes trouvées sans les afficher.
- **-i** ne différencie pas les majuscules et les minuscules.
- **-n** indique le numéro de ligne pour chaque ligne trouvée.
- **-l** dans le cas de fichiers multiples, indique dans quel fichier la ligne a été trouvée.

```
$ grep -i "^b" fic4
Boeuf
boeuf
```

b. egrep

La commande **egrep** étend les critères de recherche et peut accepter un fichier de critères en entrée. Elle est équivalente à un grep -E. Elle emploie comme critères des expressions régulières.

```
egrep -f fichier_critère Fichier_recherche
```

Caractère spécial	Signification
|	Ou logique, l'expression située avant ou après doit apparaître.
(...)	Groupement de caractères.
[...]	Un caractère à cette position parmi ceux indiqués.
. (point)	Un caractère quelconque.
+	Répétition, le caractère placé avant doit apparaître au moins une fois.
*	Répétition, le caractère situé avant doit apparaître de zéro à n fois.
?	Le caractère situé avant doit apparaître une fois au plus.
{n}	Le caractère situé avant doit apparaître exactement n fois.
{n,}	Il apparaît n fois ou plus.
{n,m}	Il apparaît entre n et m fois.
^	En début de chaîne.
$	En fin de chaîne.

Seulement bonjour et bonsoir commençant par une majuscule ou une minuscule s'ils sont seuls sur une ligne :

```
^[bB]on(jour|soir)$
```

Vérification très sommaire de la validité d'une adresse IP :

```
echo $IP | egrep '([0-9]{1,3}\.){3}[0-9]{1,3}')
```

Voici comment cette ligne est décomposée :

- ([0-9]{1,3}\.){3} :www.xxx.yyy.
- [0-9] : un caractère entre 0 et 9
- {1,3} : répété entre une et trois fois, donc x, xx ou xxx
- \. : suivi d'un point
- {3} : le tout trois fois

Puis [0-9]{1,3} : .zzz

- [0-9] : un caractère entre 0 et 9
- {1,3} : répété entre une et trois fois

c. fgrep

La commande **fgrep** est un grep simplifié et rapide (fast grep) et équivaut à un grep -F . Elle accepte aussi un fichier de critères de recherche mais il s'agit là de critères simples, sans caractères spéciaux. Vous saisissez dans le fichier de critères des lignes simples (du texte et des chiffres), une recherche par ligne. Fgrep va alors rechercher dans un fichier cible ou un flux en entrée les lignes correspondant à chacun des critères.

3. sed

L'apprentissage de sed demanderait tout un livre. Sed est un éditeur de flux (Stream Editor) permettant de filtrer et de transformer du texte. C'est un peu comme un éditeur permettant de modifier du texte via des commandes scripts, mais en une passe et sans édition interactive. Il utilise un jeu étendu de commandes issu de l'éditeur ed. Sa syntaxe de base est :

```
sed -e '<cmd>' fic
```

Pour utiliser sed, il faut apprendre et comprendre les expressions rationnelles. Le tableau de la commande **egrep** reprend la syntaxe de base des expressions. Tout ouvrage sur sed part de ces expressions, et réciproquement.

Sed est très souvent utilisé pour remplacer des valeurs par d'autres (substitution) ou supprimer des lignes particulières (bien que grep pourrait être utilisé dans ce cas). La syntaxe basique de substitution est la suivante :

```
s/<ancien>/nouveau/[g]
```

Le `g` final permet de faire un remplacement sur toute la ligne en cas de présence de plusieurs occurrences. Voici un exemple qui remplace `__NOM__` par `Toto` :

```
$ echo "Je m'appelle __NOM__. Tu t'appelles __NOM__?" | sed -e 's/__
NOM__/Toto/'
Je m'appelle Toto. Tu t'appelles __NOM ?
$ echo "Je m'appelle __NOM__. Tu t'appelles __NOM__ ?" | sed -e 's/__
NOM__/Toto/g'
Je m'appelle Toto. Tu t'appelles Toto ?
```

Vous pouvez placer une valeur numérique dans le champ nouveau pour préciser, si la recherche comporte plusieurs éléments regroupés par des parenthèses, sur quel élément recherché travailler. Voici un simple exemple qui rajoute des étoiles autour du nom toto :

```
$ echo toto | sed -e "s/\(toto\)/**\1**/"
**toto**
```

Pour supprimer toutes les lignes vides ou ne contenant que des espaces :

```
$ sed -e '/^ *$/d' fichier
```

4. Colonnes et champs

La commande **cut** permet de sélectionner des colonnes et des champs dans un fichier.

a. Colonnes

La syntaxe est la suivante :

```
cut -cColonnes [fic1...]
```

Une colonne est la position d'un caractère dans la ligne. Le premier caractère est la colonne 1, le deuxième la colonne 2, et ainsi de suite. Une ligne de 80 caractères dispose de 80 colonnes. La numérotation commence à 1. C'est la méthode idéale pour des fichiers plats et à format fixe où chaque champ débute et finit à des positions données.

Le format de sélection de colonne est le suivant :

- une colonne seule (ex. -c2 pour la colonne 2) ;
- une plage (ex. -c2-4 pour les colonnes 2, 3 et 4) ;
- une liste de colonnes (ex. -c1,3,6 pour les colonnes 1, 3 et 6) ;
- les trois en même temps (ex. -c1-3,5,6,12-).

```
$ cat liste
Produit prix    quantites
souris  30      15
disque  100     30
ecran   300     20
clavier 45      30

$ cut -c1-5 liste
Produ
souri
disqu
ecran
clavi

$ cut -c1-3,10-12,15
Prorx   quantites
sou0    15
dis0    30
ecr0    20
cla530
```

b. Champs

La commande **cut** permet aussi de sélectionner des champs. Ces champs doivent être par défaut délimités par une tabulation, mais le paramètre `-d` permet de sélectionner un autre caractère (espace, ;). La sélection des champs est identique à celle des colonnes.

Le caractère séparateur doit être unique. Il n'est pas possible d'en mettre deux ou trois, ou une chaîne de séparateurs. Pour éliminer les caractères multiples, utilisez tr. De même le séparateur par défaut est la tabulation. Or par défaut les tabulations sont souvent remplacées par des espaces au sein des éditeurs... `cut -dc -fChamps [fic1...]`

Voici quelques exemples. Le fichier liste contient des champs séparés par des tabulations.

```
$ cat liste
Produit prix    quantites
souris  30      15
dur     100     30
disque  100     30
ecran   300     20
clavier 45      30
carte   45      30

$ cut -f1 liste
Produit
souris
dur
disque
ecran
clavier
carte

$ cut -f1,3 liste
Produit quantites
souris  15
dur     30
```

```
disque  30
ecran   20
clavier 30
carte   30
```

Notez que si vous inversez l'ordre des champs (-f3,1) vous n'obtenez pas l'effet escompté : les champs sortent toujours dans le sens 1,3.

Voici comment isoler les noms d'un groupe et leurs identifiants respectifs :

```
$ cat /etc/group
seb@slyserver:~> cat /etc/group
at:!:25:
audio:x:17:
avahi:!:106:
beagleindex:!:107:
bin:x:1:daemon
cdrom:x:20:
console:x:21:
daemon:x:2:
dialout:x:16:seb,steph,henri,public
disk:x:6:

$ cut -d: -f1,3 /etc/group
at:25
audio:17
avahi:106
beagleindex:107
bin:1
cdrom:20
console:21
daemon:2
dialout:16
disk:6
```

S'il n'y a pas de délimiteur (tabulation ou autre) dans une ligne, cut affiche toute la ligne.

5. Décompte de lignes

La commande **wc** (*word count*) permet de compter les lignes, les mots et les caractères.

`wc [-l] [-c] [-w] [-w] fic1`

- `-l` : compte le nombre de lignes
- `-c` : compte le nombre d'octets
- `-w` : compte le nombre de mots
- `-m` : compte le nombre de caractères

```
$ wc liste
       12        48       234 liste
```

Le fichier liste contient 12 lignes, 48 mots et 234 caractères.

6. Tri de lignes

La commande **sort** permet de trier des lignes. Par défaut le tri s'effectue sur tout le tableau et en ordre croissant. Le

tri est possible sur un ou plusieurs champs. Le séparateur de champs par défaut est la tabulation ou au moins un espace. S'il y a plusieurs espaces, le premier est le séparateur, les autres des caractères du champ.

La syntaxe de sort a évolué depuis quelques années et Linux s'est mis en conformité. Aussi l'ancienne syntaxe basée sur +/- n'est plus utilisée. À la place, il faut utiliser le paramètre `-k`. La numérotation des champs commence à 1.

```
sort [options] [-k pos1[,pos2]] [fic1...]

$ cat liste
souris  optique 30      15
dur     30giga  100     30
dur     70giga  150     30
disque  zip     12      30
disque  souple  10      30
ecran   15      150     20
ecran   17      300     20
ecran   19      500     20
clavier 105     45      30
clavier 115     55      30
carte   son     45      30
carte    video   145     30
```

Voici comment trier par ordre alphabétique sur la première colonne :

```
$ sort -k 1 liste
carte   son     45      30
carte   video   145     30
clavier 105     45      30
clavier 115     55      30
disque  souple  10      30
disque  zip     12      30
dur     30giga  100     30
dur     70giga  150     30
ecran   15      150     20
ecran   17      300     20
ecran   19      500     20
souris  optique 30      15
```

Quelques paramètres

Option	Rôle
-d	Dictionnary sort (tri dictionnaire). Ne prend comme critère de tri que les lettres les chiffres et les espaces.
-n	Tri numérique, idéal pour les colonnes de chiffres.
-b	Ignore les espaces en début de champ.
-f	Pas de différences entre majuscules et minuscules (conversion en minuscules puis tri).
-r	Reverse, tri en ordre décroissant.
-tc	Nouveau délimiteur de champ c.

Exemple, tri numérique sur le prix par produits en ordre décroissant :

```
$ sort -n -r -k 3 liste
ecran   19      500     20
ecran   17      300     20
ecran   15      150     20
dur     70giga  150     30
carte   video   145     30
dur     30giga  100     30
clavier 115     55      30
clavier 105     45      30
```

```
carte   son     45      30
souris  optique 30      15
disque  zip     12      30
disque  souple  10      30
```

Il est aussi possible de démarrer le tri à partir d'un certain caractère d'un champ. Pour cela vous devez spécifier le « .pos » : -k1.3 commencera le tri à partir du troisième caractère du champ 1.

7. Suppression des doublons

La commande **uniq** permet de supprimer les doublons dans des flux en entrée ou des fichiers triés. Par exemple, voici comment sortir uniquement la liste des GID réellement utilisés comme groupe principal des utilisateurs :

```
$ cut -d: -f4 /etc/passwd | sort -n | uniq
0
1
2
7
8
12
13
14
25
49
51
62
...
```

8. Jointure de deux fichiers

a. Sur des champs communs

La commande **join** permet d'effectuer une jointure de deux fichiers en fonction d'un champ commun. Les deux fichiers doivent être triés sur les champs spécifiés pour la jointure.

```
join [-tc] [-1 n] [-2 m] fic1 fic2
```

L'option **-t** indique le séparateur, **-1** le champ du premier fichier et **-2** le champ du second fichier sur lesquels effectuer la jointure. Notez que **join** gère mal les doublons et risque de s'arrêter dans ce cas.

La commande **join** risque de ne pas vous fournir le résultat attendu. C'est que dès qu'elle ne trouve pas une correspondance entre deux lignes, elle s'arrête.

b. Ligne à ligne

La commande **paste** regroupe n fichiers en un. Pour cela elle concatène les lignes de chacun des fichiers en une seule ligne : ligne1 de fic1 avec ligne2 de fic2, ligne3 de fic 3, et ainsi de suite. C'est un peu l'inverse du cut. Le séparateur par défaut est la tabulation mais vous pouvez préciser un délimiteur avec -d.

```
$ cat fic1
liste_a
liste_b
liste_c

$ cat fic2
liste_a2
liste_b2
liste_c2

$ paste -d: fic1 fic2
liste_a:liste_a2
```

```
liste_b:liste_b2
liste_c:liste_c2
```

9. Découpage d'un fichier en morceaux

a. Découper

Voici une commande fort pratique, **split**, qui permet de découper un gros fichier en plusieurs morceaux d'une taille donnée. Les systèmes de fichiers ne sont pas tous égaux devant la taille maximale d'un fichier. Sous Linux le problème se pose peu, un système de fichiers de type ext3 supportant de fichiers de 1TO (TB = TeraByte = TeraOctet = 1024 Go) soit l'équivalent de 130 DVDs double couche environ. Mais les bandes magnétiques, ou dans une plus faible mesure les disques amovibles, n'ont pas tous cette possibilité.

Une clé USB ou un disque externe sont généralement « formatés » avec un système de fichiers de type VFAT issu du monde Microsoft. Ce système de fichiers provenant de DOS puis Windows 9x garantit une compatibilité entre tous les systèmes (Unix, Windows, MacOS), qui peut le plus peut le moins. VFAT (ou plutôt FAT16 ou FAT32) ne supporte que des fichiers d'une taille maximum de 4 Go. Une image ISO de DVD ou une archive de sauvegarde ne peut y rentrer d'un seul bloc. Il faut donc découper le fichier en plusieurs morceaux.

```
split [-l n] [-b n[bkm] [fichier [préfixe]]
```

La commande peut fonctionner selon deux modes :

- découpage par lignes avec `-l` : les fichiers en sortie auront tous n lignes de texte (sauf éventuellement le dernier) ;
- découpage à taille fixe avec `-b` : les fichiers auront tous une taille fixe de n octets. Le suffixe b indique une taille de n blocs (512 octets), k indique n ko (1024 octets) et m indique n Mo (1024 ko).

Comme tout filtre split peut prendre un flux en entrée, ce qui est le cas si aucun fichier n'est précisé, ou si un tiret est présent. Un préfixe définit le nom des fichiers en sortie. Voici un fichier de 1 Go à découper en tranches de 150 Mo. Le préfixe est fic. Chaque fichier en sortie s'appelle ficaa, ficab, ficac, ficad, et ainsi de suite.

```
$ ls -l grosfichier
-rw-r--r-- 1 seb users 1073741824 mar 12 19:47 grosfichier
$ split -b 150m grosfichier fic
$ ls -l fic*
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficaa
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficab
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficac
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficad
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficae
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficaf
-rw-r--r-- 1 seb users 130023424 mar 12 20:16 ficag
```

b. Reconstruire

Une ligne suffit pour reconstruire un fichier splité à l'aide des redirections :

```
$ cat fic* > newfic
$ ls -l newfic
-rw-r--r-- 1 seb users 1073741824 mar 12 20:47 newfic
```

10. Remplacement de caractères

a. Liste de caractères

La commande **tr** permet de substituer des caractères à d'autres et n'accepte que des données provenant du canal d'entrée standard, pas les fichiers.

```
tr [options] original destination
```

L'original et la destination représentent un ou plusieurs caractères. Les caractères originaux sont remplacés par les caractères de destination dans l'ordre indiqué. Les crochets permettent de définir des plages.

Par exemple, remplacer le o par le e et le i par le a.

```
$ cat liste | tr "oi" "ea"
Preduat ebjet   prax    quantates
seuras  eptaque 30      15
dur     30gaga  100     30
dur     70gaga  150     30
dasque  zap     12      30
dasque  seuple  10      30
ecran   15      150     20
ecran   17      300     20
ecran   19      500     20
clavaer 105     45      30
clavaer 115     55      30
carte   sen     45      30
carte   vadee   145     30
```

Avec cette commande vous pouvez convertir une chaîne en majuscules ou en minuscules.

```
$ cat liste | tr "[a-z]" "[A-Z]"
PRODUIT OBJET   PRIX    QUANTITES
SOURIS  OPTIQUE 30      15
DUR     30GIGA  100     30
DUR     70GIGA  150     30
DISQUE  ZIP     12      30
DISQUE  SOUPLE  10      30
ECRAN   15      150     20
ECRAN   17      300     20
ECRAN   19      500     20
CLAVIER 105     45      30
CLAVIER 115     55      30
CARTE   SON     45      30
CARTE   VIDEO   145     30
```

Supprimer les répétitions

Surtout, `tr` admet deux paramètres, `-s` (squeeze) et `-d` (delete), qui permettent de supprimer des caractères en doublons ou non. C'est parfait dans le cas de séparateurs multiples. Voici un exemple pratique où l'on cherche à isoler l'adresse IP d'une machine.

```
$ /sbin/ifconfig eth0
eth0      Lien encap:Ethernet  HWaddr 00:13:D3:D7:A4:6C
          inet adr:10.9.238.170  Bcast:10.9.239.255  asque:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15054381 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4991811 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:4157389034 (3964.7 Mb)  TX bytes:374974072 (357.6 Mb)
          Interruption:22 Adresse de base:0xcc00
```

Seule la deuxième ligne, contenant `inet`, vous intéresse :

```
$ /sbin/ifconfig eth0| grep "inet "
          inet adr:10.9.238.170  Bcast:10.9.239.255  Masque:255.255.252.0
```

Pour isoler l'adresse IP située après « `inet adr:` » le séparateur « : » semble intéressant mais dans ce cas un cut retournerait «`10.9.238.170 Bcast` » ce qui ne convient pas. L'astuce consiste à remplacer tous les espaces par un seul « : ». Le paramètre `-s` remplace une chaîne de n caractères identiques par un seul. S'il n'est pas précisé c'est le même caractère, sinon un caractère de substitution donné.

```
$ /sbin/ifconfig eth0| grep "inet " | tr -s " " ":"
:inet:adr:10.9.238.170:Bcast:10.9.239.255:Masque:255.255.252.0
```

Il n'y a plus qu'à compter : l'adresse IP est en quatrième position (le premier champ avant le premier « : » est vide).

```
$ /sbin/ifconfig eth0| grep "inet " | tr -s " " ":" | cut -d: -f4
10.9.238.170
```

b. Tabulations et espaces

La plupart des éditeurs remplacent les tabulations par des espaces. Or certaines commandes s'attendent à obtenir des tabulations comme délimiteurs de champs (comme cut). S'il est possible de s'en sortir avec tr, deux commandes sont à votre disposition pour ce cas spécifique.

La commande **expand** convertit les tabulations en espaces. La commande **unexpand** convertit les espaces en tabulations. Soit le fichier liste selon le modèle prédécent où les colonnes sont séparées par des espaces au lieu de tabulations. Dans le premier cas le résultat n'est pas du tout celui attendu. La commande **cut** tente de sortir le troisième champ d'un fichier tabulé. Comme il n'y a pas de tabulations, il affiche toute la ligne.

```
$ cut -f1 liste
Produit     objet       prix    quantites
souris      optique     30      15
dur         30giga      100     30
dur         70giga      150     30
disque      zip         12      30
disque      souple      10      30
...
```

La commande **unexpand** avec le paramètre -a remplace toutes les séquences d'au moins deux espaces par le nombre nécessaire de tabulations. Cette fois le résultat est correct.

```
$ unexpand -a liste | cut -f1
Produit
souris
dur
dur
disque
disque
...
```

11. Visualisation de texte

a. En pleine page

Rien n'empêche de détourner un quelconque flux pour l'afficher sur l'écran ou l'imprimante. Voici quelques commandes.

- page par page : **pg**, **more**, **less**
- en bloc : **cat**
- à l'envers : **tac**
- en dump hexadécimal : **hexdump**
- création d'une bannière : **banner**
- formatage pour impression : **pr**
- numéroter les lignes : **cat -n** ou **nl**

b. Début d'un fichier

Pour voir le début d'un fichier utilisez la commande **head**.

```
head [-c nbcar] [-n nblignes] [fic1...]
```

Le paramètre `-c` permet de préciser un nombre d'octets d'en-tête à afficher. Par défaut dix lignes sont affichées. Le paramètre `-n` permet d'indiquer le nombre de lignes à afficher. Vous pouvez indiquer directement le nombre de lignes :

```
head [-nblignes] [fic1...]

$ head -3 liste
Produit objet   prix    quantites
souris  optique 30      15
dur     30giga  100     30
```

c. Fin et attente de fichier

Pour voir les dernières lignes d'un fichier, utilisez la commande **tail**.

```
tail [+/-valeur[b/c]] [-f] [fic1...]
```

Comme pour head, par défaut les dix dernières lignes sont affichées. La valeur `-nblignes` permet de modifier cet état. Précisez c pour indiquer un nombre de caractères. Un b indique un nombre de blocs (512 octets par bloc).

Un `+` inverse l'ordre de la commande, et devient un `head (tail +10 <=> head -n 10)`.

Enfin l'option `-f` laisse le fichier ouvert. Ainsi si le fichier continue d'être rempli (par exemple un fichier trace), son contenu s'affichera en continu sur l'écran jusqu'à interruption volontaire par l'utilisateur ([Ctrl] **C**).

```
$ tail -5 liste
ecran   19      500     20
clavier 105     45      30
clavier 115     55      30
carte   son     45      30
carte   video   145     30

$ tail -10c liste
eo      145     30
```

12. Duplication du canal de sortie standard

Dans certains cas, comme par exemple la génération de fichiers traces, il peut être nécessaire de devoir à la fois placer dans un fichier le résultat d'une commande et de filtrer ce même résultat avec une autre commande. Utilisez pour cela la commande **tee** qui permet de dupliquer le flux de données. Elle lit le flux de données provenant d'une autre commande par le canal d'entrée, l'écrit dans un fichier et restitue ce flux à l'identique par le canal de sortie. Par défaut le fichier généré écrase l'ancien s'il existe.

```
tee [-a] nom_fic
```

Le paramètre `-a` signifie append. Dans ce cas le fichier n'est pas écrasé mais complété à la fin. Par exemple, vous voulez obtenir à la fois dans un fichier la liste des noms d'utilisateurs et afficher leur nombre sur écran.

```
$ cat /etc/passwd | cut -d: -f1 | tee users | wc -l
       65
$ cat users
root
nobody
nobodyV
daemon
bin
uucp
uucpa
auth
cron
lp
tcb
```

```
...
```

13. Comparaison de fichiers

Les deux commandes permettant de comparer le contenu de deux fichiers, ou d'un fichier et d'un flux sont les commandes **diff** et **cmp**.

a. diff

La commande **diff** indique les modifications à apporter aux deux fichiers en entrée pour que leur contenu soit identique.

```
diff [-b] [-e] fic1 fic2
```

L'option `-b` permet d'ignorer les espaces (blank), et l'option `-e` permet de générer un script ed (nous ne l'utiliserons pas). Cette commande renvoie trois types de messages :

- `APPEND : ligne1 a ligne3,ligne4, ex 5 a 6,8` veut dire : à la ligne 5 de fic1 il faut raccrocher les lignes 6 à 8 de fic2 pour que leurs contenus soient identiques.
- `DELETE : ligne1,ligne2 d ligne3, ex 7,9 d 6` veut dire : les lignes 7 à 9 de fic1 doivent être supprimées, elles n'existent pas derrière la ligne 6 de fic2.
- `CHANGE : ligne1,ligne2 c ligne3,ligne4, ex 8,12 c 9,13` veut dire : les lignes 8 à 12 de fic1 doivent être échangées contre les lignes 9 à 13 de fic2.

Dans tous les cas, le signe "<" indique les lignes de fic1 concernées, et le signe ">" les lignes de fic2 concernées.

```
$ cat liste
Produit objet   prix    quantites
souris  optique 30      15
dur     30giga  100     30
dur     70giga  150     30
disque  zip     12      30
disque  souple  10      30
ecran   15      150     20
ecran   17      300     20
ecran   19      500     20
clavier 105     45      30
clavier 115     55      30
carte   son     45      30
carte   video   145     30

$ cat liste2
Produit objet   prix    quantites
souris  boutons 30      15
dur     30giga  100     30
dur     70giga  150     30
disque  zip     12      30
disque  souple  10      30
ecran   15      150     20
ecran   17      300     20
ecran   19      500     20
ecran   21      500     20
clavier 105     45      30
clavier 115     55      30
```

Le fichier liste est l'original. Dans liste2, la deuxième ligne a été modifiée, une ligne écran a été ajoutée et les deux dernières lignes ont été supprimées.

```
$ diff liste liste2
2c2
< souris        optique 30      15
---
```

```
> souris        boutons 30      15
9a10
> ecran 21      500     20
12,13d12
< carte son     45      30
< carte video   145     30
```

- `2c2` : les lignes 2 de liste et liste2 doivent être échangées (elles doivent concorder soit en optique, soit en boutons).
- `9a10` : après la ligne 9 de liste (écran 19) il faut ajouter la ligne 10 (écran 21) de liste2.
- `12,13d12` : les lignes 12 et 13 de liste (carte son et vidéo) doivent être supprimés car elles n'existent pas après la ligne 12 de liste2.

b. cmp

La commande **cmp** compare les fichiers caractère par caractère. Par défaut la commande s'arrête dès la première différence rencontrée et indique la position de l'erreur.

```
cmp [-l] [-s] fic1 fic2
```

Le paramètre `-l` détaille toutes les différences en trois colonnes. La première colonne représente le numéro de caractère, la deuxième la valeur octale ASCII du caractère concerné de fic1 et la troisième la valeur octale ASCII du caractère concerné de fic2.

L'option `-s` retourne uniquement le code d'erreur (non visible), accessible par echo $?.

```
$ cmp liste liste2
liste liste2 differ: char 38, line 2
$ cmp -l liste liste2
    38 157 142
    39 160 157
    40 164 165
    41 151 164
    42 161 157
    43 165 156
    44 145 163
   182 143 145
   183 154 143
  ...
```

14. Délai d'attente

La commande **sleep** permet d'attendre le nombre de secondes indiqués. Le script est interrompu durant ce temps. Le nombre de secondes et un entier compris entre 0 et 4 milliards (136 ans).

```
$ sleep 10
```

Les processus

1. Définition et environnement

Un **processus** représente à la fois un programme en cours d'exécution et tout son environnement d'exécution (mémoire, état, identification, propriétaire, père...).

Voici une liste des données d'identification d'un processus :

- **Un numéro de processus unique PID** (Process ID) : chaque processus Unix est numéroté afin de pouvoir être différencié des autres. Le premier processus lancé par le système est 1 et il s'agit d'un processus appelé généralement init. On utilise le PID quand on travaille avec un processus. Lancer 10 fois le même programme (même nom) produit 10 PID différents.

- **Un numéro de processus parent PPID** (Parent Process ID) : chaque processus peut lui-même lancer d'autres processus, des processus enfants (child process). Chaque enfant reçoit parmi les informations le PID du processus père qui l'a lancé. Tous les processus ont un PPID sauf le processus 0 qui est un pseudo-processus représentant le démarrage du système (crée le 1 init).

- **Un numéro d'utilisateur et un numéro de groupe** : correspond à l'UID et au GID de l'utilisateur qui a lancé le processus. C'est nécessaire pour que le système sache si le processus a le droit d'accéder à certaines ressources ou non. Les processus enfants héritent de ces informations. Dans certains cas (que nous verrons plus tard) on peut modifier cet état.

- **Durée de traitement et priorité** : la durée de traitement correspond au temps d'exécution écoulé depuis le dernier réveil du processus. Dans un environnement multitâche, le temps d'exécution est partagé entre les divers processus, et tous ne possèdent pas la même priorité. Les processus de plus haute priorité sont traités en premier. Lorsqu'un processus est inactif, sa priorité augmente afin d'avoir une chance d'être exécuté. Lorsqu'il est actif, sa priorité baisse afin de laisser sa place à un autre. C'est l'ordonnanceur de tâches du système qui gère les priorités et les temps d'exécution.

- **Répertoire de travail actif** : à son lancement, le répertoire courant (celui depuis lequel le processus a été lancé) est transmis au processus. C'est ce répertoire qui servira de base pour les chemins relatifs.

- **Fichiers ouverts** : table des descripteurs des fichiers ouverts. Par défaut au début seuls trois sont présents : 0, 1 et 2 (les canaux standards). À chaque ouverture de fichier ou de nouveau canal, la table se remplit. À la fermeture du processus, les descripteurs sont fermés (en principe).

- On trouve d'autres informations comme la taille de la mémoire allouée, la date de lancement du processus, le terminal d'attachement, l'état du processus, les UID effectif et réel ainsi que les GID effectif et réel.

2. États d'un processus

Durant sa vie (temps entre le lancement et la sortie) un processus peut passer par divers états ou **process state** :

- exécution en mode utilisateur (**user mode**) ;

- exécution en mode noyau (**kernel mode**) ;

- en attente E/S (**waiting**) ;

- endormi (**sleeping**) ;

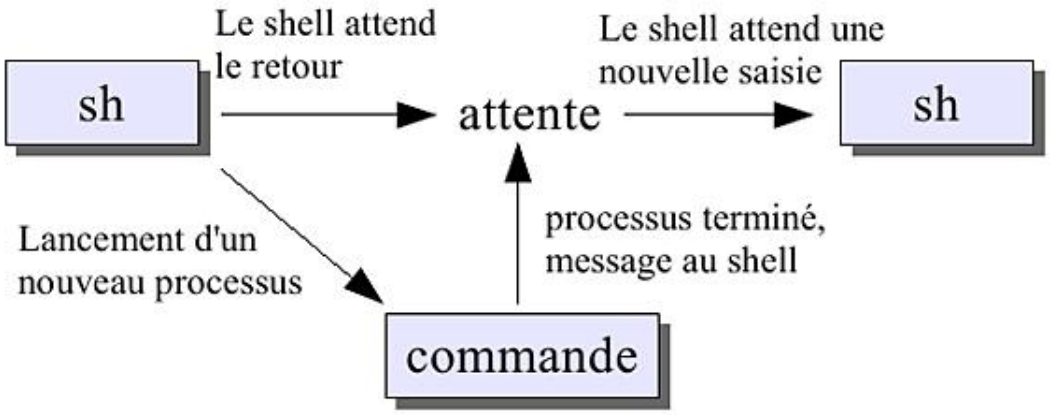
- prêt à l'exécution (**runnable**) ;

- endormi dans le swap (**mémoire virtuelle**) ;

- nouveau processus ;

- fin de processus (**zombie**).

3. Lancement en tâche de fond



En suivant ce que vous avez vu avant, le système étant multitâches un certain nombre de processus tournent déjà sur la machine sans que vous les voyiez. De même le shell que vous utilisez est lui-même un processus. Quand une commande est saisie, le shell crée un nouveau processus pour l'exécuter, ce processus devient un processus enfant du shell. Jusqu'à présent il fallait attendre la fin de l'exécution d'une commande pour en saisir une autre (sauf en utilisant « ; » pour chaîner les commandes).

Rien n'empêche le shell d'attendre le message du processus terminé pour rendre la main : de ce fait la commande une fois lancée, le shell peut autoriser la saisie d'une nouvelle commande sans attendre la fin de l'exécution de la commande précédente. Pour cela il suffit de saisir, après avoir tapé la commande, le **ET Commercial « & »**. Dans ce cas, le shell et la commande lancée fonctionneront en parallèle.

```
$ ls -R / > ls.txt 2/dev/null &
[1] 21976
$
[1]    Done                  ls -l -R / > ls.txt 2/dev/null
$ ls
fic1         fic3         liste        ls.txt       rep1         users
fic2         fic4         liste2       mypass       toto.tar.gz
```

Juste après la saisie un chiffre apparaît, il est à retenir car il s'agit du PID du nouveau processus lancé. Après une autre saisie une ligne Done indique que le traitement est terminé. La valeur [1] est propre à un shell particulier (ksh).

Quelques remarques sur l'utilisation du lancement en tâche de fond :

- Le processus lancé ne devrait pas attendre de saisie au risque de confusion entre cette commande et le shell lui-même.
- Le processus lancé ne devrait pas afficher de résultats sur l'écran au risque d'avoir des affichages en conflit avec celui du shell (par exemple apparition d'une ligne en milieu de saisie).
- Enfin, quand on quitte le shell, on quitte aussi tous ses fils : dans ce cas ne pas quitter le shell pendant un traitement important.

4. Backgroung, foreground, jobs

Vous pouvez récupérer la main sous le shell si vous avez lancé un processus au premier plan. Vous pouvez le stopper temporairement en tapant [Ctrl] **Z** :

```
$ sleep 100
<CTRL+Z>
 [1]+  Stopped                 sleep 100
```

Le processus est stoppé : son exécution est suspendue jusqu'à ce que vous le replaciez au premier plan avec la commande **fg** (*foreground*) :

```
$ fg
Sleep 100
```

Quand vous lancez une commande, vous avez remarqué le premier nombre entre crochets, c'est le numéro de job. Vous pouvez en obtenir la liste avec la commande **jobs**.

```
$ jobs
[1]-  Stopped                 sleep 100
[2]+  Stopped                 sleep 100
```

Les commandes **bg** et **fg** permettent d'agir sur ces jobs en prenant comme paramètre leur numéro. La commande **bg** est exécutée sur un job stoppé pour le relancer en arrière-plan. Le job 2 est relancé en arrière-plan :

```
$ bg 2
[2]+ sleep 100 &
$
[2]+  Done                    sleep 100
```

5. Liste des processus

La commande **ps** (*process status*) permet d'avoir des informations sur les processus en cours. Lancée seule, elle n'affiche que les processus en cours lancés par l'utilisateur et depuis la console actuelle.

```
$ ps
  PID TTY          TIME CMD
 4334 pts/1    00:00:00 bash
 5017 pts/1    00:00:00 ps
```

Pour avoir plus d'informations, utilisez le paramètre `-f`.

```
ps -f
UID        PID  PPID  C STIME TTY          TIME CMD
seb       4334 24449  0 09:46 pts/1    00:00:00 /bin/bash
seb       5024  4334  0 10:51 pts/1    00:00:00 ps -f
```

Le paramètre `-e` donne des informations sur tous les processus en cours.

```
$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
...
seb      26431     1  0 Mar04 ?        00:00:30 kded [kdeinit]
seb      26436 26322  0 Mar04 ?        00:00:03 kwrapper ksmserver
seb      26438     1  0 Mar04 ?        00:00:00 ksmserver [kdeinit]
seb      26439 26424  0 Mar04 ?        00:00:50 kwin [kdeinit
seb      26441     1  0 Mar04 ?        00:00:28 kdesktop [kdeinit]
seb      26443     1  0 Mar04 ?        00:03:40 kicker [kdeinit]
seb      26453     1  0 Mar04 ?        00:00:00 kerry [kdeinit
seb      26454 26424  0 Mar04 ?        00:00:01 evolution
seb      26465 26424  0 Mar04 ?        00:00:11 kde-window-decorator
seb      26467     1  0 Mar04 ?        00:00:02 gconfd-2 12
seb      26474     1  0 Mar04 ?        00:00:01 knotify [kdeinit]
seb      26485     1  0 Mar04 ?        00:03:06 beagled
...
```

Le paramètre `-u` permet de préciser une liste d'un ou plusieurs utilisateurs séparés par une virgule. Le paramètre `-g` effectue la même chose mais pour les groupes, `-t` pour les terminaux et `-p` pour des PID précis.

```
$ ps -u root
  PID TTY          TIME CMD
    1 ?        00:00:05 init
    2 ?        00:00:00 kthreadd
    3 ?        00:00:00 migration/0
    4 ?        00:00:09 ksoftirqd/0
```

```
    5 ?        00:00:23 events/0
    6 ?        00:00:00 khelper
   25 ?        00:00:00 kblockd/0
   26 ?        00:00:00 kacpid
   27 ?        00:00:00 kacpi_notify
  130 ?        00:00:00 cqueue/0
  131 ?        00:00:00 kseriod
  156 ?        00:00:22 kswapd0
  157 ?        00:00:00 aio/0...
```

Enfin le paramètre `-l` propose plus d'informations techniques.

```
$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4704 24449  0  75   0 -  1213 wait   pts/3    00:00:00 bash
```

Voici le détail de quelques colonnes.

Colonne	Définition
UID	User ID, nom de l'utilisateur.
PID	Process ID, numéro du processus.
PPID	Parent Process ID, numéro du processus père.
C	Facteur de priorité, plus la valeur est grande plus la priorité est élevée.
STIME	Heure de lancement du processus.
TTY	Nom du terminal depuis lequel le processus a été lancé.
TIME	Durée de traitement du processus.
CMD	Commande exécutée.
F	Drapeaux du processus (sort du cadre de l'ouvrage).
S	État du processus S (sleeping) R (running) Z (zombie).
PRI	Priorité du processus.
NI	Nice, incrément pour le scheduler.

6. Arrêt d'un processus / signaux

Lorsqu'un processus tourne en tâche de fond, il ne peux pas être arrêté par une quelconque combinaison de touches, sauf en utilisant le gestionnaire de jobs avec fg et bg. Il peut être nécessaire de lui envoyer des signaux auquel il pourra éventuellement réagir. Pour cela il faut employer la commande **kill**. Contrairement à ce que son nom semble indiquer, le rôle de cette commande n'est pas forcément de détruire ou de terminer un processus (récalcitrant ou non), mais d'envoyer des signaux aux processus.

```
kill [-l] -Num_signal PID [PID2...]
```

Le **signal** est l'un des moyens de communication entre les processus. Lorsqu'on envoie un signal à un processus, celui-doit doit l'intercepter et réagir en fonction de celui-ci. Certains signaux peuvent être ignorés, d'autres non. Suivant les Unix on dispose d'un nombre plus ou moins important de signaux. Les signaux sont numérotés et nommés, mais attention, si les noms sont généralement communs d'un Unix à l'autre, les numéros ne le sont pas forcément. L'option **-l** permet d'obtenir la liste des signaux.

Signal	Rôle

1 (SIGHUP)	Hang Up, est envoyé par le père à tous ses enfants lorsqu'il se termine.
2 (SIGINT)	Interruption du processus demandé (touche [Suppr], [Ctrl] **C**).
3 (SIGQUIT)	Idem SIGINT mais génération d'un Core Dump (fichier de débuggage).
9 (SIGKILL)	Signal ne pouvant être ignoré, force le processus à finir 'brutalement'.
15 (SIGTERM)	Signal envoyé par défaut par la commande **kill**. Demande au processus de se terminer normalement.

```
$ sleep 100&
[1] 5187
$ kill 5187
$
[1]+  Complété              sleep 100
$ sleep 100&
[1] 5194
$ kill -9 5194
[1]+  Processus arrêté      sleep 100
```

7. nohup

Quand le shell est quitté (exit, [Ctrl] D...) le signal 1 SIGHUP est envoyé aux enfants pour qu'ils se terminent aussi. Lorsqu'un traitement long est lancé en tâche de fond et que l'utilisateur veut quitter le shell, ce traitement sera alors arrêté et il faudra tout recommencer. Le moyen d'éviter cela est de lancer le traitement (processus) avec la commande **nohup**. Dans ce cas le processus lancé ne réagira plus au signal SIGHUP, et donc le shell pourra être quitté, la commande continuera son exécution.

Par défaut les canaux de sortie et d'erreur standards sont redirigés vers un fichier **nohup.out**, sauf si la redirection est explicitement précisée.

```
$ nohup ls -lR / &
10206
$ Sending output to nohup.out
```

Quand un fils se termine, il envoie le signal SIGCHLD à son père. Sauf cas prévu (le père se détache du fils) le père doit obtenir autant de SIGCHLD qu'il a eu de fils ou émis de SIGHUP. Si le père se termine avant que les fils se terminent ceux-ci deviennent des zombis : le signal SIGCHLD n'est pas reçu... Le processus fils est bien terminé, il est mort, il ne consomme aucune ressource. Il ne peut donc être tué (puisqu'il est mort) mais continue à occuper une entrée dans la table des processus. C'est init qui le récupère, et init étant toujours en attente, le zombie peut finir par disparaître.

8. nice et renice

La commande **nice** permet de lancer une commande avec une priorité pus faible, afin de permettre éventuellement à d'autres processus de tourner plus rapidement.

```
nice [-valeur] commande [arguments]
```

Une valeur positive causera une baisse de priorité, une négative l'augmentation de la priorité (si autorisé). La valeur doit être comprise entre -20 et 20. Plus la valeur est élevée et plus le traitement est ralenti.

```
$ nice -10 ls -lR / >liste 2>/dev/null&
10884
$ ps -l
       F S     UID    PID   PPID    C PRI  NI  ADDR   SZ WCHAN    TTY
           TIME CMD
80808001 U N+   75  10884  10822 28.5  54  10     0 848K aa3b3a9c ttyp4
        0:03.32 ls
```

La commande **renice** fonctionne un peu comme nice mais elle permet de modifier la priorité en fonction d'un utilisateur, d'un groupe ou d'un PID. La commande visée doit donc déjà tourner.

```
renice [-n prio] [-p] [-g] [-u] ID
```

La priorité doit être comprise entre -20 et 20. L'utilisateur standard ne peut utiliser que les valeurs entre 0 et 20 permettant de baisser la priorité. L'option -p précise un PID, -g un GID et -u un UID.

9. time

La commande **time** mesure les durées d'exécution d'une commande, idéal pour connaître les temps de traitement, et retourne trois valeurs :

- **real** : durée totale d'exécution de la commande ;
- **user** : durée du temps CPU nécessaire pour exécuter le programme ;
- **system** : durée du temps CPU nécessaire pour exécuter les commandes liées à l'OS (appels système au sein d'un programme).

Le résultat est sorti par le canal d'erreur standard 2. On peut avoir une indication de la charge de la machine par le calcul real / (user+system). Si le résultat est inférieur à 10, la machine dispose de bonnes performances, au-delà de 20 la charge de la machine est trop lourde (performances basses).

```
$ time ls -lR /home
...
real   4.8
user   0.2
sys    0.5
```

Plus loin avec le bash

1. Alias

Un alias est un raccourci d'une commande avec d'éventuels paramètres. Il se définit avec la commande **alias**. Utilisée seule elle liste les alias disponibles.

```
$ alias
alias ..='cd ..'
alias ...='cd ../..'
alias cd..='cd ..'
alias dir='ls -l'
alias l='ls -alF'
alias la='ls -la'
alias ll='ls -l'
alias ls='ls $LS_OPTIONS'
alias ls-l='ls -l'
alias md='mkdir -p'
alias o='less'
alias rd='rmdir'
...
```

Vous pouvez créer vos propres alias.

```
$ alias deltree='rm -rf'
```

2. Groupement de commandes

Le chaînage de commande est possible avec « ; ». Il est aussi possible de grouper les commandes. Quand vous exécutez les commandes suivantes :

```
$ uname -a ; pwd ; ls -l >resultat.txt &
```

Seule la dernière commande est exécutée en tâche de fond et seul son résultat est redirigé dans le fichier resultat.txt. Une solution serait :

```
$ uname -a >resultat.txt & ; pwd >>resultat.txt & ; ls -l >>resultat.txt &
[1] 18232
[2] 18238
[3] 18135
```

C'est une solution complexe et qui ne fonctionnera pas toujours. De plus même si les commandes sont lancées séquentiellement, elles tournent toutes en parallèle et c'est la première finie qui écrira en premier dans le fichier. La solution consiste en l'utilisation des parenthèses.

```
$ (uname -a ; pwd ; ls -l) > resultat.txt &
[1] 18239
$
[1]    Done                 ( uname -a; pwd; ls -l ) > resultat.txt
```

Dans ce cas, toutes les commandes placées entre les parenthèses sont lancées par un sous-shell, qui va ensuite exécuter les commandes précisées séquentiellement telles qu'indiquées. Ainsi la redirection concerne l'ensemble des commandes et rien n'empêche de lancer ce sous-shell en arrière-plan. Vous distinguez bien d'ailleurs un seul PID 18239 lors de l'exécution des commandes.

Une seconde possibilité est l'utilisation des accolades {...}. Dans ce cas aucun sous-shell n'est exécuté, et si une commande interne (cd ou autre) est exécutée, elle concerne le shell actif. L'accolade fermante doit être placée juste après un ;.

```
$ { uname -a; pwd; ls -l; } > resultat.txt
```

Vous pouvez faire facilement la différence entre les deux syntaxes avec exit. Le premier exemple semble ne rien faire, alors qu'il quitte le shell fils. Le second sort de votre shell.

```
$ (exit)
$ { exit; }
```

Attention avec les parenthèses, notamment en programmation. Comme le groupement est lancé au sein d'un autre processus, les éventuelles variables modifiées au sein du groupement ne seront pas visibles une fois l'exécution terminée.

3. Liaison et exécution conditionnelle

En plus du chaînage classique, les commandes peuvent être liées et exécutées de façon conditionnelle. La condition d'exécution d'une commande est la réussite ou non de la précédente. Chaque commande une fois exécutée renvoie un code de retour, généralement 0 si tout s'est bien passé, 1 ou 2 en cas d'erreur. Le shell peut récupérer cette valeur par la variable $?.

```
$ ls
...
$ echo $?
0
```

Les caractères **&&** et **||** permettent d'effectuer une exécution conditionnelle.

```
commande1 && commande2
commande1 || commande2
```

La commande située après **&&** sera exécutée uniquement si la commande précédente a retourné 0 (réussite). La commande située après **||** ne sera exécutée que si la commande précédente a retourné autre chose que 0.

```
$ grep "souris" liste && echo "Souris trouvee" || echo "Souris in-
trouvable"
souris  optique 30      15
Souris trouvee
$ grep "memoire" liste && echo "Memoire trouvee" || echo "Memoire
introuvable"
Memoire introuvable
```

Les variables

On en distingue trois types : utilisateur, système et spéciales. Le principe est de pouvoir affecter un contenu à un nom de variable, généralement une chaîne de caractère ou des valeurs numériques.

1. Nomenclature

Un nom de variable obéit à certaines règles :

- Il peut être composé de lettres minuscules, majuscules, de chiffres, de caractères de soulignement.
- Le premier caractère ne peut pas être un chiffre.
- Le taille d'un nom est en principe illimitée (il ne faut pas abuser non plus).
- Les conventions veulent que les variables utilisateur soient en minuscules pour les différencier des variables système. Au choix de l'utilisateur.

2. Déclaration et affectation

Une variable est déclarée dès qu'une valeur lui est affectée. L'affectation est effectuée avec le signe **=**, sans espace avant ou après le signe.

```
var=Bonjour
```

3. Accès et affichage

Vous accédez au contenu d'une variable en plaçant le signe **$** devant le nom de la variable. Quand le shell rencontre le **$**, il tente d'interpréter le mot suivant comme étant une variable. Si elle existe, alors le $nom_variable est remplacé par son contenu, ou par un texte vide dans le cas contraire. On parle aussi de référencement de variable.

```
$ chemin=/tmp/seb
$ ls $chemin
...
$ cd $chemin
$ pwd
/tmp/seb
$ cd $chemin/rep1
$ pwd
/tmp/seb/rep1
```

Pour afficher la liste des variables on utilise la commande **env**. Elle affiche les variables utilisateur et les variables système, nom et contenu.

```
$ env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man
KDE_MULTIHEAD=false
SSH_AGENT_PID=26377
HOSTNAME=p64p17bicb3
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
```

```
...
```

Une variable peut contenir des caractères spéciaux, le principal étant l'espace. L'exemple suivant ne fonctionne pas :

```
$ c=Salut les copains
les: not found
$ echo $c
```

Pour cela il faut soit verrouiller les caractères spéciaux un par un, soit les mettre entre guillemets ou apostrophes.

```
c=Salut\ les\ Copains # Solution lourde
c="Salut les copains" # Solution correcte
c='Salut les copains' # Solution correcte
```

La principale différence entre les guillemets et les apostrophes est l'interprétation des variables et des substitutions. " et ' se verrouillent mutuellement.

```
$ a=Jules
$ b=Cesar
$ c="$a $b a conquis la Gaule"
$ d='$a $b a conquis la Gaule'
$ echo $c
Jules Cesar a conquis la Gaule
$ echo $d
$a $b a conquis la Gaule
$ echo "Linux c'est top"
Linux c'est top
$ echo 'Linux "trop bien"'
Linux "trop bien"
```

4. Suppression et protection

Vous supprimez une variable avec la commande **unset**. Vous protégez une variable en écriture et contre sa suppression avec la commande **readonly**. Une variable en lecture seule, même vide, est figée. Il n'existe aucun moyen de la replacer en écriture et de la supprimer, sauf en quittant le shell.

```
$ a=Jules
$ b=Cesar
$ echo $a $b
Jules Cesar
$ unset b
$ echo $a $b
Jules
$ readonly a
$ a=Neron
a: is read only
$ unset a
a: is read only
```

5. Export

Par défaut une variable n'est accessible que depuis le shell où elle a été définie. La variable **a** est déclarée sous l'invite du shell courant puis est affichée par un script lancé depuis ce même shell. Ce dernier ne connaît pas la variable **a** : rien ne s'affiche.

```
$ a=Jules
$ echo 'echo "a=$a"' > voir_a.sh
$ chmod u+x voir_a.sh
$ ./voir_a.sh
a=
```

La commande **export** permet d'exporter une variable de manière à ce que son contenu soit visible par les scripts et autres sous-shells. Les variables exportées peuvent être modifiées dans le script, mais ces modifications ne s'appliquent qu'au script ou au sous-shell. Cette fois le premier script peut accéder à la variable **a** exportée. Mais les modifications restent locales au script. Une fois celui-ci terminé la modification disparaît.

```
$ export a
$ ./voir_a.sh
a=Jules
$ echo 'a=Neron ; echo "a=$a"' >> voir_a.sh
$ ./voir_a.sh
a=Jules
a=Neron
$ echo $a
Jules
```

6. Accolades

Les accolades de base {} permettent d'identifier le nom d'une variable. Imaginez la variable fichier contenant le nom de fichier 'liste'. Vous voulez copier liste1 en liste2.

```
$ fichier=liste
$ cp $fichier1 $fichier2
cp: opérande fichier manquant
Pour en savoir davantage, faites: « cp --help ».
```

Cela ne fonctionne pas car ce n'est pas $fichier qui est interprété mais $fichier1 et $fichier2 qui n'existent pas.

```
$ cp ${fichier}2 ${fichier}1
```

Dans ce cas, cette ligne équivaut à :

```
$ cp liste2 liste1
```

7. Accolades et remplacement conditionnel

Les accolades disposent d'une syntaxe particulière.

```
{variable:Remplacement}
```

Selon la valeur ou la présence de la variable, il est possible de remplacer sa valeur par une autre.

Remplacement	Signification
{x:-texte}	Si la variable x est vide ou inexistante, le texte prendra sa place. Sinon c'est le contenu de la variable qui prévaudra.
{x:=texte}	Si la variable x est vide ou inexistante, le texte prendra sa place et deviendra la valeur de la variable.
{x:+texte}	Si la variable x est définie et non vide, le texte prendra sa place. Dans le cas contraire une chaîne vide prend sa place.
{x:?texte}	Si la variable x est vide ou inexistante, le script est interrompu et le message texte s'affiche. Si texte est absent un message d'erreur standard est affiché.

```
$ echo $nom

$ echo ${nom:-Jean}
Jean
$ echo $nom

$ echo ${nom:=Jean}
Jean
$ echo $nom
Jean
```

```
$ echo ${nom:+"Valeur définie"}
Valeur définie
$ unset nom
$ echo ${nom:?Variable absente ou non définie}
nom: Variable absente ou non définie
$ nom=Jean
$ echo ${nom:?Variable absente ou non définie}
Jean
```

8. Variables système

En plus des variables que l'utilisateur peut définir lui-même, le shell est lancé avec un certain nombre de variables prédéfinies utiles pour certaines commandes et accessibles par l'utilisateur. Le contenu de ces variables système peut être modifié mais il faut alors faire attention car certaines ont une incidence directe sur le comportement du système.

Variable	Contenu
HOME	Chemin d'accès du répertoire utilisateur. Répertoire par défaut en cas d'utilisation de CD.
PATH	Liste de répertoires, séparés par des : où le shell va rechercher les commandes externes et autres scripts et binaires. La recherche se fait dans l'ordre des répertoires saisis.
PS1	Prompt String 1, chaîne représentant le prompt standard affiché à l'écran par le shell en attente de saisie de commande.
PS2	Prompt String 2, chaîne représentant un prompt secondaire au cas où la saisie doit être complétée.
IFS	Internal Field Separator, liste des caractères séparant les mots dans une ligne de commande. Par défaut il s'agit de l'espace, de la tabulation et du saut de ligne.
MAIL	Chemin et fichier contenant les messages de l'utilisateur.
SHELL	Chemin complet du shell en cours d'exécution.
LANG	Définition de la langue à utiliser ainsi que du jeu de caractères.
USER	Nom de l'utilisateur en cours.
LOGNAME	Nom du login utilisé lors de la connexion.
HISTFILE	Nom du fichier historique, généralement $HOME/.sh_history.
HISTSIZE	Taille en nombre de lignes de l'historique.
OLDPWD	Chemin d'accès du répertoire accédé précédemment.
PS3	Définit l'invite de saisie pour un select.
PWD	Chemin d'accès courant.
RANDOM	Génère et contient un nombre aléatoire entre 0 et 32767.

9. Variables spéciales

Il s'agit de variables accessibles uniquement en lecture et dont le contenu est généralement contextuel.

Variable	Contenu
$?	Code retour de la dernière commande exécutée.
$$	PID du shell actif.
$!	PID du dernier processus lancé en arrière-plan.
$-	Les options du shell.

```
$ echo $$
23496
$ grep memoire liste
$ echo $?
1
$ grep souris liste
souris  optique 30      15
$ echo $?
0
$ ls -lR >toto.txt 2<&1 &
26675
$ echo $!
26675
```

10. Longueur d'une chaîne

Il est possible d'obtenir la longueur d'une chaîne avec le caractère #.

```
$ a=Jules
$ echo "Longueur de $a : ${#a}"
Longueur de Jules : 5
```

11. Tableaux et champs

Deux moyens sont disponibles pour déclarer un tableau, l'un avec l'utilisation des crochets **[ ]**, l'autre avec la création globale. Le premier élément est 0 le dernier 1023. Pour accéder au contenu du tableau il faut mettre la variable ET l'élément entre accolades **{}**.

```
$ Nom[0]="Jules"
$ Nom[1]="Romain"
$ Nom[2]="Francois"
$ echo ${Nom[1]}
Romain
```

ou :

```
$ Nom=(Jules Romain Francois)
$ echo ${nom[2]}
Francois
```

Pour lister tous les éléments :

```
$ echo ${Nom[*]}
Jules Romain Francois
```

Pour connaître le nombre d'éléments :

```
$ echo ${#Nom[*]}
3
```

Si l'index est une variable, on ne met pas le $ devant celui-ci :

```
$ idx=0
$ echo ${Nom[idx]}
Jules
```

12. Variables typées

Les variables peuvent être typées en entier (integer) avec la commande **typeset -i** le permet. L'avantage est qu'il devient possible d'effectuer des calculs et des comparaisons sans passer par expr. La commande **let** ou **((...))** permet des calculs sur variables.

Opérateur	Rôle
+ - * /	Opérations simples
%	Modulo
< > <= >=	Comparaisons, 1 si vraie, 0 si faux
== !=	Égal ou différent
&& ||	Comparaisons liées par un opérateur logique
& | ^	Logique binaire AND OR XOR

```
$ typeset -i resultat
$ resultat=6*7
$ echo $resultat
42
$ resultat=Erreur
ksh: Erreur: bad number
$ resultat=resultat*3
126
$ typeset -i add
$ add=5
$ let resultat=add+5 resultat=resultat*add
$ echo $resultat
50
```

Configuration de bash

1. Fichiers de configuration

Le shell bash peut être lancé dans plusieurs modes :

- le shell interactif de connexion (login shell) ;
- le shell interactif simple ;
- le shell non interactif ;
- le mode sh ;
- etc.

Selon son mode de démarrage, le shell va chercher et exécuter divers scripts et fichiers de configuration. Un fichier de configuration est un script shell, une séquence de commandes individuelles ayant pour but de configurer l'environnement de l'utilisateur.

a. Shell de connexion

Le shell de connexion est lancé après la saisie du login et du mot de passe sur la console. C'est celui précisé à la fin de chaque ligne de /etc/passwd. Dans ce mode, le shell cherche à exécuter, dans cet ordre et s'ils sont présents :

- /etc/profile
- ~/.bash_profile
- ~/.bash_login
- ~/.profile

À la déconnexion, il tente d'exécuter :

- - ~/.bash_logout

b. Shell simple

Le shell interactif simple correspond à l'exécution du bash dans une fenêtre (xterm, konsole), une console ou à la main (taper bash dans une console). Dans ce cas seul le fichier suivant sera exécuté s'il existe :

- ~/.bashrc

Notez que dans beaucoup de distributions Linux, le .bashrc est appelé soit par .bash_profile, soit par /etc/profile, et que la configuration est donc placée dans .bash_profile qui sera alors toujours appelé.

c. Mode Bourne shell

Lorsque bash est lancé en mode Bourne Shell via la commande **sh**, en shell de connexion ou non, il tente d'exécuter les fichiers dans cet ordre :

- /etc/profile

- ~/.profile

d. Mode non interactif

Le shell peut être lancé en mode non interactif. C'est généralement le cas lorsque vous exécutez un script. Dans ce cas il n'y a aucun script exécuté par défaut au démarrage sauf si vous précisez une variable appelée BASH_ENV qui contient le chemin d'un script. Dans ce cas bash charge et exécute ce fichier avant le début de l'exécution du script ou de la commande.

2. Commandes set

Le shell dispose de dizaines d'options, dont la plupart peuvent être paramétrées à l'aide de la commande **set**. Celles qui suivent ne sont qu'une simple sélection. Le - avant l'option permet de passer celle-ci à On. Un + passe l'option à off.

Option	Résultat
-a / -o allexport	Toutes les variables seront automatiquement exportées.
-u / -o nounset	Par défaut le shell traite les variables inexistantes comme des chaînes vides. Cette option produit une erreur si la variable n'existe pas.
-x / -o xtrace	Affiche toutes les commandes au fur et à mesure de leur exécution : idéal en début de script pour débuggage.
-o vi	Manipulation de la ligne de commande avec la syntaxe de vi.
-o emacs	Manipulation de la ligne de commande avec la syntaxe de emacs.
-C / -o noclobber	Interdit les redirections en sortie si le fichier existe déjà.
history	Autorise la gestion de l'historique.

Le manuel du shell vous fournira toutes les options possibles.

Programmation shell

1. Structure et exécution d'un script

Le shell n'est pas qu'un simple interpréteur de commandes, mais dispose d'un véritable langage de programmation avec notamment une gestion des variables, des tests et des boucles, des opérations sur les variables, des fonctions...

Toutes les instructions et commandes sont regroupées au sein d'un script. Lors de son exécution, chaque ligne sera lue une à une et exécutée. Une ligne peut se composer de commandes internes ou externes, de commentaires ou être vide. Plusieurs instructions par lignes sont possibles, séparées par le **;** ou liées conditionnellement par **&&** ou **||**. Le **;** est l'équivalent d'un **saut de ligne**.

Par convention les noms des shell scripts se terminent généralement (pas obligatoirement) par « .sh » pour le Bourne Shell et le Bourne Again Shell, par « .ksh » pour le Korn Shell et par « .csh » pour le C Shell.

Pour rendre un script exécutable directement :

```
$ chmod u+x monscript
```

Pour l'exécuter :

```
$ ./monscript
```

Pour éviter le ./ :

```
$ PATH=$PATH:.
$ monscript
```

Notez que le point est placé en dernier dans le PATH. Le mettre en premier peut présenter un risque pour la sécurité : une nouvelle commande **ls** modifiée est placée dans votre répertoire.

Quand un script est lancé, un nouveau shell fils est créé qui va exécuter chacune des commandes. Si c'est une commande interne, elle est directement exécutée par le nouveau shell. Si c'est une commande externe, dans le cas d'un binaire un nouveau fils sera créé pour l'exécuter, dans le cas d'un shell script un nouveau shell fils est lancé pour lire ce nouveau shell ligne à ligne.

Une ligne de commentaire commence toujours par le caractère **#**. Un commentaire peut être placé en fin d'une ligne comportant déjà des commandes.

```
# La ligne suivante effectue un ls
ls # La ligne en question
```

La première ligne a une importance particulière car elle permet de préciser quel shell va exécuter le script :

```
#!/bin/bash
#!/bin/ksh
```

Dans le premier cas c'est un script Bourne Again, dans l'autre un script Korn.

2. Arguments d'un script

a. Paramètres de position

Les paramètres de position sont aussi des variables spéciales utilisées lors d'un passage de paramètres à un script.

Variable	Contenu
$0	Nom de la commande (du script).
$1-9	$1,$2,$3... Les neuf premiers paramètres passés au script.
$#	Nombre total de paramètres passés au script.

$*	Liste de tous les paramètres au format "$1 $2 $3 ...".
$@	Liste des paramètres sous forme d'éléments distincts "$1" "$2" "$3" ...

```
$ cat param.sh
#!/bin/bash

echo "Nom : $0"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3"
echo "Liste : $*"
echo "Elements : $@"

$ param.sh riri fifi loulou
Nom : ./param.sh
Nombre de parametres : 3
Parametres : 1=riri 2=fifi 3=loulou
Liste : riri fifi loulou
Elements : riri fifi loulou
```

La différence entre **$@** et **$*** ne saute pas aux yeux. Reprenez l'exemple précédent avec une petite modification :

```
$ param.sh riri "fifi loulou"
Nom : ./param.sh
Nombre de parametres : 2
Parametres : 1=riri 2=fifi loulou 3=
Liste : riri fifi loulou
Elements : riri fifi loulou
```

Cette fois-ci il n'y a que deux paramètres de passés. Pourtant les listes semblent visuellement identiques. En fait si la première contient bien :

```
"riri fifi loulou"
```

La deuxième contient :

```
"riri" "fifi loulou"
```

Soit bien deux éléments. Dans le premier exemple vous aviez :

```
"riri" "fifi" "loulou"
```

b. Redéfinition des paramètres

Outre le fait de lister les variables, l'instruction **set** permet de redéfinir le contenu des variables de position. Avec :

```
set valeur1 valeur2 valeur3 ...
```

$1 prendra comme contenu valeur1, $2 valeur2 et ainsi de suite.

```
$ cat param2.sh
#!/bin/bash
echo "Avant :"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"
set alfred oscar romeo zoulou
echo "apres set alfred oscar romeo zoulou"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param2.sh riri fifi loulou donald picsou
Avant :
Nombre de parametres : 5
```

```
Parametres : 1=riri 2=fifi 3=loulou 4=donald
Liste : riri fifi loulou donald picsou
apres set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou
```

c. Réorganisation des paramètres

La commande **shift** est la dernière commande permettant de modifier la structure des paramètres de position. Un simple appel décale tous les paramètres d'une position en supprimant le premier : $2 devient $1, $3 devient $2 et ainsi de suite. Le $1 originel disparaît. $#, $* et $@ sont redéfinis en conséquence.

La commande **shift** suivie d'une valeur n effectue un décalage de n éléments. Ainsi avec `shift 4` $5 devient $1, $6 devient $2, ...

```
$ cat param3.sh
#!/bin/bash
set alfred oscar romeo zoulou
echo "set alfred oscar romeo zoulou"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"
shift
echo "Après un shift"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param3.sh
set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou
Après un shift
Nombre de parametres : 3
Parametres : 1=oscar 2=romeo 3=zoulou 4=
Liste : oscar romeo zoulou
```

d. Sortie de script

La commande **exit** permet de mettre fin à un script. Par défaut la valeur retournée est 0 (pas d'erreur) mais n'importe quelle autre valeur de 0 à 255 peut être précisée. Vous récupérez la valeur de sortie par la variable $?.

```
$ exit 1
```

3. Environnement du processus

En principe seules les variables exportées sont accessibles par un processus fils. Si vous souhaitez visualiser l'environnement lié à un fils (dans un script par exemple) utilisez la commande **env**.

```
$ env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man
KDE_MULTIHEAD=false
SSH_AGENT_PID=28012
HOSTNAME=slyserver
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
TERM=xterm
```

```
PROFILEREAD=true
HISTSIZE=1000
XDM_MANAGED=/var/run/xdmctl/xdmctl-
:0,maysd,mayfn,sched,rsvd,method=classic
XDG_SESSION_COOKIE=16af07a56781b4689718210047060300-
1211264847.394692-546885666
TMPDIR=/tmp
GTK2_RC_FILES=/etc/gtk-2.0/gtkrc:/usr/share/themes//QtCurve/gtk-
2.0/gtkrc:/home/seb/.gtkrc-2.0-qtengine:/home/seb/.gtkrc-
2.0:/home/seb/.kde/share/config/gtkrc-2.0
KDE_NO_IPV6=1
GTK_RC_FILES=/etc/gtk/gtkrc:/home/seb/.gtkrc:/home/seb/.kde/share/co
nfig/gtkrc
GS_LIB=/home/seb/.fonts
WINDOWID=71303176
MORE=-sl
QTDIR=/usr/lib/qt3
XSESSION_IS_UP=yes
KDE_FULL_SESSION=true
GROFF_NO_SGR=yes
JRE_HOME=/usr/lib/jvm/jre
USER=seb
...
```

La commande **env** permet de redéfinir aussi l'environnement du processus à lancer. Cela peut être utile lorsque le script doit accéder à une variable qui n'est pas présente dans l'environnement du père, ou qu'on ne souhaite pas exporter. La syntaxe est :

```
env var1=valeur var2=valeur ... commande
```

Dans le cas de bash, env n'est pas indispensable.

```
var1=valeur var2=valeur ... commande
```

Si la première option est le signe - alors c'est tout l'environnement existant qui est supprimé pour être remplacé par les nouvelles variables et valeurs.

```
$ unset a
$ ./voir_a.sh
a=
$ env a=jojo ./voir_a.sh
a=jojo
$ echo a=$a
a=
```

4. Substitution de commande

Le mécanisme de substitution permet de placer le résultat de commandes simples ou complexes dans une variable. Vous placez les commandes à exécuter entre des accents graves « ` » ([Alt Gr] **7**).

```
$ mon_unix=`uname`
$ echo ${mon_unix}
Linux
$ machine=`uname -a | cut -d" " -f5`
echo $machine
SMP
```

Attention, seul le canal de sortie standard est affecté à la variable. Le canal d'erreur standard sort toujours vers l'écran dans ce cas.

Les accents graves ne sont pas toujours idéaux pour ces manipulations. En effet si vous travaillez à plusieurs niveaux, vous devez verrouiller ceux qui sont à l'intérieur des premiers niveaux. Aussi le bash permet d'utiliser à la place la syntaxe **$(...)** qui n'a pas ce problème.

```
$ mon_unix=$(uname)
$ echo ${mon_unix}
Linux
```

```
$ machine=$(uname -a | cut -d" " -f5)
echo $machine
SMP
```

Ne confondez pas les accolades et les parenthèses. Les premières isolent les variables, les secondes effectuent la substitution des commandes.

5. Tests de conditions

La commande **test** permet d'effectuer des tests de conditions. Le résultat est récupérable par la variable $? (code retour). Si ce résultat est 0 alors la condition est réalisée.

a. Tests sur une chaîne

- **test -z "variable"** : zero, retour OK si la variable est vide (ex : test -z "$a").
- **test -n "variable"** : non zero, retour OK si la variable n'est pas vide (texte quelconque).
- **test "variable"** = chaîne : OK si les deux chaînes sont identiques.
- **test "variable" !** = chaîne : OK si les deux chaînes sont différentes.

```
$ a=
$ test -z "$a" ; echo $?
0
$ test -n "$a" ; echo $?
1
$ a=Jules
$ test "$a" = Jules ; echo $?
0
```

Attention à bien placer vos variables contenant du texte entre guillemets. Dans le cas contraire un bug se produira si la variable est vide :

```
$ a=
$ b=toto
$ [ $a = $b ] && echo "ok"
bash: [: =: unary operator expected
```

Alors que :

```
[ "$a" = "$b" ] && echo "ok"
```

ne produit pas d'erreur.

b. Tests sur les valeurs numériques

Les chaînes à comparer sont converties en valeurs numériques. Bash ne gère que des valeurs entières. La syntaxe est :

```
test valeur1 option valeur2
```

et les options sont les suivantes :

Option	Rôle
-eq	Equal : égal
-ne	Not Equal : différent

-lt	Less than : inférieur
-gt	Greater than : supérieur
-le	Less or equal : inférieur ou égal
-ge	Greater or equal : supérieur ou égal

```
$ a=10
$ b=20
$ test  "$a" -ne "$b" ; echo $?
0
$ test "$a" -ge "$b" ; echo $?
1
$ test "$a" -lt "$b" && echo "$a est inferieur a $b"
10 est inferieur a 20
```

c. Tests sur les fichiers

La syntaxe est :

```
test option nom_fichier
```

et les options sont les suivantes :

Option	Rôle
-f	Fichier normal.
-d	Un répertoire.
-c	Fichier en mode caractère.
-b	Fichier en mode bloc.
-p	Tube nommé (named pipe).
-r	Autorisation en lecture.
-w	Autorisation en écriture.
-x	Autorisation en exécution.
-s	Fichier non vide (au moins un caractère).
-e	Le fichier existe.
-L	Le fichier est un lien symbolique.
-u	Le fichier existe, SUID-Bit positionné.
-g	Le fichier existe SGID-Bit positionné.

```
$ ls -l
-rw-r--r--   1 seb   users      1392 Aug 14 15:55 dump.log
lrwxrwxrwx   1 seb   users         4 Aug 14 15:21 lien_fic1 -> fic1
lrwxrwxrwx   1 seb   users         4 Aug 14 15:21 lien_fic2 -> fic2
-rw-r--r--   1 seb   users       234 Aug 16 12:20 liste1
```

```
-rw-r--r--   1 seb   users       234 Aug 13 10:06 liste2
-rwxr--r--   1 seb   users       288 Aug 19 09:05 param.sh
-rwxr--r--   1 seb   users       430 Aug 19 09:09 param2.sh
-rwxr--r--   1 seb   users       292 Aug 19 10:57 param3.sh
drwxr-xr-x   2 seb   users      8192 Aug 19 12:09 rep1
-rw-r--r--   1 seb   users      1496 Aug 14 16:12 resultat.txt
-rw-r--r--   1 seb   users      1715 Aug 16 14:55 toto.txt
-rwxr--r--   1 seb   users        12 Aug 16 12:07 voir_a.sh
$ test -f lien_fic1 ; echo $?
1
$ test -x dump.log ; echo $?
1
$ test -d rep1 ; echo $?
0
```

d. Tests combinés par des critères ET, OU, NON

Vous pouvez effectuer plusieurs tests avec une seule instruction. Les options de combinaison sont les mêmes que pour la commande **find**.

Critère	Action
`-a`	AND, ET logique
`-o`	OR, OU logique
`!`	NOT, NON logique
`(...)`	groupement des combinaisons. Les parenthèses doivent être verrouillées \(...\).

```
$ test -d "rep1" -a -w "rep1" && echo "rep1: repertoire, droit en
ecriture"
rep1: repertoire, droit en ecriture
```

e. Syntaxe allégée

Le mot test peut être remplacé par les crochets ouverts et fermés **[...]**. Il faut respecter un espace avant et après les crochets.

```
$ [ "$a" -lt "$b" ] && echo "$a est inferieur a $b"
10 est inferieur a 20
```

Le bash (et le ksh) intègre une commande interne de test qui se substitue au binaire test. Dans la pratique, la commande interne est entièrement compatible avec la commande externe mais bien plus rapide car il n'y a pas de création de nouveau processus. Pour forcer l'utilisation de la commande interne, utilisez les doubles crochets **[[...]]**.

```
$ [[ "$a" -lt "$b" ]] && echo "$a est inferieur a $b"
10 est inferieur a 20
```

6. if ... then ... else

La structure **if then else fi** est une structure de contrôle conditionnelle.

```
if <commandes_condition>
then
    <commandes exécutées si condition réalisée>
else
    <commandes exécutées si dernière condition pas réalisée>
fi
```

Vous pouvez aussi préciser **elif**, en fait un else if. Si la dernière condition n'est pas réalisée, on en teste une nouvelle.

```
$ cat param4.sh
#!/bin/bash
if [ $# -ne 0 ]
then
        echo "$# parametres en ligne de commande"
else
        echo "Aucun parametre; set alfred oscar romeo zoulou"
        set alfred oscar romeo zoulou
fi

echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param4.sh titi toto
2 parametres en ligne de commande
Nombre de parametres : 2
Parametres : 1=toto 2=titi 3= 4=
Liste : toto titi

$ ./param4.sh
Aucun parametre; set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou
```

7. Choix multiples case

La commande **case ... esac** permet de vérifier le contenu d'une variable ou d'un résultat de manière multiple.

```
case Valeur in
    Modele1) Commandes ;;
    Modele2) Commandes ;;
    *) action_defaut ;;
esac
```

Le modèle est soit un simple texte, soit composé de caractères spéciaux. Chaque bloc de commandes lié au modèle doit se terminer par deux points-virgules. Dès que le modèle est vérifié, le bloc de commandes correspondant est exécuté. L'étoile en dernière position (chaîne variable) est l'action par défaut si aucun critère n'est vérifié. Elle est facultative.

Caractère	Rôle
*	Chaîne variable (même vide)
?	Un seul caractère
[...]	Une plage de caractères
[!...]	Négation de la plage de caractères
|	OU logique

```
$ cat case1.sh
#!/bin/bash
if [ $# -ne 0 ]
then
        echo "$# parametres en ligne de commande"
else
        echo "Aucun parametre; set alfred oscar romeo zoulou"
        exit 1
fi
```

```
case $1 in
        a*)
                echo "Commence par a"
                ;;
        b*)
                echo "Commence par b"
                ;;
        fic[123])
                echo "fic1 fic2 ou fic3"
                ;;
        *)
                echo "Commence par n'importe"
                ;;
esac

exit 0
$ ./case1.sh "au revoir"
Commence par a
$ ./case1.sh bonjour
Commence par b
$ ./case1.sh fic2
fic1 ou fic2 ou fic3
$ ./case1.sh erreur
Commence par n'importe
```

8. Saisie de l'utilisateur

La commande **read** permet à l'utilisateur de saisir une chaîne et de la placer dans une ou plusieurs variable. La saisie est validée par [Entrée].

```
read var1 [var2 ...]
```

Si plusieurs variables sont précisées, le premier mot ira dans var1, le second dans var2, et ainsi de suite. S'il y a moins de variables que de mots, tous les derniers mots vont dans la dernière variable.

```
$ cat read.sh
#!/bin/bash
echo "Continuer (O/N) ? \c"
read reponse
echo "reponse=$reponse"
case $reponse in
        O)
                echo "Oui, on continue"
                ;;
        N)
                echo "Non, on s'arrête"
                exit 0
                ;;
        *)
                echo "Erreur de saisie (O/N)"
                exit 1
                ;;
esac
echo "Vous avez continue. Tapez deux mots ou plus :\c"
read mot1 mot2
echo "mot1=$mot1\nmot2=$mot2"
exit 0
$ ./read.sh
Continuer (O/N) ? O
reponse=O
Oui, on continue
Vous avez continue. Tapez deux mots ou plus :salut les amis
mot1=salut
mot2=les amis
```

9. Les boucles

Elles permettent la répétition d'un bloc de commandes soit un nombre limité de fois, soit conditionnellement. Toutes les commandes à exécuter dans une boucle se placent entre les commandes **do** et **done**.

a. Boucle for

La boucle **for** ne se base pas sur une quelconque incrémentation de valeur mais sur une liste de valeurs, de fichiers ...

```
for var in liste
do
    commandes à exécuter
done
```

La liste représente un certain nombre d'éléments qui seront successivement attribués à var.

Avec une variable

```
$ cat for1.sh
#!/bin/bash
for params in $@
do
        echo "$params"
done
$ ./for1.sh test1 test2 test3
test1
test2
test3
```

Liste implicite

Si vous ne précisez aucune liste à for, alors c'est la liste des paramètres qui est implicite. Ainsi le script précédent aurait pu ressembler à :

```
for params
do
        echo "$params"
done
```

Avec une liste d'éléments explicite

Chaque élément situé après le « in » sera utilisé pour chaque itération de la boucle, l'un après l'autre.

```
$ cat for2.sh
#!/usr/bin/sh
for params in liste liste2
do
        ls -l $params
done
$ ./for2.sh
-rw-r--r--   1 oracle   system       234 Aug 19 14:09 liste
-rw-r--r--   1 oracle   system       234 Aug 13 10:06 liste2
```

Avec des critères de recherche sur nom de fichiers

Si un ou plusieurs éléments de la liste correspond à un fichier ou à un motif de fichiers présents à la position actuelle de l'arborescence, la boucle for considère l'élément comme un nom de fichier.

```
$ cat for3.sh
#!/bin/bash
for params in *
do
        echo "$params \c"
        type_fic=`ls -ld $params | cut -c1`
        case $type_fic in
```

```
                -)      echo "Fichier normal" ;;
                d)      echo "Repertoire" ;;
                b)      echo "mode bloc" ;;
                l)      echo "lien symbolique" ;;
                c)      echo "mode caractere" ;;
                *)      echo "autre" ;;
        esac

done
$ ./for3.sh
case1.sh Fichier normal
dump.log Fichier normal
for1.sh Fichier normal
for2.sh Fichier normal
for3.sh Fichier normal
lien_fic1 lien symbolique
lien_fic2 lien symbolique
liste Fichier normal
liste1 Fichier normal
liste2 Fichier normal
param.sh Fichier normal
param2.sh Fichier normal
param3.sh Fichier normal
param4.sh Fichier normal
read.sh Fichier normal
rep1 Repertoire
resultat.txt Fichier normal
toto.txt Fichier normal
voir_a.sh Fichier normal
```

Avec une substitution de commande

Toute commande produisant une liste de valeurs peut être placée à la suite du « in » à l'aide d'une substitution de commande. La boucle **for** prendra le résultat de cette commande comme liste d'éléments sur laquelle boucler.

```
$ cat for4.sh
#!/bin/bash
echo "Liste des utilisateurs dans /etc/passwd"
for params in `cat /etc/passwd | cut -d: -f1`
do
        echo "$params "
done
$ ./for4.sh
Liste des utilisateurs dans /etc/passwd
root
nobody
nobodyV
daemon
bin
uucp
uucpa
auth
cron
lp
tcb
adm
ris
carthic
ftp
stu
...
```

b. Boucle while

La commande **while** permet une boucle conditionnelle « tant que ». Tant que la condition est réalisée, le bloc de commande est exécuté. On sort si la condition n'est plus valable.

```
while condition
do
    commandes
done
```

ou :

```
while
bloc d'instructions formant la condition
do
    commandes
done
```

Par exemple :

```
$ cat while1.sh
#!/bin/bash
while
        echo "Chaine ? \c"
        read nom
        [ -z "$nom" ]
do
        echo "ERREUR : pas de saisie"
done
echo "Vous avez saisi : $nom"
```

Lecture d'un fichier ligne à ligne

```
#!/bin/bash
cat toto.txt | while read line
do
           echo "$line"
done
```

ou :

```
#!/bin/bash
while read line
do
           echo "$line"
done < toto.txt
```

Il y a une énorme différence entre les deux versions. Dans la première, notez la présence du tube (pipe) : la boucle est exécutée dans un second processus. Aussi toute variable modifiée au sein de cette boucle perd sa valeur en sortie !

c. Boucle until

La commande **until** permet une boucle conditionnelle « jusqu'à ». Dès que la condition est réalisée, on sort de la boucle.

```
until condition
do
    commandes
done
```

ou :

```
until
bloc d'instructions formant la condition
do
    commandes
done
```

d. true et false

La commande **true** ne fait rien d'autre que de renvoyer 0. La commande **false** renvoie toujours 1. De cette manière il est possible de réaliser des boucles sans fin. La seule manière de sortir de ces boucles est un exit ou un break.

Par convention, tout programme qui ne retourne pas d'erreur retourne 0, tandis que tout programme retournant une erreur, ou un résultat à interpréter, retourne autre chose que 0. C'est l'inverse en logique booléenne.

```
while true
do
    commandes
    exit / break
done
```

e. break et continue

La commande **break** permet d'interrompre une boucle. Dans ce cas le script continue après la commande **done**. Elle peut prendre un argument numérique indiquant le nombre de boucles à sauter, dans le cas de boucles imbriquées (rapidement illisible).

```
while true
do
    echo "Chaine ? \c"
    read a
    if [ -z "$a" ]
    then
        break
    fi
done
```

La commande **continue** permet de relancer une boucle et d'effectuer un nouveau passage. Elle peut prendre un argument numérique indiquant le nombre de boucles à relancer (on remonte de n boucles). Le script redémarre à la commande **do**.

f. Boucle select

La commande **select** permet de créer des menus simples, avec sélection par numéro. La saisie s'effectue au clavier avec le prompt de la variable PS3. Si la valeur saisie est incorrecte, une boucle s'effectue et le menu s'affiche à nouveau. Pour sortir d'un select il faut utiliser un **break**.

```
select variable in liste_contenu
do
    traitement
done
```

Si `in liste_contenu` n'est pas précisé, ce sont les paramètres de position qui sont utilisés et affichés.

```
$ cat select.sh
#!/bin/bash
PS3="Votre choix :"
echo "Quelle donnee  ?"
select reponse in Jules Romain Francois quitte
do
        if [[ "$reponse" = "quitte" ]]
        then
                break
        fi
        echo "Vous avez choisi $reponse"
done
echo "Au revoir."
exit 0

$ ./select.sh
Quelle donnee  ?
1) Jules
2) Romain
3) Francois
4) quitte
```

```
Votre choix :1
Vous avez choisi Jules
Votre choix :2
Vous avez choisi Romain
Votre choix :3
Vous avez choisi Francois
Votre choix :4
Au revoir.
```

10. Les fonctions

Les fonctions sont des bouts de scripts nommés, directement appelés par leur nom, pouvant accepter des paramètres et retourner des valeurs. Les noms de fonctions suivent les mêmes règles que les variables sauf qu'elles ne peuvent pas être exportées.

```
nom_fonction ()
{
    commandes
    return
}
```

Les fonctions peuvent être soit tapées dans votre script courant, soit dans un autre fichier pouvant être inclus dans l'environnement. Pour cela saisissez :

```
. nomfic
```

Le point suivi d'un nom de fichier charge son contenu (fonctions et variables) dans l'environnement courant.

La commande **return** permet d'affecter une valeur de retour à une fonction. Il ne faut surtout pas utiliser la commande **exit** pour sortir d'une fonction, sinon on quitte le script.

```
$ cat fonction
ll ()
{
        ls -l $@
}
 li ()
{
        ls -i $@
}
$ . fonction
$ li
  252 case1.sh      326 for4.sh       187 param.sh      897 resultat.txt
  568 dump.log      409 lien_fic1     272 param2.sh     991 toto.txt
  286 fonction      634 lien_fic2     260 param3.sh     716 voir_a.sh
  235 for1.sh      1020 liste          42 param4.sh    1008 while1.sh
  909 for2.sh       667 liste1        304 read.sh
  789 for3.sh      1006 liste2        481 rep1
```

11. Calculs et expressions

a. expr

La commande **expr** permet d'effectuer des calculs sur des valeurs numériques, des comparaisons, et de la recherche dans des chaînes de texte.

Opérateur	Rôle
+	Addition.
-	Soustraction. L'étoile étant reconnue par le shell comme un wildcard, il faut la verrouiller avec un antislash : *.

*	Multiplication.
/	Division.
%	Modulo.
!=	Différent. Affiche 1 si différent, 0 sinon.
=	Égal. Affiche 1 si égal, 0 sinon.
<	Inférieur. Affiche 1 si inférieur, 0 sinon.
>	Supérieur. Affiche 1 si supérieur, 0 sinon.
<=	Inférieur ou égal. Affiche 1 si inférieur ou égal, 0 sinon.
>=	Supérieur ou égal. Affiche 1 si supérieur ou égal, 0 sinon.
:	Recherche dans une chaîne. Ex : expr Jules : J* retourne 1 car Jules commence par J. Syntaxe particulière : expr "Jules" : ".*" retourne la longueur de la chaîne.

```
$ expr 7 + 3
10
$ expr 7 \* 3
21
$ a=$(expr 13 - 10)
$ echo $a
3
$ cat expr1.sh
#!/bin/bash
cumul=0
compteur=0
nb_boucles=10
while [ "$compteur" -le "$nb_boucles" ]
do
        cumul=$(expr $cumul + $compteur)
        echo "$cumul=$cumul, boucle=$compteur"
        compteur=$(expr $compteur + 1)
done
$ ./expr1.sh
cumul=0, boucle=0
cumul=1, boucle=1
cumul=3, boucle=2
cumul=6, boucle=3
cumul=10, boucle=4
cumul=15, boucle=5
cumul=21, boucle=6
cumul=28, boucle=7
cumul=36, boucle=8
cumul=45, boucle=9
cumul=55, boucle=10
$ expr "Jules Cesar" : ".*"
11
```

b. Calculs avec bash

Le bash propose une forme simple de calculs sur les entiers, en plaçant l'opération entre **$((...))** :

```
$ a=1
$ a=$((a+1))
$ echo $a
2
$ b=2
```

```
$ a=$((a*b))
$ echo $a
4
```

Vous n'avez pas besoin de spécifier les $ des noms des variables entre les doubles parenthèses.

12. Une variable dans une autre variable

Voici un exemple :

```
$ a=Jules
$ b=a
$ echo $b
a
```

Comment afficher le contenu de a et pas simplement a ? En utilisant la commande **eval**. Cette commande située en début de ligne essaie d'interpréter, si possible, la valeur d'une variable précédée par deux « $ », comme étant une autre variable.

```
$ eval echo \$$b
Jules

$ cat eval.sh
cpt=1
for i in a b c
do
    eval $i=$cpt
    cpt=$((cpt+1))
done
echo $a $b $c

$ ./eval.sh
1 2 3
```

13. Traitement des signaux

La commande **trap** permet de modifier le comportement du script à la réception d'un signal.

Commande	Réaction
trap " signaux	Ignore les signaux. trap " 2 3 ignore les signaux 2 et 3.
trap 'commandes' signaux	Pour chaque signal reçu exécution des commandes indiquées.
trap signaux	Restaure les actions par défaut pour ces signaux.

Dans l'exemple suivant trap empêche l'exécution du [Ctrl] **C** (SIGINT) et intercepte le signal SIGTERM :

```
$ cat trap.sh
!#/bin/bash

sortir()
{
        echo "Signal 15 recu"
        exit 0
}

trap '' 2
trap sortir 15

while true
do
        echo "Ctrl+C impossible!"
```

```
done

$ ./trap.sh
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
...
```

Depuis une autre console :

```
$ kill -2 12456 # aucun effet
$ kill -15 12456 # SIGTERM
Sur la premiere console :
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Ctrl+C impossible!
Signal 15 recu
```

14. Commande « : »

La commande **« : »** est généralement totalement inconnue des utilisateurs Unix. Elle retourne toujours la valeur 0 (réussite). Elle peut donc être utilisée pour remplacer la commande **true** dans une boucle par exemple :

```
while :
do
    ...
done
```

Cette commande placée en début de ligne, suivie d'une autre commande, traite la commande et ses arguments mais ne fait rien, et retourne toujours 0. Elle peut être utile pour tester les variables.

```
$ : ls
$ : ${X:?"Erreur"}
X : Erreurs
```

SQL

1. Présentation

Le SQL, *Structured Query Language*, est un langage standardisé ISO d'interrogation et de manipulation de bases de données relationnelles. Ses possibilités peuvent être décomposées en quatre fonctions :

- Le langage de **définition de données** : création, modification et suppression des tables, index, contraintes, etc.
- Le langage de **manipulation de données** : certainement le plus connu, il concerne les requêtes « classiques » : ajout, suppression, modification et sélection d'enregistrements au sein de la base.
- Le langage de **contrôle de données** : mise en place et gestion des privilèges des utilisateurs de la base.
- Le langage de **contrôle des transactions** : gestion notamment des « commit » et des éventuels retours arrière, des procédures, etc.

Il peut être utile pour un administrateur de savoir créer des requêtes dans une base de données : certains outils se servent de bases SQL pour y stocker leurs données. Même dans le cas contraire, le SQL étant très utilisé et très pratique, il est utile de le connaître. Cette initiation vous montre comment effectuer les quatre requêtes de base : sélection, ajout, modification et suppression de données. Le lecteur avisé désirant en savoir plus sur le SQL et les bases de données pourra se reporter aux livres « MySQL 5 » et « SQL et Algèbre Relationnelle » aux éditions ENI.

Le modèle de base de données utilisé pour cette présentation est présent en annexe de ce livre.

2. Requêtes de sélection

a. Select

Les sélections, avec l'instruction SELECT, sont les principales requêtes utilisées en SQL, permettant d'extraire des données d'une ou de plusieurs tables selon des critères donnés.

```
SELECT nom_champ1, nom_champ2, …
FROM table;
```

Le résultat est retourné sous forme d'une table dont les en-têtes sont les noms des champs sélectionnés. Il est possible de les renommer avec AS. Par exemple :

```
SELECT nom, prenom, id AS identifiant FROM t_utilisateurs;
```

L'utilisation de l'étoile « * » comme champ permet de sélectionner tous les champs de la table. La requête suivante liste tout le contenu de la table :

```
SELECT * FROM t_utilisateurs;
```

b. Distinct

Si la requête retourne des lignes identiques, vous pouvez supprimer les doublons avec `DISTINCT`. Dans le cas suivant, il est logique de penser que plusieurs utilisateurs ont le même prénom. La cause DISTINCT va supprimer en sortie les doublons (ou plus).

```
SELECT DISTINCT prenom FROM t_utilisateurs;
```

c. Where

La clause `WHERE` spécifie les conditions de sélection des lignes.

```
SELECT nom_champ1, nom_champ2, …
FROM table
WHERE condition;
```

Les conditions s'appliquent sur tous les champs de la table, y compris sur ceux qui ne sont pas dans le Select. Il est possible d'utiliser de nombreux opérateurs : =, >=, <=, >, <, <> (différent) et la négation avec le point d'exclamation : != (différent), !> (pas supérieur), !< (pas inférieur). Les critères de conditions peuvent être liés par les opérateurs logiques AND, OR et NOT. Si le critère est un texte, vous le mettez entre guillemets simples : 'titi'.

```
SELECT prenom FROM t_utilisateurs WHERE nom='ROHAUT';
```

Attributs de Where

Vous pouvez utiliser les attributs LIKE, BETWEEN ou IN dans une clause WHERE.

- `LIKE` compare la valeur d'un champ avec une valeur texte spécifiée avec d'éventuels caractères de remplacement : le « % » pour une chaîne quelconque et le « _ » pour un caractère quelconque.
- `BETWEEN` spécifie un intervalle de données.
- `IN` propose une liste d'éléments.

Cette requête extrait tous les enregistrements dont les ids sont compris entre 0 et 100.

```
SELECT * FROM t_utilisateurs WHERE id BETWEEN 0 AND 1000;
```

Celle-ci extrait tous les enregistrements dont le prénom correspond à un prénom composé « Jean- » : Jean-Jacques, Jean-Luc, Jean-Marie, etc.

```
SELECT * FROM t_utilisateurs WHERE prenom LIKE 'Jean-%';
```

Enfin cette dernière extrait les utilisateurs habitant à Paris, Lille, Lyon et Marseille.

```
SELECT * FROM t_utilisateurs WHERE ville IN ('Paris', 'Lille',
'Lyon', 'Marseille');
```

3. Les expressions et les fonctions

Vous pouvez utiliser des expressions, notamment les calculs, au sein d'une instruction SELECT, que ce soit dans la sélection des champs ou dans la clause WHERE. La requête suivante calcule le prix TTC d'un produit, considérant que le champ tva contient le pourcentage de TVA appliqué au produit.

```
SELECT prix+(prix*tva) AS prix_net FROM t_produits WHERE
id_produit='1245';
```

Sur des chaînes de caractères, le « + » permet de concaténer :

```
SELECT nom+' '+prenom FROM t_utilisateurs;
```

Les fonctions s'appliquent comme les expressions sur les champs sélectionnés ou dans la clause WHERE. Elles sont de trois types principaux : mathématiques et statistiques, dates, chaînes de caractères. Voici quelques exemples :

Cette requête compte le nombre de lignes dans une table :

```
SELECT count(*) FROM t_utilisateurs;
```

Celle-ci extrait les prix minimum, maximum et moyen sur l'ensemble de la table des produits :

```
SELECT min(prix), max(prix), avg(prix) FROM t_produits ;
```

Cette dernière extrait le montant total du stock de produits :

```
SELECT sum(prix*qte) from t_produits;
```

Les fonctions de date s'appliquent sur les champs de type date, sauf `current_date` qui retourne la date du jour :

```
SELECT current_date;
```

La requête suivante extrait tous les utilisateurs s'étant inscrits en 2008 :

```
SELECT * from t_utilisateurs WHERE month(d_inscription)='2008';
```

Enfin, voici un exemple de requête qui extrait tous les utilisateurs dont le nom commence par un 'RI' en récupérant les deux premiers caractères du nom avec la fonction RIGHT :

```
SELECT * FROM t_utilisateurs WHERE right(nom)
```

4. La clause ORDER BY

La clause `ORDER BY` permet de trier les résultats. Vous pouvez indiquer l'ordre de tri, croissant avec `ASC`, décroissant avec `DESC`. La requête suivante trie les produits, du moins cher au plus cher, en tenant compte de la TVA :

```
SELECT prix+(prix*tva) AS prix_net FROM t_produits ORDER BY prix_net
ASC;
```

5. La clause GROUP BY

La clause `GROUP BY` regroupe les résultats par champs de valeurs identiques, les champs étant placés après la clause. La requête suivante affiche le nombre total de produits en stock par fournisseurs :

```
SELECT id_fournisseur, sum(qte) FROM t_produits GROUP BY
id_fournisseur;
```

La clause `HAVING` s'utilise avec GROUP BY et permet de réduire la sélection avec un critère sur le ou les champs de regroupement. Par exemple, dans la requête ci-dessus vous pouvez regrouper uniquement sur un seul fournisseur donné :

```
SELECT id_fournisseur, sum(qte) FROM t_produits GROUP BY
id_fournisseur HAVING id_fournisseur='1';
```

6. Les jointures

Les requêtes précédentes ne sont effectuées que sur une seule table. Il est possible d'extraire des enregistrements de plusieurs tables en même temps, si les enregistrements disposent de champs communs que l'on peut relier. La jointure établit une liaison entre l'un (ou plus) des attributs de chaque table. Le nombre de tables et de jointures n'est pas limité.

```
SELECT table1.champ1, table2.champ1
FROM table1, table2
WHERE table1.champcommun=table2.champcommun;
```

Voici une requête qui extrait le nom unique de chaque fournisseur pour chaque produit :

```
SELECT DISTINCT t_fournisseurs.nom
FROM t_fournisseurs, t_produits
WHERE t_produits.id_fournisseur=t_fournisseurs.id_fournisseur;
```

Le AS peut être utilisé pour renommer les tables dans la requête. Si le nom du champ est unique sur toutes les tables, alors la table n'a pas besoin d'être précisée. La requête peut se transformer ainsi :

```
SELECT DISTINCT nom
FROM t_fournisseurs as t1, t_produits as t2
WHERE t2.id_fournisseur=t1.id_fournisseur
```

Les jointures peuvent s'effectuer sur plus de deux tables. Ainsi pour avoir la liste des fournisseurs par clients, vous

pouvez procéder ainsi en disposant de la table des clients appelée "t_utilisateurs", la table des commandes, la table des produits et la table des fournisseurs :

```
SELECT DISTINCT t_fournisseurs.nom
FROM t_utilisateurs, t_commande, t_produits
WHERE t_utilisateurs.id=t_commande.id
AND t_commande.id_produit=t_produits.id_produit
AND t_produits.id_fournisseur=t_fournisseurs.id_fournisseur
AND t_utilisateurs.id='1'
```

Il est possible d'effectuer une jointure d'une table sur elle-même, dans ce cas la table doit être répétée deux fois (ou plus) après le FROM, mais avec deux noms différents spécifiés avec AS. L'exemple suivant extrait les noms et prénoms du parrain de l'utilisateur 2 :

```
SELECT B.nom, B.prenom
FROM t_utilisateurs AS A, t_utilisateurs AS B
WHERE B.id = A.id_parrain
AND A.id = '2';
```

7. Un Select dans un Select

Il est possible de faire des sélections imbriquées, c'est-à-dire que le retour d'une première sélection sert de critère pour une seconde sélection. Par exemple voici comment sortir les noms et prénoms de tous les parrains :

```
SELECT id
FROM t_utilisateurs
WHERE id IN (select id_parrain from t_utilisateurs WHERE
id_parrain!='')
```

Si la sous-requête ne retourne qu'un seul résultat, utilisez un symbole d'égalité, sinon utilisez IN.

8. Les insertions

Insérez des enregistrements dans une table SQL avec la commande `INSERT`.

```
INSERT INTO table (champ1, champ2, etc)
VALUES ('valeur1', 'valeur2', ...);
```

Par exemple, voici comment insérer un nouveau client dans la table t_utilisateurs :

```
INSERT INTO t_utilisateurs (id, nom, prenom, ville, id_parrain)
VALUES ('3', 'Le Canet', 'Jules', 'Beauvais', NULL);
```

9. Modifications

L'instruction `UPDATE` met à jour un ou plusieurs enregistrements d'une table. La clause `WHERE` est optionnelle. Dans ce cas, tous les enregistrements sont mis à jour.

```
UPDATE table SET champ1='valeur', champ2='valeur'
[WHERE champn='valeur']
```

Voici comment augmenter tous les prix de 5 % :

```
UPDATE t_produits SET prix=prix*1.05 ;
```

10. Suppression

L'instruction `DELETE` supprime un ou plusieurs enregistrements d'une ou de plusieurs tables. Attention, selon le modèle de base de données, aux contraintes d'intégrité fonctionnelles : dans certains cas, supprimer une référence peut soit ne pas marcher, soit supprimer n enregistrements en cascade.

```
DELETE FROM table
WHERE champ1=’valeur’;
```

Voici comment supprimer l’utilisateur 3 créé ci-dessus :

```
DELETE from t_utilisateurs WHERE id=3;
```

Représentation des disques

1. Nomenclature

Ceci est un petit rappel des points déjà rencontrés dans le chapitre Présentation de Linux. Suivant le type de contrôleur et d'interface sur lesquels les disques sont connectés, Linux donne des noms différents aux fichiers spéciaux des périphériques disques.

Chaque disque est représenté par un fichier spécial de type bloc. Chaque partition aussi.

a. IDE

Les disques reliés à des contrôleurs IDE (appelés aussi PATA, Parallel Ata, ou ATAPI) se nomment hdX :

- hda : IDE0, Master
- hdb : IDE0, Slave
- hdc : IDE1, Master
- hdd : IDE1, Slave
- etc.

Contrairement aux idées reçues, il n'y a pas de limites au nombre de contrôleurs IDE, sauf le nombre de ports d'extension de la machine (slots PCI). De nombreuses cartes additionnelles existent, de nombreuses cartes mères proposent jusqu'à quatre, six, huit connecteurs. Dans ce cas, les fichiers se nomment hde, hdf, hdg, etc.

Les lecteurs CD-Rom, DVD et graveurs sont vus comme des disques IDE et respectent cette nomenclature.

Les derniers noyaux Linux utilisent par défaut une API appelée libata pour accéder à l'ensemble des disques IDE, SCSI, USB, Firewire, etc. Si c'est votre cas (regardez les notes de version de la distribution), la nomenclature reprend celle des disques SCSI, abordée au point suivant.

b. SCSI, SATA, USB, FIREWIRE, etc.

Les disques reliés à des contrôleurs SCSI, SCA, SAS, FiberChannel, USB, Firewire (et probablement d'autres interfaces exotiques comme les lecteurs ZIP sur port parallèle) se nomment sdX. L'énumération des disques reprend l'ordre de détection des cartes SCSI et des adaptateurs (hosts) associés, puis l'ajout et la suppression manuelle via hotplug des autres.

- sda : premier disque SCSI
- sdb : deuxième disque SCSI
- sdc : troisième disque SCSI
- etc.

La norme SCSI fait une différence entre les divers supports. Aussi les lecteurs CD-Rom, DVD, HD-DVD, BlueRay et les graveurs associés n'ont pas le même nom. Les lecteurs et graveurs sont en srX (sr0, sr1, etc.). Vous pouvez aussi trouver scd0, scd1, etc. mais ce sont généralement des liens symboliques vers sr0, sr1, etc.

La commande **lsscsi** permet d'énumérer les périphériques SCSI.

```
$ lsscsi
[4:0:0:0]    disk    ATA      ST380011A        8.01  /dev/sda
[5:0:0:0]    cd/dvd  LITE-ON  COMBO SOHC-4836V S9C1  /dev/sr0
[31:0:0:0]   disk    USB2.0   Mobile Disk      1.00  /dev/sdb
```

2. Cas spéciaux

Certains contrôleurs ne suivent pas cette nomenclature. C'est par exemple le cas de certains contrôleurs RAID matériels. C'est du cas par cas. Un contrôleur Smart Array sur un serveur HP place ses fichiers de périphériques dans /dev/cciss sous les noms cXdYpZ, où X est le slot, Y le disque et Z la partition...

Manipulations de bas niveau

1. Informations

La commande **hdparm** permet d'effectuer un grand nombre de manipulations directement sur les périphériques disques gérés par la bibliothèque libata, c'est-à-dire tous les disques SATA, ATA (IDE) et SAS. La commande **sdparm** peut faire à peu près la même chose pour les disques SCSI. Notez que bien que les noms de périphériques de la libata soient identiques à ceux du SCSI, il est fort probable que de nombreuses options de configuration de hdparm ne fonctionnent pas sur des disques SCSI, la réciproque étant vraie pour sdparm avec les disques SATA ou IDE. La suite se base sur hdparm.

Pour obtenir des informations complètes sur un disque, utilisez les paramètres `-i` ou `-I`. Le premier récupère les informations depuis le noyau et obtenues au moment du boot, le second interroge directement le disque. Préférez le `-I` qui donne des informations très détaillées.

```
# hdparm -I /dev/sda

/dev/sda:

ATA device, with non-removable media
        Model Number:       ST380011A
        Serial Number:      5JVTH798
        Firmware Revision:  8.01
Standards:
        Used: ATA/ATAPI-6 T13 1410D revision 2
        Supported: 6 5 4
Configuration:
        Logical         max     current
        cylinders       16383   16383
        heads           16      16
        sectors/track   63      63
        --
        CHS current addressable sectors:   16514064
        LBA    user addressable sectors:  156301488
        LBA48  user addressable sectors:  156301488
        device size with M = 1024*1024:       76319 MBytes
        device size with M = 1000*1000:       80026 MBytes (80 GB)
Capabilities:
        LBA, IORDY(can be disabled)
        Standby timer values: spec'd by Standard, no device specific
minimum
        R/W multiple sector transfer: Max = 16  Current = 16
        Recommended acoustic management value: 128, current value: 0
        DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5
             Cycle time: min=120ns recommended=120ns
        PIO: pio0 pio1 pio2 pio3 pio4
             Cycle time: no flow control=240ns  IORDY flow control=120ns
Commands/features:
        Enabled Supported:
           *    SMART feature set
                Security Mode feature set
           *    Power Management feature set
           *    Write cache
           *    Look-ahead
           *    Host Protected Area feature set
           *    WRITE_BUFFER command
           *    READ_BUFFER command
           *    DOWNLOAD_MICROCODE
                SET_MAX security extension
           *    48-bit Address feature set
           *    Device Configuration Overlay feature set
           *    Mandatory FLUSH_CACHE
           *    FLUSH_CACHE_EXT
           *    SMART error logging
           *    SMART self-test
           *    General Purpose Logging feature set
```

```
                Time Limited Commands (TLC) feature set
                Command Completion Time Limit (CCTL)
Security:
        Master password revision code = 65534
                supported
        not     enabled
        not     locked
        not     frozen
        not     expired: security count
        not     supported: enhanced erase
HW reset results:
        CBLID- above Vih
        Device num = 0 determined by CSEL
Checksum: correct
```

2. Modification des valeurs

Plusieurs paramètres des disques peuvent être modifiés. Attention cependant ! Certaines options de hdparm peuvent se révéler être dangereuses tant pour les données contenues sur le disque que pour le disque lui-même. La plupart des paramètres sont en lecture et écriture. Si aucune valeur n'est précisée hdparm affiche l'état du disque (ou du bus) pour cette commande. Voici quelques exemples d'options intéressantes.

- **-c** : largeur du bus de transfert EIDE sur 16 ou 32 bits. 0=16, 1=32, 3=32 compatible.
- **-d** : utilisation du DMA. 0=pas de DMA, 1=DMA activé.
- **-X** : modifie le mode DMA (mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5). Vous pouvez utiliser l'un des modes précédents ou des valeurs numériques : 32+n pour les modes mdma (n variant de 0 à 2) et 64+n pour les modes udma.
- **-C** : statut de l'économie d'énergie sur le disque (unknown, active/idle, standby, sleeping). L'état peut être modifié avec -S, -y, -Y et -Z.
- **-g** : affiche la géométrie du disque.
- **-M** : indique ou modifie l'état du Automatic Acoustic Management (AAM). 0=off, 128=quiet et 254=fast. Tous les disques ne le supportent pas.
- **-r** : passe le disque en lecture seule.
- **-T** : bench de lecture du cache disque, idéal pour tester les performances de transfert entre Linux et le cache du disque. Il faut relancer la commande deux ou trois fois.
- **-t** : bench de lecture du disque, hors cache. Mêmes remarques que l'option précédente.

Ainsi la commande suivante passe le bus de transfert en 32 bits, active le mode DMA en mode Ultra DMA 5 pour le disque sda :

```
# hdparm -c1 -d3 -X udma5 /dev/sda
```

Voici quelques autres exemples :

```
# hdparm -c /dev/sda

/dev/sda:
 IO_support    =  0 (default 16-bit)

# hdparm -C /dev/sda

/dev/sda:
 drive state is:  active/idle
```

```
# hdparm -g /dev/sda

/dev/sda:
 geometry      = 9729/255/63, sectors = 156301488, start = 0
p64p17bicb3:/etc/cups #

# hdparm -T /dev/sda

/dev/sda:
 Timing cached reads:   1320 MB in  2.00 seconds = 660.30 MB/sec

# hdparm -t /dev/sda

/dev/sda:
 Timing buffered disk reads:  168 MB in  3.03 seconds =  55.49 MB/sec
```

Choisir un système de fichiers

1. Principe

a. Définition

L'action de « formater » un disque, une clé ou tout support de données consiste uniquement à créer sur un support de mémoire secondaire l'organisation logique permettant d'y placer des données. Le mot « formatage » n'est quasiment jamais utilisé sous Linux, sauf pour expliquer le principe aux personnes provenant d'autres horizons. On parle de système de fichiers qui est à la fois l'organisation logique des supports au niveau le plus bas comme au niveau de l'utilisateur.

Les informations ne sont pas écrites n'importe comment sur les disques. Une organisation est nécessaire pour y placer tant les informations sur les fichiers qui y sont stockés que les données. C'est le système de fichiers (et les pilotes associés) qui définit cette organisation. Si les principes de base sont souvent les mêmes entre les divers systèmes présents sous Linux, les implémentations et les organisations logiques des données sur le disque varient fortement. Aussi il n'existe pas un type de système de fichiers, mais plusieurs, au choix de l'utilisateur, administrateur ou ingénieur.

Le principe de base est d'associer un nom de fichier à son contenu et d'y permettre l'accès : création, modification, suppression, déplacement, ouverture, lecture, écriture, fermeture. Suivant ce principe, le système de fichiers doit gérer ce qui en découle : mécanismes de protection des accès (les permissions, les propriétaires), les accès concurrents, etc.

b. Représentation

Outre l'organisation et le stockage des informations et des données sur les fichiers, le système de fichiers doit fournir à l'utilisateur une vision structurée de ses données, permettant de les distinguer, de les retrouver, de les traiter et de les manipuler, par exemple sous forme de fichiers au sein d'une arborescence de répertoires avec les commandes associées. De même, chaque système de fichiers doit fournir le nécessaire pour que les programmes puissent y accéder.

Un système de fichiers Unix est organisé sous forme d'un arbre de répertoires et de sous-répertoires à partir d'une racine commune. C'est une arborescence. Chaque répertoire fait partie d'une organisation et propose lui-même une organisation : le système de fichiers dispose d'une hiérarchie ordonnée. L'arborescence elle-même peut être répartie sur plusieurs supports et systèmes de fichiers.

c. Les méta-données

Un fichier est décrit par des propriétés appelées les méta-données. Sous Linux, il s'agit de l'inode. Le contenu (les données) est placé dans d'autres blocs du support de stockage. Le contenu des méta-données diffère d'un système de fichiers à un autre. Cependant on y retrouve sous Linux :

- les droits ;
- les dernières dates d'accès et de modification ;
- le propriétaire et le groupe ;
- la taille ;
- le nombre de blocs utilisés ;
- le type de fichiers ;
- le compteur de liens ;
- un arbre d'adresses de blocs de données.

d. Les noms des fichiers

Les noms peuvent avoir une longueur de 255 caractères. L'extension n'a pas d'existence en tant que composante du système de fichiers contrairement à ce qui se passe sous Windows. Le type du fichier est déterminé par son contenu, notamment les premiers octets permettant de déterminer le type MIME. La commande **file** procède ainsi. L'extension n'est qu'une simple composante du nom du fichier, et incluse dans les 255 caractères. Elle est surtout utile pour que l'utilisateur différencie rapidement les fichiers entre eux.

Les noms des fichiers Unix ne sont pas placés dans les méta-données mais dans une table de catalogue. C'est pour ça qu'il est possible de donner plusieurs noms à un même fichier.

e. Le journal

Les systèmes de fichiers actuels disposent souvent de mécanismes permettant de garantir au mieux l'intégrité des données. Le système le plus courant est la journalisation (c'est un anglicisme). Le système de fichiers maintient à jour un journal, généralement d'une taille donnée et circulaire (les nouvelles informations finissent par écraser les anciennes) dans lequel il trace tous les changements intervenus avant de les effectuer réellement. En cas de coupure brutale, le système pointe les enregistrements du journal et vérifie si les opérations ont été effectuées, éventuellement il les rejoue. Le journal contient des opérations atomiques (n opérations indivisibles) et donc même si celui-ci est incomplet, la cohérence des données est assurée soit par complétion du journal quand c'est possible, soit par retour en arrière. La réparation est donc bien plus fiable et rapide.

2. Les filesystems sous Linux

a. ext2

Le « second extended filesystem » ext2 est considéré comme le système de fichiers historique de Linux, bien que celui-ci utilisait au tout début le MinixFS. La première mouture appelée ext (extended filesystem) bien que corrigeant les défauts de minix avait quelques limites qui n'en faisaient pas un véritable système de fichiers Unix. Ext2 est donc le premier système de fichiers développé spécifiquement pour Linux, d'un niveau de production et aux normes Unix (on parle de niveau de production pour indiquer un système quelconque répondant aux critères de mise en production (utilisation réelle) en entreprise). Prévu dès le début pour supporter les rajouts de fonctionnalités, il continue depuis 1993 à être utilisé et amélioré. Ext2 n'est pas journalisé.

Bien que disposant d'un successeur (ext3), il est toujours utilisé voire conseillé dans certains cas. Il est rapide et nécessite moins d'écritures que les autres, donc il occasionne moins d'usure des supports de stockage, notamment les disques SSD, les clés USB ou les cartes mémoire. Ces supports peuvent parfois ne supporter qu'un nombre restreint de cycles de lecture/écriture...

Les fichiers peuvent avoir jusqu'à une taille de 2To (2048 Go), tandis qu'une partition peut atteindre 32 To, voire 128 To , selon la taille des blocs et l'architecture.

b. ext3

Le « third extended filesystem » ext3 est le successeur de ext2 depuis 1999. Il est journalisé. Surtout, il est entièrement compatible avec ext2. Le journal est une extension de ext2. Il est possible d'utiliser un système de fichiers ext3 comme étant ext2, avec les mêmes commandes, les mêmes manipulations. Il est possible de transformer en quelques secondes un système ext2 en ext3, et vice versa. C'est l'un des systèmes de fichiers de choix pour Linux, et le plus utilisé pour sa souplesse.

Comme pour ext2, la taille maximale des fichiers est de 2 To, et celle d'une partition de 32 To, suivant les mêmes restrictions.

c. reiserfs

reiserfs a été le premier système de fichiers intégré à Linux, avant même ext3. Sa force réside, outre dans son journal, dans l'organisation indexée des entrées des répertoires (les tables catalogues contenant les associations inodes/fichiers) et la manipulation des fichiers de petite taille. Ses performances sont exceptionnelles en présence de milliers de fichiers, de faible à moyen volume. Il est redimensionnable à chaud. Il devient plus lent sur des gros fichiers.

Les fichiers peuvent atteindre 8 To, et les partitions 16 To. Les noms de fichiers peuvent avoir 4032 caractères mais sont limités par Linux à 255 caractères (plus précisément par le **VFS**, *Virtual Filesystem Switch*).

reiserfs est moins utilisé malgré ses fortes qualités pour diverses raisons dont la principale est l'impossibilité de convertir un système de fichiers ext2/ext3 en reiserfs et vice versa, et ce à cause de la forte base de machines installées en ext2/ext3.

d. xfs

xfs est le plus ancien des systèmes de fichiers journalisés sous Unix, datant de 1993. Créé par Silicon Graphics (sgi) il a été porté sous Linux en 2000. Outre ses capacités de stockages encore inimaginables aujourd'hui, il a un système de journalisation très performant et des mécanismes avancés comme la défragmentation en ligne (à chaud et au fur et à mesure des écritures), la capacité d'effectuer des snapshots (figer l'état d'un filesystem à un instant t pour le restaurer plus tard), le dimensionnement à chaud, la réservation de bande passante pour les entrées et sorties, etc.

La taille maximale théorique (parce que personne n'en a créé de si gros) des fichiers est de 8 Eo (Exaoctets). Sachant que 1 Eo vaut 1024 Po (Petaoctet) donc 1048576 To, soit, rendu à une unité plus appréhendable, environ 1000 milliards de DVD. La partition peut atteindre 16 Eo, soit la capacité maximale d'un contrôleur sur 64 bits. En 32 bits, les tailles sont « limitées » à 16 To.

L'utilisation de xfs est encore peu étendue sous Linux peut-être à cause de sa prétendue complexité pour un paramétrage avancé, mais aussi parce que Red Hat n'a pas de support officiel pour ce système de fichiers dans ses RHEL.

e. vfat

vfat (*Virtual File Allocation Table*) est un terme générique regroupant les diverses versions de FAT supportant les noms de fichiers longs (255 caractères) sous Windows. Ces systèmes de fichiers sont conservés et continuent d'être utilisés pour des raisons à la fois historiques et pratiques. La plupart des supports amovibles, disques externes, clefs USB et lecteurs MP3 utilisent un système de fichiers de ce type. Les raisons sont :

- Un système de fichiers adapté aux petits volumes.
- Un système de fichiers simple à implémenter, idéal pour des lecteurs multimédias.
- Une compatibilité entre diverses plates-formes (Windows, Linux, BSD, MacOS, etc.).

vfat souffre cependant de défauts inhérents à sa conception :

- L'ensemble des informations est stockée au sein d'une table unique, y compris le nom du fichier et chaque adresse et longueur des blocs (appelés clusters) composant les données du fichier.
- De ce fait, FAT tente de regrouper les données d'un fichier sur le plus de clusters contigus du support. En cas de nombreuses écritures (ajout, suppression, etc.), le système se retrouve fortement fragmenté.
- Toujours de ce fait, plus le support à une taille importante, plus FAT est lent, car il doit vérifier toute la table FAT pour trouver des clusters disponibles.
- La gestion des noms longs est considérée comme une bidouille par de nombreuses personnes car FAT doit continuer à assurer une compatibilité (encore aujourd'hui) avec les noms courts en 8.3.
- Contrairement aux systèmes de fichiers Unix, Linux ou Windows récents, FAT ne gère aucun attribut étendu, notamment aucune notion des droits et des propriétaires.
- FAT est limité à une taille de fichiers de 4 Go. Les fichiers plus gros (bases de données, archives, images ISO de DVD, etc.) doivent être découpés en conséquence et les logiciels (capture audio / vidéo, sgbd, etc.) doivent gérer cette limitation.

Linux gère parfaitement VFAT. Mais son utilisation sur des supports partagés entre Windows et Linux a de moins en moins de raison d'être car l'existence de ntfs3g permet d'utiliser des supports contenant un système de fichiers NTFS de manière native.

Partitionnement

1. Découpage logique

Pour la suite, le support de stockage sera considéré comme un support magnétique ou mémoire de type disque dur, SSD, carte mémoire, etc., c'est-à-dire tout ce qui peut être apparenté à un disque dur selon la vision classique : un espace de données pouvant être découpé en plusieurs entités logiques et indépendantes disposant chacunes de leur propre système de fichiers.

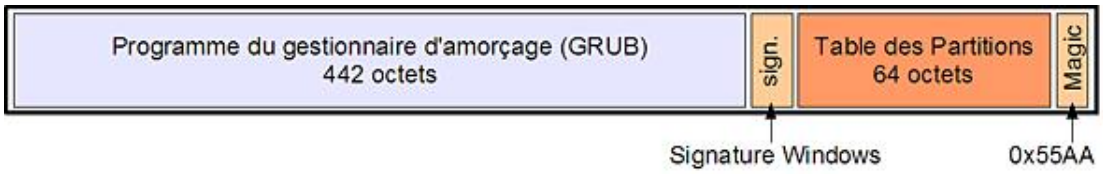
Un disque peut être vu comme une longue bande d'espace de stockage découpée en cases pouvant contenir une quantité donnée d'informations. Le disque peut être utilisé tel quel comme espace de stockage, rien n'empêche de créer un système de fichiers sur un disque sans passer par l'étape de partitionnement. Il est cependant important de donner une organisation logique à cet espace et aux systèmes de fichiers qu'il contiendra, ne serait-ce qu'au nom de la séparation des données (les fichiers de données) et des traitements (les programmes les utilisant et le système).

La partitionnement consiste en un découpage logique du disque. Le disque physique, réel, est fractionné en plusieurs disques virtuels, logiques, les partitions. Chaque partition est vue comme un disque indépendant et contient son propre système de fichiers.

2. Organisation d'un disque

a. MBR

Le premier secteur est le **MBR**, *Master Boot Record*, ou zone d'amorce. D'une taille de 512 octets il contient dans ses 444 premiers octets une routine (un programme) d'amorçage destiné soit à démarrer le système d'exploitation sur la partition active, soit un chargeur de démarrage (bootloader), puis 4 octets d'une signature optionnelle (Windows), 2 octets nuls, et les 64 octets suivants contiennent la table des quatre partitions primaires. Le tout finit par une signature 0xAA55 sur deux octets.



Format d'un MBR

b. Les partitions

Une partition est un découpage logique du disque. Il en existe trois sortes :

- Les partitions primaires, au nombre de quatre, sont celles décrites dans le MBR.
- Les partitions étendues (primaires étendues), une seule par disque (bien que théoriquement il soit possible de créer des partitions étendues au sein d'autres partitions étendues).
- Les partitions ou lecteurs logiques.

Un disque peut contenir jusqu'à 63 partitions en IDE, 15 en SCSI (c'est une limite de l'implémentation officielle du SCSI) ou via la libata. La limite actuelle est de 15 partitions pour tous les disques avec les derniers noyaux et l'API libata. Cependant quelques distributions permettent d'utiliser l'ancienne API (PATA) pour revenir à l'ancien système.

Notez bien qu'il s'agit d'une limite par disque, et pas du nombre total de partitions gérées par le système.

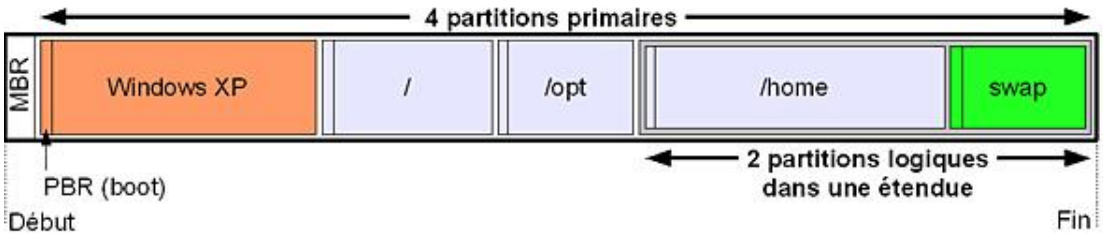
Pour dépasser la limite des 15 partitions, il est possible d'utiliser le « device mapper » de Linux, faisant appel notamment à la gestion du **LVM** (*Logical Volume Management*). Le LVM permet de regrouper plusieurs disques (Physical volumes) en une seule unité (Volume Group) vue par le système comme un énorme disque unique que vous pouvez découper en partitions (Logical Volumes), sans être limité par le nombre. Et en plus, vous pouvez rajouter des disques dans le groupe après coup, augmenter et réduire la taille des partitions à la volée, sans vous soucier de leur emplacement physique réel...

Les partitions sont numérotées de 1 à n (15 ou 63). Une partition d'une valeur supérieure ou égale à 5 indique qu'il s'agit forcément d'une partition logique. Comme il ne peut y avoir que quatre partitions primaires, la dernière (la 4) est souvent créée comme étendue :

- Partitions 1 à 3 : primaires
- Partition 4 : étendue
- Partitions 5 à n : logiques

Le numéro de la partition apparaît à la suite du nom du fichier périphérique de disque :

- hda1 : première partition primaire du premier disque IDE ;
- hdb5 : cinquième partition, première partition logique du second disque IDE ;
- sda3 : troisième partition primaire du premier disque SCSI / libata ;
- sdc8 : huitième partition, soit quatrième partition logique du troisième disque SCSI/libata.



Description schématique d'un disque

c. EBR

Chaque partition étendue devant décrire les partitions logiques qu'elle contient, elle doit aussi disposer d'une table de partition. L'**EBR** (*Extended Boot Record*) reprend la structure du MBR sauf qu'il n'y a que deux enregistrements possibles dans la table des partitions. Le premier indique effectivement la position et la taille d'une partition logique, tandis que le second est vide si c'est la seule partition logique, ou pointe sur un autre EBR. Il peut donc y avoir plusieurs EBR dans une partition étendue.

- Les EBR forment une liste chaînée, la seconde entrée de partition pointant sur l'EBR suivant.
- Il n'y a qu'une seule partition logique décrite par EBR.

d. PBR

Le **PBR** (*Partition Boot Record*), aussi appelé **VBR** (*Volume Boot Record*) ou Partition Boot Sector est le premier secteur de chaque partition primaire ou logique. Il peut contenir une routine de démarrage d'un système d'exploitation, un chargeur de démarrage, voire rien du tout si la partition n'a pas vocation à être bootée. Quand le MBR ne contient pas de routine, le bios tente de démarrer et d'exécuter la routine du PBR de la partition marquée active.

e. Types de partitions

Chaque partition dispose d'un type permettant de déterminer son contenu. C'est un identifiant numérique codé sur un octet généralement présenté en hexadécimal. Il semble important d'en fournir une liste ici pour que vous compreniez bien la finalité de cet identifiant. Les valeurs les plus communes sont en gras.

```
1  FAT12           24  NEC DOS         81  Minix / old Lin bf  Solaris
```

```
 2  XENIX root      39  Plan 9          82  Linux swap / So c1  DRDOS/
sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c4  DRDOS/
sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/
sec (FAT-
 5  Extended        41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 6  FAT16           42  SFS             86  NTFS volume set da  Non-FS
data
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set db  CP/M /
CTOS / .
 8  AIX             4e  QNX4.x 2nd part 88  Linux plein tex de  Dell
Utility
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e1  DOS
access
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          e4  Speed-
Stor
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 f  W95 Etendu (LBA 54  OnTrackDM6      a5  FreeBSD         ee  EFI GPT
10  OPUS            55  EZ-Drive        a6  OpenBSD         ef  EFI
(FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f0  Linux/
PA-RISC b
12  Compaq diagnost 5c  Priam Edisk     a8  UFS Darwin      f1  Speed-
Stor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f4  Speed-
Stor
16  Hidden FAT16    63  GNU HURD or Sys ab  Amorce Darwin   f2  DOS
secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fd  Linux
raid auto
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fe  LANstep
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid ff  BBT
1c  Hidden W95 FAT3 75  PC/IX
```

Comme le type de partition devrait refléter le système de fichiers qu'elle contient, une partition de type 0x0c devrait contenir un système de fichiers de type FAT32 LBA (gros disques). Une partition de type 0x83 devrait contenir un système de fichiers Linux. Mais lequel ? Vous avez vu qu'il en existe plusieurs...

Notez la prédominance des types de partition pour Windows. Windows se base essentiellement sur ce type pour en déterminer le contenu. Rien n'empêche de créer une partition de type Linux et d'y placer un système de fichiers FAT32. Cependant si vous faites ceci, Windows ne reconnaîtra pas la partition (considérée de type inconnu) et vous ne pourrez pas accéder au contenu.

Linux reconnaît généralement (des exceptions sont possibles) le contenu d'une partition par le système de fichiers qui y réside. Vous pouvez créer un système de fichiers ext3 dans une partition de type 0x0e et constater que tout fonctionne. Le type 0x83 peut accueillir tout système de fichiers Linux : ext2, ext3, reiserfs, jfs, xfs... Cependant pour des raisons de compatibilité veillez à respecter l'association type de partition ó système de fichiers, à rester cohérent.

3. Manipuler les partitions

a. Outils disponibles

Les outils **fdisk**, **cfdisk**, **sfdisk** ou encore **parted** permettent de manipuler les partitions, sans compter les outils graphiques disponibles durant l'installation ou dans les panneaux de configuration.

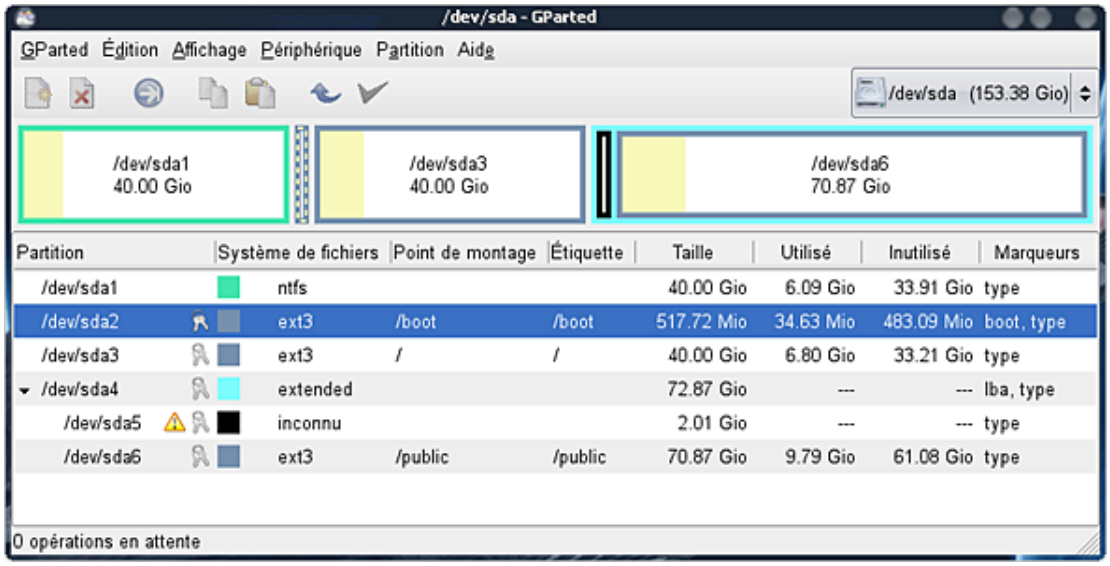
- **fdisk** est le plus ancien et le plus utilisé des outils de partitionnement. Il n'a aucun rapport avec le fdisk de Microsoft. Il est à base de menus et raccourcis textuels.

- **cfdisk** est un peu plus « visuel » et s'utilise avec les flèches directionnelles. Il permet les mêmes opérations que fdisk mais de manière plus conviviale.

- **sfdisk** fonctionne en interactif ou non, est assez compliqué mais plus précis.

- **parted** permet des opérations très avancées sur les partitions comme par exemple leur redimensionnement. Il est soit interactif (c'est un interpréteur de commandes) soit scriptable. Il existe des interfaces graphiques comme **qtparted** ou **gparted**.

Vous pouvez voir sur la capture **gparted** en action.



gparted, un éditeur de partitions graphique

b. Manipuler les partitions

Lister

C'est l'outil **fdisk**, à la fois le plus ancien et le plus standard, qui est généralement utilisé par les administrateurs et les ingénieurs système. Fdisk se lance en tant que root.

```
fdisk [-l] [disque]
```

Chaque paramètre est optionnel. Lancé tel quel fdisk se place sur le premier disque du système. Le paramètre `-l` permet de lister les partitions du disque donné, ou de tous les disques. Les informations obtenues sont les mêmes qu'en mode interactif avec l'entrée p (print) du menu.

```
# fdisk -l /dev/sda

Disque /dev/sda: 164.6 Go, 164696555520 octets
255 heads, 63 sectors/track, 20023 cylinders
Units = cylindres of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000c02ae

Périphérique Amorce    Début         Fin      Blocs    Id  Système
/dev/sda1               1        5222    41945683+   7  HPFS/NTFS
/dev/sda2   *        5223        5288      530145   83  Linux
/dev/sda3            5289       10510    41945715   83  Linux
/dev/sda4           10511       20023    76413172+   f  W95 Etendu (LBA)
/dev/sda5           10511       10772     2104483+  82  Linux swap /
Solaris
/dev/sda6           10773       20023    74308626   83  Linux
```

Les champs parlent d'eux-mêmes. Notez la partition sda4 qui est la partition étendue. Elle a comme type 0x0f, mais n'importe quel type étendu aurait fonctionné : les types 0x05 et 0x85 sont considérés comme identiques. Cependant Windows risque de ne pas reconnaître ces types et donc les partitions logiques qui y sont contenues notamment dans le cas d'un gros disque. C'est pour cela que la plupart des outils de partitionnement des distributions Linux préfèrent le type associé à Windows.

Les manipulations effectuées avec fdisk ne sont prises en compte qu'à la fin, une fois que vous sauvez vos modifications, et non au fur et à mesure. Si vous pensez vous être trompé, n'hésitez pas à quitter sans sauver ou à faire un [Ctrl] **C**. Vos modifications seront perdues, mais vous aurez sauvé vos partitions.

Liste des partitions

L'exemple suivant se base sur un disque reconnu par le système comme /dev/sdb, et ne contenant aucune partition. Le but est de créer trois partitions : une primaire, une étendue et une logique.

- Lancez **fdisk** avec le disque en argument, ne tenez pas compte des premières lignes affichées sauf si elles indiquent une erreur.

```
# fdisk /dev/sdb
...
Commande (m pour l'aide) :
```

- Vérifiez tout d'abord l'existence de partitions avec la touche p (print) puis [Entrée].

```
Commande (m pour l'aide): p

Disque /dev/sdb: 4026 Mo, 4026531840 octets
64 heads, 62 sectors/track, 1981 cylinders
Units = cylindres of 3968 * 512 = 2031616 bytes
Disk identifier: 0x0003ed63

Périphérique Amorce   Début       Fin     Blocs   Id  Système
/dev/sdb1   *          1      1981    3930273   c  W95 FAT32 (LBA)
```

Supprimer

Pour supprimer une partition, utilisez la touche d (delete) puis, si plusieurs partitions sont présentes, le numéro de partition (sdbX, X étant le numéro). Si une seule partition est présente, elle est prise par défaut.

```
Commande (m pour l'aide): d
Partition sélectionnée 1
```

Créer

Pour créer une partition, utilisez la touche n (new). Vous devez ensuite choisir le type de partition : primaire ou étendue.

```
Commande (m pour l'aide): n
Action de commande
   e   étendue
   p   partition primaire (1-4)
```

- Pour cette première partition, sélectionnez une partition primaire avec la touche p (qui veut dire primary cette fois).

- Comme le MBR contient quatre entrées vous pouvez choisir le numéro de partition à créer. Il est parfaitement possible de créer une partition sdb2 avant la sdb1. Ici, tapez **1**.

- Le premier cylindre correspond à la position de début de votre partition. Par défaut fdisk se place sur le premier cylindre disponible depuis le début du disque. Il est parfaitement possible de créer des partitions débutant au milieu d'un disque. Sélectionnez ici la valeur par défaut (1) en appuyant sur [Entrée].

- Enfin choisissez la taille de la partition. Il est préférable d'utiliser une unité lisible comme les Ko ou plutôt les Mo. Créez par exemple une partition de 1 Go, soit 1024 Mo, en saisissant **+1024M** et appuyez sur [Entrée]. La partition est maintenant définie.

```
Commande (m pour l'aide): n
Action de commande
```

```
   e   étendue
   p   partition primaire (1-4)
p
Numéro de partition (1-4): 1
Premier cylindre (1-1981, par défaut 1):
Utilisation de la valeur par défaut 1
Dernier cylindre ou +taille or +tailleM ou +tailleK (1-1981, par
défaut 1981): +1024M
```

- Vérifiez l'état de la partition (p).

```
Commande (m pour l'aide): p
...
Périphérique Amorce    Début       Fin      Blocs    Id  Système
/dev/sdb1               1       505     1001889   83  Linux
```

Sauver

Quittez **fdisk** en sauvant votre table des partitions avec la touche w (write). Fdisk écrit la nouvelle table des partitions dans le MBR et/ou les EBR. Vous risquez d'obtenir des avertissements ici indiqué en gras.

```
Commande (m pour l'aide): w
La table de partitions a été altérée!

Appel de ioctl() pour relire la table de partitions.

AVERTISSEMENT: la re-lecture de la table de partitions a échoué avec
l'erreur 16: Périphérique ou ressource occupé.
Le kernel va continuer d'utiliser l'ancienne table.
La nouvelle table sera utilisé lors du prochain réamorçage.
Synchronisation des disques.
```

Ce message signifie que, comme le périphérique disque est en cours d'accès ou d'utilisation, Linux ne voit pas encore la nouvelle table et donc les nouvelles partitions créées. Ceci peut être confirmé avec la commande suivante. Notez que la dernière ligne devrait vous montrer votre nouvelle partition, ce qui n'est pas le cas.

```
# cat /proc/partitions | grep sdb
   8    16    3932160 sdb
```

Forcer la synchronisation

Pour corriger ce dernier problème et forcer le noyau à relire la table des partitions, vous avez le choix entre deux commandes. La première est **blockdev** avec le paramètre `--rereadpt` (re-read partition table).

```
# blockdev --rereadpt /dev/sdb
```

La seconde est **partprobe**, disponible seulement si parted est installé. Elle peut réussir si blockdev a échoué. Par défaut elle relit les tables de toutes les partitions, mais vous pouvez lui spécifier le disque en argument.

```
# partprobe /dev/sdb
```

Vérifiez si la partition est bien reconnue.

```
# cat /proc/partitions | grep sdb
   8    16    3932160 sdb
   8    17    1001889 sdb1
```

Vous pouvez maintenant utiliser votre nouvelle partition et y rajouter un système de fichiers. Créez maintenant une partition étendue, puis une partition logique. Remarquez qu'une partition étendue ou logique se crée de la même manière que les autres. Une partition étendue n'est pas forcément la dernière des primaires, elle peut être la seconde par exemple. Si vous n'avez pas utilisé toute la taille pour la créer vous pouvez compléter par la suite la création de partitions primaires.

Modifier le type

La modification du type d'une partition n'entraînant pas de modification du dimensionnement de celle-ci, vous

pouvez le faire à n'importe quel moment. Voici comment procéder pour passer du type par défaut Linux au type FAT afin que Windows reconnaisse la partition.

- Lancez fdisk et appuyez sur t (type).
- Sélectionnez la partition, la 5 pour la première partition logique.

  Vous pouvez afficher tous les types en appuyant sur L, qui vous fournit la même liste que celle citée plus haut. Utilisez le type c W95 FAT32(LBA) pour être tranquille.

- Sauvez avec w.

```
Commande (m pour l'aide): t
Numéro de partition (1-5): 5
Code Hex (taper L pour lister les codes): c
Type de partition système modifié de 5 à c (W95 FAT32 (LBA))

Commande (m pour l'aide): p

Disque /dev/sdb: 4026 Mo, 4026531840 octets
64 heads, 62 sectors/track, 1981 cylinders
Units = cylindres of 3968 * 512 = 2031616 bytes
Disk identifier: 0x0003ed63

Périphérique Amorce    Début       Fin      Blocs    Id  Système
/dev/sdb1               1       505     1001889   83  Linux
/dev/sdb2             506      1981     2928384    5  Extended
/dev/sdb5             506      1981     2928353    c  W95 FAT32 (LBA)

Commande (m pour l'aide): w
```

Les types de partitions les plus utilisés pour Linux sont les suivants :

- **83** : Partition de type Linux (données)
- **82** : Partition de type swap
- **fd** : Partition de type RAID
- **8e** : Partition de type LVM

Manipuler les systèmes de fichiers

1. Définitions de base

a. Bloc

Le bloc est l'unité de base, atomique, de stockage du système de fichiers. Un fichier occupe toujours un nombre entier de blocs. Ainsi si un fichier ne contient qu'un seul octet et qu'un bloc a une taille de 4096 octets, 4095 octets sont gâchés. C'est ainsi qu'il est possible de remplir un système de fichiers avec n fichiers de 1 octets, n représentant le nombre de blocs, alors que le volume total des données n'est que de n octets !

Soit un disque contenant 102400 blocs de 4096 octets. Sa taille totale est de 400 Mo. Soient 102 400 fichiers de 384 octets. La taille totale des données est de 37,5 Mo. Or le disque est plein, tous les blocs étant utilisés ! Il y a 362,5 Mo de perdus. Il est donc très important de faire attention à la taille de blocs surtout si les fichiers à stocker sont de petite taille.

Certaines commandes calculent la taille des fichiers en blocs comme du, df ou find. Or historiquement la taille d'un bloc était de 512 octets, 1024 pour d'autres. C'est une unité dont la valeur peut changer selon la commande, et souvent sans rapport avec la taille des blocs du système de fichiers sur lequel vous travaillez. La prudence s'impose.

---

b. Superbloc

Chaque système de fichiers dispose d'au moins un superbloc. Un superbloc est une zone de méta-données qui contient plusieurs informations sur le système de fichiers :

- son type ;
- sa taille ;
- son état ;
- des informations (position) sur les autres zones de méta-données (autres superblocs, table d'inodes, etc.).

Linux tente en premier lieu de lire le superbloc primaire, le premier du disque. Il peut arriver que celui-ci soit corrompu suite à de mauvaises manipulations, un crash, une panne. Dans ce cas les données du disque ne sont plus accessibles (impossible de savoir par exemple où se trouve les inodes). Un système de fichiers Linux dispose de copies (backups) des superblocs à plusieurs endroits du disque. Les écritures sur les divers superblocs étant synchrones, ils sont tous identiques. En dernier recours si l'un deux est supprimé, il peut être recopié depuis un autre.

Vous verrez par la suite comment disposer de toutes les informations sur un système de fichiers ext2 ou ext3.

c. Table d'inodes

Un **inode** est la contraction de index node, c'est-à-dire nœud d'index. C'est une structure de données contenant les informations décrivant et représentant un fichier. Ces informations sont appelées des **attributs**. Chaque fichier dispose d'un numéro d'inode (i-number). Tous les inodes sont présents au sein d'une table d'inodes. Cette table est généralement découpée en plusieurs morceaux répartis après chaque superbloc. Une table d'inode fait partie des méta-données.

Un fichier ne peut avoir qu'un seul inode. Un inode est unique au sein d'un seul système de fichiers. Chaque système de fichiers dispose d'une table d'inodes indépendante. Si le fichier titi porte le numéro d'inode 12345 sur un premier système de fichiers, et que toto porte le numéro d'inode 12345 sur une autre, ces fichiers n'ont aucun rapport entre eux.

Cependant vous verrez plus loin que deux noms de fichiers peuvent se voir associé le même numéro d'inode au sein d'un même système de fichiers. Ces deux noms représentent alors un seul et même fichier.

Contenu

Le contenu d'un inode varie d'un système de fichiers à un autre, mais la norme POSIX impose que chacun d'eux dispose au moins des attributs suivants pour chaque fichier :

- sa taille ;
- l'identifiant du périphérique le contenant ;
- son propriétaire ;
- son groupe ;
- son numéro d'inode ;
- son mode (ses droits) d'accès ;
- sa date de dernière modification d'inode (change time) ;
- sa date de dernière modification de contenu (modification time) ;
- sa date de dernier accès (access time) ;
- un compteur de hard links (liens physiques ou durs, voir plus loin).

Un inode ne contient pas le nom du fichier.

Vous pouvez obtenir quelques informations sur un inode avec la commande **stat** :

```
# stat chapitre4.doc
  File: `chapitre4.doc'
  Size: 199168          Blocks: 408        IO Block: 4096   fichier
régulier
Device: 811h/2065d      Inode: 16629765    Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/     seb)   Gid: (  100/   users)
Access: 2008-03-17 21:39:31.000000000 +0100
Modify: 2008-03-17 21:39:30.000000000 +0100
Change: 2008-03-17 21:40:04.000000000 +0100
```

Adresses

L'inode contient aussi des champs d'adresses généralement répartis en deux types :

- des adresses pointant sur les premiers blocs de données du fichier,
- des adresses pointant sur des blocs contenant d'autres champs d'adresses,
- et dans ce dernier cas de manière récursive (adresses d'adresses pointant elles-mêmes sur d'autres adresses) formant un arbre ou chacune des feuilles (terminaisons) pointe sur un bloc de données. On parle de blocs **d'indirection** (simple, double, triple).

Voici un exemple concret de calculs d'adresses sur un inode au sein d'un système de fichiers ext2.

Un inode ext2 contient dix champs pointant sur un bloc de données chacun, et trois champs d'indirection (pointant sur des adresses).

- Le premier en simple indirection simple pointe sur 256 adresses de blocs de données.
- Le deuxième en double indirection pointe sur 256 adresses dont chacune d'elle pointe sur 256 autres adresses pointant sur des blocs de données, soit 256^2 blocs de données.
- Le troisième en triple indirection pointe sur 256 adresses, elles-mêmes pointant sur 256 adresses, elles-mêmes pointant sur 256 adresses pointant enfin sur des blocs de données, soit 256^3 blocs de données.

Soit n la taille d'un bloc en octet, la taille maximale d'un fichier est donc de **n*(10+256+256^2+256^3) octets**.

Pour un bloc de 4096 octets, cela fait environ 64 Go !

d. Tables catalogues

Un inode ne contenant pas le nom du fichier, celui-ci est placé ailleurs, dans une table de catalogue. Cette table n'est rien d'autre qu'un répertoire. Un répertoire contenant un liste de fichier, et un fichier étant représenté par un inode, chaque nom de fichier est associé au sein du répertoire à son inode.

Vous pouvez vous représenter cette table comme un tableau à deux colonnes :

Table catalogue rep1 (répertoire rep1)	
Inode	Nom
12345	Document.txt
214579	Fichier.doc
47321	Musique.mp3
98542	Copie.odt
...	...

e. Hard link

Un hard link permet d'ajouter une référence sur un inode. Le hard link rajoute une association dans une table catalogue. Les droits du fichier ne sont pas modifiés.

Un hard link ne permet pas d'affecter plusieurs noms à un même répertoire, et ne permet pas d'effectuer des liens depuis ou vers un autre système de fichiers. De plus, faites attention au compteur de lien fourni par la commande **ls -l** : un 1 indique que ce fichier ne possède pas d'autres liens, autrement dit c'est le dernier. Si vous le supprimez, il est définitivement perdu. Par contre, tant que ce compteur est supérieur à 1, si un lien est supprimé, il reste une copie du fichier quelque part.

```
$ touch fic1
$ ln fic1 fic2
$ ls -li
2394875 -rw-r--r-- 2 seb users    0 mar 21 22:40 fic1
2394875 -rw-r--r-- 2 seb users    0 mar 21 22:40 fic2
```

L'exemple précédent montre que les hard links représentent le même fichier, puisqu'il s'agit juste de noms associés au même inode. Chacun a deux liens ce qui est logique puisque les deux fichiers pointent sur le même inode. Enfin vous voyez que fic1 et fic2 ont le même inode, à savoir 2394875.

2. Créer un système de fichiers

a. mkfs, syntaxe générale

Les commandes de « formatage » telles que celles présentes sous Microsoft n'existent pas de manière identique sous Linux. Un formatage de type Microsoft est en fait la création et la vérification d'un système de fichiers sur une partition. La première étape est le remplissage des différents secteurs, blocs, et clusters de zéros (ou d'un autre motif binaire) avec une vérification du support, et la seconde l'écriture d'un système de fichiers. Cette seule dernière opération suffit à la création d'un système de fichiers vierge sur le disque ou la partition.

La commande pour créer un système de fichiers est **mkfs**. **mkfs** appelle d'autres programmes en fonction du type de système de fichiers sélectionné.

```
mkfs -t typefs options périphérique
```

C'est typefs qui détermine le type de système de fichiers et donc le programme appelé. Il existe un programme par type de système de fichiers :

- **ext2** : mkfs.ext2
- **ext3** : mkfs.ext3
- **reiserfs** : mkfs.reiserfs
- **vfat** : mkfs.vfat (pour tous les formats FAT, mais mkfs.msdos existe)
- **ntfs** : mkfs.ntfs

Notez qu'il est très important que les options de chaque système de fichiers soient indiquées APRÈS avoir précisé le système de fichiers. Si vous les précisez avant, ce sont les options de mkfs.

Plutôt que d'utiliser mkfs, vous pouvez utiliser directement les programmes correspondant au type de système de fichiers à écrire.

b. Un premier exemple en ext2

Vous allez créer un système de fichiers de type ext2 sur la première partition précédemment créée, à savoir sdb1. Voici la commande de base :

```
# mkfs -t ext2 /dev/sdb1
mke2fs 1.40.2 (12-Jul-2007)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
125440 i-noeuds, 250472 blocs
12523 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=260046848
8 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
15680 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
        32768, 98304, 163840, 229376

Écriture des tables d'i-noeuds : complété
Écriture des superblocs et de l'information de comptabilité du système
de fichiers : complété
Le système de fichiers sera automatiquement vérifié tous les 35 montages ou
après 180 jours, selon la première éventualité. Utiliser tune2fs -c ou -i
pour écraser la valeur.
```

Cette trace fournit des informations intéressantes ;

- Il est possible de donner une étiquette (un nom) au système de fichiers.
- Chaque bloc fait 4096 octets.
- Il y a 250472 blocs.
- Les i-nœuds (inodes) représentent le nombre maximal de fichiers : 125440.
- 5% de l'espace disque est réservé à root, ce qui signifie qu'un utilisateur lambda ne pourra pas remplir le disque à plus de 95%.

- Les tables d'inodes sont réparties par groupes.
- Il y a un superbloc principal et quatre superblocs de secours (un par groupe).
- Il est possible de modifier certains paramètres du système de fichiers avec la commande **tune2fs**.

c. ext2 et ext3

Les systèmes des fichiers ext2 et ext3 étant compatibles, ils partagent les mêmes paramètres, dont voici les plus courants :

Paramètre	Signification
`-b`	Taille des blocs en octet, multiple de 512. Si la taille n'est pas précisée, elle sera déterminée par la taille de la partition. Tout fichier créé sur le disque occupe au moins un bloc et donc si on manipule un grand nombre de petits fichiers il faut mettre une valeur basse (ex : 1024).
`-c`	Vérifie les mauvais blocs avant de créer le système de fichiers. On peut aussi utiliser la commande **badblocks**.
`-i`	Ratio octets/inode. La taille de la table des inodes est calculée en fonction de la taille totale du système de fichiers. Un inode occupe 128 octets. En mettre moins limite le nombre de fichiers possibles mais permet de gagner de la place. -i 4096 : un inode pour chaque 4 ko.
`-m`	Pourcentage réservé au super-utilisateur, par défaut 5%. Le mettre à zéro permet de gagner de la place et root pourra tout de même y travailler.
`-L`	Label, étiquette (nom) du système de fichiers, utile pour le montage.
`-j`	Crée un journal ext3, donc crée un système de fichiers ext3.

L'exemple suivant crée un système de fichiers journalisé ext3 (option `-j`) avec une taille de blocs de 2048 octets, et un inode pour chaque 16 Ko. La totalité du système est utilisable par les utilisateurs (aucun espace n'est réservé pour root). L'étiquette est DATA.

```
# mkfs -t ext2 -j -b 2048 -i 16384 -m 0 -L "DATA" /dev/sdb1
mke2fs 1.40.2 (12-Jul-2007)
Étiquette de système de fichiers=DATA
Type de système d'exploitation : Linux
Taille de bloc=2048 (log=1)
Taille de fragment=2048 (log=1)
62992 i-noeuds, 500944 blocs
0 blocs (0.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=513802240
31 groupes de blocs
16384 blocs par groupe, 16384 fragments par groupe
2032 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
        16384, 49152, 81920, 114688, 147456, 409600, 442368

Écriture des tables d'i-noeuds : complété
Création du journal (8192 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système
de fichiers : complété
```

Notez que la ligne de commande suivante a exactement le même effet car le système de fichiers ext3 induit le paramètre `-j` :

```
# mkfs -t ext3 -b 2048 -i 16384 -m 0 -L "DATA" /dev/sdb1
```

ext2 vers ext3

Ext3 est un système de fichiers ext2 auquel on a rajouté un journal. Vous pouvez convertir un système de fichiers ext2 en ext3 en utilisant **tune2fs**.

```
# tune2fs -j /dev/sdb1
tune2fs 1.40.2 (12-Jul-2007)
Création de l'i-noeud du journal : complété
Le système de fichiers sera automatiquement vérifié tous les 25 montages
ou après 180 jours, selon la première éventualité. Utiliser tune2fs -c
ou -i pour écraser la valeur.
```

ext3 vers ext2

Pour revenir en ext2, il faut supprimer le journal encore une fois avec tune2fs et le paramètre `-O` (grand O) :

```
# tune2fs -O ^has_journal /dev/sdb1
```

Vérifiez l'éventuelle présence d'un fichier **`.journal`** et supprimez-le. Enfin, effectuez une vérification avec fsck.

Label

Vous pouvez afficher et changer le label du système de fichiers en tapant e2label.

```
# e2label /dev/sdb1
DATA
# e2label /dev/sdb1 OLDDATA
# e2label /dev/sdb1
OLDDATA
```

Il ne faudra pas oublier de modifier les options de montage en conséquence.

Un label ne doit pas dépasser 16 caractères ou il sera tronqué.

Reiserfs

Pour créer un système de fichiers en reiserfs, c'est aussi simple qu'avec ext2 et ext3. Dans l'exemple suivant, remarquez que le programme vous demande une confirmation avant de créer le système de fichiers. Contrairement à ce qui est indiqué, il n'y a aucune raison pour rebooter. Enfin le « blabla » des premières lignes a été supprimé.

```
# mkfs -t reiserfs /dev/sdb1
mkfs.reiserfs 3.6.19 (2003 www.namesys.com)
...
Guessing about desired format.. Kernel 2.6.22.17-0.1-default is running.
Format 3.6 with standard journal
Count of blocks on the device: 250464
Number of blocks consumed by mkreiserfs formatting process: 8219
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: c1c5ac84-8ce2-4475-829d-f454b7bdb91e
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/sdb1'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/sdb1.
```

Reiserfs accepte des paramètres différents de ext2/3 :

Paramètre	Signification
`-b`	Taille des blocs en octet, multiple de 512 (puissance de 2 : 512, 1024, 2048, 4096, 8192) compris entre 512 et 8192. Si la taille n'est pas précisée, elle sera déterminée

	par la taille de la partition.
-l	Label, étiquette (nom) du système de fichiers, utile pour le montage.
-f	Force l'exécution de la commande sans poser de questions, y compris sur un disque et non une partition.
-d	Mode débug, fournit plus d'informations.

Label

Vous pouvez modifier un label **reiserfs** avec la commande **reiserfstune**.

```
reiserfstune -l HOME /dev/hda6
```

Il ne faut pas oublier de modifier les options de montage en conséquence.

d. xfs

Créez un système de fichiers xfs comme ceci.

```
# mkfs -t xfs -f /dev/sdb1
meta-data=/dev/sdb1              isize=256    agcount=8, agsize=31309
blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=250472, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal log           bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks, lazy-count=0
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

e. vfat

La création d'un système de fichiers VFAT se fait encore de la même manière. Cette fois vous allez le créer sur la partition sdb5 prévue pour ça. La commande va sélectionner automatiquement en fonction de la taille de la partition le type de FAT à créer (12, 16 ou 32). Le paramètre -v a été rajouté pour voir les traces de création.

```
# mkfs -t vfat -v /dev/sdb5
mkfs.vfat 2.11 (12 Mar 2005)
Auto-selecting FAT32 for large filesystem
/dev/sdb5 has 64 heads and 62 sectors per track,
logical sector size is 512,
using 0xf8 media descriptor, with 5856706 sectors;
file system has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 5709 sectors, and provides 730657 clusters.
Volume ID is 47df9209, no volume label.
```

Les commandes **mkfs.vfat** et **mkfs.msdos** sont des liens symboliques vers le programme mkdosfs.

Vous pouvez spécifier plusieurs paramètres, notamment si vous souhaitez forcer la création d'un type de FAT donné :

Paramètre	Signification
-c	Vérifie le périphérique avant la création.
-F	Taille de la FAT (12, 16, 32).
-I	Permet d'utiliser un disque complet et non une partition (pratique pour certains lecteurs MP3).

-n	Nom du volume (étiquette, label).
-v	Affichage des détails lors de la création.

Les mtools

Les mtools sont des outils permettant de travailler sur des systèmes de fichiers FAT et VFAT comme si vous étiez sous MSDOS ou la console Windows. Ils reprennent la syntaxe des commandes d'origine en ajoutant un m devant : mdir, mformat, mlabel, mdeltree, etc.

Les disques et partitions sont représentés par des lettres de lecteurs c :, d :, e :. Ils peuvent représenter un disque, une partition ou un répertoire. Vous devez cependant modifier un fichier de configuration /etc/mtools.conf. Par exemple pour déclarer /dev/sdb5 comme d: rajoutez ou modifiez la ligne suivante :

```
drive d: file="/dev/sdb5"
```

C'est utile pour modifier après coup certaines informations comme le nom du volume du système de fichiers vfat :

```
# mlabel -s d:
 Volume has no label
# mlabel d:
 Volume has no label
Enter the new volume label : DATAFAT
# mlabel -s d:
Volume label is DATAFAT
```

Accéder aux systèmes de fichiers

1. mount

La commande **mount** permet d'accéder aux périphériques de type blocs (les partitions) sur lesquels un système de fichiers existe. La commande **mount** attache le répertoire racine du système de fichiers à un répertoire pré-existant appelé point de montage (mountpoint).

```
mount -t typefs -o options périphérique point_de_montage
```

a. Montage par périphérique

La partition sdb1 disposant de nouveau d'un système de fichiers ext3, la commande suivante rattache la racine du système de fichiers contenu dans sdb1 au répertoire /mnt/DATA.

```
# mount -t ext3 /dev/sdb1 /mnt/DATA
```

La commande **mount** utilisée seule donne tous les détails sur les systèmes de fichiers actuellement montés (périphériques, système de fichiers, point de montage, options) :

```
# mount
/dev/sda6 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda7 on /home type ext3 (rw,acl,user_xattr)
/dev/sda1 on /windows/C type fuseblk (rw,noexec,nosuid,nodev,noatime,
allow_other,default_permissions,blksize=4096)
securityfs on /sys/kernel/security type securityfs (rw)
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

Les mêmes informations sont accessibles en affichant le contenu du fichier **/etc/mtab**.

Montage par label

On peut souligner l'intérêt pratique d'utiliser des labels pour ses systèmes de fichiers. En cas de réorganisation des disques (déplacement dans une chaîne SCSI par exemple), l'ordonnancement des périphériques est modifié. L'utilisation des noms de périphériques oblige dans ce cas à modifier le fichier **/etc/fstab** à chaque modification. Ce n'est pas le cas avec les labels. Utilisez le paramètre `-L` de mount, suivi du nom du volume.

```
# mount -t ext3 -L DATA /mnt/DATA
# mount
...
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

La liste des labels actuellement connus de Linux peut être obtenue en listant le répertoire `/dev/disk/by-label`. Notez que le label est un lien symbolique vers le fichier périphérique correspondant :

```
# ls -l /dev/disk/by-label/
total 0
lrwxrwxrwx 1 root root 10 mar 18 14:00 DATA -> ../../sdb1
lrwxrwxrwx 1 root root 10 mar 18 14:00 DATAFAT -> ../../sdb5
```

Montage par UUID

Chaque système de fichiers dispose d'un identifiant unique appelé **UUID** : *Universal Unique Identifier*, généralement un nombre aléatoire codé sur assez de bits pour que sur un ou plusieurs systèmes donnés, ils soient tous différents. Ainsi si le disque change de position logique, l'UUID ne varie pas et mount retrouve le système de fichiers, alors qu'il est théoriquement bien plus possible que deux systèmes de fichiers portent le même label.

Il existe plusieurs moyens de connaître l'UUID d'une partition. Si udev est utilisé sur votre Linux, alors la commande **vol_id** est probablement disponible. Il se peut qu'elle ne soit pas présente dans le path, par exemple sur une

openSUSE elle se situe dans **/lib/udev**.

```
# ./vol_id -u /dev/sdb1
67f6e4b8-635c-4103-9a81-877fb7db29fe
```

Si votre système de fichiers est en ext2 ou ext3 la commande **dumpe2fs** retourne énormément d'informations dont l'UUID :

```
# dumpe2fs -h /dev/sdb1 | grep UUID
dumpe2fs 1.40.2 (12-Jul-2007)
Filesystem UUID:          67f6e4b8-635c-4103-9a81-877fb7db29fe
```

Enfin tout comme pour les labels, le contenu de /dev/disk/by-uuid contient les liens symboliques des UUID pointant sur le fichier périphérique correspondant :

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 mar 18 14:00 47DF-9209 -> ../../sdb5
lrwxrwxrwx 1 root root 10 mar 18 14:00 67f6e4b8-635c-4103-9a81-
877fb7db29fe -> ../../sdb1
lrwxrwxrwx 1 root root 10 mar  3 09:23 a1cc2282-b6f4-46e1-bc94-
91585f1c5872 -> ../../sda5
lrwxrwxrwx 1 root root 10 mar  3 09:23 C2B072B5B072AF91 -> ../../sda1
lrwxrwxrwx 1 root root 10 mar  3 09:23 c56b96b5-e52f-453a-ba9d-
aa1df6f0c3c0 -> ../../sda7
lrwxrwxrwx 1 root root 10 mar  3 09:23 dd5e92d3-b931-4c18-91a3-
5edccc57ced9 -> ../../sda6
```

Notez que `vol_id` reconnaît les UUID de la plupart des systèmes de fichiers, y compris FAT et NTFS. `Vol_id` et `dumpe2fs` retournent bien plus d'informations. Par exemple `vol_id -l <périphérique>` retourne le label (volume name, étiquette).

Pour monter un système de fichiers par UUID, utilisez le paramètre `-U` de mount :

```
# mount -t ext3 -U 67f6e4b8-635c-4103-9a81-877fb7db29fe /mnt/DATA
# mount
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

Remonter un système de fichiers

Vous n'êtes pas obligé de démonter puis de remonter un système de fichiers si vous modifiez une option. Si vous modifiez une option de montage du système de fichiers (via le paramètre `-o`) vous pouvez passer l'option **remount** pour que la modification soit prise tout de suite en compte. Ne retapez pas la ligne de commande complète, mais seulement le périphérique ou le point de montage. Dans l'exemple suivant le système de fichiers est remonté en lecture seule :

```
# mount -o ro,remount /mnt/DATA
# mount
...
/dev/sdb1 on /mnt/DATA type ext3 (ro)
```

b. Options de montage

Chaque système de fichiers accepte un certain nombre d'options de montage qui peuvent être spécifiées après le paramètre `-o` de **mount**. Les options sont séparées par des virgules. Sauf indication contraire les options suivantes fonctionnent avec ext2 et ext3.

Option	Signification
defaults	Souvent présente, l'option **defaults** reprend les options rw, suid, dev, exec, auto, nouser, et async.
sync/async	Active ou désactive les écritures synchrones. Avec async les écritures passent par un tampon qui diffère les écritures (plus performant) rendant la main plus vite. Il est

	préférable d'activer les écritures synchrones sur des supports externes (clés USB, disques USB/Firewire/eSATA, etc.).
exec/noexex	Permet l'exécution/ou non des fichiers binaires sur le support.
noatime	Évite la mise à jour de l'horodatage à chaque accès à un fichier (pertinent sur les supports externes, disques SSD, pages web, newsgroups, etc.).
auto/noauto	Le système de fichiers est automatiquement monté/ne peut être monté que explicitement (voir fstab).
user/nouser	N'importe quel utilisateur peut monter le système de fichiers (implique noexec, nosuid, et nodev)/seul root a le droit de monter le système de fichiers (voir fstab).
remount	Remontage du système de fichiers pour la prise en compte de nouvelles options.
ro/rw	Montage en lecture seule ou lecture et écriture.
dev/nodev	Interpréter/Ne pas interpréter les fichiers spéciaux.
noload	Pour ext3, ne charge pas le journal.
usrquota/grpquota	Ignoré par le système de fichiers lui-même mais utilisé par le sous-système de quotas.
acl	Permet l'utilisation des Access Control Lists.
user_xattr	Pour ext3 et xfs, accepte les attributs étendus sur les fichiers, par exemple pour y coller des informations additionnelles (l'encodage du texte, etc.), des champs d'indexation (utilisés par Beagle par exemple), etc.
umask	Pour FAT/NTFS, applique un autre masque global que celui par défaut (ex 133).
dmask=/fmask=	FAT/NTFS, différencie les masques pour les répertoires et les fichiers.
uid=/gid=	FAT/NTFS, comme les droits et propriétaires ne sont pas gérés, applique un utilisateur ou un groupe par défaut sur les fichiers (ex gid=users).

c. umount

La commande **umount** détache le système de fichiers du point de montage.

```
# umount /mnt/DATA
```

Si un ou plusieurs fichiers du système de fichiers à démonter sont encore en cours d'utilisation, alors **umount** ne marchera pas. Vous devez vous assurer qu'aucun processus n'accède au système de fichiers.

```
# umount /mnt/DATA
umount: /mnt/DATA: périphérique occupé
```

La commande **lsof** vous aide à déterminer quel processus est actuellement en train d'utiliser un fichier du point de montage. Ici c'est le shell bash lancé par l'utilisateur seb qui y est présent (probablement que le répertoire courant est /mnt/DATA).

```
# lsof /mnt/DATA
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
bash    5366  seb  cwd    DIR   8,17 4096    2 /mnt/DATA
```

De manière très violente vous pouvez forcer l'arrêt des processus accédant au point de montage avec **fuser**. Il est fort probable que l'utilisateur concerné n'apprécie pas du tout (dans le cas présenté ici son shell sera arrêté et il sera déconnecté).

```
# fuser -km /mnt/DATA
```

d. /etc/fstab

Le fichier **/etc/fstab** contient une configuration statique des différents montages des systèmes de fichiers. Il est appelé à chaque démarrage du système car c'est ici qu'on indique les périphériques et leurs points de montage. Il contient six champs.

```
périphérique point_de_montage typefs options dump fsck
```

Les champs sont séparés par des espaces ou des tabulations.

Champ	Description
périphérique	Le périphérique à monter. Il peut être spécifié en tant que chemin de périphérique (/dev/hda1 par exemple), que label de système de fichiers s'il existe (LABEL=/home), ou encore en tant que UUID (UUID=xxxx).
point de montage	Le répertoire d'accès au système de fichiers monté.
typefs	Le type (ext2, etx3, reiser, vfat, etc.) du système de fichiers.
options	Les options, séparées par des virgules, vues précédemment.
dump	Fréquence de dump pour les outils de dump ou de sauvegarde.
fsck	Fréquence de vérification du système de fichiers.0=ignorer. 1=en premier, 2 en second, etc. Les systèmes ayant le même numéro sont vérifiés en parallèle.

Voici un exemple tronqué (les systèmes de fichiers virtuels n'apparaissent pas) de fichier /etc/fstab :

```
/dev/sda3  /         ext3   acl,user_xattr        1 1
/dev/sda2  /boot     ext3   acl,user_xattr        1 2
/dev/sdb1  /home     ext3   acl,user_xattr        1 2
/dev/sda6  /public   ext3   acl,user_xattr        1 2
/dev/sda1  /windows  ntfs   noauto,users,gid=users,umask=0002,utf8=true 0 0
/dev/sda5  swap      swap     defaults              0 0
```

Plutôt que de préciser des noms de périphériques statiques, il peut être préférable de préciser une étiquette (label, volume) ou un UUID.

```
LABEL=BOOT   /boot             ext3       acl,user_xattr        1 2
UUID=f0bed37c-9ddc-4764-ae7f-133205c36b5d ext3    acl,user_xattr     1 2
```

Enfin, si vous n'appréciez aucune de ces solutions mais que vous souhaitez utiliser des chemins, vous pouvez utiliser les liens présents dans /dev/disk/by-XXX où xxx représente :

- id : les identifiants de contrôleur, matériel et partition des différents volumes ;
- label : les identifiants par label ;
- uuid : les identifiants par uuid ;
- path : les identifiants par chemin matériel (bus, lun, etc.).

Voici un exemple :

```
./by-id:
lrwxrwxrwx 1 root root  9 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-> ../../sda
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-ST3160811AS_6PT1LX2M-part1 ->
```

```
../../sdb1
...
./by-label:
lrwxrwxrwx 1 root root 10 mar 20 13:44 EXTERNE -> ../../sdc1
lrwxrwxrwx 1 root root 10 mar 20 08:16 ROOT -> ../../sda3
lrwxrwxrwx 1 root root 10 mar 20 08:16 BOOT -> ../../sda2
...
./by-path:
lrwxrwxrwx 1 root root  9 mar 20 13:44 pci-0000:00:1a.7-usb-0:2:1.0-scsi-
0:0:0:0 -> ../../sdc
lrwxrwxrwx 1 root root 10 mar 20 13:44 pci-0000:00:1a.7-usb-0:2:1.0-scsi-
0:0:0:0-part1 -> ../../sdc1
lrwxrwxrwx 1 root root  9 mar 20 08:16 pci-0000:00:1f.2-scsi-0:0:0:0 ->
../../sda
...
./by-uuid:
lrwxrwxrwx 1 root root 10 mar 20 08:16 02FCDA46FCDA339F -> ../../sda1
lrwxrwxrwx 1 root root 10 mar 20 13:44 470E-63A6 -> ../../sdc1
lrwxrwxrwx 1 root root 10 mar 20 08:16 527585d3-1e52-4aba-b7fc-70f183884
58d -> ../../sda6
...
```

Montage au boot

Lors de la séquence de démarrage le fichier **/etc/fstab** est balayé par l'un des scripts, presque au tout début du boot, entre le chargement du noyau et le démarrage des services. Tous les systèmes de fichiers ne possédant pas noauto comme option sont automatiquement montés (le auto est implicite). Le premier à l'être est le système de fichiers racine /. Puis viennent ensuite le swap et les autres systèmes de fichiers s'ils sont spécifiés (ex : /home, /usr, etc.) ainsi que les systèmes de fichiers virtuels /proc, /sys , /dev/pts, etc.

Montage manuel

Le contenu de /etc/fstab peut être utilisé après l'initialisation du système pour monter et démonter ponctuellement les systèmes de fichiers qui n'ont pas par exemple l'option **noauto**, ou les supports de masse comme les lecteurs CD/DVD. Dans ce cas vous utilisez simplement les labels, les points de montage ou le périphérique sans avoir à réécrire toute la ligne de commande. Mount va chercher ses renseignements dans /etc/fstab.

```
mount /home
mount -L /u01
mount LABEL=/boot
mount /dev/hda5
```

Tout monter

Si vous avez effectué des modifications importantes dans la fstab comme le rajout de plusieurs nouveaux points de montage, vous pouvez, au lieu de monter chaque système de fichiers un par un, tous les monter d'un coup avec le paramètre `-a` de **mount** : `# mount -a`

bind

L'option particulière bind peut être très pratique pour faire apparaître une partie de système de fichiers sur plusieurs points de montages. Elle permet d'éviter l'utilisation des liens symboliques et de ses défauts (modification du fichier pointé et nom du lien lui-même). Dans l'exemple suivant le système de fichiers ayant le label **u01** est rattaché au répertoire **/u01**. Puis on rattache **/u01/applis** au répertoire **/applis**.

```
LABEL=/u01     /u01            ext3    defaults 1 2
/u01/applis    /applis         none    bind
```

e. Cas des CD et images ISO

Les CD-Rom, DVD-Roms et autres supports de ce type se montent comme n'importe quel disque. Les CD-Roms et certains DVD-Roms utilisent le système de fichiers **iso9660**.

```
# mount -t iso9660 /dev/sr0 /media/cdrom
```

La plupart des DVD-Roms utilisent plutôt le format **UDF** (*Universal Disk Format*).

```
# mount -t udf /dev/sr1 /media/dvd
```

Les distributions Linux récentes s'affranchissent du montage manuel des supports externes qu'il s'agisse d'un CD, DVD, clé USB ou disque externe. Les services udev se chargent au branchement ou à l'insertion du support de créer les fichiers spéciaux associés, et de monter et de démonter les supports automatiquement.

Une image ISO est une image du contenu d'un CD ou d'un DVD. C'est un système de fichiers iso9660 ou udf dans un fichier. Il est possible d'utiliser cette image comme un périphérique, à l'aide des périphériques de loopback. L'image est rattachée à un périphérique de loopback, et les outils passent par ce périphérique comme s'il s'agissait d'un disque.

```
# mount -o loop -t iso9660 image.iso /mnt/iso
```

Contrôler le système de fichiers

1. Statistiques d'occupation

a. Par système de fichiers

La commande **df** permet d'obtenir des statistiques d'occupation de chaque système de fichiers monté. Sans argument, df fournit des informations sur tous les systèmes de fichiers. Vous pouvez passer comme argument un périphérique monté ou un point de montage. Si vous passez un répertoire quelconque, df donne des informations sur le système de fichiers qui contient ce répertoire.

```
# df
Sys. de fich.       1K-blocs     Occupé Disponible Capacité Monté sur
/dev/sda3            41286828   6482952  32706592  17% /
udev                  1031240       124   1031116   1% /dev
/dev/sda2              521780     27092    468184   6% /boot
/dev/sdb1           153834852  49189572  96830864  34% /home
/dev/sda6            73142560  19150372  50276760  28% /public
/dev/sdc1           292890560 175894672 116995888  61% /media/EXTERNE
```

Le résultat est explicite. L'unité par défaut est le kilo-octet (identique au paramètre `-k`) bien que la norme POSIX définisse une unité de bloc à 512 octets. Vous pouvez modifier les paramètres pour demander le résultat en Mo (`-m`).

```
# df -m /home
Sys. de fich.        1M-blocs       Occupé Disponible Capacité Monté sur
/dev/sdb1               150230     48043     94557  34% /home
```

Pour que ce soit plus lisible, rajoutez le paramètre `-h` (*Human readable*).

```
# df -h /home
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/sdb1             147G   47G   93G  34% /home
```

Ne confondez pas ce dernier paramètre avec `-H` qui affiche le résultat en unités **SI** (*Système International*).

```
# df -H /home
Sys. de fich.          Tail.  Occ.  Disp. %Occ. Monté sur
/dev/sdb1              158G    51G   100G  34% /home
```

Les unités SI qui définissent les unités de poids et mesures sont basées sur des puissances de 10. Tout le monde est d'accord pour dire que 1 Kg est égal à 10^3 grammes soit 1000 grammes. Donc 1 Ko vaut 10^3 octets, soit 1000 octets... Vous n'êtes pas d'accord ? C'est qu'un ordinateur ne travaille pas avec des puissances de 10 mais des puissances de 2. Matériellement 1 Ko valait jusqu'à présent 2^{10} octets soit 1024 octets, 1 Mo valait 2^{20} octets et ainsi de suite. Cette méthode est appelée méthode traditionnelle. Le Système International préfère utiliser les termes kibioctet (kilo Binaire, Kio), mébioctets (Moi) et gibioctet (Gio) pour les représentations binaires, et Ko, Mo et Go pour les puissances de 10 comme pour les mètres et les grammes. Vous comprenez maintenant pourquoi un disque de 160 Go ne fait en fait que 152,5 Gio. Nous nous faisons tous rouler de manière légale et officielle.

Le `-T` rajoute l'affichage du type de système de fichiers.

```
# df -T /home
Sys. de fich. Type    1K-blocs     Occupé Disponible Capacité Monté sur
/dev/sdb1     ext3   153834852  49197688  96822748  34% /home
```

La commande **df** permet aussi de fournir des statistiques pour l'utilisation des inodes. Vous pouvez cumuler le paramètre `-i` avec le paramètre `-h`.

```
# df -i /home
Sys. de fich.         Inodes   IUtil.  ILib. %IUti. Monté sur
/dev/sdb1            19546112   86016 19460096    1% /home
```

```
# df -ih /home
Sys. de fich.         Inodes   IUtil.  ILib. %IUti. Monté sur
/dev/sdb1                19M     84K     19M    1% /home
```

b. Par arborescence

La commande **du** (*disk usage*) fournit des informations sur l'espace occupé par une arborescence (un répertoire et tout son contenu). Si rien n'est précisé, c'est le répertoire courant qui est utilisé. Les paramètres `-k` (Ko) et `-m` (Mo) déterminent l'unité. La taille est fournie pour chaque élément (voire arrondie). La taille totale de l'arborescence est sur la dernière ligne.

```
# du -m LIVRE_ALGO
1       LIVRE_ALGO/BACKUP/chapitre7/code_java
2       LIVRE_ALGO/BACKUP/chapitre7
1       LIVRE_ALGO/BACKUP/Introduction
1       LIVRE_ALGO/BACKUP/chapitre4/illustrations
…
42      LIVRE_ALGO/
```

Pour n'avoir que le total et pas tous les détails, utilisez le `-s`.

```
# du -ks LIVRE_ALGO
42696   LIVRE_ALGO/
```

Notez que **du** n'est pas limitée à un seul système de fichiers et continue à calculer si elle tombe sur un point de montage dans l'arborescence qu'elle analyse. Si vous voulez limiter le calcul au système de fichiers courant sans rentrer dans les points de montage présents dans l'arborescence, précisez `-x`.

```
# du -msx /
1064    /
```

2. Vérifier, régler et réparer

a. fsck

La commande **fsck** permet de vérifier et de réparer un système de fichiers.

```
fsck -t typefs périphérique
```

Le système de fichiers à vérifier ou réparer ne devrait pas être monté, ou alors seulement en lecture seule.

Tout comme mkfs, fsck appelle une autre commande selon le type de système de fichiers à vérifier : fsck.ext2, fsck.ext3, etc. Chacune peut prendre des options particulières. Si **fsck** ne reconnaît pas l'option qui lui est fournie, il la transmet au programme concerné. Si vous n'indiquez pas de type, fsck tente de le déterminer seul.

Pour cet exemple, le paramètre `-f` est passé à **fsck** pour forcer la vérification (il n'a pas été possible de produire une corruption) ainsi que le paramètre `-V` pour fournir tous les détails.

```
# fsck -fV /dev/sda2
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
Passe 1 : vérification des i-noeuds, des blocs et des tailles
Passe 2 : vérification de la structure des répertoires
Passe 3 : vérification de la connectivité des répertoires
Passe 4 : vérification des compteurs de référence
Passe 5 : vérification de l'information du sommaire de groupe

      42 inodes used (0.06%)
       1 non-contiguous inode (2.4%)
         # of inodes with ind/dind/tind blocks: 10/1/0
    8864 blocks used (6.69%)
       0 bad blocks
       1 large file
```

```
      27 regular files
       3 directories
       0 character device files
       0 block device files
       0 fifos
       0 links
       3 symbolic links (3 fast symbolic links)
       0 sockets
--------
      33 files
```

Lorsque le système de fichiers est endommagé, fsck vous pose des questions à chaque action nécessaire. Vous pouvez passer le paramètre `-p` pour tenter une réparation automatique, ou encore `-y` pour forcer les réponses à oui.

Lors du démarrage du système, celui-ci vérifie depuis combien de temps, ou au bout de combien de montage, le système de fichiers n'a pas été vérifié. Si l'intervalle de temps est trop important alors il va exécuter un fsck sur le système de fichiers concerné. Les intervalles peuvent être modifiés par la commande **tune2fs**.

b. badblocks

La commande **badblocks** tente de vérifier les blocs défectueux sur le périphérique de stockage fourni en argument. Elle peut être appelée par mkfs ou fsck si le paramètre `-c` (check) leur est fourni.

Par défaut badblocks lit l'intégralité des blocs du support et retourne un erreur si un ou plusieurs d'entre eux sont illisibles. La commande peut être lancée même si le système de fichiers est monté, sauf si vous tentez un test en lecture et écriture, même non destructif.

```
# badblocks -v /dev/sda2
Vérification des blocs 0 à 530144
Vérification des blocs défectueux (test en mode lecture seule) : done
Passe complétée, 0 blocs défectueux repérés.
```

Les paramètres `-n` (non destructif) et `-w` (write, avec motifs, destructif) tentent des écritures sur les blocs. Le premier lit et réécrit le bloc à l'identique, le second écrit plusieurs motifs (0xaa, 0x55, 0xff, 0x00) donc écrase l'ancien contenu.

L'exécution de badblocks peut être très longue, plusieurs heures sur quelques centaines de Go.

c. dumpe2fs

La commande **dumpe2fs** accepte en argument un périphérique contenant un système de fichiers ext2 ou ext3. Elle retourne un grand nombre d'informations sur le système de fichiers.

```
# dumpe2fs /dev/sda2
dumpe2fs 1.40.2 (12-Jul-2007)
Filesystem volume name:   /boot
Last mounted on:          <not available>
Filesystem UUID:          abc32a5a-a128-4492-8e03-248521015835
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal resize_inode dir_index filetype
needs_recovery sparse_super large_file
Filesystem flags:         signed directory hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              66400
Block count:              132536
Reserved block count:     6626
Free blocks:              123672
Free inodes:              66358
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      32
Blocks per group:         32768
```

```
Fragments per group:      32768
Inodes per group:         13280
Inode blocks per group:   415
Filesystem created:       Sat Feb 23 22:52:05 2008
Last mount time:          Thu Mar 20 19:13:51 2008
Last write time:          Thu Mar 20 19:13:51 2008
Mount count:              1
Maximum mount count:      500
Last checked:             Thu Mar 20 19:07:48 2008
Check interval:           5184000 (2 months)
Next check after:         Mon May 19 20:07:48 2008
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal inode:            8
Default directory hash:   tea
Directory Hash Seed:      f1584155-5760-4445-8009-0444ffa81f91
Journal backup:           inode blocks
Taille du journal:        16M

Groupe 0 : (Blocs 0-32767)
  superbloc Primaire à 0, Descripteurs de groupes à 1-1
  Blocs réservés GDT à 2-33
  Bitmap de blocs à 34 (+34), Bitmap d'i-noeuds à 35 (+35)
  Table d'i-noeuds à 36-450 (+36)
  25696 blocs libres, 13256 i-noeuds libres, 2 répertoires
  Blocs libres : 4559-5660, 6767-12287, 12289-14335, 15644-26721,
26820-32767
  I-    noeuds libres : 14, 16-20, 23-24, 33-13280
...
```

Cette sortie est très longue (elle a pourtant été tronquée), mais vous donne tous les détails possibles sur le système de fichiers. Vous pouvez isoler uniquement l'en-tête (jusqu'à la taille du journal) avec le paramètre `-h`. Le mieux, si vous cherchez une information précise, est d'utiliser la commande **grep**.

```
# dumpe2fs -h /dev/sda2|grep -i "block size"
dumpe2fs 1.40.2 (12-Jul-2007)
Block size:               4096
```

d. tune2fs

La commande **tune2fs** permet de modifier certains paramètres d'un système de fichiers ext2 ou ext3. Vous avez déjà rencontré cette commande lorsqu'il a fallu convertir l'ext2 vers l'ext3, et vice versa. Voici quelques paramètres supportés par la commande :

Paramètre	Signification
`-c n`	Nombre de fois où le système de fichiers doit être monté avant d'être automatiquement vérifié. Par exemple si n vaut 30, au trentième montage fsck sera automatiquement lancé. Si n vaut 0 ou -1, la vérification est désactivée.
`-i n`	Intervalle de temps entre deux vérifications. L'unité par défaut est le jour. Les suffixes d (jours), w (semaines) ou m (mois) peuvent être accolés au nombre. `-i` 180d signifie que le système de fichiers sera contrôlé au bout de 180 jours.
`-j`	Rajoute un journal sur un système de fichiers ext2. Il est préférable de le faire système démonté. Dans le cas contraire un fichier caché `.journal` est rajouté à la racine du système, immuable (sauf si vous détruisez le journal), qui sera intégré au sein du système à la prochaine exécution de fsck.
`-L`	Modifie l'étiquette (label, nom de volume). L'étiquette ne doit pas dépasser 16 caractères ou elle sera tronquée.
`-e err`	Indique comment doit réagir le noyau si une erreur est détectée sur le système de fichiers lors du boot. La valeur par défaut est « continue ». Les autres valeurs

	possibles sont « panic » (blocage du noyau en mode kernel panic) et « remount-ro » : passage en lecture seule.
`-m n`	Le pourcentage n représente la taille réservée que s'octroient les processus lancés par root (et root lui-même) sur la taille totale. Sur un système de fichiers réservés aux utilisateurs, mettre 0 permet de remplir le système de fichiers jusqu'à 100%. Mais il est important sur la racine, ou /var, de réserver une zone pour que certains services comme syslogd puissent continuer à écrire les traces systèmes. Par défaut 5% d'espace est réservé.
`-o [^]option`	Ajoute ou supprime (avec ^) l'option indiquée par défaut au montage. Les options peuvent être par exemple acl ou user_xattr.
`-O [^]fonction`	Ajoute ou supprime (avec ^) la fonction indiquée. La fonction la plus connue est « has_journal ». `-O has_journal` équivaut à `-j`. `-O ^has_journal` convertit ext3 en ext2.
`-U UUID`	Modifie la valeur de l'UUID à votre convenance (format hexadécimal). Il est aussi possible de le supprimer (clear), d'en générer un aléatoirement (random) ou d'en générer un en fonction de la date (time).
`-s 0/1`	Active ou désactive la « sparse super feature ». Sur des disques de grande taille, l'activation réduit le nombre de blocs de secours pour gagner de la place. Vous devez ensuite exécuter fsck.

```
# tune2fs -m 0 -s 1 -U random -e remount-ro -c 60 -I 180 /dev/sdb1
```

Notez que `-c` et `-i` vont de pair. Le premier terme échu est le premier qui effectue la vérification par fsck, puis les dates et compteurs sont remis à zéro. Les compteurs sont vérifiés lors du montage au démarrage (boot) du système. Si cela fait 300 jours que le système n'a pas rebooté et que le système de fichiers n'a pas été vérifié durant cet intervalle, il ne sera pas vérifié automatiquement durant ces 300 jours mais au prochain reboot. Certains serveurs ne rebootant que rarement (par exemple tous les deux ans) le système de fichiers n'est pas automatiquement vérifié durant tout ce temps...

Le swap

1. Pourquoi créer un swap ?

Dans un environnement 32 bits un processus peut théoriquement accéder à 4 Go d'espace mémoire. Il dispose de 4 Go de mémoire virtuelle, rien qu'à lui et à laquelle aucun autre processeur ne peut accéder. Dans la pratique il y a plusieurs freins à cette possibilité :

- L'espace mémoire adressable d'un processus est partagé entre zone de code et zone de données dont la taille peut varier selon le noyau utilisé.
- Les ordinateurs ne disposent pas tous de 4 Go de mémoire (bien qu'il soit courant de trouver des serveurs Linux disposant de 16, 32 ou même 64 Go de mémoire).
- Tous les processus doivent se partager la mémoire de l'ordinateur.

Que se passe-t-il si un processus n'a plus assez de mémoire pour traiter ses données ? Le système d'exploitation va décharger des segments de la mémoire physique dans une zone d'échange sur disque qui fera office de mémoire virtuelle tampon. Il y a donc un échange entre la mémoire physique et cette zone d'échange, appelé l'espace de swap. Ce processus permet d'utiliser plus de mémoire que l'ordinateur n'en dispose réellement, au prix d'un net ralentissement si le programme est très gourmand.

2. Taille optimale

Il n'y a pas de règles strictes sur la taille du swap. Cependant les quelques règles courantes suivantes sont valables dans la plupart des cas :

- Moins de 512 Mo de RAM : deux fois la RAM.
- 1 Go à 4 Go : la taille de la RAM.
- Plus de 4 Go : 4 Go, plus ou moins, selon l'utilisation des processus.

3. Créer une partition de swap

- Vous savez déjà créer une partition. Créez une partition avec fdisk de la taille souhaitée pour le swap, et donnez-lui le type **83**.
- Synchronisez la table des partitions avec **partprobe**.
- Utilisez la commande **mkswap** pour préparer la partition à recevoir du swap.

```
# mkswap /dev/sda5
Initialisation de la version de l'espace de swap 1, taille =
2154983 kB
pas d'étiquette, UUID=c84714e6-c42c-44d4-9fe7-10dc6afac644
```

- Votre swap est prêt.

Il est possible d'attribuer une étiquette à la partition de swap avec le paramètre `-L`.

Si vous créez plus de 1 ou 2 Go de swap, vous devriez penser à créer plusieurs partitions de swap sur des disques différents situés sur des contrôleurs matériels différents. Linux répartira la charge sur chacune de ces partitions, ce qui garantira des accès plus rapides.

4. Activer et désactiver le swap

a. Activation dynamique

Linux permet d'activer et de désactiver le swap, ou des morceaux de swap, directement sans avoir à redémarrer le système.

La commande **swapon** permet d'activer une partition de swap :

```
# swapon /dev/sda5
```

Le paramètre `-p` permet de modifier la priorité du swap. Plus la valeur, comprise entre 0 et 32767, est élevée, plus la priorité d'une zone de swap est élevée. Le système l'utilisera en priorité. Ce paramètre est utile si plusieurs partitions de swap existent sur des disques différents. Dans ce cas, privilégiez soit le disque le plus rapide, soit indiquez une priorité égale pour une meilleure répartition.

Comme avec mount, le paramètre `-L` permet d'activer une zone de swap grâce à son étiquette.

La commande **swapoff** désactive une zone de swap. Veillez à disposer de l'espace mémoire libre nécessaire, sinon la commande ne fonctionnera pas.

Le contenu de **/proc/swaps** reflète l'état actuel des zones de swap actives.

```
# cat /proc/swaps
Filename                        Type            Size    Used    Priority
/dev/sda5                       partition       1461872 2012    -1
```

b. Dans /etc/fstab

Les zones de swap se placent dans le fichier **/etc/fstab**. Voici un exemple :

```
/dev/sda5 swap swap defaults 0 0
```

Les options **noauto** et **pri=X** peuvent être précisées. L'option **pri** permet de définir la priorité de la zone de swap.

Lors du démarrage, le système exécute **swapon -a** qui active toutes les partitions de swap présentes dans la fstab sauf si noauto est précisé.

Lors de l'arrêt, le système exécute **swapoff -a** qui désactive complètement le swap.

5. En cas d'urgence : fichier de swap

Que faire si vous manquez d'espace de swap et qu'il n'est plus possible de créer une nouvelle partition ? Linux sait utiliser un fichier d'échange (comme Windows, voire celui de Windows). S'il reste de la place sur un de vos systèmes de fichiers vous pouvez créer dessus un fichier d'échange d'une taille prédéfinie. Ce swap sera moins performant qu'une partition de swap (problème de fragmentation, temps d'accès au système de fichiers).

Voici les manipulations pour un petit swap de 32 Mo :

```
# free | grep Swap
Swap:      2104472       4344    2100128
# dd if=/dev/zero of=/swap bs=1024 count=32768
32768+0 enregistrements lus
32768+0 enregistrements écrits
33554432 bytes (34 MB) copied, 0,35697 s, 94,0 MB/s
slyserver:~ # mkswap /swap
Initialisation de la version de l'espace de swap 1, taille = 33550 kB
pas d'étiquette, UUID=b2e5e99e-09a1-4b2d-ac76-59f76526453a
slyserver:~ # chmod 600 /swap
slyserver:~ # sync
slyserver:~ # swapon -v /swap
swapon sur /swap
slyserver:~ # free | grep Swap
Swap:      2137232       4308    2132924
```

Modifiez éventuellement le fichier **/etc/fstab**, en espérant que le swap soit activé après le montage des systèmes de fichiers. Le swap est activé au boot, généralement après le montage de /. Mais s'il est ailleurs (autre point de montage) alors comme le swap est activé avant les autres points de montage, il en résultera une erreur. Il est donc préférable de créer le fichier dans le système de fichiers racine /.

```
/swap swap swap defaults 0 0
```

6. État de la mémoire

a. free

La commande **free** vous fournit des informations sur la mémoire physique (RAM) de votre ordinateur ainsi que sur l'occupation du swap. L'unité par défaut est le Ko (identique avec le paramètre `-k`), mais elle peut être modifiée en Mo (`-m`) voire en Go (`-g`).

```
# free -k
             total       used       free     shared   buffers    cached
Mem:       2062484    2045732      16752          0    707512    776528
-/+ buffers/cache:     561692    1500792
Swap:      2104472        132    2104340
```

Attention cependant à bien interpréter les colonnes. Dans le résultat précédent disposant de 2 Go de mémoire le système indique que seulement 16 Mo sont libres ! C'est que le système Linux a tendance à se réserver tout l'espace disponible sous forme de buffers (tampons) et de cache. Le contenu du cache est volatile, donc Linux peut libérer en très grande partie cet espace pour l'allouer aux programmes et aux données. C'est aussi le cas pour les buffers. Lorsque la commande précédente a été saisie, l'ordinateur était en train de copier une piste de DVD sur le disque dur. Quelques instants après la fin de la copie, le résultat était le suivant :

```
             total       used       free     shared   buffers    cached
Mem:       2062484    1586772     475712          0      9708    996000
-/+ buffers/cache:     581064    1481420
Swap:      2104472         44    2104428
```

Les buffers ont été libérés, une partie est repartie en cache lors du début de l'encodage de la piste en DivX. Donc lors de vos calculs de mémoire libre, prenez en compte le fait que les caches peuvent être libérés. Le système dispose d'environ 1,4 Go d'espace mémoire (en additionnant Free + cached) que Linux peut attribuer aux programmes.

b. Mémoire réservée

2 Go correspondent à 2097152 Ko. Or le total présente une différence d'environ 34 Mo. Cette mémoire est réservée au noyau et ne peut pas être utilisée par d'autres programmes. Elle est utile pour les traitements du noyau, son chargement, l'initrd. Voyez le résultat de la commande suivante (volontairement tronqué) :

```
# dmesg | grep -i memory
...Memory: 2057756k/2096640k available (2053k kernel code, 38496k
reserved, 1017k data, 316k init)
Freeing initrd memory: 4411k freed
Freeing unused kernel memory: 316k freed
...
```

Le système se réserve environ 38 Mo de mémoire, puis libère la mémoire dont il n'a plus besoin, pour obtenir le résultat attendu.

c. meminfo

Le système de fichiers virtuel **/proc** contient des informations détaillées sur la mémoire au travers du pseudo-fichier **/proc/meminfo**. Il semble bien difficile de trouver quelque chose de plus complet. La sortie suivante est un résultat sur un système Linux en 64 bits. En 32 bits, deux lignes supplémentaires (highmem et lowmem) indiquent les zones réservées aux données et au noyau. Les premières lignes et les lignes concernant le swap sont identiques au résultat de la commande **free**.

```
# cat /proc/meminfo
MemTotal:      2062484 kB
```

```
MemFree:         16452 kB
Buffers:          4152 kB
Cached:        1456476 kB
SwapCached:          0 kB
Active:        1307912 kB
Inactive:       614356 kB
SwapTotal:     2104472 kB
SwapFree:      2104428 kB
Dirty:          188324 kB
Writeback:           0 kB
AnonPages:      461688 kB
Mapped:         120268 kB
Slab:            62476 kB
SReclaimable:    38384 kB
SUnreclaim:      24092 kB
PageTables:      15344 kB
NFS_Unstable:        0 kB
Bounce:              0 kB
CommitLimit:   3135712 kB
Committed_AS:   845496 kB
VmallocTotal: 34359738367 kB
VmallocUsed:     56016 kB
VmallocChunk: 34359678971 kB
HugePages_Total:     0
HugePages_Free:      0
HugePages_Rsvd:      0
Hugepagesize:     2048 kB
```

Les quotas disques

1. Définitions

Les **quotas** permettent de poser des limites à l'utilisation de systèmes de fichiers. Ces limites sont de deux types :

- **inodes** : limite le nombre de fichiers.
- **blocs** : limite la taille disque.

Les quotas sont implémentés par système de fichiers individuel et pas pour l'ensemble des systèmes de fichiers. Chaque utilisateur peut être géré de manière totalement indépendante. Il en est de même pour les groupes. Pour chaque utilisation (inode ou bloc), vous pouvez mettre en place deux limites dans le temps :

- **Limite dure** (hard) : quantité maximale d'inodes ou de blocs utilisés que l'utilisateur ou le groupe ne peuvent absolument pas dépasser. Dans ce cas, plus rien ne sera possible (création de fichier ou fichier dont la taille dépasse la limite).
- **Limite douce** (soft) : quantité maximale d'inodes ou de blocs utilisés que l'utilisateur ou le groupe peuvent temporairement dépasser. Dans ce cas, les créations et modifications seront possibles jusqu'à un certain point : limite dure et délai de grâce.
- **Un délai de grâce** est mis en place. Durant ce temps, l'utilisateur peut continuer à travailler sur le système de fichiers. Le but est qu'il revienne à terme sous la limite douce. Le délai dépassé, la limite douce devient la limite dure. Quoi qu'il arrive, l'utilisateur ne pourra jamais dépasser la limite dure.

Les quotas sont implémentés dans le noyau Linux et au sein des systèmes de fichiers. Pour les utiliser, les outils de quotas (packages quota) doivent être installés. Les manipulations suivantes sont effectuées sur un système de fichiers ext3.

2. Mise en place

Vous allez mettre en place les quotas sur la partition /home en respectant les étapes suivantes :

- Modifiez les options de partition dans **/etc/fstab**. On rajoute dans les options usrquota (utilisateur) ou grpquota (groupe), ou les deux.

```
LABEL=/home /home ext3 defaults,usrquota 1 2
```

- Remontez le système de fichiers.

```
# mount -o remount /home
```

- Créez les fichiers contenant les informations de quota (base de données de quotas).

```
# cd /home
# touch aquota.user aquota.group
```

- Mettez à jour la base de données avec la commande **quotacheck**.

```
# quotacheck -c /home
```

- Démarrez (ou arrêter) les quotas. Cette opération n'est pas nécessaire après un redémarrage de Linux car la mise en place des quotas est comprise dans les scripts de démarrage. La commande **quotaon** démarre les quotas pour le système de fichiers indiqué (-a pour tous). La commande **quotaoff** stoppe les quotas.

```
quotaon /home
```

- Éditez les quotas pour les utilisateurs ou les groupes. La commande **edquota** est utilisée. En pratique, si tous les utilisateurs doivent avoir les mêmes quotas ou avec quelques variantes, on crée un utilisateur lambda dont on recopiera les propriétés.

Établir les quotas pour roger :

```
# edquota roger # = edquota -u roger
```

Les quotas de arthur sont identiques à ceux de roger :

```
# edquota -p roger arthur
```

- Établissez le délai de grâce. Le délai accepte les unités « seconds », « minutes », « hours », « days », « weeks », « monthes ».

```
# edquota -t
```

- Vérifiez les quotas. Les utilisateurs peuvent vérifier l'état de leurs quotas avec la commande **quota**. L'administrateur peut générer un rapport avec **repquota**. Enfin la commande **warnquota** qui peut être exécutée via cron peut envoyer un mail aux utilisateurs pour les prévenir en cas de dépassement.

L'édition des quotas se fait avec l'éditeur par défaut du système qui est généralement vi (le comportement peut être modifié via les variables EDITOR et VISUAL). Les blocs de quotas sont des blocs de 1 Ko.

```
# edquota roger
Disk quotas for user roger (uid 502):
Filesystem  blocks    softhard       inodes    softhard
/dev/hda5   1695256   25000003000000    12345     00
```

Avec l'éditeur, on peut modifier les valeurs soft et hard qui correspondent aux limites douces et dures pour le nombre de blocs et le nombre d'inodes. Ci-dessus, il a été établi une limite douce à environ 2,4 Go (2500000 ko) et dure à environ 2,9 Go (3000000 ko) d'occupation du système de fichiers pour roger. Il n'y a pas de quotas sur le nombre d'inodes (valeur à 0).

Le contenu des champs blocks et inodes est dynamique et ne doit pas être touché, ce qui n'a de toute façon aucun effet.

```
# edquota -t
Filesystem            Block grace period     Inode grace period
/dev/hda3                     7days                  7days

# repquota /home
*** Report for user quotas on device /dev/hda5
Block grace time: 7 days; Inode grace time: 7 days
                        Block limits                       File limits
User           Used     soft      hard    grace    used    soft    hard
----------------------------------------------------------------------
root     --   12345        0         0               5       0       0
roger    -- 1695256  2500000   3000000            12345       0       0
```

Une dernière nécessité est d'utiliser régulièrement la commande **quotacheck** pour maintenir la cohérence des informations de quotas des systèmes de fichiers. En effet, en cas arrêt des quotas ou de problème (arrêt inopiné par exemple), il peut parfois être nécessaire de vérifier et de réactualiser les informations.

```
# quotacheck -avug
```

Les droits d'accès

1. Les droits de base

a. Droits et utilisateurs

Le rôle d'un système d'exploitation est aussi d'assurer la sécurité et l'accès aux données, ce qui est possible grâce au mécanisme des droits. Chaque fichier ou répertoire se voit attribuer des droits qui lui sont propres, des autorisations d'accès individuelles. Lors d'un accès le système vérifie si celui-ci est permis.

À sa création par l'administrateur, un utilisateur se voit affecter un **UID** (*User Identification*) unique. Les utilisateurs sont définis dans le fichier **/etc/passwd**. De même chaque utilisateur est rattaché à un groupe au moins (groupe principal), chaque groupe possédant un identifiant unique, le **GID** (*Group Identification*). Les groupes sont définis dans **/etc/group**.

La commande **id** permet d'obtenir ces informations. En interne, le système travaille uniquement avec les UID et GID, et pas avec les noms eux-mêmes.

```
$ id
uid=1000(seb) gid=100(users) groupes=7(lp),16(dialout),33(video),
100(users)
```

À chaque fichier (inode) sont associés un UID et un GID définissant son propriétaire et son groupe d'appartenance. Vous affectez des droits pour le propriétaire, pour le groupe d'appartenance et pour le reste du monde. On distingue trois cas de figure :

- UID de l'utilisateur identique à l'UID défini pour le fichier. Cet utilisateur est propriétaire du fichier.
- Les UID sont différents : le système vérifie si le GID de l'utilisateur est identique au GID du fichier. Si oui l'utilisateur appartient au groupe associé au fichier.
- Dans les autres cas (aucune correspondance) : il s'agit du reste du monde (others), ni le propriétaire, ni un membre du groupe.

d	rwxr-xr-x	29	seb	users	4096	Mar 15 22:13	Documents

Sur cette ligne du tableau, le répertoire Documents appartient à l'utilisateur seb et au groupe users, et possède les droits rwxr-xr-x.

b. Signification

Droit	Signification
Général	
r	Readable (lecture).
w	Writable (écriture).
x	Executable (exécutable comme programme).
Fichier normal	
r	Le contenu du fichier peut être lu, chargé en mémoire, visualisé, recopié.
w	Le contenu du fichier peut être modifié, on peut écrire dedans. La suppression n'est pas forcément liée à ce droit (voir droits sur répertoire).
x	Le fichier peut être exécuté depuis la ligne de commande, s'il s'agit soit d'un

	programme binaire (compilé), soit d'un script (shell, perl...).
Répertoire	
r	Les éléments du répertoire (catalogue) sont accessibles en lecture. Sans cette autorisation, ls et les critères de filtre sur le répertoire et son contenu ne sont pas possibles. L'accès individuel à un fichier reste possible si vous connaissez son chemin.
w	Les éléments du répertoire (catalogue) sont modifiables et il est possible de créer, renommer et supprimer des fichiers dans ce répertoire. **C'est ce droit qui contrôle l'autorisation de suppression d'un fichier.**
x	Le catalogue peut être accédé par CD et listé. Sans cette autorisation il est impossible d'accéder au répertoire et d'agir sur son contenu qui devient verrouillé.

Ainsi pour un fichier :

rwx	**r-x**	**r--**
Droits de l'utilisateur, en lecture, écriture et exécution.	Droits pour les membres du groupe en lecture et exécution.	Droits pour le reste du monde en lecture uniquement.

2. Modification des droits

Lors de sa création, un fichier ou un répertoire dispose de droits par défaut. Utilisez la commande **chmod** (*change mode*) pour modifier les droits sur un fichier ou un répertoire. Il existe deux méthodes pour modifier ces droits : par la forme symbolique et par la base 8. Seul le propriétaire d'un fichier peut en modifier les droits (plus l'administrateur système). Le paramètre `-R` change les droits de manière récursive.

a. Par symboles

La syntaxe est la suivante :

```
chmod modifications Fic1 [Fic2...]
```

S'il faut modifier les droits de l'utilisateur, utilisez le caractère **u**, pour les droits du groupe le caractère **g**, pour le reste du monde le caractère **o** et pour tous le caractère **a**.

Pour ajouter des droits, on utilise le caractère **+**, pour en retirer le caractère **-**, et pour ne pas tenir compte des paramètres précédents le caractère **=**.

Enfin, le droit d'accès par lui-même : **r**, **w** ou **x**.

Vous pouvez séparer les modifications par des virgules et cumuler plusieurs droits dans une même commande.

```
$ ls -l
total 0
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic3
$ chmod g+w fic1
$ ls -l fic1
-rw-rw-r-- 1 seb users 0 mar 21 22:03 fic1
$ chmod u=rwx,g=x,o=rw fic2
$ ls -l fic2
-rwx--xrw- 1 seb users 0 mar 21 22:03 fic2
$ chmod o-r fic3
$ ls -l fic3
-rw-r----- 1 seb users 0 mar 21 22:03 fic3
```

Si vous voulez supprimer tous les droits, ne précisez rien après le signe = :

```
$chmod o=fic2
$ ls -l fic2
-rwx--x--- 1 seb users 0 mar 21 22:03 fic2
```

b. Par base 8

La syntaxe est identique à celle des symboles. À chaque droit correspond une valeur octale, positionnelle et cumulable. Pour encoder trois droits rwx, il faut trois bits, chacun prenant la valeur 0 ou 1 selon que le droit est absent ou présent. 2^3=8, d'où une notation octale possible.

- Le r vaut 4.
- Le w vaut 2.
- Le x vaut 1.

Le tableau suivant vous aide :

Propriétaire			**Groupe**			**Reste du monde**		
r	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**
400	200	100	40	20	10	4	2	1

Pour obtenir le droit final il suffit d'additionner les valeurs. Par exemple si vous voulez rwxrw-rw- alors obtenez 400+200+100+40+10+4+1=755, et pour rw-r--r-- 400+200+40+4=644.

```
$ chmod 755 fic1
$ chmod 644 fic2
$ ls -l fic1 fic2
-rwxr-xr-x 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
```

La modificale octale des droits n'est pas fine et ne permet pas de modifier un seul droit. C'est l'intégralité des droits qui est modifiée en une fois.

3. Masque des droits

a. Restreindre des droits automatiquement

Lors de la création d'un fichier ou d'un répertoire, des droits leur sont automatiquement assignés. Généralement, c'est rw-r--r-- (644) pour un fichier et rwxr-xr-x (755) pour un répertoire. Ces valeurs sont contrôlées par un masque, lui-même modifiable par la commande **umask**. La commande **umask** prend comme paramètre une valeur octale dont chaque droit individuel sera supprimé des droits d'accès maximum du fichier ou du répertoire.

- Par défaut, tous les fichiers sont créés avec les droits 666 (rw-rw-rw-).
- Par défaut tous les répertoires sont créés avec les droits 777 (rwxrwxrwx).
- Puis le masque est appliqué.
- Le masque est le même pour l'ensemble des fichiers.
- Un masque ne modifie pas les droits des fichiers existants, mais seulement ceux des nouveaux fichiers.

Les droits par défaut (maximum) des fichiers et des répertoires ne sont pas identiques. C'est logique : comme le droit x permet de rentrer dans un répertoire, il est normal que celui-ci en dispose par défaut. Ce même droit est inutile par défaut sur les fichiers : seule une très petite minorité des fichiers sont des scripts et des programmes binaires.

Le masque par défaut est 022, soit `----w--w-`. Pour obtenir cette valeur, tapez **umask** sans paramètre.

```
$ umask
0022
```

b. Calcul de masque

Pour un fichier

```
Défaut  rw-rw-rw- (666)
Retirer ----w--w- (022)
Reste   rw-r--r-- (644)
```

Pour un répertoire

```
Défaut  rwxrwxrwx (777)
Retirer ----w--w- (022)
Reste   rwxr-xr-x (755)
```

Notez qu'appliquer un masque n'est pas soustraire, mais supprimer des droits de ceux par défaut, droit par droit. Par exemple :

```
Défaut  rw-rw-rw- (666)
Retirer ----wxrwx (037)
Reste   rw-r----- (640)
```

Et non 629, ce qui est impossible en octal...

4. Changer de propriétaire et de groupe

Il est possible de changer le propriétaire et le groupe d'un fichier à l'aide des commandes **chown** (*change owner*) et **chgrp** (*change group*). Le paramètre `-R` change la propriété de manière récursive.

```
chown utilisateur fic1 [Fic2...]
chgrp groupe fic1 [Fic2...]
```

En précisant le nom d'utilisateur (ou de groupe), le système vérifie d'abord son existence. Vous pouvez préciser un UID ou un GID, dans ce cas le système n'effectuera pas de vérification.

Pour les deux commandes, les droits précédents et l'emplacement du fichier ne sont pas modifiés. Il est possible de modifier en une seule commande à la fois le propriétaire et le groupe.

```
chown utilisateur[:groupe] fic1 [fic2...]
chown utilisateur[.groupe] fic1 [fic2...]
```

Seul root a le droit de changer le propriétaire d'un fichier. Mais un utilisateur peut changer le groupe d'un fichier s'il fait partie du nouveau groupe.

```
$ chgrp video fic1
$ ls -l fic1
-rwxr-xr-x 1 seb video 0 mar 21 22:03 fic1
```

5. Droits d'accès étendus

a. SUID et SGID

Il est possible d'établir des **droits d'accès étendus** à l'exécution d'une commande. Ces droits d'accès étendus appliqués à une commande permettent à cette commande de s'exécuter avec les droits du propriétaire ou du groupe d'appartenance de la commande, et non plus avec les droits de l'utilisateur l'ayant lancée.

L'exemple le plus simple est le programme **passwd** permettant de changer son mot de passe. Si la commande était exécutée avec les droits d'un utilisateur classique, **passwd** ne pourrait pas ouvrir et modifier les fichiers **`/etc/passwd`** et **`/etc/shadow`** :

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1440 fév 24 10:35 /etc/passwd
```

Vous constatez que ce fichier appartient à root, et que seul root peut y écrire. Un utilisateur simple ne peut lire que son contenu sans interagir. La commande **passwd** ne devrait donc pas pouvoir modifier les fichiers. Voyez les droits de la commande **passwd** (/bin/passwd ou /usr/bin/passwd) :

```
> ls -l /usr/bin/passwd
-rwsr-xr-x 1 root shadow 78208 sep 21 23:06 /usr/bin/passwd
```

Un nouveau droit est apparu : le droit **s** pour les droits de l'utilisateur root. Ce nouvel attribut permet l'exécution de la commande avec des droits d'accès étendus. Le temps du traitement, le programme est exécuté avec les droits du propriétaire du fichier ou de son groupe d'appartenance. Dans le cas de passwd, il est lancé avec les droits de root le temps de son traitement.

Le droit s sur l'utilisateur est appelé le **SUID-Bit** (*Set User ID Bit*), et sur le groupe le **GUID-Bit** (*Set Group ID Bit*)

La commande **chmod** permet de placer les SUID-Bit et GUID-Bit.

```
chmod u+s commande
chmod g+s commande
```

Les valeurs octales sont 4000 pour le SUID-Bit et 2000 pour le GUID-Bit.

```
chmod 4755 commande
chmod 2755 commande
```

Seul le propriétaire ou l'administrateur peut positionner ce droit. Positionner le SUID-bit ou le SGID-Bit n'a de sens que si les droits d'exécution ont préalablement été établis (attribut x sur le propriétaire ou le groupe). Si ceux-ci ne sont pas présents ; le **s** est remplacé par un **S**.

b. Real / effectif

Dans les données d'identification du processus vous avez vu la présence de **UID et de GID réels et effectifs**. Quand une commande est lancée avec un SUID-Bit ou un SGID-Bit positionné, les droits se trouvent modifiés. Le système conserve les UID et GID d'origine de l'utilisateur ayant lancé la commande (UID et GID réels) transmis par le père, les numéros UID et GID effectifs étant ceux du propriétaire ou du groupe d'appartenance du programme.

Ex : toto (UID=100, GID=100) lance passwd, appartenant à root (UID=1, GID=1) avec le SUID-Bit positionné.

```
UID réel : 100
GID réel : 100
UID effectif : 1
GID effectif : 100
```

Si le SGID-Bit était positionné seul :

```
UID réel : 100
GID réel : 100
UID effectif : 100
GID effectif : 1
```

Il est à noter que les SUID-Bit et SGID-bit ne sont pas transmis aux enfants d'un processus. Dans ce cas les enfants seront exécutés avec les droits de l'utilisateur ayant lancé la commande de base.

c. Sticky bit

Le **sticky bit** (*bit collant*) permet d'affecter une protection contre l'effacement du contenu d'un répertoire. Imaginez un répertoire /tmp où tous les utilisateurs ont le droit de lire et d'écrire des fichiers.

```
$ ls -ld /tmp
drwxrwxrwx   6 root     system     16384 Aug 14 13:22 tmp
```

Dans ce répertoire tout le monde peut supprimer des fichiers y compris ceux qui ne lui appartiennent pas (droit w présent partout et pour tous). Si l'utilisateur toto crée un fichier, l'utilisateur titi peut le supprimer même s'il ne lui appartient pas.

Le sticky bit appliqué à un répertoire, ici /tmp, empêche cette manipulation. Si le fichier peut encore être visualisé et

modifié, seul son propriétaire (et l'administrateur) pourra le supprimer.

```
$ chmod u+t /tmp
ls -ld /tmp
drwxrwxrwt 35 root root 77824 mar 21 22:30 /tmp
```

En octal, on utilisera la valeur 1000 (`chmod 1777 /tmp`).

Bien qu'appliqué à l'utilisateur, le sticky bit, représenté par un **t** apparaît au niveau de « others ».

d. Droit s et répertoire

Si vous donnez le droit s au groupe sur un répertoire, tous les fichiers créés au sein de ce répertoire, et quel que soit le groupe de la personne créant ce fichier, seront du même groupe que ce répertoire.

```
$ mkdir rep
$ chmod 770 rep
$ ls -ld rep
drwxrwx--- 2 seb users 4096 mar 21 22:36 rep
$ chgrp video rep
$ chmod g+s rep
$ ls -ld rep
drwxrws--- 2 seb video 4096 mar 21 22:37 rep
$ cd rep
$ touch toto
$ ls -l toto
-rw-r--r-- 1 seb video 0 mar 21 22:37 toto
```

Processus de démarrage

1. Le BIOS

a. Rôle

Le **BIOS** (*Basic Input Output System*) est l'interface logicielle entre le matériel et le logiciel à un niveau très basique. Il fournit l'ensemble des instructions de base utilisées par le système d'exploitation. Il fournit le niveau d'interface le plus bas aux pilotes et périphériques.

Le BIOS est présent sur une mémoire **EEPROM** (*Electrical Erasable Programmable Read-Only Memory*) de l'ordinateur. Quand l'ordinateur est électriquement allumé, ou lors d'un reset, un signal appelé *powergood* est envoyé au microprocesseur. Celui-ci déclenche alors l'exécution du BIOS.

Le BIOS effectue un auto-test de l'allumage (POST) puis recherche les périphériques, notamment ceux utilisés pour démarrer. Les informations sur le matériel sont stockées de manière permanente dans une petite mémoire CMOS alimentée par une batterie. À la fin du processus, le périphérique de démarrage est sélectionné.

Le BIOS lit et exécute le premier secteur physique du média de démarrage. Il s'agit généralement des 512 premiers octets du premier disque dur (le MBR) ou de la partition active (le PBR), comme le chapitre Les disques et les systèmes de fichiers vous l'a décrit.

b. Réglages basiques

Chaque BIOS est différent selon les constructeurs de cartes mères et les éditeurs (AMIBios, Phoenix, Award, etc.). Cependant de nombreux réglages sont identiques ou en tout cas se ressemblent en passant de l'un à l'autre.

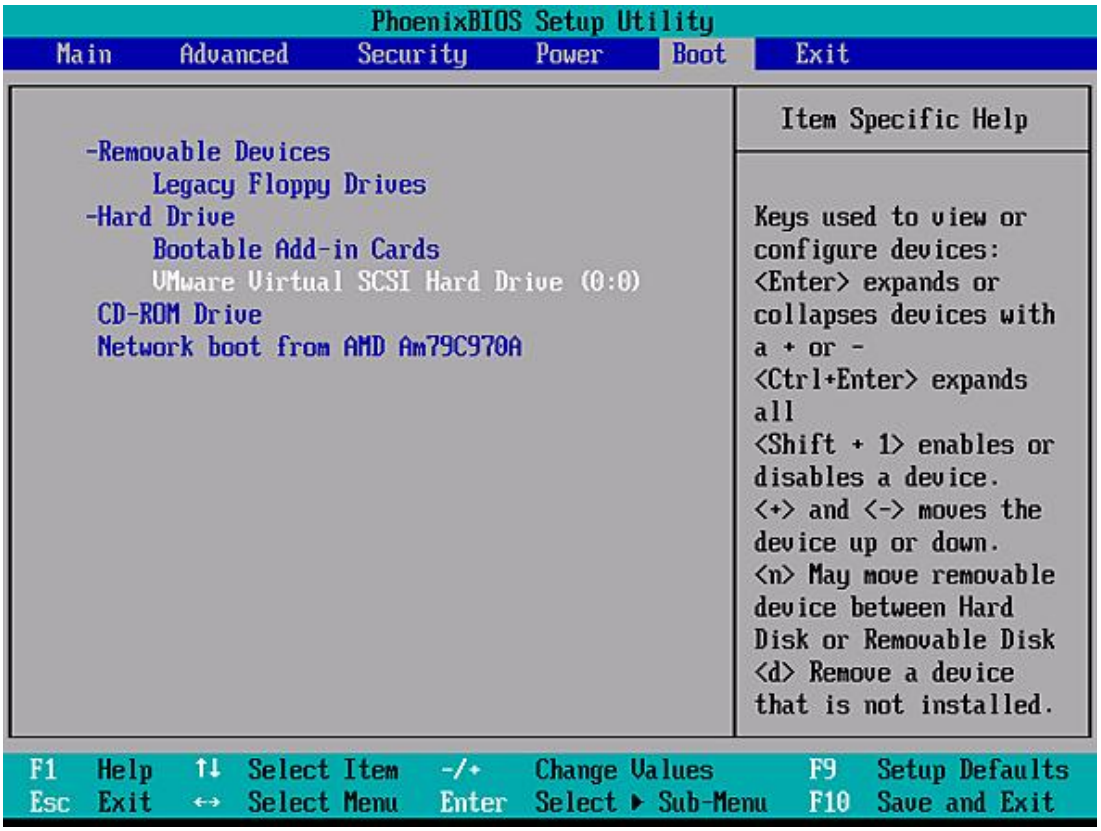
C'est par le BIOS que s'effectue la détection des disques durs et le choix du support de démarrage. Linux supporte l'IDE, le SATA et le SCSI. Il se peut cependant qu'en SATA votre chipset ne soit pas reconnu. Dans ce cas, la plupart des BIOS permettent de passer le contrôleur SATA en mode d'émulation IDE. Linux les reconnaîtra comme tels. Cependant vous ne perdrez rien à tester une première installation avec le support natif des disques en SATA activé.

En principe, Linux gère bien le support des chipsets SATA compatibles **AHCI** (*Advanded Host Controller Interface*), un standard aux spécifications publiques. Activez ce choix dans le BIOS s'il vous est proposé, il apparaît sous ce nom, et parfois en **native**. Si rien ne fonctionne, tentez le mode **combined**, puis **legacy IDE**. Vous trouverez de l'aide sur le SATA sur le site suivant : http://linux-ata.org/faq.html

Pour démarrer l'installation de Linux depuis un CD-Rom ou un DVD-Rom vous devez modifier l'ordre de démarrage dans la séquence de boot pour démarrer en premier sur le lecteur CD ou DVD.

Si votre clavier est de type USB, ou sans fil mais avec un adaptateur sans fil USB, vous devez activer le **USB legacy support** (cette fonction s'appelle parfois **USB DOS function ou USB keyboard enable**). Il permet d'activer au boot le support des claviers mais aussi des supports de stockage USB (clés, disques durs, cartes mémoire). Ceci n'empêche pas la prise en charge de l'USB par le système : une fois l'OS démarré, ce sont les pilotes USB du noyau et des modules qui prennent en charge l'USB.

Vous n'avez pas, en principe, à toucher aux autres réglages. Évitez notamment de jouer à l'apprenti sorcier en modifiant les réglages avancés du chipset et autres ressources dont vous ne comprenez pas l'utilité. Vous pouvez cependant dans un souci d'économie de ressources désactiver les ports de la carte mère que vous n'utilisez pas : port parallèle, port série, etc.



Écran du BIOS Phoenix modifiant l'ordre de boot

L'overclocking nécessite du matériel adapté : processeur, carte mère, mémoire et alimentation doivent être de haute qualité et le PC bien ventilé. L'overclocking est à la base de nombreuses sources d'instabilité et de plantages, tant sous Windows que sous Linux. La mémoire notamment est soumise à rude épreuve. Elle est la principale cause d'instabilité. Même sans overclocking, il n'est pas vain d'investir dans des composants de qualité.

2. Le chargeur de démarrage

Le BIOS active le chargeur de programme initial (*Initial Program Loader*, IPL) à partir des premiers 512 octets du support de démarrage. Sur Linux, le chargeur est décomposé en deux parties. Le chargeur initial des 512 octets ne contient pas assez de code pour proposer des menus et lancer le chargement d'un système d'exploitation. Il charge la seconde phase, basée sur un fichier de configuration.

La seconde phase fournit une interface pour lancer un système d'exploitation parmi un choix donné. Vous pouvez en profiter pour passer des paramètres au noyau Linux et au processus init.

Le BIOS n'intervient qu'au démarrage de la machine, à l'utilisation du chargeur de démarrage et aux toutes premières étapes du chargement du noyau. Ensuite, il devient inutile. Le noyau dispose de ses propres fonctions de détection bien qu'il s'appuie sur la configuration du BIOS. En effet ce dernier, sur plate-forme Intel, s'exécute en mode réel et Linux en mode protégé.

3. GRUB

a. Configuration

Le chargeur par défaut sur la plupart des distributions Linux s'appelle **GRUB** (*Grand Unified Bootloader*). Il est hautement paramétrable, notamment en acceptant une protection par mot de passe crypté, un interpréteur de commandes ou encore des graphiques. Il est basé sur un fichier texte de configuration et il n'y a pas besoin de réinstaller GRUB à chaque modification.

Voici un exemple de configuration partant du principe que la première partition du premier disque est /boot, et que la seconde contient une installation de Windows.

```
timeout=10
default=0
title Red Hat
    root (hd0,0)
    kernel /vmlinuz-2.6.12-15 ro root=LABEL=/
    initrd /initrd-2.6.12-15.img
title Windows XP
    rootnoverify (hd0,1)
    chainloader +1
```

Voici la syntaxe générale d'un fichier GRUB :

Paramètre GRUB	Signification
timeout	Nombre de secondes avant le démarrage par défaut.
default n	Démarrage par défaut (0=1er titre, 1=2ème titre, etc.).
gfxmenu	Chemin vers un menu graphique.
title xxxx	Début d'une section, entrée du menu de GRUB.
root(hdx,y)	Tous les accès fichiers spécifiés dessous le seront à partir de cette partition (cf. signification plus bas). Ici hd0,0 représente la première partition du premier disque détecté par le BIOS. C'est la partition /boot.
kernel	Le nom de l'image du noyau Linux, suivi de ses paramètres. Le / n'indique pas la racine du système de fichiers mais celle de (hd0,0), donc /boot/vmlinuz...
initrd	Initial ramdisk. Le noyau va charger ce fichier comme disque en mémoire pour y trouver une configuration et des pilotes initiaux.
rootnoverify	La racine spécifiée, à ne pas monter par GRUB (il ne supporte pas NTFS).
chainloader +1	Démarrer le premier secteur de la racine spécifiée ci-dessus.

Voici la signification des noms de périphériques sous GRUB.

- (fd0) : premier lecteur de disquettes détecté par le BIOS (/dev/fd0 sous Linux).
- (hd0,0) : première partition sur le premier disque dur détecté par le BIOS que ce soit IDE ou SCSI (/dev/hda1 ou /dev/sda1 suivant le cas).
- (hd1,4) : cinquième partition sur le second disque dur détecté par le BIOS (/dev/hdb5 ou /dev/sda5).

b. Installation

La configuration de **GRUB** réside dans **`/etc/grub.conf`** ou **`/boot/grub/menu.lst`** (le premier est un lien sur l'autre). GRUB peut s'installer sur un **MBR** (*Master Boot Record*, les 512 premiers octets d'un disque) ou un **PBR** (*Partition Boot Record*, les 512 premiers octets d'une partition).

Pour installer ou réinstaller GRUB en cas de MBR corrompu, par exemple sur **`/dev/sda`** utilisez la commande **grub-install** :

```
# /sbin/grub-install /dev/sda
```

c. Démarrage et édition

Au démarrage de GRUB, un menu s'affiche. Il peut être graphique ou textuel, selon la configuration. Vous devez choisir une image de démarrage avec les flèches de direction parmi celles proposées. En appuyant sur la touche [Entrée] vous démarrez l'image sélectionnée.

Vous pouvez éditer les menus directement pour modifier par exemple les paramètres passés au noyau Linux ou init. Dans ce cas, sélectionnez une entrée de menu et appuyez sur la touche **e** (edit). Ici, toutes les lignes de la section sont affichées. Vous pouvez appuyer sur :

- e : pour éditer la ligne (la compléter) ;
- d : pour supprimer la ligne ;
- o : pour ajouter une ligne ;
- b : pour démarrer l'image (booter).

Par exemple, pour démarrer en mode urgence (emergency) :

- Allez sur la ligne Linux ou Red Hat et appuyez sur **e**.
- Allez sur la ligne kernel et appuyez sur **e**.
- À la fin de la ligne rajoutez 1 ou Single et appuyez sur [Entrée].
- Appuyez sur **b**.

Vous pouvez aussi accéder à un interpréteur de commandes en appuyant sur [Echap]. Attention seules les commandes GRUB sont reconnues.

4. Initialisation du noyau

Au chargement du noyau une multitude d'informations défile sur l'écran. Vous ne pouvez pas figer ces informations à ce moment-là. Par contre juste après le passage à l'étape suivante (init) toutes les traces du noyau sont placées dans le fichier **`/var/log/dmesg`**.

- Le matériel est détecté et initialisé.
- initrd est chargé, les modules présents éventuellement chargés.
- Le noyau monte le système de fichiers racine en lecture seule.
- Il crée la première console.
- Le premier processus est lancé (normalement init).

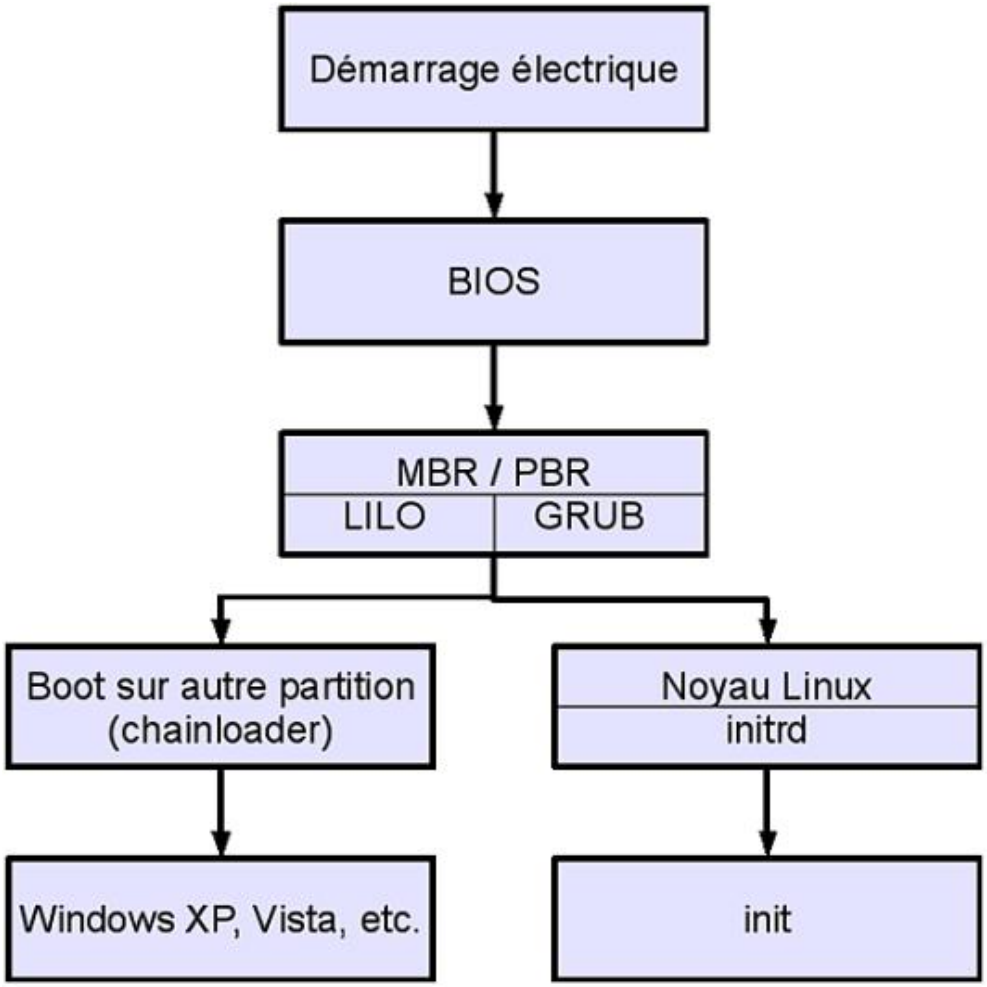


Schéma de la séquence de démarrage

Il existe d'autres chargeurs dont un appelé **LILO** *(Linux Loader)*. Celui-ci a été jusqu'à peu le chargeur le plus utilisé. Ces dernières années, il a été cependant presque totalement remplacé par GRUB à cause de nombreuses limitations.

init

1. Rôle

Le programme init, premier démarré et dernier stoppé au sein du système, est celui qui lance toutes les autres tâches. Le rôle initial de init est de démarrer et d'arrêter tous les services. C'est init qui va exécuter les diverses tâches initiales nécessaires au bon fonctionnement de Linux via l'exécution de plusieurs commandes et scripts.

Une fois le système démarré et les services lancés, init reste actif pour gérer les changements d'état des processus qu'il contrôle et des niveaux d'exécution.

Le programme init n'est pas toujours le même d'une distribution à une autre. Sur la plupart des distributions professionnelles et/ou majeures (Mandriva, Red Hat/Fedora, openSUSE, Debian, etc.) le principe est globalement le même : init de type System V (basé sur la notion de niveaux d'exécution). La distribution Ubuntu utilise Upstart qui gère les priorités, les événements et dépendances entre les services, mais qui reste calquée sur le même principe. La distribution Slackware utilise un autre principe issu du fonctionnement de BSD.

Le processus **init** est le père de tous les processus. Il a toujours le PID 1. Sa configuration est présente dans le fichier **`/etc/inittab`**. Si ce fichier est corrompu et inutilisable, il faudra démarrer en mode single (S, s, 1, Single) et le réparer, ou au pire démarrer depuis un support externe ou un disque de secours. C'est un fichier central du système d'exploitation.

2. Niveaux d'exécution

Un niveau d'exécution, ou runlevel, est un état dans lequel se trouve Unix/Linux. Cet état est contrôlé par init. Chaque état dispose de sa propre configuration (soit par inittab, soit par des scripts appelés initscripts). Un niveau d'exécution peut par exemple être utilisé pour lancer Unix en mono-utilisateur, en multi-utilisateurs, avec ou sans réseau, avec ou sans mode graphique. Tous les niveaux sont personnalisables par l'administrateur. Ces niveaux sont généralement définis comme ceci par convention sur les distributions Red Hat/Fedora, Mandriva, openSUSE et associées :

Niveau	Effet
0	Halt : stoppe le système d'exploitation, éteint la machine.
1	Mode mono-utilisateur utilisé pour la maintenance, mode console.
2	Multi-utilisateur, sans réseau, console.
3	Multi-utilisateur, avec réseau, console.
4	Idem que le 3, laissé à la convenance de l'administrateur.
5	Multi-utilisateur, avec réseau, avec environnement graphique X Window.
6	Reboot : redémarrage de la machine.
S,s	Single user mode, le mode le plus bas en cas de soucis.

Les niveaux 7 à 9 sont parfaitement valides mais pas utilisés par défaut. Le niveau d'exécution par défaut est positionné dans **`/etc/inittab`** sur la ligne initdefault.

```
id:5:initdefault:
```

Remplacez 5 par le niveau souhaité au démarrage.

Un gag, posé lors de certains exercices éliminatoires de certifications, consiste à créer une situation de panne où la valeur par défaut est positionnée à 0 ou 6. Dans le premier cas la machine s'éteint dès l'exécution de init, dans l'autre elle redémarre en boucle...

La distribution Debian (et les distributions qui en dérivent) considère aussi les niveaux 2 à 5 comme multi-utilisateur

mais n'établit pas de différences entre ces niveaux, et démarre par défaut au niveau 2 où tout est lancé, y compris éventuellement l'interface graphique.

Comme chaque niveau peut être totalement modifié et reconfiguré, il est possible de tout redéfinir et donc de faire en sorte qu'une Debian démarre comme une Red Hat, ou qu'une Red Hat démarre comme une Debian. Pour des raisons de conformité et de support, considérez qu'il est important de rester conforme au « standard » de la distribution que vous utilisez.

3. /etc/inittab

Le comportement du processus init et des runlevels est défini dans le fichier **`/etc/inittab`**. La syntaxe d'une ligne est la suivante :

```
Id:[niveaux]:action:commande
```

Champ	Description
Id	Identifiant de ligne sur quatre caractères, juste indicatif (sous Linux avec getty/mingetty : numéro de terminal).
Niveaux	Indique si la commande doit être prise en compte pour le niveau demandé, c'est la liste des niveaux sans séparateur.
Action	Type d'action à effectuer selon les circonstances pour cette ligne.
Commande	La commande à exécuter avec ses paramètres et les redirections.

L'action est très importante car elle définit les activités de init lors du démarrage et du changement de niveau. Voici les principales :

Action	Signification
initdefault	Définit le niveau par défaut lors du boot et du lancement d'init.
sysinit	Exécuté une seule et unique fois lors du démarrage du système.
boot	Idem mais après sysinit.
bootwait	Idem, mais init attend la fin de l'exécution de la commande avant de continuer à dérouler le fichier inittab.
off	La ligne est ignorée.
once	La commande est exécutée à chaque changement de niveau pour les niveaux spécifiés.
wait	Idem, mais init attend la fin de l'exécution avant de continuer.
respawn	La commande est lancée pour les niveaux concernés. Si le processus se termine, il est automatiquement relancé. C'est le cas pour les terminaux si un utilisateur s'en déconnecte.
powerwait	La commande est lancée si le serveur passe sur alimentation de secours (UPS).
powerfail	Idem, mais sans attente de la fin d'exécution de la commande.
powerokwait	La commande est lancée lorsque le courant est rétabli.
powerfailnow	Commande de dernier recours lorsque l'alimentation de secours est presque vide.
ctrlaltdel	Init reçoit un signal SIGINT issu d'une séquence [Alt][Ctrl][Suppr].

Voici un exemple issu d'une installation openSUSE 10.3 :

```
# Niveau d'exécution à 5 (multiuser graphique)
id:5:initdefault:

# Premier script execute au démarrage
si::bootwait:/etc/init.d/boot

# Gestion des services par niveau d'exécution
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

# Cas du mode single, console de secours pour root
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin

# Action sur Alt+Ctrl+Del
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# Que faire en cas de coupure de courant
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# Lancement des consoles virtuelles Alt+Fx
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

4. Changement de niveau

Vous pouvez changer de niveau à la volée après le démarrage de la machine avec la commande **/sbin/init** ou **/sbin/telinit**, cette dernière étant un simple lien symbolique vers init. La commande suivante passe au niveau 5.

```
# telinit 5
```

Les valeurs `q`, `u` ou `-t` peuvent être précisées :

- `Q` ou `q` : init relit le fichier **/etc/inittab**, s'il a été modifié, en corrigeant ses tables internes.
- `U` ou `u` : init se relance sans relire inittab et sans changer de niveau. Si des services ont été rajoutés ou supprimés du niveau en cours, init prend en compte la modification.
- `-t` : quand init a terminé l'arrêt des services (ou plutôt quand le script rc l'a fait, voir un peu plus loin), init envoie le signal SIGTERM à tous les processus restants, leur demandant de se terminer proprement, attend le nombre de secondes spécifié (5 par défaut), puis envoie SIGKILL.

Le niveau d'exécution actuel est visible avec la commande **/sbin/runlevel**. La première valeur retournée est le niveau précédent le niveau actuel. Un N signifie qu'il n'y a pas de précédent niveau. La seconde valeur est le niveau actuel.

```
# runlevel
N 5
```

5. Paramétrage système de base

Quel que soit le niveau d'exécution précisé par défaut, init lance toujours la commande associée aux actions sysinit, bootwait ou boot de **/etc/inittab** lors du démarrage du système, l'action sysinit étant la première.

- Sous Red Hat : `si::sysinit:`**`/etc/rc.d/rc.sysinit`**
- Sous openSUSE : `si::bootwait:`**`/etc/init.d/boot`**
- Sous Debian : `si::sysinit:`**`/etc/init.d/rcS`**

Sous Red Hat, c'est un seul script monolithique qui s'occupe de toute la configuration de base. Sous Debian, le script appelle tous les scripts du niveau S (single). Sous openSUSE le script met en place le strict nécessaire puis exécute le contenu de **/etc/rc.d/boot.d** qui établit le reste de la configuration de base.

Dans tous les cas, les tâches suivantes sont exécutées à peu près dans cet ordre :

- Configuration des paramètres du noyau présents dans **/etc/sysctl.conf** (ex : IP Forwarding).
- Mise en place des fichiers périphériques (/dev via udev par exemple).
- Configuration de l'horloge du système.
- Chargement des tables de caractères du clavier.
- Activation des partitions d'échange SWAP.
- Définition du nom d'hôte.
- Contrôle et montage du système de fichiers racine (en lecture-écriture cette fois).
- Ajout des périphériques RAID et/ou LVM. Ceci peut déjà être mis en place lors du chargement de inittab.
- Activation des quotas de disque.
- Contrôle et montage des autres systèmes de fichiers.
- Nettoyage des verrous (stale locks) et des fichiers PID en cas d'arrêt brusque.

Il est possible avec certaines distributions de passer en mode interactif. Au début du boot, après le lancement d'init, il peut vous être demandé de taper sur la lettre i puis de répondre par oui ou par non aux différentes actions.

6. Niveaux d'exécution System V

a. rc

Le script **/etc/init.d/rc** prend comme paramètre le niveau d'exécution par défaut selon la ligne **initdefault** de **/etc/inittab** ou celui spécifié lors de l'appel manuel des commandes **init** ou **telinit**. Le script rc initialise le niveau d'exécution voulu et est responsable du démarrage et de l'arrêt des services associés quand le niveau d'exécution change.

```
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

Les services sont analysés à chaque niveau d'exécution. Lors du passage d'un niveau à un autre, et quel que soit l'ordre (du 2 au 5, du 5 au 3, etc.) le script rc compare les services qui doivent être arrêtés ou démarrés entre

l'ancien et le nouveau niveau. Si un service est commun aux deux niveaux, il est maintenu. Si un nouveau service doit être lancé dans le nouveau niveau, il le lance. Si un service doit être arrêté car il est absent du nouveau niveau, il l'arrête.

Ce fonctionnement, standard à toutes les distributions Linux de type System V, n'est pas commun à tous les Unix. HP-UX (un Unix de HP) considère qu'il doit y avoir une progression constante dans les niveaux, passant successivement du 1 au 3 (1 puis 2 puis 3) et chargeant successivement les services. À l'arrêt il redescend jusqu'au niveau 0 en terminant successivement les services. La différence est de taille : il ne compare pas les niveaux et n'effectue pas d'arrêt/relance entre chaque niveau...

7. Gestion des niveaux et des services

a. Services dans init.d

Le niveau d'exécution définit les services à démarrer pour ce niveau. C'est le script rc qui charge les services. Les services sont contrôlés (démarrage, arrêt, relance, status, etc.) à l'aide de scripts présents dans **/etc/init.d**.

```
# cd /etc/init.d
# ls -l
-rwxr-xr-x  1 root root  1128 aou  9  2004 acpid
-rwxr-xr-x  1 root root   834 sep 28  2004 anacron
-rwxr-xr-x  1 root root  1429 jun 22  2004 apmd
-rwxr-xr-x  1 root root  1176 avr 14  2006 atd
-rwxr-xr-x  1 root root  2781 mar  5  2007 auditd
-rwxr-xr-x  1 root root 17058 sep  5  2007 autofs
-rwxr-xr-x  1 root root  1368 fev  2  2007 bluetooth
-rwxr-xr-x  1 root root  1355 mai  2  2006 cpuspeed
-rwxr-xr-x  1 root root  1904 jui 16  2007 crond
-rwxr-xr-x  1 root root  2312 oct 30 13:46 cups
...
```

Pour chaque niveau d'exécution *n*, il existe un répertoire rc*n*.d qui contient des liens symboliques (raccourcis) vers les services présents dans **/etc/init.d** à lancer ou arrêter. Ce répertoire peut être à différents endroits selon la distribution :

- Red Hat : **/etc/rc.d/rcn.d** avec des liens sur **/etc/rcn.d**
- openSUSE : **/etc/init.d/rcn.d** sachant que **/etc/rc.d** pointe sur **/etc/init.d**
- Debian : **/etc/rcn.d**

Le préfixe du nom de chaque lien définit son ordre de lancement ou son ordre d'arrêt. Le nom est sous la forme suivante :

`[SK]nnservice`

- **S** : start.
- **K** : kill (stop).
- **nn** : ordre numérique de démarrage ou d'arrêt. (00=premier, 99=dernier).
- **service** : nom du service.

Par exemple le lien S10network indique que le service network, responsable de la mise en place du réseau, sera démarré en ordre 10, après les S01, S05, etc. mais avant les S11, S15, S20, etc.

```
# ls -1 S*
S00microcode_ctl
S01sysstat
S02lvm2-monitor
```

```
S05kudzu
S06cpuspeed
S08iptables
S09isdn
S09pcmcia
S10network
S12syslog
S13irqbalance
S13portmap
S14nfslock
S15mdmonitor
S18rpcidmapd
...
```

Quand rc est exécuté, il va tout d'abord lister tous les liens commençant par K* à l'aide d'une boucle for. Puis il fait la même chose pour S*, cette fois en lançant les services. Voici un bout du fichier rc pour mieux comprendre la séquence de démarrage :

```
# test existence de /etc/rcn.d
if [ -d /etc/rc${level}.d ]
then
    # Liste tous les scripts commençant par S dans ce répertoire
    for i in /etc/rc${level].d/S*
    do
        # le script existe et n'est pas vide : on l'exécute
        if [ -s ${i} ]
             then
                 sh ${i} start
        fi
    done
fi
```

b. Contrôle manuel des services

Via le script

Les services peuvent être lancés dans tous les cas individuellement, ou à l'aide d'outils selon la distribution. La première méthode est la seule par défaut sous Debian. Chaque service présent dans /etc/init.d accepte au moins deux paramètres :

- **start** : le service démarre.
- **stop** : le service s'arrête.

Si vous souhaitez démarrer et arrêter le service sshd (serveur ssh) à la main :

```
# /etc/init.d/sshd start
Starting SSH daemon                                        done
# /etc/init.d/sshd stop
Shutting down SSH daemon                                   done
```

Certains services peuvent accepter d'autres paramètres :

```
# /etc/init.d/sshd
Usage: /etc/init.d/sshd {start|stop|status|try-
restart|restart|force-reload|reload|probe}
```

- **status** : fournit l'état du service (démarré ou non). Selon les services des informations supplémentaires peuvent être fournies.
- **probe** : indique s'il y a nécessité de recharger la configuration, si des fichiers de configuration ont par exemple été modifiés.
- **reload / forcereload** : indique au service de relire sa configuration (via un signal 1).

- **restart** : arrête et relance le service, quelle que soit l'issue de l'arrêt.
- **try-restart** : arrête et relance le service seulement si l'arrêt a bien eu lieu.

La distribution openSUSE crée des liens symboliques **rc<service>** permettant de saisir **rcsshd** par exemple pour le contrôle manuel des services.

Via la commande service

La commande **service** est disponible sous Red Hat et openSUSE. Elle permet simplement de se passer du chemin vers le script de lancement du service et d'utiliser simplement son nom :

```
# service sshd stop
Shutting down SSH daemon                                    done
# service sshd start
Starting SSH daemon                                         done
```

Pour contrôler la configuration des services de System V lancés par init, il n'est pas conseillé de tout faire à la main mais plutôt d'utiliser les outils du système concerné quand ils existent, en mode texte ou graphique.

c. Modification des niveaux d'exécution

Red Hat et openSUSE

Sous Red Hat/Fedora et openSUSE la commande **chkconfig** permet d'ajouter, de supprimer, d'activer ou de désactiver des scripts, par niveau d'exécution. Cette commande est très pratique pour configurer les services parce qu'elle sait gérer tant les services System V que les services xinetd.

```
chkconfig [option] [service]
```

Bien que la syntaxe soit identique dans les deux distributions, chkconfig ne fonctionne pas de la même manière. Sous Red Hat une ligne spéciale est insérée en début de script qui indique à chkconfig les paramètres par défaut (runlevels, positions de démarrage et d'arrêt). Sous openSUSE, chkconfig est un frontend pour la commande **insserv**. Cette dernière exploite aussi l'en-tête des scripts, mais de manière plus complexe (elle gère l'ordonnancement et le parallélisme par exemple).

Voici le début du script de lancement de service sshd sous Red Hat. Le script démarre et s'arrête dans les niveaux 2, 3, 4 et 5. Il démarre en position 55 (S55sshd) et s'arrête en position 25 (K25sshd).

```
# chkconfig: 2345 55 25
# description: OpenSSH server daemon
```

Voici la même chose sous openSUSE. chkconfig et insserv gèrent eux-mêmes l'ordre de démarrage et d'arrêt grâce aux champs Required-Start et Required-Stop. Les services réseaux et remote_fs doivent être démarrés avant sshd. Le service démarre aux niveaux 3 et 5 et est stoppé aux niveaux 0 (arrêt), 1 (single user), 2 (sans réseau) et 6 (reboot).

```
### BEGIN INIT INFO
# Provides: sshd
# Required-Start: $network $remote_fs
# Required-Stop: $network $remote_fs
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Description: Start the sshd daemon
### END INIT INFO
```

Voici la liste des options de chkconfig :

- **--list** : liste de l'ensemble de la configuration.
- **--list** service : la configuration d'un service donné.

- **--add** service : ajoute le service indiqué dans la configuration System V.
- **--del** service : supprime le service de la configuration System V.
- **--level xxx** service **on/off** : active ou désactive le service pour les niveaux d'exécution indiqués.

```
# chkconfig --list
rwhod       0:arrêt  1:arrêt  2:arrêt  3:arrêt  4:arrêt  5:arrêt  6:arrêt
atd         0:arrêt  1:arrêt  2:arrêt  3:marche 4:marche 5:marche 6:arrêt
snmpd       0:arrêt  1:arrêt  2:marche 3:marche 4:marche 5:marche 6:arrêt
ntpd        0:arrêt  1:arrêt  2:marche 3:marche 4:marche 5:marche 6:arrêt
keytable    0:arrêt  1:marche 2:marche 3:marche 4:marche 5:marche 6:arrêt
syslog      0:arrêt  1:arrêt  2:marche 3:marche 4:marche 5:marche 6:arrêt
...

# chkconfig --list smb
smb         0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt

# chkconfig --level 35 smb on
# chkconfig --list smb
smb         0:arrêt  1:arrêt  2:arrêt  3:marche 4:arrêt  5:marche 6:arrêt

# /sbin/chkconfig --add httpd
```

chkconfig ne lance aucun service. Il ne fait que paramétrer les niveaux d'exécution. Pour lancer un service, on utilisera le script associé ou la commande service.

Sous Debian

La commande **update-rc.d** crée les liens nécessaires dans les divers répertoires rc*n*.d, un peu comme chkconfig, mais de manière plus « brute » : les scripts n'ont pas d'information particulière dans leur en-tête et la commande ne fonctionne qu'au niveau du système de fichiers, mettant en place les divers liens selon vos indications.

Voici deux exemples de syntaxe. La première inscrit un service avec des paramètres par défaut. Dans ce cas le service est configuré pour démarrer sur les niveaux de 2 à 5 et s'arrêter aux niveaux 0, 1 et 6. La position d'arrêt/relance est à 20.

```
# update-rc.d ssh defaults
```

Dans le second exemple, le service est inséré avec des options complètes. Au démarrage le service est en position 10 sur les niveaux 3, 4 et 5. À l'arrêt le service est à la position 5 sur les niveaux 0, 1 et 6. N'oubliez pas les points.

```
# update-rc.d ssh start 10 3 4 5 . stop 05 0 1 6 .
```

Le paramètre remove supprime les liens des divers répertoires. Cependant le script **/etc/init.d/xxx** associé doit lui-même ne plus exister. Dans le cas contraire, utilisez le paramètre `-f` pour forcer la suppression des liens (le script lui-même reste en place).

```
# update-rc.d -f ssh remove
```

8. Consoles virtuelles

Les consoles virtuelles permettent d'obtenir des terminaux virtuels sur une machine. Elles sont définies dans **/etc/inittab**. Elles sont disponibles via les périphériques **/dev/ttyn** où n est le numéro de console.

La couche graphique n'est généralement pas installée ni lancée sur les serveurs en entreprise. Vous accédez au serveur soit directement (ou par kvm), soit par le réseau (ssh par exemple) ou encore via des solutions intégrées (ports d'administration du serveur, comme ILO sur les machines HP).

- Passez d'une console à l'autre avec la séquence de touches [Alt][F*n*] (ex : [Alt][F2]) depuis la console ou [Ctrl][Alt][F*n*] depuis X Window.

- Utilisez les touches [Alt][Flèche à droite] et [Alt][Flèche à gauche] pour passer à la console suivante ou précédente.
- /dev/ttyn représente la console virtuelle n.
- /dev/tty0 représente la console courante.
- Comme il y a 12 touches de fonction, il peut y avoir par défaut 12 terminaux virtuels.
- Cependant 6 « seulement » sont activés par défaut.
- X Window est lancé par défaut sur la première console disponible, généralement la 7.

Les consoles sont lancées par inittab et par les processus **getty** ou **mingetty**. Ce sont les seules entrées de inittab ou le label a son importance : il correspond au numéro de la console. Notez l'utilisation de respawn. Comme c'est la console qui établit la demande de login puis est substituée par le shell, lorsque le shell se termine, un processus mingetty est automatiquement relancé pour accepter une nouvelle connexion.

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

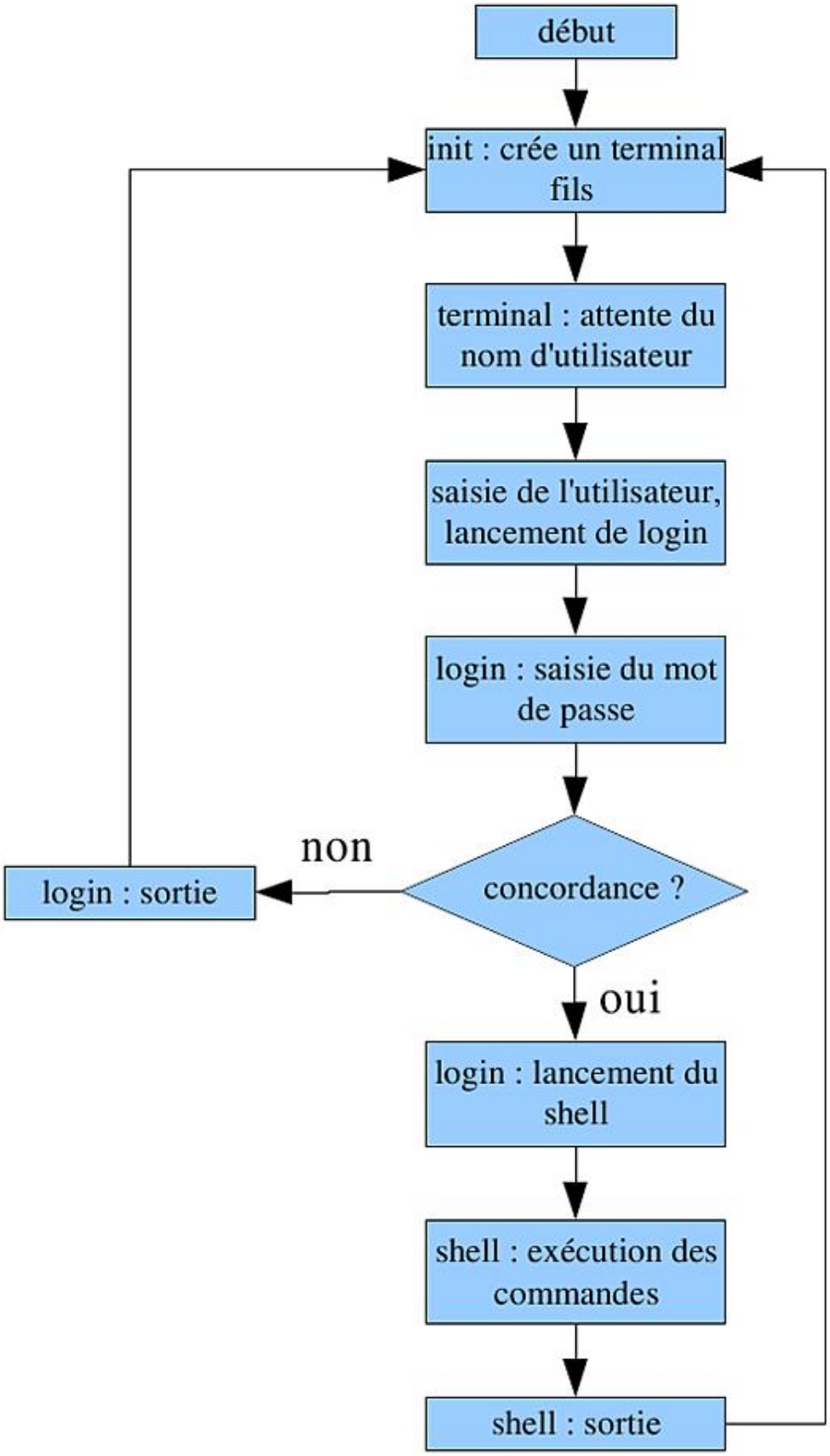
9. Les logins

Une fois les terminaux (getty) lancés par init, un prompt attend la saisie du nom (le login) de l'utilisateur. Avant ce prompt le contenu du fichier **/etc/issue** est affiché. Le nom validé, le terminal exécute la commande **login** qui va demander la saisie du mot de passe. Si le mot de passe est correct (vérification dans **/etc/passwd** et **/etc/shadow** ou utilisation des modules PAM), alors login affiche le contenu de **/etc/motd** et exécute un shell (toujours défini dans /etc/passwd).

Notez que getty et login n'effectuent pas de fork : les processus lancés ne sont pas des fils mais se substituent au processus courant (API exec). Il n'y a pas de relations pères-fils entre les processus (getty->login->shell) mais chacun se substitue au précédent, et le processus garde le même PID. C'est ainsi que init sait quand une connexion est terminée.

Une fois la session terminée (fin du shell), init relance un terminal pour une nouvelle connexion (commande respawn).

C'est getty qui va permettre un bon fonctionnement du terminal de l'utilisateur, en s'adaptant aux divers paramètres de celui-ci (VT100, VT220, XTERM, CONSOLE...). De plus getty peut parfaitement écouter un port série et supporter une connexion modem (et lancer ensuite une session ppp par exemple).



Séquence d'ouverture de session via login

10. Arrêt

Plusieurs méthodes permettent d'arrêter proprement une machine sous Linux. Tout d'abord pour mémoire les arrêts sont aussi gérés par init avec les niveaux 0 et 6. Les deux sont en pratique quasiment identiques sauf pour la dernière action.

- Runlevel 0 : l'ordinateur est électriquement éteint.
- Runlevel 6 : l'ordinateur reboote.

C'est ainsi que la commande suivante éteint l'ordinateur :

```
# init 0
```

Et que celle-ci le reboote :

```
# init 6
```

Cependant la commande la plus correcte, la plus propre et la plus sécuritaire pour arrêter le système est **shutdown**. Shutdown appelle init, mais accepte des paramètres supplémentaires. Sa syntaxe base est `shutdown <param> <délai> <message>`

Les paramètres sont :

Paramètre	Action
`-k`	N'effectue pas le shutdown mais envoie le message à tout le monde.
`-r`	C'est un reboot.
`-h`	(halt) c'est un arrêt.
`-f`	Empêche l'exécution de fsck au boot.
`-F`	Force l'exécution de fsck au boot.
`-c`	Annule le shutdown sans délai, mais un message est possible.

Le délai peut être spécifié de différentes manières :

- **hh:mm** : une heure précise.
- **+m** : dans m minutes.
- **now** : un alias pour +0, c'est-à-dire tout de suite.

L'exemple suivant programme un reboot pour dans 10 minutes avec un message d'avertissement.

```
# shutdown -r +10 "Reboot pour maintenance dans 10 minutes"

Broadcast message from root (pts/2) (Fri Apr  4 15:00:34 2008):

Reboot pour maintenance dans 10 minutes
The system is going DOWN for reboot in 10 minutes!
```

L'exemple suivant annule le reboot.

```
# shutdown -c "Maintenance annulée"

Shutdown cancelled.

Broadcast message from root (pts/2) (Fri Apr  4 15:02:21 2008):
```

```
Maintenance annulée
```

Les commandes **reboot** et **halt** sont appelées en fin d'init 6 et 0, respectivement. Si elles sont appelées dans un autre niveau que 6 ou 0, elles sont l'équivalent d'un appel à shutdown :

- **halt** : shutdown -h
- **reboot** : shutdown -r

Consulter les traces du système

1. dmesg

La commande **dmesg** permet de récupérer les messages du noyau émis au démarrage de la machine, puis les messages émis par la suite. Le tampon de dmesg est circulaire. Au bout d'un certain nombre de messages, les premiers disparaissent. Cependant ces traces ne sont pas perdues car le service syslogd (chapitre Le réseau) écrit ces éléments dans des fichiers.

Cette commande est généralement la première lancée par un administrateur, ingénieur ou exploitant du système Linux pour vérifier la présence d'éventuelles erreurs. En effet, après le boot les messages continuent d'arriver, notamment lors de la connexion à chaud de périphériques, au chargement de certains modules, lorsque des crashs se produisent, lors d'une corruption du système de fichiers, etc.

L'exemple suivant est volontairement tronqué à une cinquantaine de lignes, la sortie originale en contenant plus de 500. Le début représente le tout début de l'exécution du noyau (informations fournies par le BIOS). Le milieu montre la détection du premier disque dur et de ses partitions. La fin montre ce qu'il se passe à l'insertion d'une clé USB, après le boot au cours d'une utilisation normale, et à sa déconnexion.

```
# dmesg
Linux version 2.6.22.17-0.1-default (geeko@buildhost) (gcc version 4.2.1
(SUSE Linux)) #1 SMP 2008/02/10 20:01:04 UTC
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
 BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000e4000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000003bfd0000 (usable)
 BIOS-e820: 000000003bfd0000 - 000000003bfde000 (ACPI data)
 BIOS-e820: 000000003bfde000 - 000000003c000000 (ACPI NVS)
 BIOS-e820: 00000000ff780000 - 0000000100000000 (reserved)
63MB HIGHMEM available.
896MB LOWMEM available.
found SMP MP-table at 000ff780
...
(quelques dizaines de lignes)
...
sd 4:0:0:0: [sda] 156301488 512-byte hardware sectors (80026 MB)
sd 4:0:0:0: [sda] Write Protect is off
sd 4:0:0:0: [sda] Mode Sense: 00 3a 00 00
sd 4:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
 sda:<6>ehci_hcd 0000:00:13.2: new USB bus registered, assigned bus
number 2
ehci_hcd 0000:00:13.2: irq 19, io mem 0xff6fc000
 sda1 sda2 < sda5 sda6 sda7 >
sd 4:0:0:0: [sda] Attached SCSI disk
...
(quelques dizaines de lignes)
...
usb-storage: device scan complete
usb 2-6: new high speed USB device using ehci_hcd and address 8
usb 2-6: new device found, idVendor=126f, idProduct=0161
usb 2-6: new device strings: Mfr=0, Product=2, SerialNumber=3
usb 2-6: Product: USB Mass Storage Device
usb 2-6: SerialNumber: 0c1738a65e944d
usb 2-6: configuration #1 chosen from 1 choice
scsi10 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 8
usb-storage: waiting for device to settle before scanning
scsi 10:0:0:0: Direct-Access     USB2.0   Mobile Disk      1.00 PQ:
0 ANSI: 2
sd 10:0:0:0: [sdc] 1007616 512-byte hardware sectors (516 MB)
sd 10:0:0:0: [sdc] Write Protect is off
sd 10:0:0:0: [sdc] Mode Sense: 00 00 00 00
sd 10:0:0:0: [sdc] Assuming drive cache: write through
sd 10:0:0:0: [sdc] 1007616 512-byte hardware sectors (516 MB)
sd 10:0:0:0: [sdc] Write Protect is off
```

```
sd 10:0:0:0: [sdc] Mode Sense: 00 00 00 00
sd 10:0:0:0: [sdc] Assuming drive cache: write through
 sdc: sdc1
sd 10:0:0:0: [sdc] Attached SCSI removable disk
sd 10:0:0:0: Attached scsi generic sg3 type 0
usb-storage: device scan complete
usb 2-5: USB disconnect, address 7
```

Pour exploiter le résultat, l'idéal est soit de rediriger celui-ci dans un fichier pour une analyse à froid, soit d'utiliser la commande **grep** à bon escient, si vous savez ce que vous cherchez.

```
# dmesg|grep CPU
Initializing CPU#0
CPU: After generic identify, caps: 078bfbff e3d3fbff 00000000 00000000
00000001 00000000 00000001
CPU: L1 I Cache: 64K (64 bytes/line), D cache 64K (64 bytes/line)
CPU: L2 Cache: 256K (64 bytes/line)
CPU: After all inits, caps: 078bfbff e3d3fbff 00000000 00000410 00000001
00000000 00000001
Intel machine check reporting enabled on CPU#0.
CPU0: AMD Sempron(tm) Processor 3200+ stepping 02
Brought up 1 CPUs
Switched to NOHz mode on CPU #0
```

2. /var/log/messages

Quelle que soit la distribution employée, **/var/log/messages** est le fichier central des messages du système, qu'ils proviennent du noyau ou des services. Le contenu de ce fichier, géré par syslog, reflète l'état global du système (et pas uniquement du noyau) au cours de son utilisation. Sur un système classique, son contenu reprend celui issu de la commande **dmesg** et celui de divers services.

Les lignes sont horodatées. En effet sans action spéciale (voir logrotate au chapitre Le réseau) le fichier grossit dans le temps et n'est pas purgé. Un fichier messages peut contenir plusieurs milliers de lignes, surtout si un problème survient !

```
# wc -l < messages
8453
```

Aussi tout comme avec la commande **dmesg**, prenez soin d'effectuer un grep pour sélectionner vos lignes (ou tail, head, etc.).

```
# tail -100 /var/log/messages | grep fglrx
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] interrupt source 20008000
successfully disabled!
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] enable ID = 0x00000000
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] Receive disable interrupt
message with irqEnableMask: 20008000; dwIRQEnableId: 00000004
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Maximum main memory to use
for locked dma buffers: 867 MBytes.
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] GART Table is not in FRAME_
BUFFER range
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Reserve Block - 0 offset =
0X0 length = 0X40000
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Reserve Block - 1 offset =
0X3ff5000 length = 0Xb000
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] interrupt source 20008000
successfully enabled
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] enable ID = 0x00000004
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Receive enable interrupt
message with irqEnableMask: 20008000
```

Services et modules noyau

1. Présentation

Le noyau est le cœur du système d'exploitation Linux. Linux en tant que tel est uniquement le nom du noyau développé à l'origine par Linus Torvalds. Le système d'exploitation Linux est composé du noyau et des outils d'exploitation de base.

Le noyau de Linux est libre. Ses sources sont disponibles. Il est donc possible de le recompiler pour l'adapter finement à ses besoins, de le modifier, d'y rajouter des extensions.

Le noyau Linux fait partie de la famille des noyaux monolithiques. C'est-à-dire que toutes ses fonctionnalités et composants sont regroupés dans un programme unique. Cependant depuis la version 2.0 (ou plutôt la version de développement 1.3 pour être plus précis) le noyau est modulaire.

Le noyau est appelé **kernel**. Il est présent dans **/boot** et son nom, par convention, commence souvent par **vmlinuz-X.Y.Z.p-Vtxt**.

On obtient la version du noyau avec la commande **uname**.

```
$ uname -r
2.4.9-e.57smp
```

Les lettres ont une signification particulière.

- **X** : version majeure du noyau. Entre la version 1 et la version 2, le passage au fonctionnement modulaire a été déterminant, ainsi que la réimplémentation de la couche réseau.
- **Y** : une valeur paire représente une branche stable du noyau. Une version impaire représente une branche de développement (attention !). Chaque incrément pair (0,2,4,6) représente une évolution importante du noyau.

La version 2.6 ne dispose pas encore de branche de développement car elle évolue trop vite. Les développeurs ont décidé d'implémenter leurs nouveautés directement dans la version stable.

- **Z** : version mineure du noyau. Quand un lot de modifications par rapport à une version précédente nécessite la diffusion d'un nouveau noyau, alors on incrémente ce chiffre. Par exemple, un lot regroupant une modification du système son (Alsa qui passe de 1.0.8 à 1.0.9), du système de fichier (ajout de ReiserFS 4), et ainsi de suite...
- **p** : version corrigée ou intermédiaire présente depuis la version 2.6. Quand le noyau nécessite une mise à jour mineure (correction d'un ou deux bugs, etc.) mais pas ou peu d'ajouts de fonctionnalités, on incrémente cette valeur.
- **V** : comme pour les packages, version propre à l'éditeur de la distribution.
- **txt** : on rajoute parfois un texte pour donner des précisions sur le noyau. Par exemple, smp indique un noyau multi-processeur.

2. uname

La commande **uname** (unix name) permet d'obtenir toutes les informations concernant la version d'Unix (de Linux ici), de manière précise et/ou complète.

Paramètre	Résultat
-m (machine)	Type matériel de la machine.
-n (nodename)	Nom d'hôte de la machine.

-r (release)	Version (numéro) du noyau.
-s (system name)	Nom du système d'exploitation. Par défaut.
-p (processor)	Type de processeur.
-i	Plate-forme matérielle.
-v (version)	Version du système.
-a (all)	Toutes les informations.

```
$ uname
Linux
$ uname -m
X86_64
$ uname -n
slyserver
$ uname -r
2.6.22.17-0.1-default
$ uname -s
Linux
$ uname -p
$ uname -i
X86_64
$ uname -o
GNU/Linux
$ uname -v
#1 SMP 2008/02/10 20:01:04 UTC
$ uname -a
Linux slyserver 2.6.22.17-0.1-default #1 SMP 2008/02/10 20:01:04 UTC
x86_64 intel x86_64 GNU/Linux
```

3. Gestion des modules

Les composants de base (scheduler, gestion de la mémoire, des processus, API, etc.) sont toujours présents au sein d'un programme unique. Mais certains pilotes de périphériques, systèmes de fichiers, extensions, protocoles réseaux, etc. peuvent être présents sous forme de modules. Les modules communiquent avec le noyau via une API commune. Ils s'exécutent dans l'espace du noyau. Ils sont paramétrables. Ils peuvent être chargés et déchargés à la demande évitant ainsi un redémarrage de la machine. L'ajout d'un nouveau module (depuis ses sources par exemple) ne nécessite pas de redémarrage.

Les modules sont présents dans **/lib/modules/$(uname -r)**.

```
# cd /lib/modules/$(uname -r)
# pwd
/lib/modules/2.6.22.17-0.1-default
```

Les modules ont un nom qui finit par « .ko » pour kernel object. La terminaison d'origine était « .o » pour les noyaux 2.0 à 2.4. Il s'agit bien de cela : des fichiers objets dynamiquement liés (linked) à leur chargement au noyau, proposant ainsi une API supplémentaire.

```
# cd /lib/modules/$(uname -r)/kernel/fs/vfat
# file vfat.ko
vfat.ko: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV),
not stripped
```

Le mot « relocatable » indique que vous êtes en présence d'un fichier objet.

a. lsmod

La commande **lsmod** liste les modules actuellement chargés, avec leurs dépendances éventuelles.

```
# lsmod
Module                  Size  Used by
fglrx                1482380  70
nls_iso8859_1           8192  1
nls_cp437               9856  1
vfat                   16128  1
fat                    52636  1 vfat
snd_pcm_oss            50432  0
snd_mixer_oss          20096  1 snd_pcm_oss
snd_seq                54452  0
snd_seq_device         12172  1 snd_seq
iptable_filter          6912  0
ip_tables              16324  1 iptable_filter
ip6_tables             17476  0
x_tables               18308  2 ip_tables,ip6_tables
cpufreq_conservative    11272  0
cpufreq_userspace       8704  0
...
```

La première colonne indique le nom du module chargé. Son nom reflète bien souvent ce à quoi il sert. La seconde colonne donne la taille du module. La troisième colonne fournit un compteur d'utilisation (combien de composants du système accèdent aux modules). La dernière colonne fournit la liste des modules utilisant (donc dépendant) du premier.

Dans l'exemple précédent le module fat est utilisé par le module vfat.

lsmod ne fait que remettre en forme le contenu du fichier virtuel **`/proc/modules`**.

```
# cat /proc/modules
fglrx 1482380 60 - Live 0xf9a55000 (P)
nls_iso8859_1 8192 0 - Live 0xf94d0000
nls_cp437 9856 0 - Live 0xf94ed000
snd_pcm_oss 50432 0 - Live 0xf9887000
snd_mixer_oss 20096 1 snd_pcm_oss, Live 0xf94d3000
snd_seq 54452 0 - Live 0xf94de000
snd_seq_device 12172 1 snd_seq, Live 0xf94b9000
...
```

b. depmod

La commande **depmod** met à jour l'arbre des dépendances entre les modules en modifiant le fichier **`modules.dep`**.

Le fichier **`/lib/modules/$(uname -r)/modules.dep`** contient deux colonnes. La première est le chemin du module, la seconde est la liste des dépendances : les modules qui doivent aussi être chargés pour que le premier fonctionne. Voici l'exemple de la ligne correspondant au module vfat :

```
# grep vfat modules.dep
/lib/modules/2.6.22.17-0.1-default/kernel/fs/vfat/vfat.ko:
/lib/modules/2.6.22.17-0.1-default/kernel/fs/fat/fat.ko
```

Le module vfat.ko dépend du module fat.ko. Il faut donc que le module fat.ko soit chargé en premier pour que vfat.ko fonctionne.

L'usage le plus courant de depmod est avec le paramètre `-a` qui reconstruit les dépendances de tous les modules correspondant au noyau actuel. Cette action est exécutée à chaque démarrage du système, mais si vous compilez et/ou installez de nouveaux modules, vous devez relancer à la main cette commande pour prendre en compte les nouvelles dépendances.

```
# depmod -a
```

c. modinfo

La commande **modinfo** fournit toutes les informations nécessaires sur un module :

- le nom du fichier correspondant,

- une description du module,
- son auteur,
- sa licence,
- ses dépendances,
- ses paramètres,
- ses alias matériels.

Les modules ne fournissent pas tous ces informations. Certains modules n'ont pas de paramètres. Dans le premier exemple, le module vfat n'a pas de paramètres, ceux-ci étant mis en place comme options de montage.

```
# modinfo vfat
filename:       /lib/modules/2.6.22.17-0.1-default/kernel/fs/fat/fat.ko
license:        GPL
srcversion:     886B1B65F96E53415B5811C
depends:
supported:      yes
vermagic:       2.6.22.17-0.1-default SMP mod_unload 586
```

Dans ce deuxième exemple, le module dispose de paramètres. Ce module permet d'utiliser une Webcam contenant une puce ov511. Les paramètres sont volontairement tronqués.

```
# modinfo ov511
filename:       /lib/modules/2.6.22.17-0.1-
default/kernel/drivers/media/video/ov511.ko
license:        GPL
description:    ov511 USB Camera Driver
author:         Mark McClelland <mark@alpha.dyndns.org> & Bret Wallach
& Orion Sky Lawlor <olawlor@acm.org> & Kevin Moore & Charl P. Botha
<cpbotha@ieee.org> & Claudio Matsuoka <claudio@conectiva.com>
srcversion:     E0DC673BFADEA96F2DED84F
alias:          usb:v0813p0002d*dc*dsc*dp*ic*isc*ip*
alias:          usb:v05A9pA518d*dc*dsc*dp*ic*isc*ip*
alias:          usb:v05A9p0518d*dc*dsc*dp*ic*isc*ip*
alias:          usb:v05A9pA511d*dc*dsc*dp*ic*isc*ip*
alias:          usb:v05A9p0511d*dc*dsc*dp*ic*isc*ip*
depends:        compat_ioctl32,videodev,usbcore
supported:      yes
vermagic:       2.6.22.17-0.1-default SMP mod_unload 586
parm:           autobright:Sensor automatically changes brightness (int)
parm:           autogain:Sensor automatically changes gain (int)
parm:           autoexp:Sensor automatically changes exposure (int)
parm:           debug:Debug level: 0=none, 1=inits, 2=warning, 3=config,
4=functions, 5=max (int)
parm:           snapshot:Enable snapshot mode (int)
parm:           cams:Number of simultaneous cameras (int)
parm:           compress:Turn on compression (int)
parm:           led:LED policy (OV511+ or later). 0=off, 1=on (default),
2=auto (on when open) (int)
```

Un même module peut gérer plusieurs types de matériel. Il existe plusieurs Webcams disposant d'une puce ov511 ou affiliée pouvant être gérée par le module ov511. Pour que le noyau sache quel module charger lors de la détection de la webcam, ou pour que le module sache quel type de matériel il doit gérer lors de son chargement, il dispose d'alias matériels, usb, pci, scsi, etc., permettant de reconnaître les périphériques qu'il doit gérer.

Les paramètres sont passés au module via les commandes **insmod** ou **modprobe**, ou à l'aide du fichier **`/etc/modprobe.conf`**.

d. insmod

La commande **insmod** charge un module donné sans gérer les dépendances. Elle prend en paramètre un nom de module, avec son éventuel chemin. C'est à vous de gérer l'ordre de chargement des modules pour éviter des

erreurs liées à des symboles non résolus causées par un problème de dépendance.

Dans l'exemple suivant, les modules fat et vfat n'étant pas chargés, une tentative a lieu pour charger le module vfat.ko seul. Une erreur se produit car ce module dépend de la présence de fat.ko. Le retour de dmesg est éloquent à ce sujet.

```
# lsmod|grep fat
# ls
vfat.ko
# insmod vfat.ko
insmod: error inserting 'vfat.ko': -1 Unknown symbol in module
# dmesg | tail -20
...
vfat: Unknown symbol fat_dir_empty
vfat: Unknown symbol fat_fs_panic
vfat: Unknown symbol fat_get_dotdot_entry
vfat: Unknown symbol fat_free_clusters
vfat: Unknown symbol fat_scan
vfat: Unknown symbol fat_date_unix2dos
vfat: Unknown symbol fat_search_long
vfat: Unknown symbol fat_getattr
vfat: Unknown symbol fat_attach
vfat: Unknown symbol fat_build_inode
vfat: Unknown symbol fat_fill_super
vfat: Unknown symbol fat_alloc_new_dir
vfat: Unknown symbol fat_notify_change
vfat: Unknown symbol fat_remove_entries
vfat: Unknown symbol fat_add_entries
vfat: Unknown symbol fat_sync_inode
vfat: Unknown symbol fat_detach
```

Dans ce second exemple, le module fat est tout d'abord chargé, puis le module vfat. Il n'y a pas d'erreur car toutes les dépendances sont présentes.

```
# cd ../fat
# ls
fat.ko
# insmod fat.ko
# cd ../vfat
# insmod vfat.ko
# lsmod|grep fat
vfat                   16128  0
fat                    52636  1 vfat
```

Vous devriez envisager le paramètre `-k` qui permet l'autoclean des modules. Cette option indique au système de décharger automatiquement un module (compteur à zéro) s'il n'est plus utilisé, permettant de gagner quelques ressources et d'obtenir un système plus propre.

Pour transmettre des paramètres au module, indiquez ceux-ci à la suite de la commande. Par exemple pour le module ov511, vous souhaitez que la led soit active uniquement lorsque la webcam est active et que la compression est activée (pour atteindre les 25 fps annoncés) :

```
# insmod ov511.ko led=2 compress=1
```

e. rmmod

La commande **rmmod** décharge le module donné. C'est l'inverse de insmod et, comme cette dernière, rmmod ne gère pas les dépendances :

- Il n'est pas possible de décharger un module en cours d'utilisation.
- Il n'est pas possible de décharger un module s'il est utilisé par un autre module, même si ce dernier n'est pas utilisé (problème de dépendance).

Dans cet exemple, une clé USB contenant un système de fichiers vfat est branchée et montée. Une première tentative de supprimer vfat échoue.

```
# mount | grep vfat
```

```
/dev/sdb1 on /media/disk type vfat  ...
# rmmod vfat
ERROR: Module vfat is in use
```

Dans ce deuxième exemple, la clé est débranchée. Les modules fat et vfat sont devenus inutiles. On tente de supprimer le module fat. Le système retourne une erreur liée aux dépendances.

```
# rmmod fat
ERROR: Module fat is in use by vfat
```

Dans ce dernier exemple, le module vfat est déchargé, puis le module fat, dans cet ordre.

```
# rmmod vfat
# rmmod fat
```

f. modprobe

La commande **modprobe** charge le module donné ainsi que toutes ses dépendances et des paramètres contenus dans `/etc/modprobe.conf`. Le paramètre `-r` permet de décharger un modules et ceux qui en dépendent (s'ils ne sont pas utilisés).

Le chargement du module vfat à l'aide de modprobe va automatiquement charger le module fat.

```
# lsmod|grep fat
# modprobe vfat
# lsmod|grep fat
vfat                   16128  0
fat                    52636  1 vfat
```

Maintenant, voyant que seul vfat utilise fat (compteur à 1) mais que rien n'utilise vfat (compteur à zéro), vous pouvez tenter de décharger vfat et les modules dont il dépend s'ils ne sont plus utilisés.

```
# modprobe -r vfat
# lsmod|grep fat
```

Vous pouvez passer des options au module, dans ce cas vous devez vérifier si une ligne « install » correspondant au module existe dans le fichier modprobe.conf et la modifier pour qu'elle accepte les paramètres, ou utiliser une ligne « options ».

g. modprobe.conf

La configuration des modules est placée dans **`/etc/modprobe.conf`**. Dans ce fichier vous pouvez définir des alias de modules (très pratique pour les cartes réseau), passer des options aux modules, ajouter des actions au chargement et déchargement d'un module.

Alias et options

Exemple : vous avez deux cartes réseaux. Le pilote de la première carte est contenu dans le module **e1000**. Vous voulez qu'il soit chargé en utilisant le nom **eth0**. Le pilote de la seconde carte est dans le module **airo** et vous souhaitez le charger en utilisant le nom **eth1**. Vous allez aussi spécifier des paramètres au module ov511 dont vous avez récupéré les informations avec modinfo.

```
# cat /etc/modprobe.conf
...
alias eth0 e1000
alias eth1 airo
options ov511 led=2 compress=1
...
```

Ensuite vous chargez les modules soit grâce à leur nom, soit grâce à leur alias. Dans tous les cas, les paramètres sont automatiquement pris en compte.

```
# modprobe eth0
# modprobe eth1
# modprobe ov511
```

Les options des modules peuvent aussi être utilisés avec les alias.

```
alias webcam ov511
options webcam led=2 compress=1
...
```

install et remove

Les commandes **install** et **remove** du fichier **modprobe.conf** sont les plus puissantes. Lors du chargement d'un module, si modprobe trouve une commande install associée, il ne charge pas le module mais exécute les commandes indiquées. La commande peut être ce que vous voulez, comme par exemple le chargement d'un module plus adapté, la mise en place d'une configuration spécifique, l'exécution d'un script, etc.

L'exemple suivant tente de charger le module ov511_new s'il existe, sinon il bascule sur ov511, lorsque vous invoquez modprobe webcam.

```
install webcam /sbin/modprobe ov511_new || /sbin/modprobe ov511
```

modprobe peut prendre un paramètre pratique lui permettant d'ignorer les lignes install de modprobe.conf : **--ignore-install**. L'exemple suivant charge le module ahci puis le module ata_piix lorsque vous tentez de charger ce dernier. Sans le paramètre, modprobe tournerait en boucle, relisant et interprétant la ligne install à chaque tentative de chargement de ata_piix.

```
install ata_piix /sbin/modprobe ahci 2>&1 |:; /sbin/modprobe --
ignore-install ata_piix
```

La commande **remove** fait la même chose, mais au déchargement du module avec **modprobe -r**.

CMDLINE_OPTS

Si une ligne install est présente dans modprobe.conf et que vous tentez de charger le module correspondant avec modprobe et des paramètres, ces derniers ne seront pas pris en compte sauf si vous avez rajouté derrière le module la chaîne $CMDLINE_OPTS.

```
install webcam /sbin/modprobe ov511_new $CMDLINE_OPTS ||
/sbin/modprobe ov511 $CMDLINE_OPTS
```

4. Chargement des modules au boot

a. initrd

Certains modules peuvent être nécessaires au démarrage de la machine, notamment pour monter un système de fichiers. Comment monter la partition racine en ext3 alors que le module gérant ce type de système de fichiers est sur cette partition (et pas en dur dans le noyau) ? Ces modules sont placés dans une image de disque mémoire initiale ou **initrd** (initial ramdisk). Ces fichiers compressés sont chargés au démarrage en mémoire et sont des ramdisks. Ils contiennent des instructions et modules qui sont chargés au démarrage.

```
$ ls -l /boot/initrd*
-rw-r--r-- 1 root root 4151529 fév 13 10:48 /boot/initrd-2.6.22.17-0.1-
default
file /boot/initrd-2.6.22.17-0.1-default
/boot/initrd-2.6.22.17-0.1-default: gzip compressed data, from Unix,
last modified: Wed Feb 13 10:48:34 2008, max compression
```

Il est possible d'extraire le contenu de initrd et même de le modifier et le reconstruire pour l'adapter. Ce fichier est une archive cpio compressée avec gzip.

```
# zcat /boot/initrd-2.6.22.17-0.1-default | cpio -id --no-absolute-
filenames
17588 blocks
p64p17bicb3:/home/seb/initrd # ls -l
total 196
drwxr-xr-x 2 root root   4096 avr  2 12:22 bin
drwxr-xr-x 2 root root   4096 avr  2 12:22 boot
-rw-r--r-- 1 root root 133490 avr  2 12:22 bootsplash
```

```
drwxr-xr-x 2 root root   4096 avr  2 12:22 config
drwxr-xr-x 2 root root   4096 avr  2 12:22 dev
drwxr-xr-x 6 root root   4096 avr  2 12:22 etc
-rwxr-xr-x 1 root root   2067 avr  2 12:22 init
drwxr-xr-x 5 root root   4096 avr  2 12:22 lib
drwxr-xr-x 2 root root   4096 avr  2 12:22 proc
drwxr-xr-x 2 root root   4096 avr  2 12:22 root
-rw-r--r-- 1 root root   3641 avr  2 12:22 run_all.sh
drwxr-xr-x 2 root root   4096 avr  2 12:22 sbin
drwxr-xr-x 2 root root   4096 avr  2 12:22 sys
drwsrwxrwx 2 root root   4096 avr  2 12:22 tmp
drwxr-xr-x 3 root root   4096 avr  2 12:22 usr
drwxr-xr-x 5 root root   4096 avr  2 12:22 var
```

Quand initrd est chargé et monté, le noyau tente d'exécuter le script init présent à la racine du pseudo système de fichiers. C'est lui qui doit charger les modules nécessaires et établir la toute première configuration de base, avant que le noyau ne monte le système de fichier racine et exécute /sbin/init. Dans la pratique, c'est ce fichier (ou l'un de ceux qu'il appelle) qui monte le système de fichiers racine et qui passe la main à /sbin/init.

Sur certaines distributions, notamment anciennes, c'est un script appelé /sbin/init qui était appelé. Ceci a été supprimé pour éviter de confondre le script /sbin/init de initrd avec le « vrai » init du système. Si un fichier **linuxrc** est présent à la racine de l'initrd, il est exécuté. En cas de doute sur le nom du script lancé, celui-ci est dans les sources du noyau, init/main.c, variable ram_execute_command.

```
$ cd /usr/src/linux
$ grep -n ram_execute_commande init/main.c
...
865:            ramdisk_execute_command = "/init";
...
```

La commande **mkinitrd** permet de reconstruire un fichier initrd standard, mais il est bien souvent possible, selon la distribution, de lui fournir une liste de modules à y placer et donc à charger :

- Sous Red Hat, c'est l'option **--preload=module**, qui peut être utilisée plusieurs fois, qui précise quels sont les modules à charger.
- Sous openSUSE, c'est l'option **-m "module1 module2 etc.**" qui précise cette liste. Voyez cependant la méthode utilisant les variables de sysconfig.
- Sous Debian, il faut placer les noms des modules dans le fichier **/etc/mkinitrd/modules**.

Voici le résultat d'une commande mkinitrd sous openSUSE. La ligne « Kernel Modules » précise les modules chargés par initrd.

```
# mkinitrd

Kernel image:   /boot/vmlinuz-2.6.22.17-0.1-default
Initrd image:   /boot/initrd-2.6.22.17-0.1-default
Root device:    /dev/disk/by-id/scsi-SATA_ST380011A_5JVTH798-part6
(/dev/sda6) (mounted on / as ext3)
Resume device:  /dev/sda5
Kernel Modules: processor thermal scsi_mod libata sata_sil
pata_atiixp fan jbd mbcache ext3 edd sd_mod usbcore ohci-hcd uhci-hcd
ehci-hcd ff-memless hid usbhid
Features:       block usb resume.userspace resume.kernel
Bootsplash:     SuSE (1280x1024)
17588 blocks
```

Un initrd bien construit (ou plutôt une commande **mkinitrd** correcte) recopie aussi la commande **modprobe** et le fichier de configuration **modprobe.conf** qui peut (et doit) contenir les paramètres des modules si besoin.

b. Red Hat /etc/rc.modules

La méthode préférée sous Red Hat est de charger les modules depuis l'initrd. Cependant vous avez la possibilité de créer un script **/etc/rc.modules** qui contiendra les commandes nécessaires au chargement des modules,

principalement avec modprobe comme vu précédemment. Ce fichier est exécuté par rc.sysinit au démarrage du système, via l'inittab. Il doit être rendu exécutable.

c. openSUSE : /etc/sysconfig/kernel

La distribution openSUSE place sa configuration dans une arborescence **/etc/sysconfig**. Le fichier **/etc/sysconfig/kernel** contient quelques éléments de configuration du noyau mais surtout deux variables chargées de spécifier les modules à charger :

- **INITRD_MODULES** est la liste des modules chargés par l'initrd.
- **MODULES_LOADED_ON_BOOT** est la liste des modules chargé par init (donc après initrd).

Vous verrez une différence entre la liste des modules indiquée pour l'initrd et la liste réelle chargée lors de la création de celui-ci. La commande **mkinitrd** de openSUSE permet de spécifier des fonctionnalités préétablies lors de la création de l'initrd (option **-f "feature1 feature2 etc"**). La fonctionnalité « usb » va par exemple charger tous les modules liés au support USB, ce qui est vital si vous disposez d'un clavier USB ou si vous bootez depuis une clé ou un disque externe.

d. Debian : /etc/modules

Sous Debian il suffit de rajouter les noms des modules à charger dans le fichier **/etc/modules**. Après le nom des modules vous pouvez indiquer des paramètres. Pour chaque ligne un modprobe est exécuté. Vous pouvez rajouter un commentaire après un #. C'est le fichier **/etc/init.d/modutils** qui charge les modules au démarrage via init.

```
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line.  Comments begin with
# a "#", and everything on the line after them are ignored.

ide-cd
ide-detect
ov511
```

5. Paramètres dynamiques

a. /proc et /sys

/proc et **/sys** sont des systèmes de fichiers virtuels contenant des informations sur le noyau en cours d'exécution. La version 2.4 du noyau ne connaît que **/proc** où toutes les informations sont regroupées. La version 2.6 du noyau a modifié la fonction de **/proc** pour déléguer une partie de ses tâches à **/sys**.

S'agissant de systèmes de fichiers virtuels, ils ne prennent aucune place ni en mémoire, ni sur un disque quelconque. Il ne faut pas se laisser berner par la taille des pseudo-fichiers contenus dedans. Ne tentez pas de supprimer **/proc/kcore** pour gagner de la place ! Tous ces fichiers (ou presque) peuvent être lus et affichés directement.

Les fichiers de **/proc** vous donneront énormément d'informations sur le système :

- **interrupts** : les paramètres IRQ.
- **cpuinfo** : détails sur vos processeurs.
- **dma** : les paramètres DMA.
- **ioports** : les ports mémoires E/S.
- **devices** : les périphériques présents.

- **meminfo** : l'état global de la mémoire.
- **loadavg** : la charge du système.
- **uptime** : uptime du système, attente.
- **version** : détails de la version de Linux.
- **modules** : identique au résultat de lsmod.
- **swaps** : liste et état des partitions d'échange.
- **partitions** : liste et état des partitions connues du système.
- **mounts** : montages des systèmes de fichiers.
- **pci** : détails du bus PCI.

```
$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 8
model name      : Pentium III (Coppermine)
stepping        : 6
cpu MHz         : 996.895
cache size      : 256 KB
physical id     : 1360587528
siblings        : 1
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 mmx fxsr sse
bogomips        : 1985.74

$ cat /proc/version
Linux version 2.4.9-e.57smp (bhcompile@tweety.build.Red Hat.com) (gcc
version 2.96 20000731 (Red Hat Linux 7.2 2.96-129.7.2)) #1 SMP Thu
Dec 2 20:51:12 EST 2004
```

/proc contient des sous-répertoires qui regroupent des informations par thème.

- **/proc/scsi** : informations sur le bus SCSI.
- **/proc/ide** : informations sur le bus IDE.
- **/proc/net** : informations sur le réseau.
- **/proc/sys** : paramètres et configuration dynamique du noyau.
- **/proc/<PID>** : informations sur le processus PID.

Certaines entrées des systèmes de fichiers **/proc/sys** et **/sys** sont différentes des autres car leur contenu peut être modifié et les modifications sont prises en compte directement par le noyau sans avoir à redémarrer la machine.

Par exemple voici comment activer le forwarding IP et passer le nombre de handles de fichiers de 8192 à 16384.

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
# echo "16384" > /proc/sys/fs/file-max
```

b. sysctl

Les valeurs modifiées ne sont pas enregistrées. En cas de redémarrage il faut recommencer. Le fichier **rc.sysinit** appelle la commande **sysctl** qui agit sur ces paramètres. Pour que les valeurs restent permanentes (remises en place à chaque démarrage) il faut modifier le fichier **/etc/sysctl.conf**. Vous pouvez rechercher les modifications manuellement.

```
# sysctl -e -p /etc/sysctl.conf

# sysctl -a
...
dev.raid.speed_limit_max = 100000
dev.raid.speed_limit_min = 100
net.token-ring.rif_timeout = 60000
net.ipv4.conf.eth1.arp_filter = 0
net.ipv4.conf.eth1.tag = 0
net.ipv4.conf.eth1.log_martians = 0
net.ipv4.conf.eth1.bootp_relay = 0
net.ipv4.conf.eth1.proxy_arp = 0
net.ipv4.conf.eth1.accept_source_route = 1
net.ipv4.conf.eth1.send_redirects = 1
net.ipv4.conf.eth1.rp_filter = 1
net.ipv4.conf.eth1.shared_media = 1
net.ipv4.conf.eth1.secure_redirects = 1
...
```

Compiler un noyau

1. Obtenir les sources

a. Sources officielles

Les sources officielles du noyau sont disponibles depuis le site kernel.org. Elles portent le nom de vanilla. Un noyau (ou kernel) vanilla est un noyau brut, sans ajout de patches, issu directement des développements des contributeurs du noyau, et n'a pas été adapté à une quelconque distribution.

Ce sont les sources du noyau qui sont fournies. Vous devez configurer, compiler et installer un noyau, et éventuellement créer un initrd avant de pouvoir l'utiliser.

Le noyau est fourni sous forme d'archive compressée que vous devez ouvrir avec les outils adaptés.

```
$ ls
linux-2.6.24.4.tar.bz2
$ tar xvjf linux-2.6.24.4.tar.bz2
linux-2.6.24.4/
linux-2.6.24.4/.gitignore
linux-2.6.24.4/.mailmap
linux-2.6.24.4/COPYING
linux-2.6.24.4/CREDITS
linux-2.6.24.4/Documentation/
linux-2.6.24.4/Documentation/00-INDEX
linux-2.6.24.4/Documentation/ABI/
linux-2.6.24.4/Documentation/ABI/README
linux-2.6.24.4/Documentation/ABI/obsolete/
linux-2.6.24.4/Documentation/ABI/obsolete/dv1394
linux-2.6.24.4/Documentation/ABI/removed/
linux-2.6.24.4/Documentation/ABI/removed/devfs
linux-2.6.24.4/Documentation/ABI/removed/raw1394_legacy_isochronous
linux-2.6.24.4/Documentation/ABI/stable/
linux-2.6.24.4/Documentation/ABI/stable/syscalls
linux-2.6.24.4/Documentation/ABI/stable/sysfs-module
linux-2.6.24.4/Documentation/ABI/testing/
linux-2.6.24.4/Documentation/ABI/testing/debugfs-pktcdvd
linux-2.6.24.4/Documentation/ABI/testing/sysfs-bus-usb
linux-2.6.24.4/Documentation/ABI/testing/sysfs-class
linux-2.6.24.4/Documentation/ABI/testing/sysfs-class-pktcdvd
linux-2.6.24.4/Documentation/ABI/testing/sysfs-devices
linux-2.6.24.4/Documentation/ABI/testing/sysfs-kernel-uids
linux-2.6.24.4/Documentation/ABI/testing/sysfs-power
...
```

Les sources du noyau sont placées dans **/usr/src/linux**. Si plusieurs versions des sources de noyaux sont présentes, Linux est un lien symbolique vers la source du noyau courant.

```
# ls -l
lrwxrwxrwx  1 root root        19 fév 13 10:54 linux -> linux-2.6.22.17-0.1
drwxr-xr-x  4 root root      4096 fév 13 10:54 linux-2.6.22.16-0.2
drwxr-xr-x 20 root root      4096 avr  7 10:57 linux-2.6.22.17-0.1
drwxr-xr-x  3 root root      4096 fév 13 10:52 linux-2.6.22.17-0.1-obj
drwxrwxr-x 20 root root      4096 mar 24 19:49 linux-2.6.24.4
drwxr-xr-x  6 root root      4096 fév  1 14:37 linux-2.6.24-g13f09b95-15
lrwxrwxrwx  1 root root        23 fév 13 10:54 linux-obj -> linux-2.6.22
.17-0.1-obj
```

b. Sources de la distribution

Chaque distribution est fournie avec un noyau bien souvent patché. Ces modifications peuvent revêtir plusieurs aspects : ajout de pilotes, backports (mises à jour rétroactives, rajout de fonctionnalités issues d'une version ultérieure) de noyaux plus récents, correctifs de sécurité, rajout de fonctionnalités, etc. Elles sont appliquées sur un noyau vanilla, et souvent dans un ordre précis.

Compiler un noyau n'est pas anodin. Si vous disposez d'un contrat de support avec l'éditeur de la distribution (par exemple pour une version Serveur de Red Hat), et si vous rencontrez des problèmes avec le noyau par défaut, vous devriez plutôt envisager de :

- vérifier si une mise à jour officielle corrige votre problème ;
- remonter votre problème à la hotline de l'éditeur.

Avant de penser à installer un noyau recompilé par vos soins.

Si vous souhaitez tout de même recompiler le noyau, la démarche est la même que pour les sources officielles. Seulement vous devez installer le package des sources, ce qui devrait avoir pour effet d'installer en même temps tous les outils nécessaires à la compilation.

Le package s'appelle kernel-sources sur openSUSE et Red Hat. Pour ce dernier, rendez-vous sur le site de Red Hat car le package kernel-sources n'est pas nécessaire pour compiler des nouveaux modules (il faut le package kernel-devel) et il n'est pas très simple de les obtenir. Sous Debian le package se nomme linux-source-2.x.y (x et y représentant la version du noyau). Vous devez installer ces packages avec les outils adaptés à votre distribution.

2. Les outils nécessaires

Pour compiler le noyau Linux, il est nécessaire de disposer de certains outils :

- le compilateur C ;
- les bibliothèques de développement C standard ;
- la bibliothèque ncurses (pour menuconfig) ;
- la bibliothèque qt3 (pour xconfig) ;
- les outils Make ;
- les modutils ;
- mkinitrd ;
- de la concentration ;
- des nerfs solides ;
- de la patience ;

Ces outils sont fournis en standard avec la distribution, sauf les trois derniers...

3. Configuration

a. Le .config

Avant de lancer la compilation du noyau, vous devez sélectionner les options, fonctionnalités et pilotes à conserver ou non. Ceci se fait par divers moyens exposés par la suite. Cette configuration est ensuite sauvée dans un fichier **`/usr/src/linux/.config`**. Ce fichier contient un grand nombre de variables, chacune d'entre elles correspondant à une option du noyau, et pouvant prendre trois valeurs :

- **y** : la fonctionnalité est présente et intégrée au sein du noyau monolithique, ou, si elle dépend d'un module, intégrée au sein de ce module ;

- **m** : la fonctionnalité sera compilée sous forme de module ;
- **n** : la fonctionnalité est absente.

Dans le dernier cas, cette valeur est rarement présente. Il suffit que l'option soit absente du fichier pour qu'elle ne soit pas activée lors de la compilation. Dans ce cas la ligne correspondante est simplement commentée avec un # devant.

L'exemple suivant montre que l'option MCORE2 (optimisation pour les Intel Core2) n'est pas active.

```
$ grep "^#" .config | grep -i core2
# CONFIG_MCORE2 is not set
```

L'exemple suivant montre les lignes de configuration associées aux fonctionnalités du système de fichiers ext3. Le support du système de fichier ext3 est compilé sous forme de module (première ligne). Les fonctionnalités supplémentaires de ext3 (attributs étendus, ACL, sécurité, attributs nfs4) dépendent de la première ligne, elles seront donc présentes au sein du module et non dans le noyau, ou comme module à part.

```
# grep -i ext3 .config
CONFIG_EXT3_FS=m
CONFIG_EXT3_FS_XATTR=y
CONFIG_EXT3_FS_POSIX_ACL=y
CONFIG_EXT3_FS_NFS4ACL=y
CONFIG_EXT3_FS_SECURITY=y
```

Certaines options dépendent d'autres. Sauf si vous savez exactement ce que vous faites, vous ne devriez pas éditer le fichier **.config** à la main pour y effectuer des modifications, au risque de casser des dépendances lors de la compilation. Le plus simple reste de passer par les étapes de configuration exposées par la suite.

Évitez de modifier **.config** à la main, passez plutôt par les outils fournis depuis les sources ou par votre distribution.

b. Récupérer la configuration du noyau

La configuration actuelle du noyau peut être accessible depuis plusieurs endroits. Si un noyau (ou ses sources) provient d'un package de la distribution, il est probable que le fichier .config soit déjà présent au sein de **/usr/src/linux**, ou ailleurs, auquel cas vous vous rapporterez à la documentation officielle.

Sous openSUSE par exemple le répertoire **/boot** contient une copie du .config ayant servi à la compilation du noyau.

```
# ls /boot/config*
/boot/config-2.6.22.17-0.1-default
```

Dans ce cas vous pouvez réutiliser cette configuration à la main :

```
# cp /boot/config-2.6.22.17-0.1-default /usr/src/linux/.config
```

Les noyaux sont souvent configurés avec deux options intéressantes.

```
# grep -i KCONF .config
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
```

La première permet de placer le contenu du .config dans le noyau lui-même lors de la compilation. La seconde permet d'accéder à cette configuration depuis le système de fichiers virtuel **/proc** via le fichier **/proc/config.gz**. Ce pseudo-fichier est compressé au format gzip. Pour le lire, utilisez la commande **zcat**.

```
# zcat /proc/config.gz > /usr/src/linux/.config
```

c. make oldconfig

La méthode précédente présente quelques inconvénients notamment lorsqu'il s'agit de récupérer la configuration d'un noyau plus récent ou plus ancien :

- des fonctionnalités peuvent avoir disparu ;
- d'autres ont pu être ajoutées.

Dans ce cas, partir d'un fichier de configuration inadapté peut avoir des conséquences néfastes. Pour éviter tout problème, le mieux est d'utiliser une possibilité offerte par les sources du noyau (ou plutôt du Makefile) : récupérer l'ancienne configuration, l'analyser, indiquer les changements survenus et demander quoi faire. Ceci se fait via la commande **make oldconfig**.

Dans l'exemple suivant, la configuration d'un noyau 2.6.22 d'origine openSUSE est reprise pour un noyau vanilla 2.6.24. De nombreux avertissements apparaissent (volontairement tronqués) car le noyau vanilla ne contient pas certaines options issues de patches spécifiques au noyau de la distribution. Ensuite (lignes en gras), le nouveau noyau dispose de nouvelles possibilités absentes à l'origine. Vous devez alors répondre à chacune des questions ce qui est assez fastidieux.

```
# make oldconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/basic/docproc
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/kxgettext.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/lex.zconf.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf -o arch/x86/Kconfig
#
# using defaults found in /boot/config-2.6.22.17-0.1-default
#
/boot/config-2.6.22.17-0.1-default:36:warning: trying to assign
nonexistent symbol SUSE_KERNEL
/boot/config-2.6.22.17-0.1-default:39:warning: trying to assign
nonexistent symbol IPC_NS
/boot/config-2.6.22.17-0.1-default:47:warning: trying to assign
nonexistent symbol UTS_NS
/boot/config-2.6.22.17-0.1-default:125:warning: trying to assign
nonexistent symbol X86_XEN
/boot/config-2.6.22.17-0.1-default:177:warning: trying to assign
nonexistent symbol X86_MINIMUM_CPU_MODEL
/boot/config-2.6.22.17-0.1-default:252:warning: trying to assign
nonexistent symbol PM_SYSFS_DEPRECATED
/boot/config-2.6.22.17-0.1-default:253:warning: trying to assign
nonexistent symbol SOFTWARE_SUSPEND
...
/boot/config-2.6.22.17-0.1-default:3814:warning: trying to assign
nonexistent symbol KDB
/boot/config-2.6.22.17-0.1-default:3824:warning: symbol value 'm'
invalid for SECURITY_CAPABILITIES
/boot/config-2.6.22.17-0.1-default:3825:warning: symbol value 'm'
invalid for SECURITY_ROOTPLUG
/boot/config-2.6.22.17-0.1-default:3827:warning: trying to assign
nonexistent symbol SECURITY_APPARMOR
*
* Linux Kernel Configuration
*
*
* General setup
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL)
[Y/n/?] y
Local version - append to kernel release (LOCALVERSION) [-default]
-default
Automatically append version information to the version string
(LOCALVERSION_AUTO) [N/y/?] n
Support for paging of anonymous memory (swap) (SWAP) [Y/n/?] y
System V IPC (SYSVIPC) [Y/n/?] y
POSIX Message Queues (POSIX_MQUEUE) [Y/n/?] y
BSD Process Accounting (BSD_PROCESS_ACCT) [Y/n/?] y
  BSD Process Accounting version 3 file format (BSD_PROCESS_ACCT_V3)
```

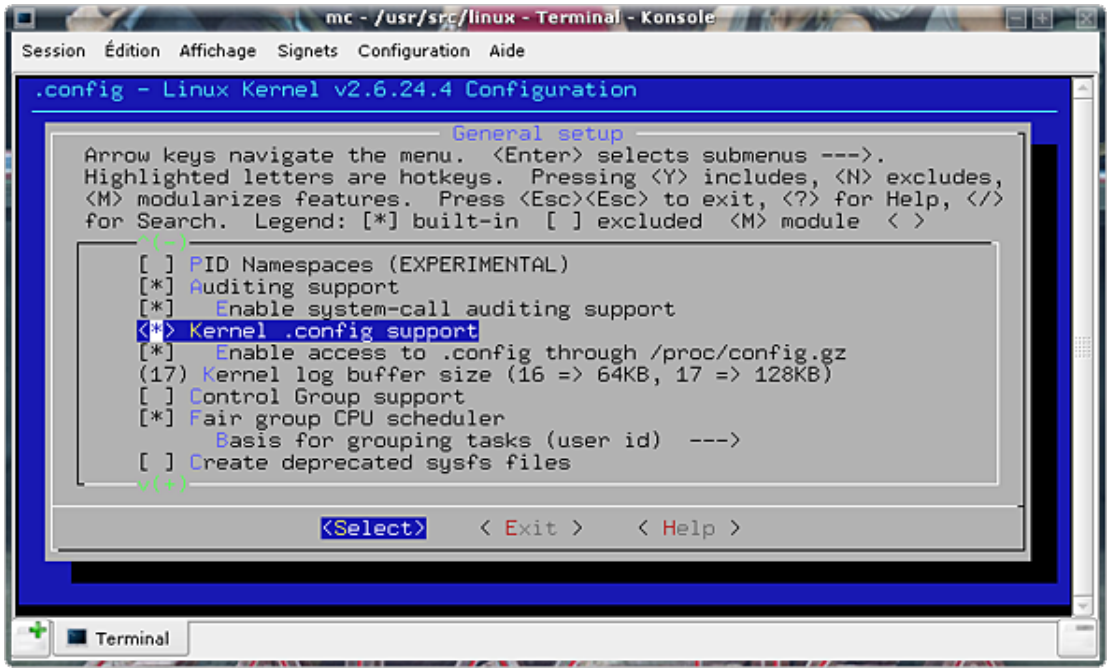
```
[Y/n/?] y
Export task/process statistics through netlink (EXPERIMENTAL) (TASKSTATS)
[Y/n/?] y
  Enable per-task delay accounting (EXPERIMENTAL) (TASK_DELAY_ACCT)
[Y/n/?] y
  Enable extended accounting over taskstats (EXPERIMENTAL) (TASK_XACCT)
[N/y/?] n
User Namespaces (EXPERIMENTAL) (USER_NS) [N/y/?] (NEW) n
PID Namespaces (EXPERIMENTAL) (PID_NS) [N/y/?] (NEW) n
Auditing support (AUDIT) [Y/n/?] y
  Enable system-call auditing support (AUDITSYSCALL) [Y/n/?] y
Kernel .config support (IKCONFIG) [Y/n/m/?] y
  Enable access to .config through /proc/config.gz (IKCONFIG_PROC)
[Y/n/?] y
Kernel log buffer size (16 => 64KB, 17 => 128KB) (LOG_BUF_SHIFT) [17] 17
Control Group support (CGROUPS) [N/y/?] (NEW) y
  Example debug cgroup subsystem (CGROUP_DEBUG) [N/y/?] (NEW) n
  Namespace cgroup subsystem (CGROUP_NS) [N/y/?] (NEW) n
  Cpuset support (CPUSETS) [Y/n/?] y
Fair group CPU scheduler (FAIR_GROUP_SCHED) [Y/n/?] (NEW) y
...
```

Si vous appuyez sur la touche [Entrée], c'est l'option par défaut qui est activée. Vous pouvez laisser le doigt appuyé sur la touche [Entrée] jusqu'à la fin, puis ensuite modifier la configuration avec l'interface textuelle ou graphique.

d. make menuconfig

La méthode précédente est pratique dans le cas d'une migration. Cependant vous voudrez peut-être utiliser quelque chose de plus convivial. Si vous ne disposez pas d'interface graphique (elle est souvent absente sur les serveurs car inutile) vous pouvez configurer votre noyau en passant par une interface en mode console. Voilà qui devrait rappeler MS-DOS à certains.

```
# make menuconfig
```



Les options de compilation du noyau via menuconfig

Le mode d'emploi est inclus dans l'interface (touches [Entrée], [Y], [N], [M], [?], [/], [Echap]). Notez que si vous appuyez sur la touche [Espace] vous faites défiler les choix possibles. N'hésitez pas à utiliser la touche d'aide pour (tenter de) comprendre à quoi sert une option.

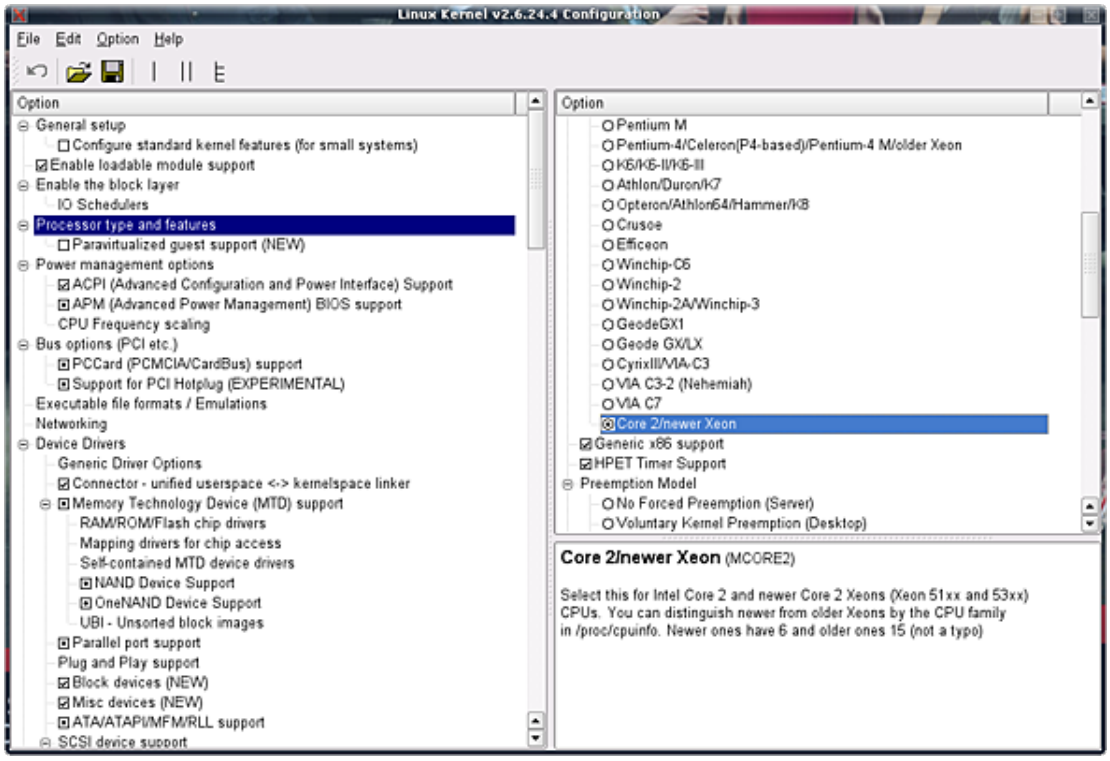
Une fois vos choix effectués, quittez et le fichier .config sera automatiquement généré.

e. make xconfig

Si vous disposez d'un environnement graphique, voici le meilleur choix possible. Vous disposez d'une interface graphique pour configurer votre noyau. La liste des entrées principales est à gauche sous forme d'arborescence. Vous choisissez à droite les options associées. Dans le cadre inférieur vous obtenez de l'aide pour chacune des options. Les menus vous permettent d'effectuer une recherche parmi les valeurs.

Depuis l'interface graphique vous pouvez charger une configuration issue d'un autre fichier de configuration, ou exporter celle-ci vers un fichier de votre choix.

```
# make xconfig
```



Les options de compilation via xconfig

f. Quelques choix d'optimisation

Les distributions Linux fournissent des noyaux génériques. Certaines proposent des noyaux spécifiques à certains types de processeurs (Intel, AMD, etc.). Dans tous les cas ou presque, des versions 32 bits et 64 bits sont fournies, ces dernières permettant d'exploiter les processeurs Intel ou AMD récents (Athlon64, Prescott, Core2, etc.).

Ces noyaux sont conçus pour fonctionner sur un maximum de machines, processeurs et matériels. Ils sont compilés avec des options standards et la quasi-intégralité du support des divers périphériques. Ils ne sont donc pas optimisés pour certains usages. Vous trouverez très intéressante la possibilité d'adapter le noyau à votre machine ce qui peut vous apporter :

- Une parfaite adéquation des optimisations du noyau pour votre matériel.
- Un gain certain de performances suivant le type d'usage (serveur, station de travail).
- Le support de nouveaux matériels (noyau plus récent).
- Un allègement du volume disque occupé par les modules inutiles.

Dans tous les cas, si vous compilez un nouveau noyau, ne désinstallez ou n'écrasez pas l'ancien. Conservez-le : en cas de problème avec votre noyau personnalisé, vous aurez alors la possibilité de revenir en arrière.

Parmi les pistes pour améliorer votre noyau :

- Activez les optimisations du noyau pour votre processeur. Dans l'entrée **Processor type and features** sélectionnez par exemple **Core2 / Newer Xeon** si votre machine dispose d'un processeur Core 2 Duo ou supérieur.
- Modifiez la valeur du **Preemption Model** en **Preemptible kernel** et cochez **Preempt the Big Kernel Clock** afin de réduire l'effet de latence du noyau (pour le rendre préemptible par les processus).
- Modifiez la valeur du **Timer Frequency** à 300 Hz ou 1000 Hz afin de réduire le temps de réponse aux événements.
- Dans **CPU Frequency Scaling**, vous pouvez légèrement augmenter la vitesse de démarrage du noyau en passant la valeur de **Default CPUfreq Governor** à **performance**. Ceci ne change rien par la suite où vous pourrez régler après coup la vitesse du processeur.
- Supprimez enfin les divers modules dont vous n'aurez jamais besoin. Inutile par exemple de conserver les cartes Token Ring ou ISDN si vous ne les utiliserez jamais.

Sur ce dernier point, faites tout de même attention à ne pas supprimer n'importe quoi. Gardez par exemple à l'esprit que si vous changez de machine, contrairement à Windows vous n'avez pas forcément besoin de réinstaller Linux si le noyau est suffisamment générique (il vous faudra peut-être booter en single pour modifier les modules chargés au démarrage) - l'auteur a migré d'un Pentium 4 d'il y a quatre ans vers un Core 2 Duo (avec changement de carte mère, carte graphique, etc.) sans avoir entièrement réinstallé son système. Le nouveau système a peut-être besoin de modules malheureusement supprimés.

Pensez aussi aux supports et périphériques en hotplug ou en hotswap. Si vous avez par erreur supprimé le pilote ou une option du pilote, vous devrez recompiler le noyau ou redémarrer sur l'ancien, que vous aurez conservé.

Une bonne idée avec les noyaux par défaut des distributions, est de toucher uniquement aux options d'optimisation comme celles vues ci-dessus, en laissant tous les autres choix par défaut.

---

32 ou 64 bits ?

Quasiment toutes les machines vendues ce jour disposent d'un jeu d'instructions 64 bits, y compris les ordinateurs portables. Pourtant les ordinateurs grand public sont quasiment tous livrés avec des systèmes en 32 bits (Windows XP ou Vista principalement) et ce malgré le fait que des versions 64 bits existent. Le 64 bits a actuellement des avantages et des inconvénients. Ces derniers sont liés à l'historique des logiciels et des systèmes d'exploitation :

- Les machines sont livrées avec un OS 32 bits.
- Les constructeurs fournissent des pilotes pour ces OS 32 bits.
- Les bibliothèques fournies sont compilées en 32 bits.
- Les applications liées à ces bibliothèques doivent donc être compilées en 32 bits.
- Les constructeurs et éditeurs ne font pas l'effort de fournir des versions 64 bits, la base installée étant trop faible.
- Les utilisateurs n'installent pas de version 64 bits, par manque de support de ces dernières.

La boucle s'entretient elle-même, ce qui contraint plus ou moins l'utilisateur à brider son système. Le problème est le même avec les processus en HyperThreading ou multicœurs. L'OS est bien souvent programmé en multithread, mais pas les applications (les jeux sont les bons derniers dans ce cas), ce qui réduit d'autant l'intérêt.

Pourtant une machinant fonctionnant en 64 bits a beaucoup d'avantages :

- Un gain important dans la gestion de la mémoire (plus de limitation à 4 Go).
- Un gain important de performances sur les applications optimisées.
- Une garantie de compatibilité dans l'avenir.
- Plus de "big crunch" (passage à zéro des compteurs de temps sous Unix ayant pour effet de repasser toutes les dates au 13 décembre 1901) Unix en janvier 2038 !

À titre de comparaison, le transcodage d'un DVD en DIVX via dvdrip (options : une passe, désentrelacement intelligent, grande taille redimentionnement HQ, mp3 192 kbits) tourne à une vitesse de 27 à 30 images par seconde sur une openSUSE 10.3 en 32 bits (Core 2 Duo e6750). Sur la même machine, la vitesse est de 40-42 images par seconde (sur une installation optimisée pour travailler en 64 bits). Soit un gain moyen de 25 à 30%. La même chose a été constatée avec un calcul super_pi optimisé. Le gain peut être réel, même s'il n'est pas significatif dans un environnement bureautique.

Comme les sources du noyau sont fournies et que la plupart des pilotes y sont inclus, soit fournis sous forme de sources, tout matériel supporté en 32 bits par Linux l'est en principe en 64 bits. Certains modules propriétaires (ati, nvidia) sont fournis en 64 bits. Les distributions 64 bits fournissent le nécessaire (bibliothèques) pour faire fonctionner les applications 32 bits nativement.

Aussi, si vous en avez la possibilité, testez une version 64 bits de Linux.

Si vous recompilez un noyau avec les extensions 64 bits depuis une distribution 32 bits, seul le noyau fonctionnera en 64 bits. Il faudrait alors installer ensuite toutes les bibliothèques et tous les outils en 64 bits, ce qui est complexe et long. Le mieux dans ce cas est de réinstaller le système.

---

4. Compilation

Maintenant que le noyau est configuré, vous pouvez lancer la compilation. Cette étape est la plus longue. Le noyau contient des millions de lignes en langage C et en assembleur. Suivant vos options (et surtout les modules inclus), la charge de votre machine et la vitesse de votre processeur, cette étape peut prendre de quelques minutes à quelques heures ! Dans tous les cas, vous avez le temps pour faire autre chose.

```
# make
scripts/kconfig/conf -s arch/x86/Kconfig
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/linux/utsrelease.h
  UPD     include/linux/utsrelease.h
  SYMLINK include/asm -> include/asm-x86
  CC      arch/x86/kernel/asm-offsets.s
  GEN     include/asm-x86/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  HOSTCC  scripts/genksyms/genksyms.o
  SHIPPED scripts/genksyms/lex.c
  SHIPPED scripts/genksyms/parse.h
  SHIPPED scripts/genksyms/keywords.c
  HOSTCC  scripts/genksyms/lex.o
  SHIPPED scripts/genksyms/parse.c
  HOSTCC  scripts/genksyms/parse.o
  HOSTLD  scripts/genksyms/genksyms
  CC      scripts/mod/empty.o
  HOSTCC  scripts/mod/mk_elfconfig
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/file2alias.o
  HOSTCC  scripts/mod/modpost.o
  HOSTCC  scripts/mod/sumversion.o
  HOSTLD  scripts/mod/modpost
  HOSTCC  scripts/kallsyms
  HOSTCC  scripts/conmakehash
  HOSTCC  scripts/bin2c
  CC      init/main.o
  CHK     include/linux/compile.h
  UPD     include/linux/compile.h
```

```
  CC      init/version.o
  CC      init/do_mounts.o
...
  BUILD   arch/x86/boot/bzImage
Root device is (8, 6)
Setup is 11012 bytes (padded to 11264 bytes).
System is 1567 kB
Kernel: arch/x86/boot/bzImage is ready  (#1)
  Building modules, stage 2.
  MODPOST 1950 modules
...
  LD [M]  sound/usb/snd-usb-lib.ko
  CC      sound/usb/usx2y/snd-usb-usx2y.mod.o
  LD [M]  sound/usb/usx2y/snd-usb-usx2y.ko
```

La première colonne est l'action effectuée sur le fichier représenté par la deuxième colonne. CC indique une compilation, LD l'édition des liens, un [M] l'action sur un module, etc.

5. Installation

Si la compilation s'est effectuée sans erreurs, il vous reste deux actions à effectuer. Tout d'abord installez les modules. La commande suivante les installe dans **/lib/module/<version_noyau>** et crée le fichier des dépendances associés.

```
# make modules_install
  INSTALL arch/x86/crypto/aes-i586.ko
  INSTALL arch/x86/crypto/twofish-i586.ko
  INSTALL arch/x86/kernel/apm.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/acpi-cpufreq.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/cpufreq-nforce2.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/e_powersaver.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/gx-suspmod.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/longhaul.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/longrun.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko
  INSTALL arch/x86/kernel/cpu/cpufreq/powernow-k6.ko
...
  INSTALL sound/usb/snd-usb-lib.ko
  INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
  DEPMOD  2.6.24.4-default
```

Cette seconde commande recopie le noyau et le nécessaire associé dans /boot. Selon les distributions, elle va aussi créer l'initrd associé, et modifier la configuration du chargeur de démarrage GRUB.

```
# make install
sh /usr/src/linux-2.6.24.4/arch/x86/boot/install.sh 2.6.24.4-default
arch/x86/boot/bzImage System.map "/boot"

Kernel image:   /boot/vmlinuz-2.6.24.4-default
Initrd image:   /boot/initrd-2.6.24.4-default
Root device:    /dev/disk/by-id/scsi-SATA_ST380011A_5JVTH798-part6
(/dev/sda6) (mounted on / as ext3)
Resume device:  /dev/sda5
Kernel Modules: processor thermal scsi_mod libata sata_sil pata_atiixp
fan jbd mbcache ext3 edd sd_mod usbcore ohci-hcd uhci-hcd ehci-hcd
ff-memless hid usbhid
Features:       block usb resume.userspace resume.kernel
Bootsplash:     SuSE (1280x1024)
36359 blocks
```

Respectez l'ordre précédent. Si vous installez le noyau avant ses modules, l'initrd ne pourra pas être construit car les modules devant y être présents ne sont pas encore installés. Il se peut que l'initrd ne soit pas généré par défaut, auquel cas vous devrez après l'installation relancer cette commande avec les paramètres correspondant au noyau installé.

```
# ls -l /boot/*2.6.24*
-rw-r--r-- 1 root root 7877336 avr  8 19:08 /boot/initrd-2.6.24.4-
```

```
default
-rw-r--r-- 1 root root  872762 avr  8 19:06 /boot/System.map-2.6.24.4-
default
-rw-r--r-- 1 root root 1615232 avr  8 19:07 /boot/vmlinuz-2.6.24.4-
default
```

Vérifiez si la configuration du chargeur de démarrage a été modifiée.

```
# grep 2.6.24 /boot/grub/menu.lst
title openSUSE 10.3 - 2.6.24.4
    kernel /boot/vmlinuz-2.6.24.4-default root=/dev/disk/by-id/scsi-
SATA_ST380011A_5JVTH798-part6 vga=0x31a resume=/dev/sda5 splash=silent
showopts
    initrd /boot/initrd-2.6.24.4-default
title Failsafe -- openSUSE 10.3 - 2.6.24.4
    kernel /boot/vmlinuz-2.6.24.4-default root=/dev/disk/by-id/scsi-
SATA_ST380011A_5JVTH798-part6 vga=normal showopts ide=nodma apm=off
acpi=off noresume nosmp noapic maxcpus=0 edd=off 3
    initrd /boot/initrd-2.6.24.4-default
```

Vous n'avez pas à réinstaller GRUB quand vous modifiez son fichier de configuration.

Si vous utilisez LILO, vous devez, outre modifier **`/etc/lilo.conf`**, le réinstaller avec **`/sbin/lilo`**.

6. Test

Si toutes les étapes précédentes ont fonctionné, il n'y a plus qu'à redémarrer votre ordinateur et à sélectionner votre nouveau noyau au chargement. Si le boot se termine correctement (accès à la console ou environnement graphique), ouvrez une console et vérifiez la version de votre système.

```
$ uname -a
Linux slyserver 2.6.24.4-default #1 SMP PREEMPT Tue Apr 8 19:06:10
CEST 2008 x86_64 intel x86_64 GNU/Linux
```

7. Autres options

Après une compilation, les divers fichiers intermédiaires (fichiers objets) prennent énormément de place. Pour les supprimer, utilisez la commande suivante : `# make clean`

Si votre distribution est à base de rpm, vous pouvez créer les packages du noyau (sources, headers, noyau) avec la commande suivante : `# make rpm`

Les fichiers périphériques

1. Introduction

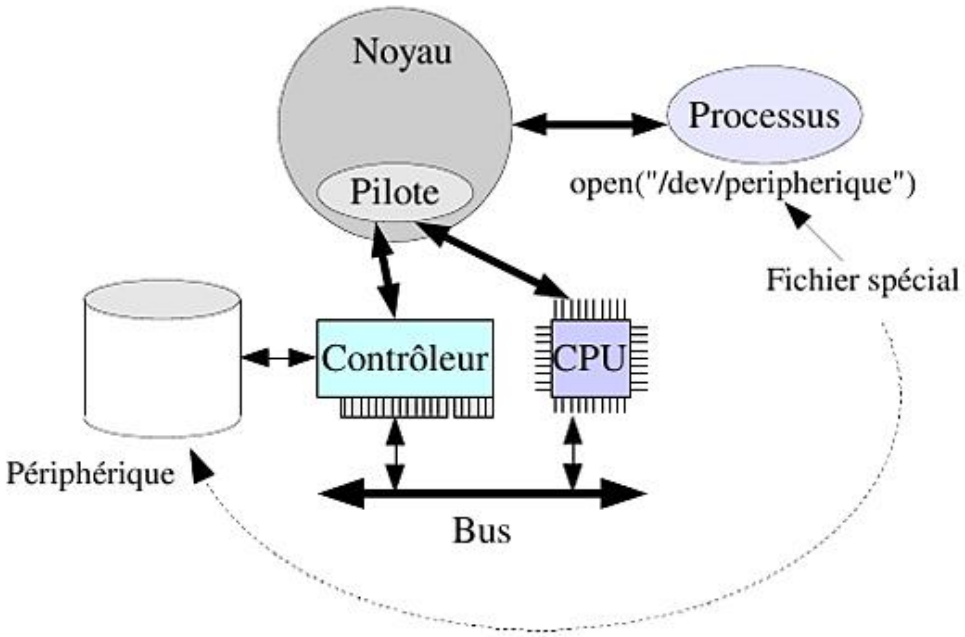
Revenez sur le fonctionnement des périphériques au sein d'un ordinateur. Le principe est généralement le même sur tous les ordinateurs.

Les périphériques sont reliés à un contrôleur, par exemple un contrôleur IDE ou SATA pour les disques IDE, un contrôleur SCSI pour les disques, lecteurs et autres scanners SCSI, ou encore un contrôleur USB. Un contrôleur sait généralement contrôler plusieurs périphériques qui lui sont rattachés.

Le contrôleur communique avec le micro-processeur et la mémoire à l'aide de bus (bus de commandes et bus de données).

Côté Linux le contrôleur et ses périphériques sont gérés à l'aide de pilotes (un pilote pour le contrôleur, et un ou plusieurs pilotes pour les périphériques qui y sont rattachés, par exemple un pilote pour le contrôleur SCSI, puis un pilote pour les disques, un autre pour les scanners, et encore un autre pour un CD-Rom). Le pilote est souvent un module complémentaire du noyau, livré par le constructeur ou déjà présent.

Le périphériques sont vus comme des fichiers. Du coup, les processus accèdent aux périphériques par l'intermédiaire de ces fichiers à l'aide des primitives en langage C dont le code est dans le noyau. Le processus doit d'abord ouvrir le fichier spécial du périphérique (primitive open), puis lire (read) ou écrire (write) des données de ou vers le périphérique comme il le ferait avec un fichier normal. Ces opérations de lecture/écriture sont ensuite interprétées par le pilote du périphérique.



Linux accède aux périphériques via des fichiers spéciaux

C'est dans le fichier spécial **`/dev/peripherique`** que le système de gestion de fichiers trouve les informations nécessaires pour s'adresser au pilote concerné par le périphérique ouvert par un processus.

2. Fichiers spéciaux

Les fichiers spéciaux périphériques sont par convention placés dans le répertoire /dev et disposent, comme n'importe quel autre fichier, d'un inode unique. Vous pouvez donc connaître ses attributs à l'aide de la commande **ls -l**.

Le premier caractère identifie le type de périphérique :

- **c** : type de périphérique en mode caractère ;
- **b** : type de périphérique en mode bloc.

Ces modes différencient le type d'échange de données entre le module de gestion de fichiers et le pilote du périphérique. En mode caractère, il n'y a pas d'utilisation des buffers du système et l'échange se fait octet par octet.

En mode bloc le système accède au périphérique via un index qui représente les coordonnées du bloc de données sur le support. Il est donc plus rapide pour les périphériques comme les disques.

Les deux autres attributs essentiels d'un fichier périphérique sont la paire d'informations que vous trouvez à la place de la taille du fichier : le numéro **majeur** et le numéro **mineur**.

- Le numéro majeur identifie le pilote et par conséquence le contrôleur de périphérique.
- Le numéro mineur identifie généralement le périphérique mais il peut aussi désigner une particularité du périphérique, comme la partition d'un disque, un emplacement précis, le numéro de carte (en cas de présence de plusieurs cartes contrôleurs identiques, de plusieurs cartes son, etc.).

Voici quelques fichiers spéciaux courants, selon la distribution :

- **`/dev/mem`** : la mémoire physique.
- **`/dev/kmem`** : la mémoire virtuelle du noyau.
- **`/dev/console`** : la console maître (/dev/syscon).
- **`/dev/tty`** : l'entrée/sortie standard du processus en cours.
- **`/dev/mouse`** : la souris, souvent un raccourci.
- **`/dev/swap`** : le disque swap primaire.
- **`/dev/null`** : la poubelle UNIX. On peut y écrire. La lecture provoque un EOF.
- **`/dev/root`** : système de fichier spécial root.
- **`/dev/dump`** : le disque dans lequel le noyau fait son dump en cas de panique système.
- **`/dev/rmt0`** : lecteur de bande magnétique ou de cartouche en mode caractère.
- **`/dev/fd0`** : lecteur de disquettes en mode bloc.
- **`/dev/pts/1`** : idem mais pour Unix SYSTEM V (et Linux).
- **`/dev/lp0`** : une imprimante parallèle.
- **`/dev/ttyS0`** : port COM1.
- **`/dev/ttyS1`** : port COM2.
- **`/dev/psaux`** : port PS2 pour la souris.
- **`/dev/sound`** : carte son.
- **`/dev/dsp`** : contrôleur DSP de la carte son.
- **`/dev/sequencer`** : séquenceur MIDI de la carte son.
- **`/dev/ide/*`** : les périphériques IDE.
- **`/dev/scsi/*`** : les périphériques SCSI.

- **/dev/usb/*** : les périphériques USB.
- **/dev/hdX** : les disques IDE.
- **/dev/sdX** : les disques SATA ou SCSI.
- etc.

3. Créer un fichier spécial

La commande **mknod** permet de créer un fichier spécial. Bien que Linux dispose de méthodes particulières qui prennent souvent en charge automatiquement la création des fichiers périphériques (udev), il peut arriver qu'une documentation précise l'ajout manuel d'un périphérique. Dans ce cas attention ! Le système de fichiers /dev est souvent de type udev et totalement dynamique ! Il faudra alors trouver une autre solution.

```
mknod /dev/peripherique type majeur mineur
```

Les pilotes sont soit intégrés au noyau à la compilation de celui-ci, soit compilés sous forme de modules complémentaires chargés dynamiquement. Suivant les distributions, le répertoire /dev est parfois un système de fichiers dynamique (devfs, udev) donc le contenu varie suivant la présence ou non des périphériques. C'est ainsi que la création du fichier périphérique est prise en charge par le pilote de périphérique et un démon particulier devfsd ou udevd. Cela autorise par exemple le hotplug, comme le branchement de périphériques USB à la demande : le noyau détecte le matériel, charge le bon pilote, et ce pilote crée dynamiquement le fichier périphérique.

Parfois, seule une partie du répertoire /dev est dynamique comme le support de l'USB avec le système de fichier usbdevfs.

4. Connaître son matériel

a. Bus PCI

La commande **lspci** fournit des informations détaillées sur les cartes et adaptateurs reliés au bus PCI. Les adaptateurs peuvent être ceux connectés sur les ports d'extension de la carte mère mais aussi ceux intégrés à la carte mère (contrôleurs IDE/SATA, cartes réseaux, etc.). Les bus AGP et PCI Express sont considérés comme étant des bus PCI.

```
# lspci
00:00.0 Host bridge: ATI Technologies Inc RS480 Host Bridge (rev 10)
00:01.0 PCI bridge: ATI Technologies Inc RS480 PCI Bridge
00:11.0 IDE interface: ATI Technologies Inc 437A Serial ATA Controller
00:12.0 IDE interface: ATI Technologies Inc 4379 Serial ATA Controller
00:13.0 USB Controller: ATI Technologies Inc IXP SB400 USB Host
Controller
00:13.1 USB Controller: ATI Technologies Inc IXP SB400 USB Host
Controller
00:13.2 USB Controller: ATI Technologies Inc IXP SB400 USB2 Host

Controller
00:14.0 SMBus: ATI Technologies Inc IXP SB400 SMBus Controller (rev 10)
00:14.1 IDE interface: ATI Technologies Inc Standard Dual Channel PCI IDE
Controller
00:14.3 ISA bridge: ATI Technologies Inc IXP SB400 PCI-ISA Bridge
00:14.4 PCI bridge: ATI Technologies Inc IXP SB400 PCI-PCI Bridge
00:14.5 Multimedia audio controller: ATI Technologies Inc IXP SB400
AC'97 Audio Controller (rev 01)
00:18.0 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
HyperTransport Technology Configuration
00:18.1 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
Address Map
00:18.2 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
DRAM Controller
00:18.3 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
```

```
Miscellaneous Control
01:05.0 VGA compatible controller: ATI Technologies Inc RS480 [Radeon
Xpress 200G Series]
02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8169
Gigabit Ethernet (rev 10)
```

Vous pouvez détailler encore plus avec l'option **-v** et préciser un adaptateur avec ses identifiants. Pour obtenir les informations détaillées sur le contrôleur Ethernet (02:03.0) procédez comme ceci :

```
# lspci -v -s 02:03.0
02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8169
Gigabit Ethernet (rev 10)
        Subsystem: Unknown device 1631:d008
        Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 20
        I/O ports at 8800 [size=256]
        Memory at ff3ffc00 (32-bit, non-prefetchable) [size=256]
        Expansion ROM at ff300000 [disabled] [size=128K]
        Capabilities: [dc] Power Management version 2
```

Quand c'est possible, la ligne Subsystem indique le constructeur et le modèle exact du périphérique. Le périphérique est identifié par une paire de valeurs hexadécimales. La première est l'identifiant unique du constructeur (vendor). La seconde est l'identifiant du modèle. Certains modèles ont parfois les mêmes identifiants alors que le matériel peut être différent (changement de chipset). La correspondance est statique (les associations sont inscrites au sein du code compilé). Elle sert entre autres au chargement des bons modules. Si les informations ne sont pas suffisantes, spécifiez l'option **-vv** :

```
# lspci -vv -s 02:03.0
02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8169
Gigabit Ethernet (rev 10)
        Subsystem: Unknown device 1631:d008
        Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV+ VGASnoop-
ParErr- Stepping- SERR+ FastB2B-
        Status: Cap+ 66MHz+ UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort-
 <TAbort- <MAbort- >SERR- <PERR+
        Latency: 64 (8000ns min, 16000ns max), Cache Line Size: 32 bytes
        Interrupt: pin A routed to IRQ 20
        Region 0: I/O ports at 8800 [size=256]
        Region 1: Memory at ff3ffc00 (32-bit, non-prefetchable)
[size=256]
        Expansion ROM at ff300000 [disabled] [size=128K]
        Capabilities: [dc] Power Management version 2
                Flags: PMEClk- DSI- D1+ D2+ AuxCurrent=375mA PME(D0-
,D1+,D2+,D3hot+,D3cold+)
                Status: D0 PME-Enable- DSel=0 DScale=0 PME-
```

b. Bus USB

La commande **lsusb** fait la même chose que lspci mais pour le bus USB :

```
# lsusb
Bus 003 Device 002: ID 04b4:6830 Cypress Semiconductor Corp. USB-2.0 IDE
Adapter
Bus 003 Device 001: ID 0000:0000
Bus 008 Device 002: ID 126f:0161 TwinMOS
Bus 008 Device 001: ID 0000:0000
Bus 006 Device 001: ID 0000:0000
Bus 007 Device 001: ID 0000:0000
Bus 005 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 004 Device 003: ID 046d:092e Logitech, Inc.
Bus 004 Device 002: ID 046d:c50e Logitech, Inc. MX-1000 Cordless Mouse
Receiver
Bus 004 Device 001: ID 0000:0000
Bus 002 Device 002: ID 045e:00dd Microsoft Corp.
Bus 002 Device 001: ID 0000:0000
```

Quand c'est possible, Linux indique quels sont les noms des périphériques via une base d'identifiants, de la même manière que pour les cartes PCI. Ces identifiants servent aussi à Linux pour déterminer quel pilote USB charger.

Comme pour lspci, vous obtiendrez plus d'informations avec -v, ainsi que -d :

```
# lsusb -d 046d:092e
Bus 004 Device 003: ID 046d:092e Logitech, Inc.
# lsusb -v -d 045e:00dd

Bus 002 Device 002: ID 045e:00dd Microsoft Corp.
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         8
  idVendor           0x045e Microsoft Corp.
  idProduct          0x00dd
  bcdDevice            1.73
  iManufacturer           1 Microsoft
  iProduct                2 Comfort Curve Keyboard 2000
  iSerial                 0
  bNumConfigurations      1
...
```

c. Ressources matérielles

Le système de fichiers virtuel /proc regorge d'informations sur votre matériel. En voici une liste non exhaustive.

Interruptions

```
# cat /proc/interrupts
           CPU0       CPU1
  0:     692294     559242   IO-APIC-edge      timer
  1:          1          1   IO-APIC-edge      i8042
  8:          1          0   IO-APIC-edge      rtc
  9:          0          0   IO-APIC-fasteoi   acpi
 12:          0          4   IO-APIC-edge      i8042
 16:        591     430773   IO-APIC-fasteoi   ahci, uhci_hcd:usb1,
ohci1394, nvidia
 17:         39      15025   IO-APIC-fasteoi   libata
 18:        285      39829   IO-APIC-fasteoi   ehci_hcd:usb3,
uhci_hcd:usb4, uhci_hcd:usb7
 19:          0          0   IO-APIC-fasteoi   uhci_hcd:usb6
 21:      11107         28   IO-APIC-fasteoi   uhci_hcd:usb2
 22:     397327       3429   IO-APIC-fasteoi   libata, libata, HDA
Intel
 23:          1       2533   IO-APIC-fasteoi   uhci_hcd:usb5,
ehci_hcd:usb8
4347:         40      24310   PCI-MSI-edge      eth0
NMI:          0          0
LOC:    1251342    1251335
ERR:          0
```

Canaux DMA

```
# cat /proc/dma
 4: cascade
```

Plages d'adresses d'entrées-sorties

```
# cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
```

```
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0290-0297 : pnp 00:06
  0295-0296 : w83627ehf
03c0-03df : vesafb
03f8-03ff : serial
0400-041f : 0000:00:1f.3
  0400-041f : i801_smbus
0480-04bf : 0000:00:1f.0
0800-087f : 0000:00:1f.0
  0800-0803 : ACPI PM1a_EVT_BLK
  0804-0805 : ACPI PM1a_CNT_BLK
  0808-080b : ACPI PM_TMR
  0810-0815 : ACPI CPU throttle
  0820-082f : ACPI GPE0_BLK
0cf8-0cff : PCI conf1
9400-940f : 0000:00:1f.2
9480-948f : 0000:00:1f.2
  9480-948f : libata
...
```

Périphériques (bloc, caractère)

```
# cat /proc/devices
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 /dev/vc/0
  4 tty
  4 ttyS
...
 14 sound
...
171 ieee1394
180 usb
189 usb_device
195 nvidia
253 rtc
254 usb_endpoint

Block devices:
  7 loop
  8 sd
  9 md
 11 sr
 65 sd
...
253 device-mapper
254 mdp
```

Partitions

```
# cat /proc/partitions
major minor  #blocks  name

   8     0  160836480 sda
   8     1   41945683 sda1
   8     2     530145 sda2
   8     3   41945715 sda3
   8     4          1 sda4
   8     5    2104483 sda5
   8     6   74308626 sda6
   8    16  156290904 sdb
   8    17  156288321 sdb1
```

```
   8    32  293036184 sdc
   8    33  293033601 sdc1
   8    48     503808 sdd
   8    49     503792 sdd1
```

Processeurs

```
# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 15
model name      : Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz
stepping        : 11
cpu MHz         : 1998.000
cache size      : 4096 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
syscall nx lm constant_tsc pni monitor ds_cpl vmx smx est tm2 ssse3
cx16 xtpr lahf_lm
bogomips        : 6804.61
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:
...
```

d. Autres outils

Certaines distributions sont fournies avec, ou disposent dans leurs dépôts d'outils complémentaires comme **hwinfo** ou **dmidecode**.

hwinfo

L'outil **hwinfo** détecte votre matériel et vous en fournit la liste (d'une manière courte avec l'option `--short`). Il se base sur une interrogation du matériel qui lui retourne ses informations :

```
# hwinfo --short
cpu:
                       Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz,
2000 MHz
...
keyboard:
  /dev/input/event4    Microsoft Comfort Curve Keyboard 2000
mouse:
  /dev/input/mice      Logitech MX-1000 Cordless Mouse Receiver
monitor:
                       Generic Monitor
graphics card:
                       ASUSTeK GeForce 8600 GT
sound:
                       ASUSTeK 82801I (ICH9 Family) HD Audio Controller
storage:
                       ASUSTeK 82801IB (ICH9) 2 port SATA IDE Controller
...
network:
  eth0                 ASUSTeK L1 Gigabit Ethernet Adapter
...
disk:
```

```
  /dev/sda             HDT722516DLA380
...
partition:
  /dev/sda1            Partition
 ...
cdrom:
  /dev/sr0             TSSTcorp CD/DVDW SH-S183A
...
usb controller:
                       ASUSTeK 82801I (ICH9 Family) USB UHCI Controller #4
    ...
BIOS:
                       BIOS
bridge:
                       ASUSTeK 82G33/G31/P35/P31 Express DRAM Controller
...
Port
                       Intel 82801I (ICH9 Family) PCI Express Port 1
...
memory:
                       Main Memory
firewire controller:
                       ASUSTeK IEEE 1394 Host Controller
unknown:
...
                       Keyboard controller
...
  /dev/ttyS0           16550A
                       Logitech Camera
  /dev/input/event5    Microsoft Comfort Curve Keyboard 2000
```

Pour obtenir plus de détails, supprimez l'option **--short**, et spécifiez éventuellement quel composant vous voulez détailler. Par exemple **--cpu** pour le processeur, **--memory** pour la mémoire, etc. La liste est dans le manuel de la commande.

```
# hwinfo --cpu
01: None 00.0: 10103 CPU
  [Created at cpu.301]
  Unique ID: rdCR.j8NaKXDZtZ6
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.15.11 "Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz"
  Features:
fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,
clflush,dts,acpi,mmx,fxsr,sse,sse2,ss,ht,tm,syscall,nx,lm,constant_tsc,
pni,monitor,ds_cpl,vmx,smx,est,tm2,ssse3,cx16,xtpr,lahf_lm
  Clock: 2666 MHz
  BogoMips: 6804.61
  Cache: 4096 kb
  Units/Processor: 2
  Config Status: cfg=new, avail=yes, need=no, active=unknown
...
# hwinfo --memory
01: None 00.0: 10102 Main Memory
  [Created at memory.61]
  Unique ID: rdCR.CxwsZFjVASF
  Hardware Class: memory
  Model: "Main Memory"
  Memory Range: 0x00000000-0x7ff7ffff (rw)
  Memory Size: 2 GB
  Config Status: cfg=new, avail=yes, need=no, active=unknown
```

dmidecode

L'outil **dmidecode** n'interroge pas les composants matériels mais lit et interprète la table **DMI** (*Desktop Management Interface*) de l'ordinateur, parfois aussi appelée **SMBIOS** (*System Management BIOS*). Il fournit non seulement des informations sur l'état matériel actuel de la machine, mais aussi sur ses extensions possibles (par exemple la vitesse maximale du processeur, la quantité de mémoire possible, etc.). Contrairement à **hwinfo** qui interroge un composant,

par exemple le CPU, **dmidecode** lit les informations telles que détectées par le BIOS et la carte mère. C'est rapide, parfois plus précis que **hwinfo**, mais parfois faux donc à vérifier.

La sortie étant bien plus longue qu'avec **hwinfo**, précisez quelles informations vous souhaitez avec `-s` ou `-t` (consultez le manuel) :

```
# dmidecode -t processor
# dmidecode 2.9
SMBIOS 2.4 present.

Handle 0x0004, DMI type 4, 35 bytes
Processor Information
        Socket Designation: LGA775
        Type: Central Processor
        Family: Pentium 4
        Manufacturer: Intel
        ID: FB 06 00 00 FF FB EB BF
        Signature: Type 0, Family 6, Model 15, Stepping 11
        Flags:
                FPU (Floating-point unit on-chip)
                VME (Virtual mode extension)
                DE (Debugging extension)
                PSE (Page size extension)
                TSC (Time stamp counter)
                MSR (Model specific registers)
                PAE (Physical address extension)
                MCE (Machine check exception)
                CX8 (CMPXCHG8 instruction supported)
                APIC (On-chip APIC hardware supported)
                SEP (Fast system call)
                MTRR (Memory type range registers)
                PGE (Page global enable)
                MCA (Machine check architecture)
                CMOV (Conditional move instruction supported)
                PAT (Page attribute table)
                PSE-36 (36-bit page size extension)
                CLFSH (CLFLUSH instruction supported)
                DS (Debug store)
                ACPI (ACPI supported)
                MMX (MMX technology supported)
                FXSR (Fast floating-point save and restore)
                SSE (Streaming SIMD extensions)
                SSE2 (Streaming SIMD extensions 2)
                SS (Self-snoop)
                HTT (Hyper-threading technology)
                TM (Thermal monitor supported)
                PBE (Pending break enabled)
        Version: Intel(R) Core(TM)2 Duo CPU E6750 @ 2.66GHz
        Voltage: 1.3 V
        External Clock: 427 MHz
        Max Speed: 3800 MHz
        Current Speed: 3416 MHz
        Status: Populated, Enabled
        Upgrade: Socket LGA775
        L1 Cache Handle: 0x0005
        L2 Cache Handle: 0x0006
        L3 Cache Handle: 0x0007
        Serial Number: To Be Filled By O.E.M.
        Asset Tag: To Be Filled By O.E.M.
        Part Number: To Be Filled By O.E.M.
```

Notez que les valeurs en gras ne sont pas les mêmes que celles retournées par les autres commandes. Il n'y a pas d'erreur ici : le processeur est overclocké à 3,4Ghz...

```
# dmidecode -s processor-frequency
3416 MHz
```

5. Le support de l'USB et du hotplug

a. Les modules

L'**USB** (*Universal Serial Bus*) est un bus de données en mode série et "plug and play". Le principe est qu'une fois l'adaptateur USB branché, le système charge l'éventuel pilote correspondant et que l'appareil fonctionne tout de suite. En USB 2.0, les taux de transferts peuvent atteindre 480 Mbits/s. Le futur USB3 devrait permettre une vitesse de 4.8 Gbits/s.

En USB1.0 et USB 1.1, deux types de contrôleurs se partagent le marché : **UHCI** et **OHCI**.

- Universal Controller Host Interface : développé par Intel.
- Open Controller Host Interface : les autres.

En USB 2.0, le contrôleur se nomme **EHCI** : *Enhenced Host Controller Interface*.

Linux gère les trois types de contrôleur.

Le module de base se nomme **usbcore**. Il propose l'ensemble des API nécessaires aux autres modules :

- **ohci_hcd** : support OHCI.
- **uhci_hcd** : support UHCI.
- **ehci_hcd** : support EHCI.
- **usb_storage** : couche d'accès aux supports de masse : disques externes, clés USB, etc.
- **usbhid** : couche d'accès au support des périphériques **HID** (*Human Interface Device*) du type claviers, souris, joystick, etc.
- **snd-usb-audio** : support des cartes son USB.
- **usbvideo** : support des cartes vidéo et d'acquisition USB.
- **irda-usb** : support des ports infrarouges USB.
- **usbnet** : support des cartes réseaux USB.
- etc.

b. Chargement

Les modules USB (et d'autres liés au matériel) sont chargés de manières différentes :

- Via le **ramdisk initial** : la plupart des claviers et souris sont actuellement de type USB. Pour cette raison le support de base de l'USB et le module usbhid sont très souvent chargés via l'initrd (Initial Ramdisk). C'est pour cela qu'il y a souvent un petit décalage entre le moment où GRUB charge le noyau et celui où il est possible d'utiliser le clavier, entre la fin du support de l'USB par le BIOS (passage du noyau en mode protégé) et la prise en charge du support USB par le noyau.
- Via **init** : un service chargé de la détection du matériel et/ou du chargement des pilotes charge la liste des modules correspondant au matériel.
- Via **kmod** : le chargement automatique des modules du noyau. Quand le noyau détecte la présence d'un nouveau périphérique USB, il peut charger le module correspondant. Rappelez-vous qu'un module fournit les identifiants du matériel qu'il prend en charge. Le fichier **modules.usbmap** fournit cette correspondance. Le noyau exécute alors **modprobe**.
- Via **udev** ou **hotplug** : des règles permettent de spécifier des actions à l'arrivée de nouveaux périphériques, dont par exemple le chargement de modules complémentaires.

- À la main.

c. hotplug, usbmgr

Avant l'arrivée de udev et de HAL, c'était le projet **Linux Hotplug Project** qui était majoritairement utilisé. hotplug ne gérait pas que l'USB mais tout type de matériel connecté, à chaud ou non (coldplug) au PC.

hotplug n'est plus utilisé sur aucune des distributions récentes à base de noyau 2.6 et a laissé la place au tandem udev/HAL.

Usbmgr était un système de hotplug prévu pour la gestion de l'USB. Il n'est plus utilisé.

d. udev

udev est un système de fichiers dynamique qui remplace l'ancien devfs des noyaux 2.4. Udev gère toute l'arborescence /dev. C'est lui qui va créer et modifier les fichiers périphériques présents dans ce répertoire.

Contrairement à hotplug qui s'exécutait en mode noyau, udev est lancé comme un service classique, associé à un système de fichiers particulier. Il est situé dans l'espace utilisateur.

```
# mount | grep udev
udev on /dev type tmpfs (rw,mode=0755)
```

Le noyau génère des événements et des messages. Udev lit les messages émis par le noyau et les interprète. Il dispose de règles qu'il applique selon le type de message. Par exemple voici trois règles simples :

```
KERNEL=="raw1394*",             GROUP="video"
KERNEL=="dv1394*",              SYMLINK+="dv1394/%n", GROUP="video"
KERNEL=="video1394*",           SYMLINK+="video1394/%n", GROUP="video"
```

- **KERNEL** : nom de l'événement du noyau. Ici apparition d'un périphérique Firewire.
- **GROUP** : le fichier périphérique appartiendra au groupe indiqué.
- **SYMLINK** : udev va créer un lien symbolique du fichier périphérique vers l'emplacement indiqué. Ici le %n représente le numéro dans l'ordre de détection.

Ce n'est qu'un exemple. La formation LPI ne vous demande pas d'écrire des règles élaborées mais de comprendre le mécanisme associé. Avec les règles précédentes, si vous connectez un caméscope numérique via la prise DV à votre carte Firewire, le noyau va générer un événement dont le nom commence par « video1394 ». La règle correspondante va être exécutée :

- Apparition de /dev/video1394.
- Modification de son groupe en vidéo.
- Création d'un lien symbolique entre /dev/video1394 et /dev/video1394/0 (le premier).

Les règles sont placées dans **`/etc/udev/rules.d`**.

```
$ ls -l
total 180
-rw-r--r-- 1 root root   191 sep 21  2007 05-udev-early.rules
-rw-r--r-- 1 root root   366 oct  2  2007 40-alsa.rules
-rw-r--r-- 1 root root  2276 sep 24  2007 40-bluetooth.rules
-rw-r--r-- 1 root root  4937 sep 21  2007 50-udev-default.rules
-rw-r--r-- 1 root root   571 sep 22  2007 51-lirc.rules
-rw-r--r-- 1 root root   399 oct 10  2007 55-hpmud.rules
-rw-r--r-- 1 root root 83444 fév 17 14:22 55-libsane.rules
-rw-r--r-- 1 root root   119 sep 21  2007 56-idedma.rules
-rw-r--r-- 1 root root   119 sep 21  2007 60-cdrom_id.rules
-rw-r--r-- 1 root root  1424 sep 21  2007 60-persistent-input.rules
-rw-r--r-- 1 root root  4347 sep 21  2007 60-persistent-storage.rules
```

```
-rw-r--r-- 1 root root   918 sep 21  2007 64-device-mapper.rules
-rw-r--r-- 1 root root   725 sep 21  2007 64-md-raid.rules
-rw-r--r-- 1 root root  1290 sep 22  2007 70-kpartx.rules
-rw-r--r-- 1 root root   611 oct  5  2007 70-persistent-cd.rules
-rw-r--r-- 1 root root   325 oct  5  2007 70-persistent-net.rules
...
```

Vous pouvez modifier les règles, créer les vôtres. Dans ce cas, créez un fichier appelé **99-local.rules** (par exemple) et mettez-y les vôtres.

À titre d'exemple les différents chemins possibles d'accès aux disques et partitions présentés dans le chapitre Les disques et le système de fichiers - Accéder aux systèmes de fichiers sont issus de règles udev. La règle suivante (qui dépend d'autres règles en amont, ce qui peut être déduit par le test d'une variable udev DETYPE déjà positionnée) importe le résultat d'une commande **vol_id** que vous connaissez déjà pour créer les liens par UUID et par label.

```
# by-label/by-uuid (filesystem properties)
ENV{DEVTYPE}=="partition", IMPORT{program}="vol_id --export $tempnode"
```

Vous pouvez connaître tous les périphériques contrôlés par udev avec udevinfo et notamment sur un périphérique donné comme ceci :

```
# udevinfo --query=all -n /dev/sda
P: /block/sda
N: sda
S: disk/by-id/scsi-SATA_ST380011A_5JVTH798
S: disk/by-id/ata-ST380011A_5JVTH798
S: disk/by-path/pci-0000:00:14.1-scsi-0:0:0:0
S: disk/by-id/edd-int13_dev80
E: DEVTYPE=disk
E: ID_VENDOR=ATA
E: ID_MODEL=ST380011A
E: ID_REVISION=8.01
E: ID_SERIAL=SATA_ST380011A_5JVTH798
E: ID_SERIAL_SHORT=5JVTH798
E: ID_TYPE=disk
E: ID_BUS=scsi
E: ID_ATA_COMPAT=ST380011A_5JVTH798
E: ID_PATH=pci-0000:00:14.1-scsi-0:0:0:0
E: ID_EDD=int13_dev80
```

Administration des utilisateurs

1. Principe

a. Identification et authentification

L'**identification**, c'est savoir qui est qui, afin de déterminer les droits de la personne qui se connecte. Un utilisateur est identifié par un login.

L'**authentification**, c'est apporter la preuve de qui on est, par exemple via un secret partagé entre l'utilisateur et le système, et connus d'eux seuls. L'utilisateur est authentifié par un mot de passe.

b. Les utilisateurs

Un utilisateur est l'association d'un nom de connexion, le login, à un UID et au moins un GID.

- **UID** : User ID.
- **GID** : Group ID.

Les UID et les GID sont en principe uniques. Le login est unique. Il est cependant envisageable d'associer plusieurs logins au même UID, le système travaillant parfois avec le login.

L'UID identifie l'utilisateur (ou le compte applicatif) tout au long de sa connexion. Il est utilisé pour le contrôle de ses droits et de ceux des processus qu'il a lancé. Ce sont les UID et GID qui sont stockés au sein de la table des inodes, dans la table des processus, etc., et non les logins.

L'utilisateur dispose des attributs de base suivants :

- un nom de connexion appelé le login ;
- un mot de passe ;
- un UID ;
- un GID correspondant à son groupe principal ;
- un descriptif ;
- un répertoire de connexion ;
- une commande de connexion ;

D'autres attributs sont disponibles via l'utilisation de la sécurité des mots de passe via shadow (voir la section concernée).

Les UID d'une valeur inférieure à 100 sont en principe associés à des comptes spéciaux avec des droits étendus. Ainsi l'UID de root, l'administrateur, est 0. Selon les distributions, à partir de 100, 500 ou 1000, et ce jusqu'à environ 60000, ce sont les UID des utilisateurs sans pouvoirs particuliers.

Un login a en principe une taille de 8 caractères. En fait Linux et d'autres systèmes acceptent une taille plus grande, mais avec la plupart des commandes l'affichage, voire la gestion des logins, est limité à 8 caractères.

Un login accepte la plupart des caractères. Il ne doit pas commencer par un chiffre. Il est possible de modifier la liste des caractères autorisés et de forcer la longueur et la complexité via les mécanismes d'authentification PAM et le fichier **`/etc/login.defs`**.

c. Les groupes

Chaque utilisateur fait partie d'au moins un groupe. Un groupe regroupe des utilisateurs. Comme pour les logins, le

GID du groupe accompagne toujours l'utilisateur pour le contrôle de ses droits. Un utilisateur peut faire partie de plusieurs groupes, auquel cas il faut distinguer son groupe primaire des groupes secondaires.

Les groupes sont aussi des numéros. Il existe des groupes spécifiques pour la gestion de certaines propriétés du système et notamment l'accès à certains périphériques.

Le groupe primaire est celui qui est toujours appliqué à la création d'un fichier. Si l'utilisateur seb a pour groupe primaire users, alors les fichiers créés par seb auront comme groupe d'appartenance users.

Un utilisateur dispose de tous les droits associés à ses groupes secondaires. Si seb a comme groupe secondaire video et qu'un fichier dispose des droits d'écriture pour ce groupe, alors seb aura le droit de modifier son contenu.

La commande **id** permet de connaître les informations essentielles sur un utilisateur : uid, gid, groupes secondaires.

```
$ id seb
uid=1000(seb) gid=100(users) groupes=100(users),16(dialout),3(sys),
33(video)
```

Un fichier est créé par seb. Son propriétaire est seb et son groupe est le groupe principal de seb : users.

```
$ touch test
$ ls -l test
-rw-r--r-- 1 seb users 0 avr 10 14:30 test
```

d. Les mots de passe

Les mots de passe permettent d'authentifier les utilisateurs. Ils doivent être assez complexes pour ne pas être découverts facilement, mais assez intuitifs pour qu'ils s'en souviennent. Les mots de passe sont cryptés (MD5, DES par exemple) et ne sont pas directement lisibles sous leur forme cryptée par l'utilisateur afin que personne ne puisse tenter de le décrypter via un quelconque traitement.

Un utilisateur devrait changer régulièrement son mot de passe, ne jamais l'écrire quelque part ni le conserver sur lui. Vous verrez par la suite qu'il est possible de contraindre l'utilisateur à appliquer des règles de nommage et de durée de conservation.

Voici par exemple le résultat crypté par Blowfish (reconnaissable par le $2a$ commençant la chaîne) d'un mot de passe :

```
aher874oP47  vaut
$2a$10$CqutecUAlTGSFs2BPnVl..ntI8OEdy5j6gLI/cIKhHP4XZISdlGZO
```

2. Les fichiers

a. /etc/passwd

Le fichier **/etc/passwd** contient la liste des utilisateurs du système local. Il est lisible par tout le monde. Les informations qu'il contient sont publiques et utiles tant pour le système que pour les utilisateurs. Chaque ligne représente un utilisateur et est composée de sept champs.

`Login:password:UID:GID:comment:homedir:shell`

- Champ 1 : le login ou nom d'utilisateur.
- Champ 2 : sur les vieilles versions, le mot de passe crypté. Si un x est présent, le mot de passe est placé dans /etc/shadow. Si c'est un point d'exclamation le compte est verrouillé.
- Champ 3 : le User ID.
- Champ 4 : le GID, c'est-à-dire le groupe principal.
- Champ 5 : un commentaire ou descriptif. C'est un champ d'information.
- Champ 6 : le répertoire de travail, personnel, de l'utilisateur. C'est le répertoire dans lequel il arrive lorsqu'il se connecte.

- Champ 7 : le shell par défaut de l'utilisateur. Mais ce peut être toute autre commande, y compris une commande interdisant la connexion.

b. /etc/group

Le fichier **/etc/group** contient la définition des groupes d'utilisateurs et pour chacun la liste des utilisateurs dont il est le groupe secondaire. Chaque ligne est composée de quatre champs :

```
Group:password:GID:user1,user2,...
```

- Champ 1 : le nom du groupe.
- Champ 2 : le mot de passe associé. Voyez l'explication juste en dessous.
- Champ 3 : le Group Id.
- Champ 4 : la liste des utilisateurs appartenant à ce groupe.

Il est inutile de replacer dans le quatrième champ les utilisateurs ayant ce groupe pour groupe principal, c'est induit.

Vous pouvez être surpris de voir la présence d'un champ de mot de passe pour les groupes. Dans la pratique il est très rarement utilisé. Comme il est bien entendu impossible de se connecter comme un groupe, l'explication est ailleurs. Un utilisateur a le droit de changer de groupe afin de prendre, temporairement tout du moins, un groupe secondaire comme groupe principal avec la commande **newgrp**.

Dans ce cas, l'administrateur peut mettre en place un mot de passe sur le groupe pour protéger l'accès à ce groupe en tant que groupe principal.

c. /etc/shadow

Le fichier **/etc/shadow** accompagne le fichier **/etc/passwd**. C'est là qu'est stocké, entre autres, le mot de passe crypté des utilisateurs. Pour être plus précis il contient toutes les informations sur le mot de passe et sa validité dans le temps. Chaque ligne est composée de 9 champs séparés par des : :

```
bean:$2a$10$AjADxPEfE5iUJcltzYA4wOZO.f2UZ0qP/8EnOFY.P.m10HifS7J8i:13
913:0:99999:7:::
```

- Champ 1 : le login.
- Champ 2 : le mot de passé crypté. Le xx initial indique le type de cryptage.
- Champ 3 : nombre de jours depuis le 1er janvier 1970 du dernier changement de mot de passe.
- Champ 4 : nombre de jours avant lesquels le mot de passe ne peut pas être changé (0 : il peut être changé n'importe quand).
- Champ 5 : nombre de jours après lesquels le mot de passe doit être changé.
- Champ 6 : nombre de jours avant l'expiration du mot de passe durant lesquels l'utilisateur doit être prévenu.
- Champ 7 : nombre de jours après l'expiration du mot de passe après lesquels le compte est désactivé.
- Champs 8 : nombre de jours depuis le 1er janvier 1970 à partir du moment où le compte a été désactivé.
- Champ 9 : réservé.

Dans l'exemple de la ligne bean, le mot de passe a été changé 13913 jours après le 01/01/1970. Le mot de passe doit être changé avant 0 jours mais il est toujours valide car le champ suivant indique qu'il faut le changer au bout

de 99999 jours (273 ans) et le champ 5 est vide (pas d'obligation de changement de mot de passe). Le compte est désactivé après 7 jours, ce qui évidemment ne risque pas d'arriver…

Pour connaître la date en fonction du 01/01/1970 utilisez la commande date comme ceci, en ajoutant le nombre de jours désiré :

```
$ date --date "1 jan 1970 +13984 days"
mar avr 15 00:00:00 CEST 2008
```

d. /etc/gshadow

Le fichier **/etc/gshadow** est le pendant du fichier précédent mais pour les groupes. Il n'est cependant pas supporté par défaut sur la plupart des distributions Linux et ne sera pas expliqué ici. Les mots de passe des groupes sont placés dans **/etc/group**.

3. Gestion des utilisateurs

a. Ajout

La création d'un utilisateur pourrait être entièrement effectuée à la main car Linux (et les autres Unix) s'appuient sur une suite de commandes qui ne font « que » modifier des fichiers plats déjà existants et qui créent et recopient des fichiers et dossiers au bon endroit avec les bons droits.

La création d'un utilisateur consiste à :

- rajouter une ligne dans **/etc/passwd**,
- rajouter d'une ligne dans **/etc/shadow**,
- rajouter d'éventuelles informations dans **/etc/group**,
- créer le répertoire personnel et mettre à jour son contenu avec **/etc/skel**,
- changer les permissions et le propriétaire du répertoire personnel,
- changer le mot de passe (encodé).

Vous pouvez créer directement un compte en éditant les fichiers avec un éditeur, bien que ce soit plutôt déconseillé. Si vous souhaitez le faire tout de même, utilisez la commande **vipw** qui va mettre à jour les divers caches associés à la gestion des comptes.

La commande **vipw** admet trois arguments :

- `-p` : édition de **/etc/passwd**.
- `-g` : édition de **/etc/group**.
- `-s` : édition de **/etc/shadow**.

Tout ceci peut être effectué avec la commande **useradd**. Elle ajoute un nouveau compte et effectue les principales opérations :

- création de l'utilisateur et remplissage des fichiers,
- création d'un groupe privé d'utilisateur (de même nom que celui-ci),
- création du répertoire personnel, remplissage et modification des droits.

```
useradd <options> login
```

Si aucune option n'est précisée, les valeurs par défaut sont récupérées au sein du fichier **/etc/defaults/useradd**.

Les options principales suivantes sont acceptées :

Option	Rôle
-m	Crée aussi le répertoire personnel. Elle est parfois comprise par défaut, mais il vaut mieux vérifier si le répertoire personnel est présent après l'utilisation de la commande si vous n'utilisez pas cette option.
-u	Précise l'UID numérique de l'utilisateur, pour le forcer. Autrement l'UID est calculé selon les règles du fichier login.defs et les UID existants.
-g	Précise le groupe principal de l'utilisateur, par GID ou par son nom (variable GROUP).
-G	Précise les groupes additionnels (secondaires, de l'utilisateur) séparés par des virgules (variable GROUPS).
-d	Chemin du répertoire personnel. Généralement /home/<login>, mais n'importe quel chemin peut être précisé (variable HOME/<login>).
-c	Un commentaire associé au compte. Il peut être quelconque mais est parfois utilisé par certaines commandes comme **finger**. Son contenu peut être modifié par l'utilisateur avec la commande **chfn**.
-k	Chemin du répertoire contenant le squelette de l'arborescence du répertoire utilisateur. C'est généralement /etc/skell (variable SKEL).
-s	Shell (commande de connexion) par défaut de l'utilisateur (variable SHELL). L'utilisateur peut le changer via la commande **chsh**.
-p	Le mot de passe de l'utilisateur. Attention ! le mot de passe doit déjà être crypté ! À moins de recopier le mot de passe d'un compte générique, vous préférerez utiliser ensuite la commande **passwd**.

La commande suivante crée le compte robert avec la plupart des options de base précisées. C'est juste un exemple car, sauf parfois le -m, si vous ne précisez rien ce sont les options par défaut par rapport à celles précisées dans le fichier **/etc/defaults/useradd**.

```
# useradd -m -u 1010 -g users -G video,dialout,lp -s /bin/bash -d
/home/robert -c "Compte de Robert" robert
# grep robert /etc/passwd
robert:x:1010:100:Compte de Robert:/home/robert:/bin/bash
```

La commande ne crée pas de mot de passe. Il faut le faire à la main avec la commande **passwd**.

```
# passwd robert
Changing password for robert.
Nouveau mot de passe :
Retaper le nouveau mot de passe :
Mot de passe changé.
```

b. Sécurité des mots de passe

Changer de mot de passe

La commande **passwd** permet de gérer les mots de passe mais aussi les autorisations de connexion et la plupart des champs présents dans **/etc/shadow**.

Tout utilisateur a le droit de changer son mot de passe, dans le délai précisé par le champ 4 de /etc/shadow. L'action par défaut est de changer le mot de passe de l'utilisateur courant. L'ancien mot de passe est demandé par sécurité (notamment pour empêcher une personne mal intentionnée de modifier votre mot de passe derrière votre

dos). La saisie est masquée.

```
$ id
uid=1000(seb) gid=100(users) ...
$ passwd
Changing password for seb.
Ancien mot de passe :
Nouveau mot de passe :
Retaper le nouveau mot de passe :
Mot de passe changé.
```

Les modules **PAM** (*Pluggable Authentification Module*) peuvent imposer des contraintes plus ou moins sévères pour le choix du mot de passe : de telle longueur, pas basé sur un mot du dictionnaire, etc. Voyez ce qu'il se passe en tentant d'utiliser successivement toto (trop court), azerty (trop long) et Martine (dictionnaire) :

```
$ passwd
Changing password for seb.
Ancien mot de passe :
Nouveau mot de passe :
Mot de passe incorrect : trop court
Nouveau mot de passe :
Mot de passe incorrect : trop simple
Nouveau mot de passe :
Mot de passe incorrect : basé sur un mot du dictionnaire
passwd: Nombre maximum de tentatives épuisées pour le service
```

L'utilisateur root a le droit de modifier les mots de passe de tous les utilisateurs du système, sans avoir à connaître le précédent mot de passe. Mieux : il peut forcer l'utilisation d'un mot de passe même si celui-ci n'est pas validé par PAM :

```
# passwd seb
Changing password for seb.
Nouveau mot de passe :
Mot de passe incorrect : basé sur un mot du dictionnaire
Retaper le nouveau mot de passe :
Mot de passe changé.
```

Gérer les informations de validité

Tous les champs de **`/etc/shadow`** peuvent être modifiés par la commande **passwd**. Voici quelques options disponibles.

Option	Rôle
-l	Lock : verrouille le compte en rajoutant un ! devant le mot de passe crypté.
-u	Unlock : déverrouille le compte. Il n'est pas possible de déverrouiller un compte qui n'a pas de mot de passe, il faut utiliser en plus -f pour cela.
-d	(root) Supprime le mot de passe du compte.
-n <j>	(root) Durée de vie minimale en jours du mot de passe.
-x <j>	(root) Durée de vie maximale en jours du mot de passe.
-w <j>	(root) Nombre de jours avant avertissement.
-i <j>	(root) Délai de grâce avant désactivation si le mot de passe est expiré.
-S	(root) Statut du compte.

Dans l'exemple suivant le compte bean est modifié comme ceci :

- Il doit attendre 5 jours après saisie d'un nouveau mot de passe pour pouvoir le changer,

- Son mot de passe est valide 45 jours,
- Il est prévenu 7 jours avant qu'il doit changer de mot de passe,
- S'il ne change pas de mot de passe après 45 jours, il dispose encore de 5 jours avant d'être désactivé.

```
# passwd -n 5 -x 45 -w 7 -i 5 bean
Password expiry information changed.
```

Voici la ligne de **/etc/shadow** associée.

```
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:13
984:5:45:7:5::
```

La commande **chage** permet de faire à peu près la même chose. Elle n'est accessible que par root. Lancée sans autre argument que le login de l'utilisateur, elle est interactive. Notez à la fin la possibilité de modifier la date du dernier changement du mot de passe et une date fixe d'expiration du mot de passe (champ 8) :

```
# chage bean
Changing aging information for bean.
        Minimum Password Age [7]:
        Maximum Password Age [40]:
        Password Expiration Warning [10]:
        Password Inactive [5]:
        Last Password Change (YYYY-MM-DD) [2008-04-10]:
        Account Expiration Date (YYYY-MM-DD) [1969-12-31]: 2010-01-01
Aging information changed.
```

Voici la ligne /etc/shadow résultante :

```
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:13
979:7:40:10:5:14610:
```

Les paramètres suivants sont acceptés :

Option	Rôle
-m	Mindays : équivaut à passwd -n.
-M	Maxdays : équivaut à passwd -x.
-d	Date de dernière modification du mot de passe (depuis le 01/01/1970).
-E	Date d'expiration du mot de passe (depuis le 01/01/1970).
-I	Inactive : équivaut à passwd -i.
-W	Warndays : équivaut à passwd -w.
-l	List : affiche tous les détails.

Les détails sont bien plus lisibles avec chage qu'avec passwd :

```
# passwd -S bean
bean PS 04/10/2008 7 40 10 5
# chage -l bean
Minimum:        7
Maximum:        40
Warning:        10
Inactive:       5
Last Change:            avr 10, 2008
Password Expires:       mai 20, 2008
Password Inactive:      mai 25, 2008
Account Expires:        jan 01, 2010
```

Un utilisateur quelconque peut afficher ses propres détails, mais son mot de passe lui sera demandé.

c. Modification

Utilisez la commande **usermod** pour modifier un compte. Elle prend la même syntaxe et les mêmes options que useradd mais dispose aussi d'une syntaxe complémentaire qui nécessite quelques précisions.

Option	Rôle
-L	Lock du compte, comme passwd -l.
-U	Unlock du compte, comme passwd -u.
-e <n>	Expire : le mot de passe expire n jours après le 01/01/1970.
-u <UID>	Modifie l'UID associé au login. Le propriétaire des fichiers appartenant à l'ancien UID au sein du répertoire personnel est modifié en conséquence.
-l <login>	Modifie le nom de login.
-m	Move : implique la présence de -d pour préciser un nouveau répertoire personnel. Le contenu de l'ancien répertoire est déplacé dans le nouveau.

d. Suppression

Supprimez un utilisateur avec la commande **userdel**. Par défaut le répertoire personnel n'est pas supprimé. Vous devez pour ceci passer l'option `-r`.

```
# userdel -r bean
```

4. Gestion des groupes

a. Ajout

Vous pouvez créer un groupe directement dans le fichier **`/etc/group`** ou bien passer par les commandes associées. Si vous éditez le fichier à la main, utilisez la commande **vigr** (ou vipw -g).

La commande **groupadd** permet de créer un groupe. Sa syntaxe simple accepte l'argument `-g` pour préciser un GID précis.

```
# grep amis /etc/group
amis:!:1234:
```

b. Modification

La commande **groupmod** permet de modifier un groupe. Ses paramètres sont les suivants :

Option	Rôle
-n <nom>	Renomme le groupe.
-g <GID>	Modifie le GID. Attention, le groupe d'appartenance des fichiers concernés n'est pas modifié.
-A <user>	Ajoute l'utilisateur spécifié dans le groupe (groupe secondaire).

-R <user>	Supprime l'utilisateur spécifié du groupe.

```
# groupmod -R seb amis
# grep amis /etc/group
amis:!:1234:
```

c. Suppression

La commande **groupdel** supprime un groupe. La commande vérifie d'abord si le groupe que vous voulez supprimer est le groupe principal d'un utilisateur. Dans ce cas le groupe ne peut pas être supprimé.

Par contre aucune action autre que celle consistant à supprimer la ligne correspondant dans /etc/group n'est effectuée : c'est à vous de vérifier le système de fichiers (et la configuration des applications si besoin) pour supprimer toute trace de ce groupe.

```
# groupdel amis
```

5. Commandes additionnelles

a. Conversion des fichiers

Vous utiliserez probablement rarement les commandes suivantes. Elles sont surtout utiles dans le cadre de migrations de serveurs Linux vers d'autres Unix, et vice versa. Quelques systèmes Unix n'utilisent pas par défaut (il faut l'activer après) la gestion des comptes avec les fichiers **shadow**. Dans ce cas il peut être nécessaire de convertir les fichiers **/etc/shadow** et **/etc/passwd** en un seul et unique **/etc/passwd**. C'est le rôle de la commande **pwunconv**.

Dans l'exemple suivant, le fichier **/etc/passwd** est converti. Une fois la commande exécutée, toute trace de **/etc/shadow** a disparu.

```
# pwunconv
# grep bean /etc/passwd
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:1001:100:
toto:/home/bean:/bin/bash
# ls -l /etc/shadow
ls: ne peut accéder /etc/shadow: Aucun fichier ou répertoire de ce type
```

Attention : la commande **pwunconv** est destructive. Toutes les informations des durées de validité des mots de passe sont détruites. Il n'est plus possible d'utiliser les options diverses de chage ou de passwd concernant les durées de validité.

La commande **pwconv** fait l'inverse : elle crée le fichier **/etc/shadow** associé à **/etc/passwd**, y déplace les mots de passe et y place les réglages par défaut tels que définis dans le fichier **/etc/login.defs**.

```
# grep bean /etc/passwd
bean:x:1001:100:toto:/home/bean:/bin/bash
p64p17bicb3:/home/seb # grep bean /etc/shadow
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:13
984:0:99999:7:::0
```

Les commandes **grpconv** et **grpunconv** font la même chose pour les groupes, mais ici sans succès puisque la plupart des distributions ne supporte pas les groupes en shadow.

b. Vérifier la cohérence

Il peut être utile de lancer des outils de vérification de la cohérence des fichiers des groupes et des mots de passe. Si vous avez l'habitude de modifier ces fichiers à la main, rien ne garantit que tout est correct : un groupe peut être manquant, un shell inexistant, un répertoire personnel absent, etc. La commande **pwck** effectue une vérification des fichiers **/etc/passwd** et **/etc/shadow** et reporte les erreurs.

Voici un exemple où les données de l'utilisateur bean ont été volontairement altérées pour tester la commande : le groupe n'existe pas, et le shell non plus. Sur ces entrefaites un autre problème a été découvert sur la machine de

test.

```
# pwck
Checking `/etc/passwd'
User `suse-ncc': directory `/var/lib/YaST2/suse-ncc-fakehome' does
not exist.
User `bean': unknown group `14400'
User `bean': shell `/bin/bashr' is not executable.
Checking `/etc/shadow'.
```

La commande **grpck** fait la même chose pour les groupes. Dans ce cas les contrôles sont moins étendus, se limitant aux doublons et à l'existence des utilisateurs pour les groupes secondaires.

```
# grpck
Checking `/etc/group'
Group `users': unknown user `zorg'
```

c. Vérifier les connexions

Vous pouvez tracer les connexions sur votre machine à l'aide de deux commandes. La commande **lastlog** se base sur le contenu de /var/log/lastlog. Elle accepte les paramètres `-u` (précision d'un utilisateur) et `-t` pour rechercher les connexions des n derniers jours.

```
$ lastlog -u seb
Username                Port     Latest
seb                     pts/4    jeu avr 10 15:13:46 +0200 2008
$ lastlog -t 10
Username                Port     Latest
root                    pts/3    jeu avr 10 14:48:13 +0200 2008
seb                     pts/4    jeu avr 10 15:13:46 +0200 2008
```

La commande **last** fait à peu près la même chose, mais se base sur /var/log/wtmp qui fournit des informations supplémentaires comme l'origine de la connexion (IP, nom de la console, etc.) et les dates de connexion et de déconnexion, ainsi que la durée de connexion et si l'utilisateur est encore connecté.

```
$ last
seb      pts/1                        Tue Apr 15 14:39   still logged in
seb      pts/4                        Tue Apr 15 12:06   still logged in
seb      pts/6                        Tue Apr 15 10:07   still logged in
seb      pts/3                        Mon Apr 14 13:36 - 15:18 (1+01:42)
seb      pts/1                        Thu Apr 10 15:39 - 11:58 (4+20:18)
seb      pts/4      localhost         Thu Apr 10 15:13 - 15:39  (00:25)
seb      pts/4      localhost         Thu Apr 10 15:13 - 15:13  (00:00)
seb      pts/4      localhost         Thu Apr 10 15:12 - 15:13  (00:00)
...
```

d. Actions de l'utilisateur

L'utilisateur dispose de certaines actions sur les informations de son compte. Il peut notamment :

- changer son shell de connexion,
- changer ses informations personnelles,
- changer de groupe principal,
- prendre l'identité de quelqu'un d'autre.

Changer de shell

La commande **chsh** permet à l'utilisateur de modifier définitivement (ou jusqu'à la prochaine commande chsh) de shell de connexion. Il ne peut pas choisir n'importe quoi. Le shell (ou toute autre commande) doit être présent dans **`/etc/shells`**. Cette liste est accessible via le paramètre `-l` de la commande. La modification est faite au sein

de **/etc/passwd**. Seul root a le droit de le modifier pour d'autres utilisateurs. Le nouveau shell est précisé avec le paramètre -s.

```
$ chsh -l
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
...
$ id
uid=1004(bean) gid=100(users) ...
$ chsh -s /bin/ksh
Changing login shell for bean.
Mot de passe :
Shell changed.
# grep bean /etc/passwd
bean:x:1004:100:toto:/home/bean:/bin/ksh
```

Changer le commentaire

Le commentaire du fichier **/etc/passwd** peut être modifié par l'utilisateur à l'aide de la commande **chfn**. Il est préférable de l'utiliser de manière interactive (le passage de paramètre en mode non interactif est réservé à root).

```
$ chfn
Changing finger information for bean.
Mot de passe :
Enter the new value, or press ENTER for the default
        Full Name: toto
        Room Number []: Mister Bean
        Work Phone []: 0102030405
        Home Phone []: 0605040302
        Other:
Finger information changed.
$ grep bean /etc/passwd
bean:x:1004:100:Mister Bean,0102030405,0605040302:/home/bean:/bin/bash
```

Changer de groupe principal

La commande **newgrp** permet de changer à titre temporaire de groupe principal, à condition que le nouveau groupe précisé soit un groupe secondaire de l'utilisateur et/ou que l'utilisateur dispose du mot de passe du groupe. Utilisée seule, **newgrp** revient au groupe d'origine. Les modifications sont temporaires, le fichier des mots de passe n'est pas modifié.

```
$ id
uid=1004(bean) gid=100(users) groupes=16(dialout),33(video),100(users)
$ newgrp video
$ id
uid=1004(bean) gid=33(video) groupes=16(dialout),33(video),100(users)
```

Que se passe-t-il si vous tentez de prendre comme groupe principal un groupe ne faisant pas partie de vos groupes secondaires mais qui est protégé par un mot de passe ? Dans l'exemple suivant, bean tente de prendre comme groupe principal grptest.

```
$ id
uid=1004(bean) gid=100(users) groupes=16(dialout),33(video),100(users)
$ newgrp grptest
Password:
$ id
uid=1004(bean) gid=1000(grptest) groupes=16(dialout),33(video),
100(users),1000(grptest)
```

Changer d'identité

L'utilisateur peut endosser, le temps d'une commande ou de toute une session, l'identité d'une autre personne. Il s'agit généralement de root, car vous savez qu'il ne faut jamais (ou au moins éviter) de se connecter en permanence en tant que root. Donc pour les tâches administratives il faut pouvoir devenir root (ou un autre utilisateur) le temps nécessaire.

La commande **su** (*substitute user*) permet d'ouvrir une session, ou d'exécuter un shell ou une commande donnée avec une autre identité. Évidemment, vous devez connaître le mot de passe de cet utilisateur.

```
su [-c commande] [-s shell] [-] [utilisateur]
```

Si aucun utilisateur n'est précisé, c'est root qui est utilisé.

```
$ id
uid=1000(seb) gid=100(users) ...
seb@p64p17bicb3:~> su
Mot de passe :
p64p17bicb3:/home/seb # id
uid=0(root) gid=0(root) groupes=0(root)
```

Notez que c'est l'environnement de l'utilisateur d'origine qui est utilisé par défaut. L'environnement de root (ou de tout autre utilisateur) n'est pas chargé :

```
# echo $LOGNAME $USER
seb seb
```

Pour charger l'environnement complet de l'utilisateur cible, rajoutez le tiret en paramètre :

```
$ su - bean
Mot de passe :
$ echo $USER $LOGNAME
bean bean
```

Pour exécuter ponctuellement une commande avec une autre identité, utilisez `-c`.

```
$ su -c "make install"
```

La commande **sg** (*subsitute group*) est identique à **su** mais elle prend un nom de groupe en argument.

6. Configuration avancée

Les descriptions des fichiers qui suivent dépendent fortement des versions des commandes, de leurs options de compilation et de la politique de sécurité appliquée par les éditeurs des diverses distributions. Il est possible que certains fichiers ne soient pas présents, et que d'autres contiennent des paramètres différents.

a. /etc/default/useradd

Le fichier **`/etc/default/useradd`** contient un certain nombre de variables définissant les règles par défaut à appliquer à la création d'un utilisateur :

- son groupe,
- la racine de son répertoire personnel (là où celui-ci sera situé),
- s'il est actif ou non,
- le shell,
- son ou ses groupes secondaires,
- l'endroit où est situé le squelette des comptes (structure de base d'un répertoire utilisateur),

- la création ou non d'un spool (dépôt) de courrier,
- etc.

```
$ cat /etc/defaults/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
GROUPS=video,dialout
CREATE_MAIL_SPOOL=no
```

b. /etc/default/passwd

Le fichier **/etc/default/passwd** contient quelques règles utilisées par la commande **passwd** pour le cryptage des mots de passe. Il est possible de définir des règles de cryptage globales, mais aussi par type de fichier, et de passer quelques options selon la méthode.

```
$ cat /etc/default/passwd
# Cryptage par defaut
CRYPT=md5

# Cryptage pour les fichiers (/etc/shadow)
CRYPT_FILES=blowfish

# option pour blowfish
BLOWFISH_CRYPT_FILES=10

# Pour NIS
CRYPT_YP=des
```

c. /etc/default/su

Le fichier **/etc/default/su** permet de configurer le fonctionnement de la commande **su**. Par défaut **su** avec le paramètre - met en place un nouveau PATH car il charge l'environnement de l'utilisateur ciblé. Vous pouvez modifier ceci et mettre en place votre propre PATH, ou conserver l'ancien.

```
# Change le PATH meme sans le tiret
ALWAYS_SET_PATH=no

# Path par défaut
PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin

# Path par défaut pour root
SUPATH=/usr/sbin:/bin:/usr/bin:/sbin:/usr/X11R6/bin
```

d. /etc/login.defs

Le fichier **/etc/login.defs** est utilisé par de nombreuses commandes comme **login**, **useradd**, **groupadd**, **passwd** pour définir quelques valeurs par défaut et la validité des logins. Son contenu peut varier suivant les distributions. Il peut contenir :

- une règle de validité des comptes (caractères autorisés, longueur, etc.),
- les UID min et max lors de la création d'un utilisateur,
- les GID min et max lors de la création d'un groupe,

- les commandes à appeler pour les ajouts/modifications/créations d'utilisateur,
- les règles par défaut pour la validité des mots de passe,
- la création ou non d'un répertoire personnel,
- etc.

Le contenu du fichier login.defs peut être en conflit avec certaines options des autres fichiers présents dans /etc/default. Dans ce cas, vous devez vérifier dans les manuels (ou manpages) de votre distribution quel fichier prévaut.

```
# Login autorisé meme si l'accès au home est impossible
DEFAULT_HOME        yes

# Path par défaut pour la commande login
ENV_PATH             /usr/local/bin:/usr/bin:/bin

# Idem pour la connexion root
ENV_ROOTPATH         /sbin:/bin:/usr/sbin:/usr/bin

# Délai en secondes entre deux tentatives de login
FAIL_DELAY        3

# Les utilisateurs dans le fichier ne voient pas les messages de cnx
HUSHLOGIN_FILE    /etc/hushlogins

# Affiche la date de dernière connexion
LASTLOG_ENAB        yes

# Les tentatives de connexion des logins inexistants sont tracées
LOG_UNKFAIL_ENAB    no

# trois tentatives de login en cas de mauvais mot de masse
LOGIN_RETRIES        3

# Timeout de 60 secondes
LOGIN_TIMEOUT        60

# Emplacement du motd (ou des , séparés par des :)
MOTD_FILE    /etc/motd

# type des terminaux par défaut
TTYTYPE_FILE    /etc/ttytype

# Permission par défaut des terminaux
TTYGROUP    tty
TTYPERM        0620

# Réclame ou non un mot de passe pour chsh et chfn
CHFN_AUTH        yes

# Restriction du chfn (f:nom, r:bureau, w:tel travail, h:tel maison)
CHFN_RESTRICT        rwh

# PASS_MAX_DAYS    Duree max du mot de passe
# PASS_MIN_DAYS    Délai min entre deux changements de mot de passe
# PASS_WARN_AGE    Avertissement avant le changement
PASS_MAX_DAYS      99999
PASS_MIN_DAYS      0
PASS_WARN_AGE      7

# UID/GID Min et Max par défaut et systeme pour useradd
SYSTEM_UID_MIN          100
SYSTEM_UID_MAX          499
```

```
UID_MIN                  1000
UID_MAX                 60000

#
# Min/max values for automatic gid selection in groupadd
#
# SYSTEM_GID_MIN to SYSTEM_GID_MAX inclusive is the range for
# GIDs for dynamically allocated administrative and system groups.
# GID_MIN to GID_MAX inclusive is the range of GIDs of dynamically
# allocated groups.
#
SYSTEM_GID_MIN           100
SYSTEM_GID_MAX           499
GID_MIN                 1000
GID_MAX                60000

# Regexp pour les noms de logins autorisés via login/useradd
CHARACTER_CLASS        [A-Za-z_][A-Za-z0-9_.-]*[A-Za-z0-9_.$-]\?

# Umask par défaut pour la création du homedir
UMASK            022

# Commande réellement exécutée lors de l'ajout d'un groupe
GROUPADD_CMD             /usr/sbin/groupadd.local

# Commande réellement exécutée lors de l'ajout d'un utilisateur
USERADD_CMD        /usr/sbin/useradd.local

# Commande exécutée avant la suppression d'un groupe
USERDEL_PRECMD        /usr/sbin/userdel-pre.local

# Commande exécutée après la suppression d'un groupe
USERDEL_POSTCMD        /usr/sbin/userdel-post.local
```

En plus des commandes **useradd** et **userdel**, voici une liste non exhaustive des commandes qui utilisent les paramètres de ce fichier :

- **login** : DEFAULT_HOME, ENV_PATH, ENV_ROOTPATH, FAIL_DELAY, HUSHLOGIN_FILE, LASTLOG_ENAB, LOG_UNKFAIL_ENAB, LOGIN_RETRIES, LOGIN_TIMEOUT, MOTD_FILE, TTYPERM, TTYTYPE_FILE ;
- **newusers** : PASS_MAX_DAYS, PASS_MIN_DAYS, PASS_WARN_AGE, UMASK ;
- **passwd** : OBSCURE_CHECKS_ENAB, PASS_MAX_LEN, PASS_MIN_LEN, PASS_ALWAYS_WARN, CRACKLIB_DICTPATH, PASS_CHANGE_TRIES ;
- **pwconv** : PASS_MAX_DAYS, PASS_MIN_DAYS, PASS_WARN_AGE.

7. Notifications à l'utilisateur

a. /etc/issue

Lorsqu'un utilisateur se connecte depuis la console, un message est généralement affiché juste avant l'invite de saisie de son login. Ce message est contenu dans le fichier **/etc/issue**. C'est un message d'accueil et à ce titre il peut contenir tout ce que vous voulez. Par défaut, il contient généralement le nom de la distribution Linux et le numéro de version du noyau.

Voici l'exemple du contenu de **/etc/issue** sur une distribution Mandriva :

```
$ cat issue
Mandriva Linux release 2008.1 (Official) for i586
Kernel 2.6.24.4-desktop-1mnb on an i686 / \l
```

b. /etc/issue.net

Le message d'accueil peut être différent lorsqu'un utilisateur se connecte depuis une console distante (telnet, ssh, etc.). C'est souvent le même mais sans les caractères de contrôles liés à un shell donné. Pour modifier ce message spécifique, éditez le contenu du fichier **/etc/issue.net**.

c. /etc/motd

Motd signifie Message of the day, le message du jour. Une fois l'utilisateur connecté depuis une console (locale ou distante), un message peut être affiché. L'administrateur peut modifier ce message en éditant le fichier **/etc/motd**. Par défaut il est vide. Vous pouvez par exemple modifier ce fichier pour prévenir vos utilisateurs qu'un reboot de maintenance aura lieu tel jour à telle heure, ceci évitant d'envoyer n mails…

8. L'environnement utilisateur

a. /etc/skel

À la création d'un utilisateur et de son répertoire personnel, l'environnement de l'utilisateur est mis en place. L'environnement contient par exemple les variables d'environnement, les alias, l'exécution de divers scripts. Il est contenu dans des fichiers chargés au démarrage de l'interpréteur de commandes (shell).

Lors de la création d'un compte, les divers fichiers de configuration sont copiés depuis le contenu du répertoire /etc/skel (skeleton) vers le répertoire personnel. Si vous souhaitez modifier les environnements de façon globale AVANT la création des utilisateurs, vous pouvez placer dans /etc/skel tous les fichiers que vous souhaitez et les modifier selon votre convenance. Ainsi si par exemple vous souhaitez que tout le monde dispose des mêmes icônes par défaut sur son bureau, et la même configuration par défaut du bureau, placez-y les répertoires **Desktop** et **.kde** d'un compte modèle.

b. Scripts de configuration

Le chapitre Le shell et les commandes GNU vous a présenté les différents fichiers dont le contenu est exécuté par le shell. À la connexion d'un utilisateur, les scripts suivants sont exécutés dans cet ordre :

- **/etc/profile** : définit les variables d'environnement importantes comme PATH, LOGNAME, USER, HOSTNAME, HISTSIZE, MAIL et INPUTRC.
- **/etc/profile.d/*** : /etc/profile appelle tous les scripts présents dans ce répertoire. Ces scripts peuvent compléter la configuration globale en ajoutant par exemple la configuration des paramètres linguistiques, des alias globaux, etc.
- **~/.bash_profile** : si le shell est bash, c'est le script suivant à être exécuté. Il est dans le répertoire utilisateur et appelle un autre script : **~/.bashrc** qui appelle lui-même **/etc/bashrc**. Vous pouvez définir dans .bash_profile des variables supplémentaires, alors que vous aurez tendance à définir dans **~/.bashrc** des alias et des fonctions. Il n'y a pas de règles strictes.
- **/etc/bashrc** est utilisé pour définir les fonctions et alias pour tout le système et tous les utilisateurs sous bash.

c. Groupes privés et setgid

La politique de Red Hat et d'autres distributions associées, comme Fedora, pour la sécurité des utilisateurs est de systématiquement leur créer un groupe privé et de leur attribuer un masque 002. De ce fait, les fichiers sont mieux protégés par défaut puisqu'aucun groupe ne peut accéder aux fichiers par défaut : les fichiers créés n'appartiennent qu'à un groupe qui n'a qu'un membre.

Comme un utilisateur peut faire partie de plusieurs groupes, il peut en théorie étendre l'accès à ses fichiers et répertoires avec la commande **chgrp** qui change le groupe d'un fichier. De même il peut changer temporairement de groupe principal avec la commande **newgrp**.

Cependant l'utilisateur préfère souvent ne pas se « casser la tête » et donner tous les droits avec un **chmod** (ex. chmod 777) ce qui constitue une faille dans la sécurité.

Dans ce cas, la solution est d'utiliser le bit **setgid** sur un répertoire.

- On définit un groupe commun à tous les utilisateurs devant pouvoir accéder à un répertoire et à ses fichiers.
- On crée un répertoire dont le groupe propriétaire est le groupe commun.
- On donne au répertoire les droits avec l'umask 002 (`rwxrwxr-x`) ou autre, mais avec tous les droits sur le groupe.
- On ajoute au répertoire le droit setgid (s) sur le groupe.

```
# mkdir rep
# chgrp grpcommun rep
# chmod 2770 rep
```

Lorsque setgid est positionné sur un répertoire, tous les fichiers créés dans ce répertoire appartiennent au groupe du répertoire et non pas au groupe de l'utilisateur. Ainsi tous les membres du groupe commun pourront accéder aux données. Le fichier continue cependant à appartenir à son créateur.

Associé au Sticky Bit, vous obtenez un bon niveau de protection et d'accès aux données :

- tous les membres du groupe peuvent y créer des fichiers et des répertoires,
- tous les fichiers appartiennent automatiquement au même groupe,
- seuls les propriétaires des fichiers peuvent les supprimer.

```
# chmod 3770 rep
# ls -ld rep
drwxrws--T 2 seb grpcommun 1024 2008-04-24 11:39 rep/
$ ls -l test
-rw-r--r-- 1 seb grpcommun 0 2008-04-24 11:43 test
```

9. Aperçu de PAM

PAM (*Pluggable Authentification Modules*) est un ensemble de modules et une bibliothèque permettant de mettre en place des mécanismes d'authentification avancés pour tous les outils nécessitant une sécurité accrue. L'authentification est basée sur des modules. Chaque module peut utiliser des mécanismes différents pour tenter d'authentifier un utilisateur. L'un se basera sur une authentification Unix classique, un autre sur LDAP, un troisième sur Active Directory, et un dernier sur la reconnaissance d'une emprunte digitale. Ce mécanisme permet aussi la vérification via un dongle USB...

Le principe est assez simple, la configuration plus compliquée. Un outil appelle la bibliothèque **`libpam.so`** pour un besoin d'authentification. La bibliothèque lit un fichier de configuration où l'appel à plusieurs modules peut être demandé. Chaque module appelé retourne vrai ou faux. Suivant la configuration, vrai peut être un pré-requis pour continuer avec un autre module ou être suffisant pour se connecter.

Les fichiers de configuration sont placés dans **`/etc/pam.d`**. Il en existe généralement un par application utilisant PAM et il porte le même nom. Si le fichier est manquant c'est « **`other`** » qui est utilisé. La syntaxe est la suivante.

```
Type_module contrôle module arguments
```

Type_module

- **`auth`** : module d'authentification (par exemple demande de login et de mot de passe).
- **`account`** : autorisation, gestion de comptes (vérification de l'utilisateur pour le service donné. Est-il autorisé ?).
- **`password`** : vérification et mise à jour des informations de sécurité (ex : le mot de passe est-il encore valable et si non, demande d'un nouveau mot de passe).
- session : modification de l'environnement de l'utilisateur.

contrôle

- **required** : réussite requise. En cas d'échec, les modules restants sont tout de même appelés mais quoi qu'il arrive au final PAM retournera un échec.
- **requisite** : un échec termine immédiatement l'authentification. La réussite l'autorise à continuer.
- **sufficient** : une réussite contourne les autres modules. Autrement dit PAM retourne ok quoi qu'il arrive en cas de réussite ici.
- **optional** : le résultat est ignoré.

module

- **pam_unix.so** : authentification standard via la fonction C getpw().
- **pam_env.so** : définition des variables d'environnement.
- **pam_securetty.so** : interdit une connexion super-utilisateur (root) depuis un terminal non sécurisé. La liste des terminaux autorisés est placée dans /etc/securetty.
- **pam_stack.so** : appelle un autre service PAM pour le chargement de modules supplémentaires.
- **pam_nologin.so** : interdit la connexion d'utilisateurs si le fichier **/etc/nologin** est présent. Dans ce cas son contenu est affiché.
- **pam_deny.so** : retourne toujours un échec.
- **pam_console.so** : donne des permissions supplémentaires à un utilisateur local.

Les paramètres dépendent de chaque module. Voici un exemple dans le cas classique d'une ouverture de session par la console. Dans ce cas, le shell de connexion appelle la commande **login**. Attention ce n'est qu'un exemple !

```
$ cat login
#%PAM-1.0
auth       requisite    /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_unix.so
auth         required     /lib/security/pam_deny
auth       required     /lib/security/pam_nologin.so
```

- La première ligne vérifie si root tente de se connecter depuis un terminal non sécurisé (ex : telnet, rlogin, rsh, etc.). Si c'est le cas, PAM retourne directement faux et l'authentification échoue directement.
- La seconde ligne charge l'environnement de l'utilisateur. Un échec ici n'empêche pas l'exécution des lignes suivantes, mais au bout du compte PAM retournera faux.
- La troisième ligne tente une authentification via les mécanismes Unix classiques (fichier des mots de passe, NIS, etc.). La réussite ici stoppe la procédure : l'utilisateur est directement authentifié. Autrement, on passe à la ligne suivante.
- La quatrième ligne retourne toujours faux. Les modules d'authentification précédents ayant échoué, la connexion de l'utilisateur échoue. La ligne suivante est quand même exécutée.
- Si le fichier **/etc/nologin** existe, il est affiché.

Il est possible d'interdire l'accès à une liste d'utilisateurs donnés. Placez les noms des utilisateurs interdits dans /etc/nologinusers et ajoutez la ligne suivante dans le fichier de configuration PAM. Dans notre exemple, il

faudrait l'ajouter entre la deuxième et la troisième ligne.

```
auth required /lib/security/pam_listfile.so onerr=suceed item=user
sense=deny file=/etc/nologinusers
```

La distribution Red Hat est configurée de telle manière que le fichier **/etc/security/system-auth** est appelé dans toutes les configurations PAM ou presque. Exemple du fichier de configuration login :

```
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
...
```

Dans ce cas, vous auriez mis notre ligne en dessous de celle de pam_nologin.so. Cependant il faut garder à l'esprit que le fichier de configuration **/etc/pam.d/login** n'est utilisé que pour l'authentification depuis la commande **login** et donc une connexion depuis un terminal (console texte). Depuis une fenêtre de connexion graphique (X-Window, x/k/gdm) vous n'aurez pas l'effet souhaité. Dans ce cas, la modification aurait dû être effectuée dans **/etc/pam.d/system-auth**.

```
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        required      /lib/security/pam_deny.so
...
```

Attention encore une fois à placer la ligne au bon endroit. Après ce bloc, la ligne n'aura aucun effet à cause de la ligne **sufficient** si l'utilisateur a un login et un mot de passe valides. Vous placerez donc la ligne au début des lignes **auth**.

L'impression

1. Principe

Il existe trois standards d'impression sous Unix, un sous System V, un autre sous BSD et un dernier fédérateur.

Quel que soit le standard, le principe est le même. À chaque imprimante déclarée (généralement dans /etc/printcap) correspond une file d'attente (**queue**). L'ensemble de ces files d'attente est géré par un service indépendant. Ces deux principes permettent une impression multi-utilisateur (les travaux d'impression sont en file d'attente, **job queues**), et en réseau (le service peut être utilisé depuis une autre machine distante).

En règle générale toutes les imprimantes savent directement imprimer du texte brut ASCII en 80 colonnes. Pour imprimer des documents formatés ou des images, on peut utiliser un pilote. On parle en fait de **filtre d'impression**. Le filtre peut être un script ou un binaire qui récupère le flux entrant (texte, image, document, postscript...), l'identifie et à l'aide de traitements associés le transforme en langage compréhensible par l'imprimante (Postscript, PCL, Canon, Epson, WPS...).

Si vous avez le choix et les moyens n'hésitez pas à prendre une imprimante compatible Postscript qui est un gage de parfaite compatibilité. Le site linuxprinting.org dispose d'une base complète de compatibilité des imprimantes sous Linux.

Linux accepte les commandes issues des Unix de type System V et BSD. Pendant longtemps le sous-système d'impression était basé sur les services BSD et le démon **lpd**. Depuis quelques années, toutes les distributions se basent sur CUPS, rétro-compatible (pour les commandes en tout cas) avec les anciens systèmes d'impression.

2. System V

Les commandes de gestion des files d'attente et des impressions sous System V sont les suivantes :

- `lp [-dImprimante] [-nChiffre] fic1` : imprime le contenu du fichier fic1. L'option `-d` permet de choisir l'imprimante, `-n` le nombre d'exemplaires.
- `lpstat [-d] [-s] [-t] [-p]` : informations sur l'impression. L'option `-d` affiche le nom de l'imprimante par défaut, `-s` un état résumé des imprimantes, `-t` la totalité des informations (statut, travaux...), `-p [liste]` uniquement les informations sur les imprimantes incluses dans la liste.
- `cancel [ids] [printers] [-a] [-e]` : supprime les tâches d'impression ids des imprimantes printers. L'option `-a` supprime tous les travaux de l'utilisateur, `-e` tous les travaux (seulement pour l'administrateur).
- On peut trouver les commandes **enable** et **disable** qui prennent comme paramètre le nom de la file d'attente, permettant d'en activer ou désactiver l'accès.

Le démon (ou daemon) s'appelle généralement **lpd** (*line printer daemon*) ou **lpsched** (*line printer scheduler*).

- **lpadmin** permet d'administrer les services d'impression comme les files d'attentes associées à une imprimante et la file d'attente par défaut. Ex : `lpadmin -p queue1 -v imprimante-m modèle`.
- `lpadmin -x file` : suppression de la file d'attente.
- `lpadmin -d file` : définir la file d'attente par défaut.
- `lpadmin -p file -u allow:liste` : autorisation d'imprimer pour les utilisateurs précisés.
- `lpadmin -p file -u deny:liste` : interdiction d'imprimer pour les utilisateurs précisés.
- `lpshut` arrête le service d'impression. Au redémarrage du démon, les impressions en cours au moment de l'arrêt sont reprises.

- `accept` et `reject` permettent de valider une file d'attente pour l'impression ou de la fermer.
- `lpmove` permet de transférer des requêtes d'impression d'une file d'attente vers une autre.

3. BSD

- `lpr [-Pimprimante] [-#copies] fic1` : imprime le contenu du fichier fic1. L'option `-P` permet de spécifier l'imprimante, `-#` le nombre de copies.
- `lpq [-Pimprimante]` : indique l'état et la liste des travaux pour l'imprimante éventuellement spécifiée par l'option `-P`.
- `lprm [-Pimprimante] [-] [ids]` : permet de supprimer un travail de l'imprimante spécifiée par l'option `-P`, l'option - supprime tous les travaux de l'utilisateur, ids représente une liste de travaux à supprimer.
- La commande **lpc** est une sorte de petit shell permettant de contrôler les imprimantes et les travaux.

Le service s'appelle généralement **lpd**.

4. CUPS

a. Présentation

CUPS (*Common Unix Printing System*) est un système d'impression Unix, orienté réseau :

- Basé sur le protocole **IPP** (*Internet Printing Protocol*) basé lui-même sur le protocole **HTTP/1.1**.
- Simple d'utilisation, notamment grâce à une configuration et une administration centralisée depuis une interface HTTP, des règles de conversion basées sur les types MIME, et des fichiers de description d'imprimante standards (**PPD**, *PostScript Printer Description*).
- CUPS reprend les commandes System V et BSD déjà abordées pour plus de simplicité.
- Les traces des impressions sont disponibles au format **CLF** (*Common Log Format*) de serveur Web et exploitables par les mêmes outils.
- CUPS est capable d'interagir avec les serveurs d'impression LPD pour garder une compatibilité ascendante.
- CUPS dispose de sa propre API permettant de créer des interfaces utilisateur pouvant s'intégrer dans des environnement graphiques ou des interfaces d'administration.
- Les pools d'impression permettent la redirection automatique des tâches.
- L'authentification est possible par utilisateur, hôte ou certificat numérique.

Le service d'impression se nomme **cupsd**.

```
$ ps -ef |grep cupsd
root      3128     1  0 Apr29 ?        00:00:01 /usr/sbin/cupsd
```

Il n'y a pas besoin d'outils graphiques pour administrer un serveur CUPS bien que pour des raisons de facilité la plupart des distributions, voire des environnements graphiques, en proposent. CUPS propose une interface d'administration WEB directement accessible depuis le port 631 du serveur. L'interface fonctionne avec n'importe quel navigateur HTTP.

`http://localhost:631`

Le fichier de configuration est **/etc/cups/cupsd.conf**. L'équivalent du fichier **/etc/printcap** est présent dans **/etc/cups/printers.conf**.

```
<DefaultPrinter lj2100>
Info Laserjet 2100
DeviceURI socket://192.168.1.10:9100
State Idle
StateTime 1203806079
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
</Printer>
```

b. Ajout d'une imprimante

Vous avez deux solutions pour ajouter une imprimante :

- Éditer les fichiers à la main.
- Passer par l'interface Web ou un outil de votre distribution.

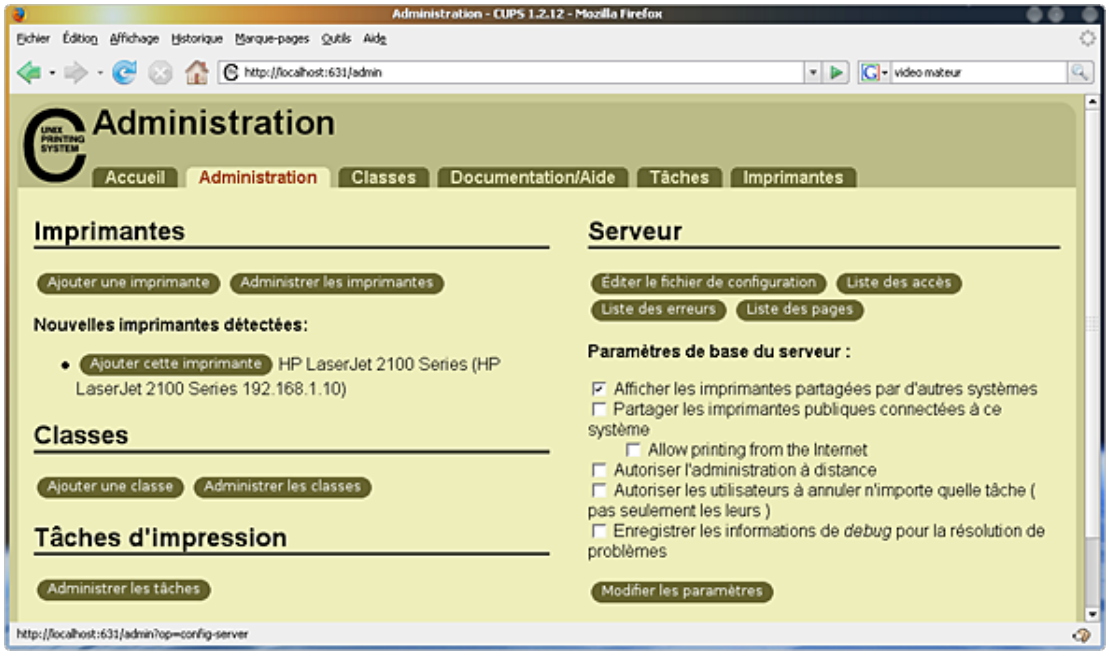
Dans le premier cas, vous devez modifier le fichier printers.conf pour ajouter une section pour votre imprimante puis copier dans /etc/cups/ppd le fichier PPD correspondant à votre imprimante et le renommer en utilisant le nom de section (ex : lj2100.ppd) du fichier printers.conf. Enfin vous devez rechercher la configuration de CUPS. Sous Red Hat ou openSUSE par exemple :

```
# service cups reload
```

Comme l'interface Web est généralement activée par défaut, vous pouvez passer par un navigateur Web. Il est possible que lors de l'accès aux pages d'administration des identifiants vous soient demandés. En principe ceux de root suffisent mais vous pouvez créer des comptes (ou rajouter des utilisateurs) chargés de la gestion des impressions avec la commande **lppasswd**.

```
# lppasswd -a seb
Enter password:
Enter password again:
```

- Sur la page principale cliquez sur l'onglet **Administration**.

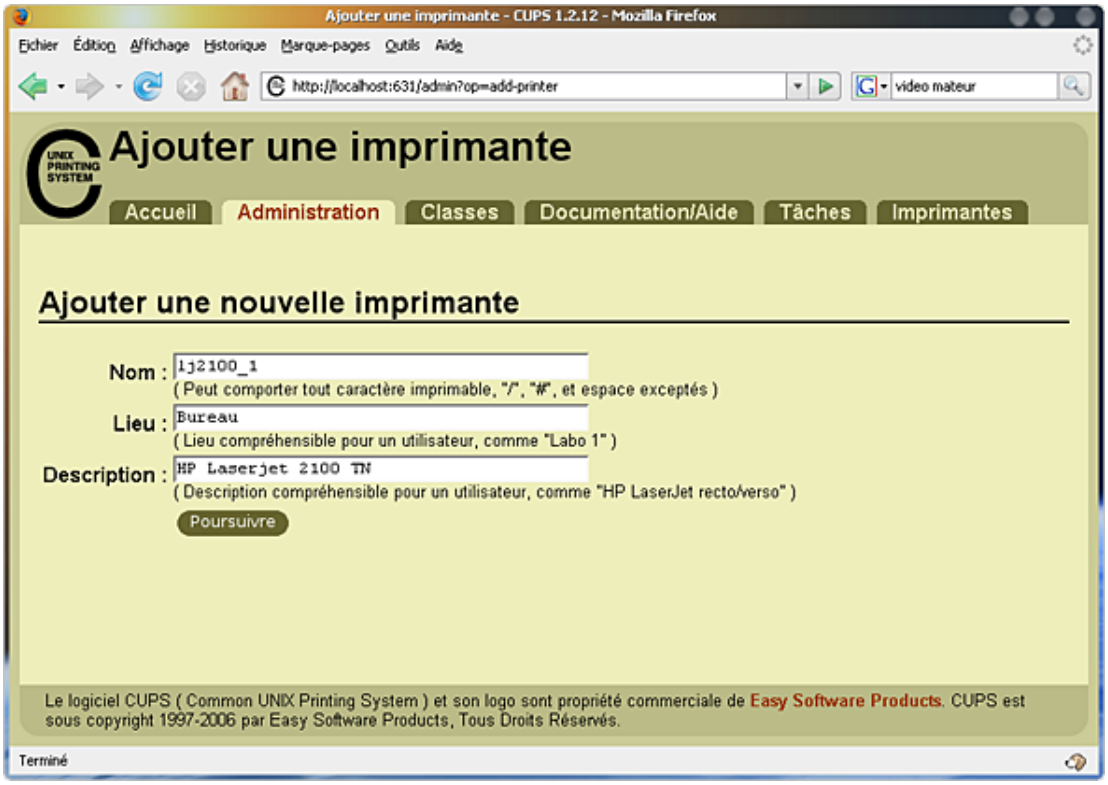


Accueil du frontend Web de CUPS

Il est possible que CUPS détecte les imprimantes locales ou sur le réseau lorsqu'elles proposent des services LPD (port 9100) ou IPP car le service est à l'écoute en permanence. Dans l'exemple en cours une imprimante a été détectée et le fait de cliquer sur **Ajouter cette imprimante** simplifie grandement la tâche.

Pour l'exemple cependant, basé sur l'ajout manuel d'une imprimante HP Laserjet 2100, vous allez ajouter une imprimante en cliquant sur **Ajouter une imprimante**.

- Saisissez les informations suivantes :
  - Le **nom** de l'imprimante est le nom de la file qui sera visible dans les outils d'impression. Vous ne devez pas mettre d'espaces.
  - Le **lieu** peut être laissé vide mais est utile pour la localiser (dans le cas d'une imprimante distante).
  - La **description** est ce que vous voulez, par exemple le modèle complet de l'imprimante.
- Cliquez ensuite sur **Poursuivre**.



Ajouter un imprimante, première étape

- L'étape suivante consiste à choisir votre type de connexion dans le menu matériel. L'imprimante est connectée en réseau. Dans ce cas sélectionnez **AppSocket/HP JetDirect**. Cliquez sur **Poursuivre**.

Ajouter une imprimante sur un port réseau

Pour une connexion réseau, vous avez plusieurs possibilités. La plupart des imprimantes connectées à un réseau de type Ethernet ou Wi-Fi proposent des services d'impression LPD ou Socket (impression directe). Dans ce dernier cas c'est l'imprimante elle-même qui gère les tâches d'impression. Pour les autres (IPP, http, Samba, etc.), veuillez

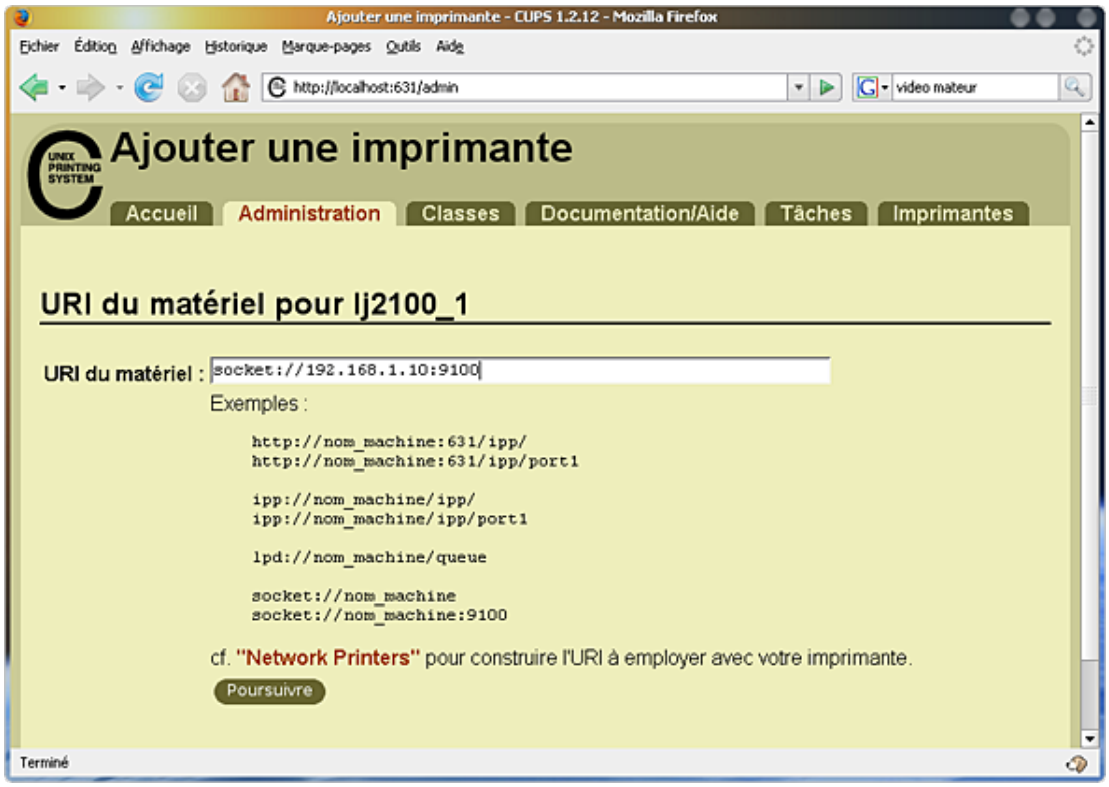
consulter le mode d'emploi de votre imprimante. Quelques imprimantes professionnelles proposent une interface de configuration Web (comme les routeurs grands public) pour paramétrer les ports.

Si votre imprimante distante est déjà configurée sur un autre serveur CUPS, vous pouvez passer par ce serveur et les **URI** de type http, ipp ou lpd pour imprimer dessus.

L'imprimante de test Laserjet 2100 dispose d'une carte réseau et d'un serveur d'impression jetDirect intégré. L'URI saisie est `socket://192.168.1.10:9100`.

L'IP est celle de l'imprimante sur le réseau, le port 9100 étant le port standard dans ce cas.

- Cliquez ensuite sur **Poursuivre**.



Adresse et port réseau de l'imprimante

- Choisissez maintenant un pilote correspondant à votre modèle d'imprimante. Vous remarquerez que pour plusieurs modèles il existe plusieurs pilotes. Celui recommandé est généralement indiqué comme tel (**recommanded**). Vous devriez vérifier quel pilote est réellement recommandé pour votre imprimante sur le site suivant :

  http://www.linux-foundation.org/en/OpenPrinting

  Qui informe que pour cette imprimante :

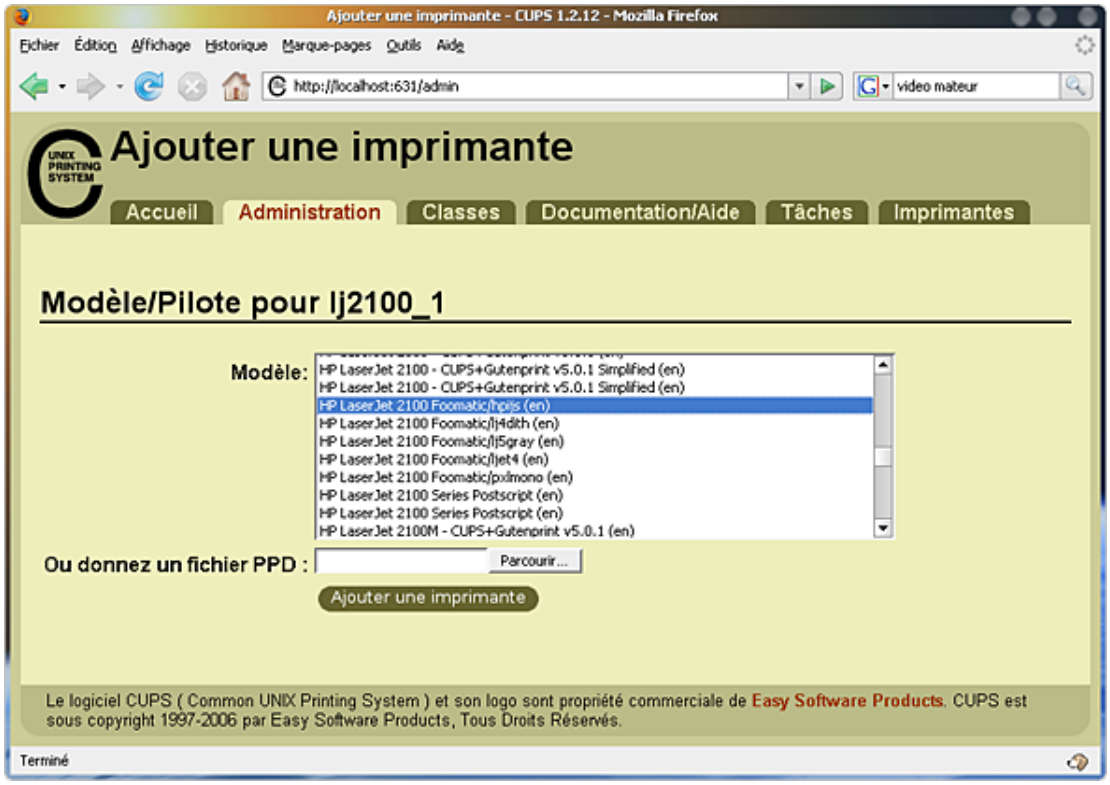
  For basic printing functionality use the Postscript PPD. For advanced functionality such as printer status and maintenance features, use the **HPLIP** driver (which includes **HPIJS**).

  Le choix se porte donc sur le pilote **HP Laserjet 2100 Foomatic/hpijs**.

Les imprimantes (et les scanners) de marque HP sont particulièrement bien supportés sous Linux au travers du projet libre de l'éditeur **HPLIP** (*HP Linux Imaging and Printing*), y compris les solutions intégrées (imprimante, scanner, fax, photocopieur, impression photo, etc.). La liste du matériel supporté est accessible via http://hplip.sourceforge.net/.

Vous pouvez préciser un autre fichier PPD. Plusieurs constructeurs proposent ce genre de fichiers pour leur imprimante. Les fichiers PPD sont présents par défaut dans /usr/share/cups/model.

- Cliquez sur **Ajouter une imprimante**.



Sélection du pilote d'impression

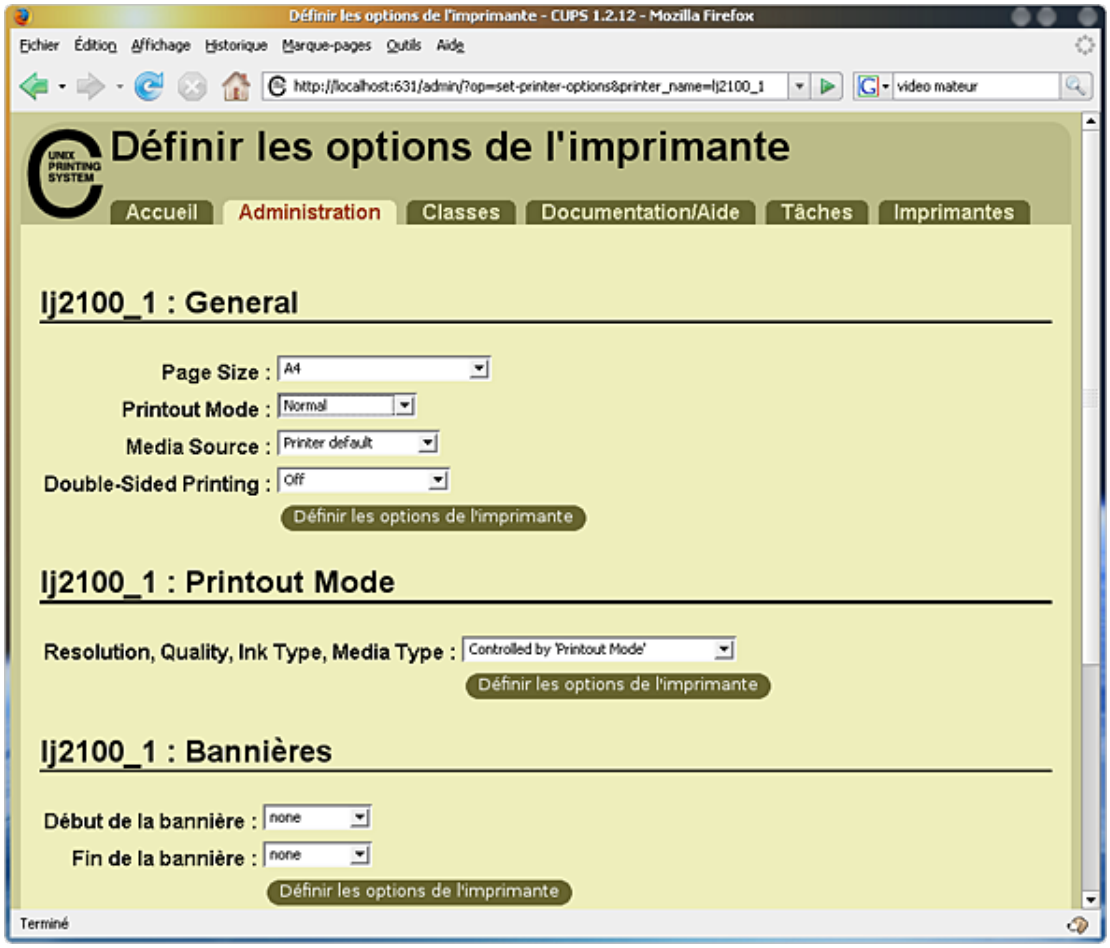
Vous arrivez sur l'écran présentant la liste des imprimantes actuellement configurées. Vous pouvez tester votre imprimante en imprimant une page de test. Vous pouvez aussi modifier les options par défaut de l'imprimante. Ces options sont de plusieurs types :

- Options générales pour indiquer le type de papier par défaut (A4), la qualité de sortie, le bac par défaut (les Laserjet 2100TN ont deux bacs et une entrée en façade), l'impression en Duplex (si le kit est présent), etc.

- Le mode de sortie ou les options diverses, pour régler par exemple la résolution et le type de papier par défaut, un filtrage quelconque, etc.

- Les bannières : pour séparer les impressions (cas des liasses par exemple) vous pouvez placer des bannières avant et après l'impression. Les fichiers des bannières peuvent être personnalisés dans certaines limites (voyez le manuel de CUPS pour cela).

- La facturation. CUPS peut être configuré pour gérer la facturation par service/machines/utilisateurs, dans le cas de l'utilisation de certains services avancés.

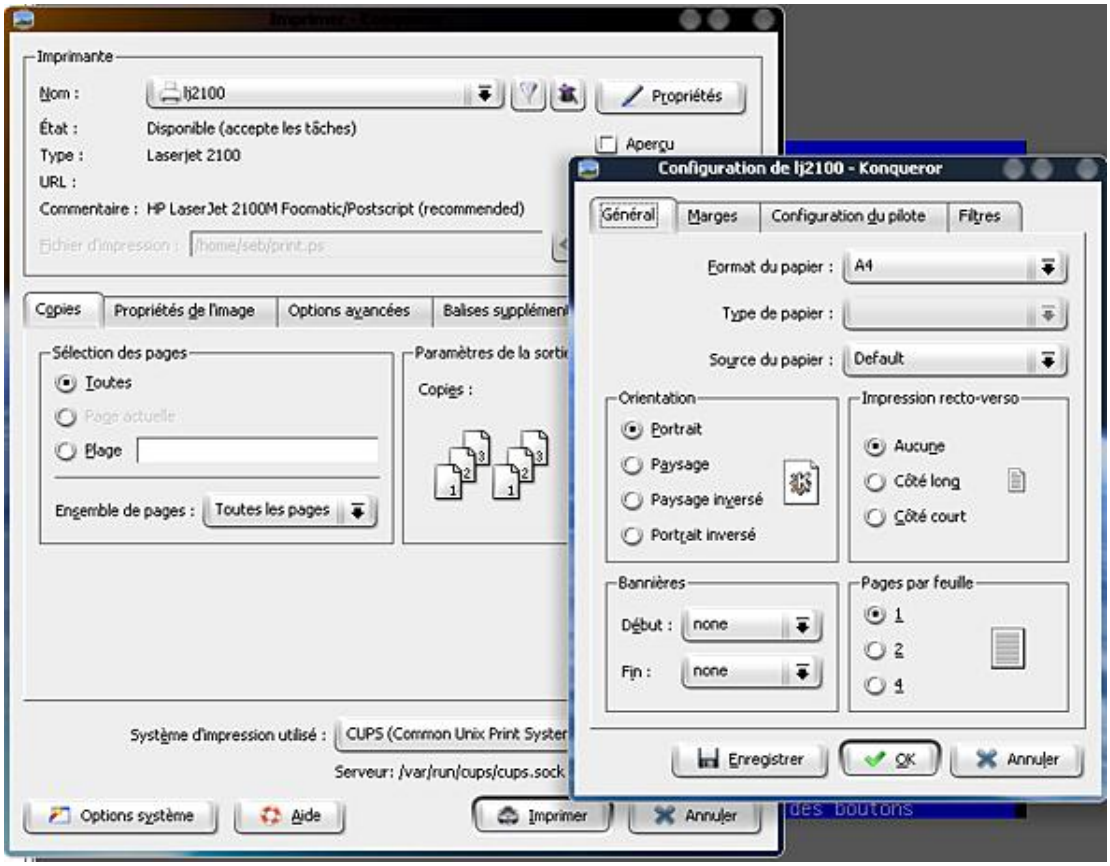


Réglages de l'imprimante

Vous devriez laisser, outre le format de papier, les options par défaut. La plupart des environnements de bureau et des logiciels proposent de modifier les options d'impression via CUPS comme sur les systèmes de type Windows.

La capture suivante a été effectuée sous KDE. Dans **Konqueror** l'entrée **Imprimer** du menu **Fichier** a été appelée lors de l'affichage d'une image. Toutes les options accessibles par le bouton **Propriétés** à côté du nom de l'imprimante sont issues des possibilités offertes par CUPS et le pilote d'impression PPD. D'une imprimante à l'autre et même d'un pilote à l'autre (si plusieurs sont disponibles) vous ne bénéficierez pas des mêmes options.

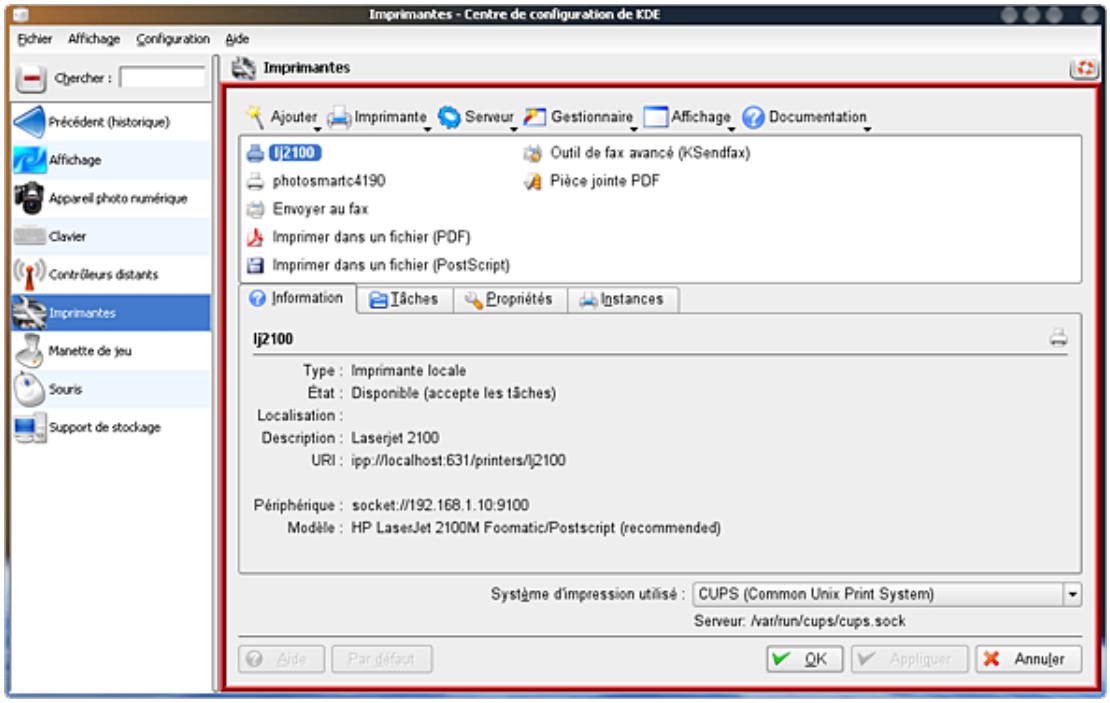
Si vous cliquez sur l'onglet **Configuration du pilote** vous obtiendrez le même choix d'options que ce que l'interface Web de CUPS vous propose.



Boîtes de dialogue d'impression sous KDE

Les distributions et les environnements graphiques proposent souvent leurs propres outils d'impression ou de gestion des imprimantes. Ainsi avec **KDE**, rendez-vous dans le **Centre de configuration**, **Périphériques** puis **Imprimantes**. Vous avez dès lors accès à toutes les imprimantes connues sur le système et même plus puisque certaines pseudo-imprimantes (qui ne sont pas gérées par CUPS) permettent d'imprimer dans des fichiers en PDF, d'envoyer une impression par mail, d'envoyer un fax, etc.

En cliquant sur **Mode superutilisateur** en bas à gauche et en saisissant votre mot de passe root, vous pouvez ajouter des imprimantes via le bouton associé en haut à gauche. Les étapes reprennent de manière mieux intégrée celles de CUPS.



Automatisation

1. Avec cron

a. Présentation

Le service **cron** permet la programmation d'événements à répétition. Il fonctionne à l'aide d'une table, appelée une **crontab**. C'est un fichier texte, éditable avec un simple éditeur, par exemple vi. Pour modifier votre crontab personnelle utilisez la commande **crontab** pour éditer la table, avec le paramètre -e.

Les fichiers crontabs sont sauvés dans /var/spool/cron.

Le service **cron** doit tourner pour que les crontabs soient actives.

```
$ ps -ef|grep cron
root      3634     1  0 18:28 ?        00:00:00 /usr/sbin/cron
```

b. Formalisme

Le format d'un enregistrement de crontab est le suivant :

Minutes	Heures	Jour du mois	Mois	Jour semaine	Commande
1	2	3	4	5	6

Utilisez le format suivant pour les valeurs périodiques :

- Une valeur pour indiquer quand il faut exécuter la commande. Ex : la valeur 15 dans le champ minute signifie la quinzième minute.
- Une liste de valeurs séparées par des virgules. Ex : 1,4,7,10 dans le champ mois pour janvier, avril, juillet, octobre.
- Un intervalle de valeurs. Ex : 1-5 dans le champ jour de la semaine indique du lundi (1) au vendredi (5). Le 0 est le dimanche et le 6 le samedi.
- Le caractère * pour toutes les valeurs possibles. Ex : * dans le champ jour du mois indique tous les jours du ou des mois.

c. Exemples

Exécution de df tous les jours, toute l'année, tous les quarts d'heure :

```
0,15,30,45 * * * * df > /tmp/libre
```

Exécution d'une commande tous les jours ouvrables à 17 heures :

```
0 17 * * 1-5 fin_travail.sh
```

Lister les crontabs actives :

```
$ crontab -l
```

Supprimer la crontab active :

```
$ crontab -r
```

Éditer la crontab d'un utilisateur particulier :

```
# crontab -u user
```

d. crontab système

La configuration crontab générale pour le système est dans /etc/crontab.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Ici tous les jours à 4h02 du matin **`run-parts /etc/cron.daily`** est exécuté. Le script **run-parts** accepte en paramètre un répertoire et exécute tous les programmes présents dans ce répertoire.

```
$ ls cron.daily/
00-logwatch  0anacron   makewhatis.cron  slocate.cron
00webalizer  logrotate  rpm              tmpwatch
```

Parmi les programmes exécutés, remarquez **logrotate** qui permet d'effectuer des sauvegardes et de renommer des fichiers logs et des journaux du système afin que ceux-ci ne deviennent pas inexploitables à cause de leur taille. Le programme **tmpwatch** est chargé de nettoyer le système des fichiers inutilisés (dans /tmp par exemple).

Enfin, le répertoire /etc/cron.d contient des crontabs supplémentaires.

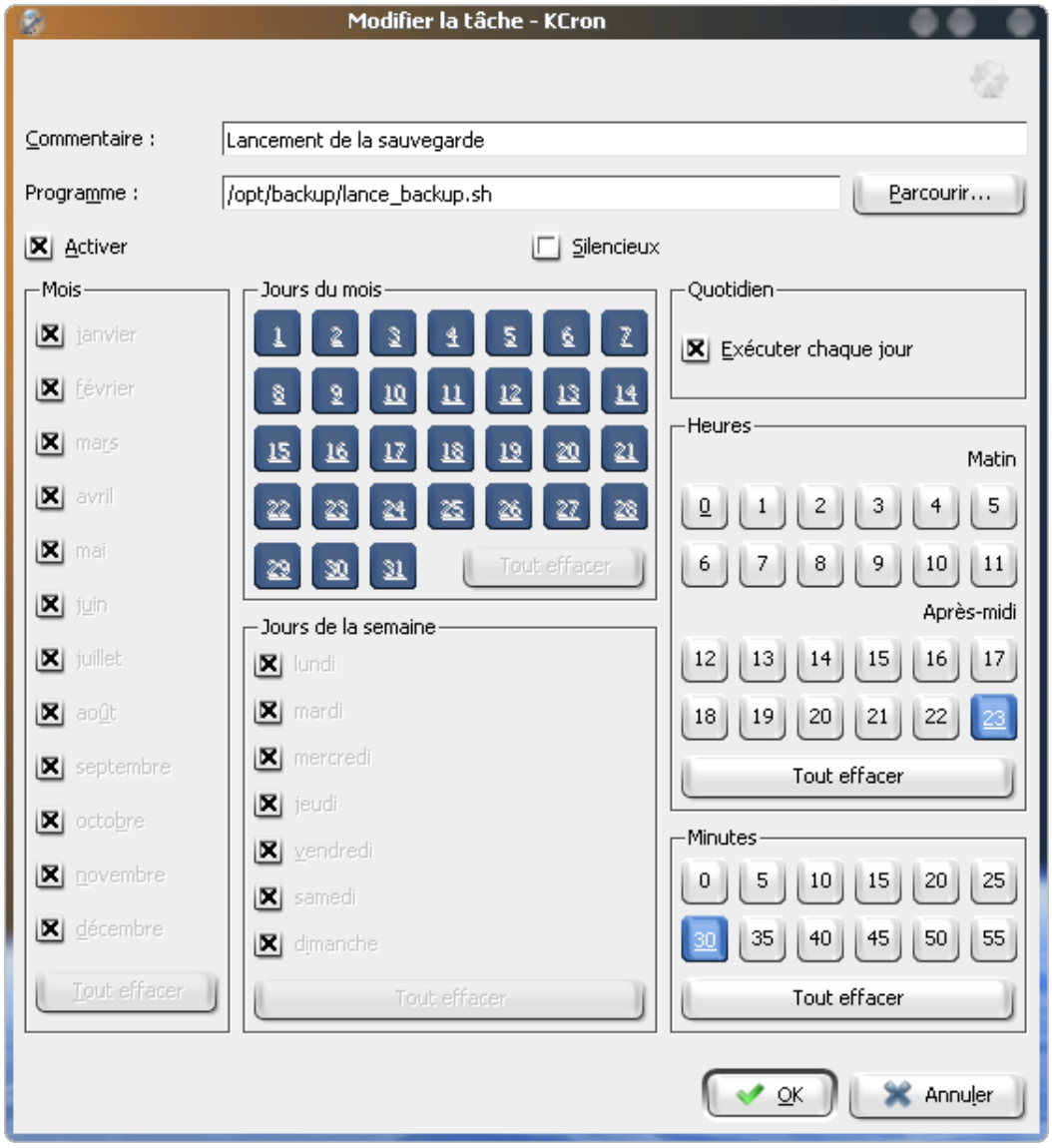
e. Contrôle d'accès

Vous pouvez contrôler l'accès à la commande **crontab** par utilisateur avec les fichiers **`/etc/cron.allow`** et **`/etc/cron.deny`**.

- Si cron.allow est présent, seuls les utilisateurs qui y sont explicitement indiqués peuvent utiliser **at** (cf. Automatisation - Avec at dans ce chapitre).
- Si cron.allow est absent, cron vérifie la présence d'un fichier cron.deny. Tous les utilisateurs n'y étant pas sont autorisés à utiliser cron. S'il est vide la commande **cron** est autorisée pour tout le monde.
- Si les deux fichiers sont absents, seul root peut utiliser cron.

f. Crontab en mode graphique

Quelques outils permettent d'éditer une crontab de manière visuelle sans passer par un éditeur de texte. L'outil kcron sous KDE est très populaire et très bien adapté pour cela.



L'outil kcron sous KDE

L'exemple de la capture présente un enregistrement de la crontab d'un utilisateur en cours d'édition. La tâche lance_backup.sh est exécutée tous les jours à 23h30. Les lignes correspondantes dans la crontab (au format texte) sont les suivantes :

```
# Lancement de la sauvegarde
30 23 * * *     /opt/backup/lance_backup.sh
```

2. Avec at

a. Présentation

La commande **at** et les commandes associées permettent une gestion des traitements batchs. Contrairement à la crontab les modifications sont volatiles : elles sont perdues lorsque la session est terminée. C'est à vous de placer la liste des commandes dans un éventuel fichier et de le charger au besoin via les scripts de votre profil.

Pour que at fonctionne le service **atd** (*at daemon*) doit fonctionner.

```
$ ps -ef | grep atd
at        7988     1  0 21:05 ?        00:00:00 /usr/sbin/atd
```

b. Formalisme

Pour simplifier, il y a deux moyens d'utiliser at :

- en lui passant de manière interactive une ligne de commande,
- en lui passant un fichier exécutable contenant les commandes à exécuter.

Dans les deux cas, vous devez fournir à at une heure d'exécution. Le formalisme de cette heure est assez souple. Pour programmer l'exécution d'une ligne de commande à 21h20 de manière interactive :

```
$ at 21:20
warning: commands will be executed using /bin/sh
at> echo salut
at> <EOT>
job 4 at 2008-05-08 21:20
```

Après avoir saisi la ou les commandes à exécuter à 21h20, appuyez sur [Entrée] et sur une ligne vide appuyez sur [Ctrl] **D** (fin de saisie). La commande **at** confirme la programmation de la commande.

Pour programmer l'exécution d'une commande (script ou binaire) à 21h25 :

```
$ at -f /home/seb/test.sh 21:25
warning: commands will be executed using /bin/sh
job 6 at 2008-05-08 21:25
```

Heure

L'heure peut être formatée ainsi :

- HHMM ou HH:MM.
- L'heure peut être au format 12 ou 24h. Au format 12 heures, vous pouvez préciser AM (matin) ou PM (après-midi).
- Midnight (minuit), noon (midi), teatime (16h00, typiquement anglais).
- MMJJAA, MM/JJ/AA ou JJ.MM.AA pour une date précise.
- Now : maintenant.
- + n minutes/hours/days/weeks : l'heure courante auquel on ajoute n minutes/heures/jours/semaines.

Si l'heure précisée est inférieure à l'heure actuelle, la commande est exécutée le lendemain.

```
$ at 21:30 09.05.2008
warning: commands will be executed using /bin/sh
at> echo salut !
at> <EOT>
job 9 at 2008-05-09 21:30
$ at now + 2 days
warning: commands will be executed using /bin/sh
at> echo dans deux jours
at> <EOT>
job 10 at 2008-05-10 21:29
```

Il existe aussi la commande **batch** qui ne prend pas d'heure. Elle exécute la commande dès que la charge de la machine l'autorise. L'heure peut être précisée, dans ce cas elle sera considérée comme « à partir de cet heure, dès que possible ».

c. Contrôle des tâches

La commande **atq** (*at queue*) permet de lister les tâches programmées :

```
$ atq
10      2008-05-10 21:29 a seb
9       2008-05-09 21:30 a seb
```

Les jobs (tâches) sont placées dans le répertoire /var/spool/atjobs, à raison de un exécutable par tâche.

```
# ls -l /var/spool/atjobs/
-rwx------ 1 seb users 5620 mai  8 21:29 a000090133cf92
-rwx------ 1 seb users 5628 mai  8 21:30 a0000a0133d531
```

Si vous regardez le contenu de l'exécutable, vous voyez que votre commande n'est pas seule. Elle est située à la fin, mais le script positionne tout l'environnement lors de la création de l'entrée at.

```
#cat a0000a0133d531
#!/bin/sh
# atrun uid=1000 gid=100
# mail      seb 0
umask 22
LESSKEY=/etc/lesskey.bin; export LESSKEY
NNTPSERVER=news; export NNTPSERVER
INFODIR=/usr/local/info:/usr/share/info:/usr/info; export INFODIR
MANPATH=/usr/local/man:/usr/share/man:/opt/gnome/share/man; export
MANPATH
KDE_MULTIHEAD=false; export KDE_MULTIHEAD
... (environ 80 lignes) ...
cd /home/seb || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
echo salut !
```

La commande **atrm** permet de supprimer une tâche :

```
$ atrm 10
$ atrm 9
$ atq
```

d. Contrôle d'accès

Vous pouvez contrôler l'accès à la commande **at** par utilisateur avec les fichiers **`/etc/at.allow`** et **`/etc/at.deny`**.

- Si at.allow est présent, seuls les utilisateurs qui y sont explicitement indiqués peuvent utiliser at.
- Si at.allow est absent, at vérifie la présence d'un fichier at.deny. Tous les utilisateurs n'y étant pas sont autorisés à utiliser at. S'il est vide la commande at est autorisée pour tout le monde.
- Si les deux fichiers sont absents, seul root peut utiliser at.

Les traces (logs) du système

1. Principe

Lorsque le système démarre, fonctionne et effectue tout type d'action, ses actions et celles de la plupart de ses services sont tracées dans divers fichiers. Deux services sont spécialisés dans la réception des messages à écrire dans ces fichiers :

- **klogd** : *kernel log daemon*, chargé de la gestion des informations émises par le noyau.
- **syslogd** : *system log daemon*, chargé de la gestion des informations émises par tout type de service et éventuellement le noyau.

Certaines distributions utilisent maintenant **syslog-ng** dont les règles de traitement des messages, basées sur des expressions régulières, ont fortement évolué. Le principe reste cependant exactement le même.

Historiquement le service syslogd gérait aussi les messages émis par le noyau. Il en est toujours capable, mais la quantité de messages émis, les différents niveaux de sévérité et les nouvelles méthodes d'accès aux messages du noyau font qu'il a semblé important et pertinent de séparer la gestion des messages du noyau de ceux émis par les services.

Les messages importants émis par un composant du système devraient passer par le service syslogd. Ceci n'empêche pas, au contraire, qu'un service puisse gérer ses propres traces dans ses propres fichiers. Les traces applicatives ne devraient pas être placées dans les traces de système. Les traces d'accès aux pages Web d'un serveur Apache n'ont rien à y faire. Par contre les traces de connexion au système (via la console, ssh, telnet, etc.) ont un intérêt important et doivent être présentes dans les fichiers logs du système.

Dans la suite de l'ouvrage, les traces seront appelées par leur nom d'usage courant : les **logs**.

2. Les messages

Le service **klogd** gère les messages émis par le noyau. Il dispose de deux sources d'accès aux messages :

- le système de fichiers virtuel /proc, utilisé par défaut s'il est présent, et notamment /proc/kmsg ;
- les appels systèmes via l'API du noyau, notamment sys_syslog, si /proc est absent ou si le paramètre `-s` a été passé à klogd.

Les messages du noyau ont des niveaux de priorité différents, étagés de 0 (haute priorité) à 7 (message de débogage) :

Niveau	Alias système	Signification
0	**EMERG**	Le système est inutilisable.
1	**ALERT**	Une action doit être prise immédiatement.
2	**CRIT**	Problème critique.
3	**ERR**	Erreur.
4	**WARNING**	Avertissement.
5	**NOTICE**	Normal mais nécessite une attention particulière.
6	**INFO**	Information standard.
7	**DEBUG**	Trace de débogage du noyau.

Le service klogd renvoie les messages de niveau 0 à 6 à syslogd qui redirigera ceux-ci dans les fichiers de logs et éventuellement sur les consoles concernées. Les informations de débogage de niveau 7 ne sont pas tracées par défaut.

Le service **syslogd** (ou syslog-ng) reçoit les messages issus des services mais aussi de klogd. Il les dispatche ensuite selon l'émetteur, la sévérité, dans des fichiers, des consoles, sous forme de mails aux utilisateurs du système (root par exemple), etc.

Les actions les plus courantes sont l'écriture des logs dans des fichiers, la redirection de messages sur une console (la 10 ou la 12 bien souvent) ou l'envoi de messages à root.

3. Configuration de syslog

Le fichier de configuration **`/etc/syslog.conf`** permet de définir l'origine, l'importance et la destination de chaque message, sous forme de deux champs.

- L'origine définit en fait un ensemble de **systèmes** et de **sous-systèmes** (noyau, services). La liste, extensible, est composée à l'origine des éléments suivants. L'étoile définit l'ensemble des sous-systèmes.

Sous-système	Signification
auth/authpriv	Service de sécurité et d'authentification.
cron	Service cron.
daemon	Les démons du système.
kern	Le noyau.
lpr	Le service d'impression.
mail	La messagerie.
news	Le réseau.
syslog	Syslog lui-même.
user	Messages des processus utilisateurs.
uucp	Unix to Unix CoPy.
local0->7	Messages issus de klogd, le chiffre représente le niveau.

- L'importance ou **niveau** définit le niveau de sévérité du message. L'étoile définit l'ensemble de tous les niveaux. Il y a équivalence entre les niveaux émis par klogd et syslogd.

Niveau	Signification
emerg	Le système est inutilisable.
alert	Une intervention immédiate est indispensable.
crit	Erreur critique pour le sous-système.
err	Erreur de fonctionnement.
warning	Avertissement.

notice	Évènement normal méritant d'être signalé.
info	Pour information seulement.
debug	Message envoyé pour la mise au point.
none	Ignorer les messages.

- La destination ou **action** peut être un fichier, un message à un utilisateur, la console, une liste d'utilisateurs... L'étoile indique tout le monde.

Les messages syslog sont inscrits dans les fichiers **/var/log/messages** et **/var/log/syslog** ou dans tout autre fichier paramétré dans **/etc/syslog.conf**.

L'exemple suivant provient d'une installation Red Hat AS 4u6.

```
# Tout (sauf mail.*) est place dans /var/log/messages
*.info;mail.none;authpriv.none;cron.none     /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                   /var/log/secure

# Mails
mail.*                                       -/var/log/maillog

# Crontab
cron.*                                       /var/log/cron

# Messages d'alerte
*.emerg                                      *

# erreurs uucp et news
uucp,news.crit                               /var/log/spooler

# Messages de boot
local7.*                                     /var/log/boot.log
```

Vous pouvez vous-même, directement ou dans vos scripts, envoyer des messages à **syslogd** par la commande **logger**.

4. Les fichiers de traces

Les logs systèmes sont situés par convention dans /var/log. Tous les logs de ce répertoire ne proviennent pas de syslogd. C'est le cas par exemple des informations de connexion. Voici un exemple du contenu de ce répertoire. Il contient plusieurs fichiers textes et des répertoires. Des services peuvent décider, sans passer par **syslogd**, de concentrer et d'écrire leurs messages dans cette arborescence.

```
# cd /var/log ; ls -l
-rw-r-----  1 root  root      2460 fev  7 05:34 acpid
drwxr-x---  2 root  root      4096 mar  5  2007 audit
-rw-------  1 root  root       116 mar 27 04:02 boot.log
-rw-------  1 root  root     75487 mar 28 11:10 cron
drwxr-xr-x  2 lp    sys       4096 mar 27 04:02 cups
-rw-r--r--  1 root  root     28359 fev  7 05:34 dmesg
drwx------  2 root  root      4096 aou  7  2007 httpd
-r--------  1 root  root  18747276 mar 28 11:08 lastlog
drwxr-xr-x  2 root  root      4096 jui  1  2007 mail
-rw-------  1 root  root      4537 mar 28 04:02 maillog
-rw-------  1 root  root    178348 mar 28 11:10 messages
drwx------  2 root  root      4096 oct 16 23:21 samba
-rw-------  1 root  root    214999 mar 28 11:08 secure
-rw-r--r--  1 root  root      2734 mar 28 11:01 snmpd.log
-rw-------  1 root  root         0 mar 23 04:02 spooler
drwxr-x---  2 squid squid     4096 jan 22  2007 squid
-rw-------  1 root  root     62165 mar 28 09:13 sudo.log
```

```
drwxr-xr-x  2 root  root      4096 oct  5  2004 vbox
-rw-rw-r--  1 root  utmp    127872 mar 28 11:10 wtmp
-rw-------  1 root  root     40557 mar 28 11:03 xferlog
```

Archivage et backup

1. Les outils de sauvegarde

La sauvegarde est un travail important de l'administrateur puisqu'en cas de gros problème, on passe généralement par une restauration du système depuis une sauvegarde, ou une image du système lorsque celui-ci était encore intègre (bon fonctionnement, pas de corruption). Chaque Unix est fourni avec des commandes et des procédures de sauvegarde qui lui sont propres. On distingue tout de même quelques outils communs.

a. Commandes, plans, scripts

- Pour la sauvegarde de fichiers et d'arborescences, utilisez les commandes **tar** et **cpio**. Ces commandes sauvent une arborescence, et pas un système de fichiers. On peut faire coïncider les deux.
- Pour la sauvegarde physique de disques et de systèmes de fichiers (des dumps), utilisez la commande **dd**.

Une sauvegarde incrémentale consiste à sauvegarder une première fois la totalité des données, puis ensuite uniquement les fichiers modifiés. On trouve aussi sous forme de logiciels libres ou dans le commerce des solutions plus pointues de sauvegarde (Networker par exemple).

L'administrateur aura parfois à définir des scripts de sauvegarde et de restauration adaptés au cas par cas (partition système, données applicatives...) et à automatiser quand c'est possible l'exécution de ceux-ci en fonction de la date, l'heure ou la charge de la machine.

Il sera aussi très important de définir un plan de sauvegarde, en se posant les bonnes questions :

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- Combien de temps conservera-t-on les sauvegardes, à quel endroit, en combien d'exemplaires ?
- À quel endroit sera stocké l'historique des sauvegardes ?
- Quel est le support le plus approprié ?
- Quels sont les besoins, en capacité, du support de sauvegarde ?
- Combien de temps prévoit-on pour sauvegarder un fichier, un système de fichiers et est-ce raisonnable ?
- La sauvegarde doit-elle être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?

Chaque cas étant unique, cet ouvrage ne peut répondre à toutes ces questions. Les réponses dépendent de l'environnement cible (production, intégration, tests, etc.). Cependant envisagez toujours la mise en place d'une sauvegarde système (racine, /opt, /usr, /var, /boot, etc.) après une installation et avant une modification importante, au cas où il faudrait revenir en arrière.

b. Voici quelques autres commandes

- **mt** : contrôle d'une bande magnétique.
- **touch** : met la date de dernière modification à l'heure actuelle, pour forcer une sauvegarde incrémentale.
- **find** : sélectionne les fichiers à sauvegarder.

- **compress** et **uncompress** : compression et décompression des fichiers.
- **gzip**, **gunzip**, **zcat**, compression et décompression au format GnuZip.

2. Tar

La commande **tar** est simple et efficace. Elle crée des archives des fichiers, y compris l'arborescence de fichiers, sur tout type de support y compris dans une autre fichier (archive à l'extension .tar). L'archive ainsi créée peut s'étendre sur plusieurs volumes : quand la bande ou la disquette est pleine, c'est à l'utilisateur d'en insérer une nouvelle et la sauvegarde/restitution continue.

a. Archiver

La syntaxe est la suivante :

```
tar cvf nom_archive Fichier(s)
```

Par exemple pour placer dans une archive tar le répertoire Desktop :

```
$ tar cvf desktop.tar Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

Les paramètres sont les suivants :

- c : création d' archive,
- v : mode bavard, tar indique ce qu'il fait,
- f : le paramètre suivant est le nom de l'archive.

b. Lister

La syntaxe est :

```
tar tvf nom_archive
```

Pour lister le contenu de l'archive précédente :

```
$ tar tvf desktop.tar
drwx------ seb/users       0 2008-04-17 09:44 Desktop/
-rw-r--r-- seb/users     191 2007-10-20 20:10 Desktop/fusion-icon.desktop
-rw-r--r-- seb/users    4786 2007-09-26 00:43 Desktop/konsole.desktop
-rw-r--r-- seb/users     665 2008-04-08 15:14 Desktop/Support.desktop
-rw-r--r-- seb/users    1051 2007-10-05 10:16 Desktop/Office.desktop
-rw-r--r-- seb/users    4586 2007-12-05 11:37 Desktop/Terminal.desktop
-rw-r--r-- seb/users     829 2007-10-17 12:12 Desktop/MozillaFirefox.desktop
-rw-r--r-- seb/users    3952 2007-10-05 10:16 Desktop/Printer.desktop
-rw-r--r-- seb/users    2053 2007-10-05 10:16 Desktop/.directory
-rw-r--r-- seb/users     450 2007-10-23 11:58 Desktop/myComputer.desktop
-rw-r--r-- seb/users     218 2008-02-22 08:43 Desktop/trash.desktop
```

```
-rw-r--r-- seb/users     328 2008-04-08 15:14 Desktop/SuSE.desktop
-rw-r--r-- seb/users     472 2008-04-17 09:44 Desktop/Windows.desktop
```

Le paramètre t liste le contenu de l'archive.

c. Restauration

Pour restaurer le contenu d'une archive la syntaxe est :

```
tar xvf nom_archive fichiers
```

Pour restaurer l'archive précédente :

```
tar xvf desktop.tar
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

Le paramètre x permet l'extraction de l'ensemble des fichiers de l'archive, ou du ou des fichiers spécifiés à la suite du nom de l'archive.

d. Autres paramètres

La commande **tar** de gnu permet de gérer les formats de compression directement :

- z : l'archive est compressée au format gzip.
- Z : l'archive est compressée au format compress.
- j : l'archive est compressée au format bzip2.

Ainsi les commandes précédentes pour le format de compression gzip deviennent :

```
$ tar cvzf desktop.tar.gz Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
$ ls -l desktop.tar*
-rw-r--r-- 1 seb users 30720 mai  9 11:16 desktop.tar
-rw-r--r-- 1 seb users  7556 mai  9 11:22 desktop.tar.gz
```

Notez la différence de taille. Les options de compression peuvent être utilisées avec c, t et x. Notez que c'est l'archive finale qui est compressée, par les fichiers individuellement. Il peut être préférable de ne pas spécifier d'option de compression si vous sauvez sur une bande dont le lecteur gère lui-même la compression.

Si votre archive est compressée et qu'elle est à destination d'un autre système, ou que vous souhaitez garder une compatibilité avec les paramètres par défaut de tar, vous pouvez procéder comme ceci :

```
$ gzip -cd desktop.tar.gz | tar xvf -
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

Le paramètre `-d` précise à **gzip** de décompresser le fichier, tandis que `-c` passe le résultat par la sortie standard. Le `-` final indique à tar de récupérer le flux par l'entrée standard.

3. cpio

La commande **cpio** sauvegarde sur la sortie standard les fichiers dont on saisit les noms sur l'entrée standard, par défaut l'écran et le clavier. Vous devez donc utiliser les redirections. Cpio ne compresse pas les archives. C'est à vous de le faire.

a. Archiver

La syntaxe générale est :

```
cpio -oL
```

Les paramètres les plus utilisés sont :

- `-o` : output, création de la sauvegarde en sortie.
- `-L` : sauve les fichiers liés et pas les liens symboliques.
- `-v` : mode bavard « verbose », informations détaillées.
- `-c` : sauvegarde des attributs des fichiers sous forme ASCII (pour l'échange entre divers OS).

Voici comment archiver et compresser le répertoire Desktop :

```
find Desktop -print| cpio -ocv | gzip > archive.cpio.gz
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
> ls -l archive.cpio.gz
-rw-r--r-- 1 seb users 7377 mai  9 11:33 archive.cpio.gz
```

b. Lister

La syntaxe générale est :

```
cpio -it archive
```

Les paramètres sont :

- `-i` : lecture de l'archive en entrée.
- `-t` : comme pour tar, liste le contenu de l'archive.

```
$ cat archive.cpio.gz | gzip -cd | cpio -it
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
```

c. Restaurer

La syntaxe générale est :

```
cpio -i[umd]
```

- `-u` : restauration inconditionnelle, avec écrasement des fichiers qui existent déjà. Par défaut les fichiers ne sont pas restaurés si ceux présents sur le disque sont plus récents ou du même âge.
- `-m` : les fichiers restaurés conservent leur dernière date de modification.
- `-d` : cpio reconstruit l'arborescence des répertoires et sous-répertoires manquants.

Pour restaurer l'archive précédente :

```
$ cat archive.cpio.gz | gzip -cd | cpio -iuvd
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
```

4. dd

La commande **dd** (*device to device*) est destinée à la copie physique, bloc par bloc, d'un fichier périphérique vers un fichier périphérique ou quelconque. À l'origine elle était utilisée pour la lecture et l'écriture sur bande magnétique, mais elle peut être employée avec n'importe quel fichier. La commande **dd** permet de réaliser des copies physiques de disques et de systèmes de fichiers.

Argument	Rôle
if=fichier	Nom du fichier en entrée (celui à copier).
of=fichier	Nom du fichier en sortie.
bs=n	Taille du bloc en octets.
count=n	Nombre de blocs à copier.
skip=n	Nombre de bloc à sauter au début du fichier d'entrée.
conv=	Conversion de l'entrée.
seek=	Nombre de blocs à sauter au début du fichier de sortie.
-s	Shell (commande de connexion) par défaut de l'utilisateur (variable SHELL). L'utilisateur peut le changer via la commande **chsh**.
-p	Le mot de passe de l'utilisateur. Attention ! le mot de passe doit déjà être crypté ! Aussi à moins de recopier le mot de passe d'un compte générique, vous préférerez utiliser ensuite la commande **passwd**.

L'option **conv** admet les paramètres suivants :

- `ascii` : convertir l'EBCDIC en ASCII.
- `ebcdic` : convertir l'ASCII en EBCDIC.
- `block` : compléter les blocs se terminant par un saut de ligne avec des espaces, jusqu'à atteindre la taille mentionnée par bs.
- `unblock` : remplacer les espaces en fin de blocs (de taille cbs) par un saut de ligne.
- `lcase` : transformer les majuscules en minuscules.
- `ucase` : transformer les minuscules en majuscules.
- `noerror` : continuer même après des erreurs de lecture.
- `notrunc` : ne pas limiter la taille du fichier de sortie.
- `sync` : compléter chaque bloc lu avec des NULs pour atteindre la taille ibs.

Ici vous allez placer le secteur de boot de la partition (où est installé lilo ou grub) dans un fichier. Le fichier ainsi créé pourra être utilisé avec le chargeur de NT/2000/XP pour démarrer sous Linux.

```
# dd if=/dev/sda1 of=boot.lnx bs=442 count=1
```

Pour créer un fichier vide d'une taille de 1Mo :

```
$ dd if=/dev/zero of=vide bs=1024 count=1024
1024+0 enregistrements lus
1024+0 enregistrements écrits
1048576 bytes (1,0 MB) copied, 0,0199192 s, 52,6 MB/s
```

L’horloge

1. Connaître l’heure

a. date

Pour connaître l’heure, utilisez la commande **date**. Elle permet de donner la date actuelle, mais aussi de calculer d’autres dates en fonction soit de la date actuelle, soit en fonction d’une date quelconque. Date permet aussi de modifier la date et l’heure du système

```
$ date
sam mai  10 13:58:38 CEST 2008
```

Par défaut la date affichée est la date (et l’heure) locale, configurée en fonction du fuseau horaire. Pour afficher l’heure UTC :

```
$ date --utc
sam mai  10 11:01:10 UTC 2008
```

Le format de la date peut être modifié à volonté à l’identique de ce qui peut se faire avec la fonction C strftime. Dans ce cas la syntaxe est :

```
date +"format".
```

Voici quelques exemples de format possible :

Format	Résultat
%H	L’heure au format 00..23.
%M	Minutes 00..59.
%S	Secondes 00..60.
%T	Heure actuelle sur 24 heures.
%r	Heure actuelle sur 12 heures.
%Z	Fuseau horaire.
%a	Jour abrégé (lun, mar, etc.).
%A	Jour complet.
%b	Mois abrégé.
%B	Mois complet.
%d	Jour du mois.
%j	Jour de l’année.
%m	Numéro du mois.
%U	Numéro de la semaine 00..53.
%y	Deux derniers chiffres de l’année.
%Y	Année complète.

Pour afficher une date complète :

```
$ date +"Nous sommes le %A %d %B %Y, il est %H heures, %M minutes et
%S secondes"
Nous sommes le samedi 10 mai 2008, il est 14 heures, 10 minutes et
20 secondes
```

Vous pouvez modifier la base de calcul en passant le paramètre `--date` suivi d'une date ou d'un calcul. Les mots clés today, yesterday, tomorrow, day(s), week(s), month(es), year(s), hour(s), minute(s), second(s) sont acceptés, avec + (ajout à la date) ou - ou ago (retranche à la date précisée). Si la date n'est pas précisée, c'est la date en cours.

Dans 10 jours :

```
date --date "10 days"
mar mai 20 13:13:05 CEST 2008
```

Demain :

```
date --date "tomorrow"
```

Hier :

```
date --date "yesterday"
```

Une semaine après Noël 2008 :

```
date --date "12/25/2008 23:59:00 + 1 week"
jeu jan  1 23:59:00 CET 2009
```

b. hwclock

La commande **hwclock** permet d'interroger directement l'horloge matérielle RTC. Le paramètre `--show` (par défaut) affiche la date actuelle. Elle est différente du temps système provenant de ntp ou date. La fin de l'affichage donne d'ailleurs le décalage.

```
# hwclock --show
sam 10 mai 2008 13:16:14 CEST  -0.390436 secondes
```

Il n'est pas possible de formater le résultat de la commande.

2. Modifier l'horloge matérielle

L'horloge matérielle peut être modifiée uniquement en tant que root via les commandes **date** (horloge système interne au noyau) et **hwclock** (horloge matérielle).

a. Via date

Modifiez la date et l'heure en passant le paramètre `-s` :

```
# date -s "05/09/2008 14:00"
ven mai  9 14:00:00 CEST 2008
# date
ven mai  9 14:00:03 CEST 2008
```

b. Via hwclock

La commande **hwclock** modifie l'horloge matérielle (RTC) et/ou l'horloge système. L'heure matérielle étant indépendante de l'heure système, les résultats peuvent surprendre :

```
# hwclock --set --date "05/10/2008 14:00"
```

```
# date
sam mai  10 13:30:53 CEST 2008
# hwclock
sam 10 mai 2008 14:00:13 CEST  -0.658230 secondes
```

Vous pouvez synchroniser l'heure système et l'heure matérielle dans les deux sens. Pour que l'heure matérielle soit synchronisée à partir de l'heure système :

```
# hwclock --systohc
# hwclock
sam 10 mai 2008 13:34:00 CEST  -0.931220 secondes
```

Pour effectuer l'inverse :

```
# hwclock --hctosys
```

3. NTP

a. Principe

NTP (*Network Time Protocol*) est un protocole qui permet de synchroniser les horloges des ordinateurs via le réseau, notamment TCP/IP et donc Internet. Nos ordinateurs utilisant des horloges au quartz, celles-ci, selon la qualité des composants, peuvent parfois fortement et rapidement avancer ou retarder.

Il y a de nombreux domaines où il est inadmissible de ne pas avoir un système à l'heure notamment pour des raisons de synchronisation très précises.

Un serveur NTP diffuse l'heure au format UTC. Le client récupère l'heure et l'adapte en fonction de son fuseau horaire. Le serveur ne gère pas non plus les changements d'heure.

Pour peu que le serveur NTP soit à jour, l'heure est très précise. Elle est codée sur 64 bits :

- les 32 premiers bits donnent le nombre de secondes depuis le 1er janvier 1900 à minuit (donc le bug NTP aura lieu avant le bug Unix) ;
- les 32 derniers bits donnent la précision des secondes.

Le nouveau protocole NTP4 donne une précision des secondes sur 64 bits, évitant ainsi un bug gênant dans le futur.

Vous trouverez à l'URL suivante une liste de serveurs NTP français.

http://www.cru.fr/services/ntp/serveurs_francais0

b. Client ntp

Le service ntpd permet de synchroniser une machine auprès d'un serveur de temps.

```
$ ps -ef|grep ntp
root      6523  5378  0 14:04 pts/2    00:00:00 ntpd
```

Le fichier de configuration est **/etc/ntpd.conf**. En principe ce fichier contient déjà un certain nombre de lignes qu'il faut éviter de toucher. Vous pouvez, voire devez, rajouter une ligne pointant sur le serveur de temps que vous avez choisi (par exemple chronos.espci.fr) :

```
server chronos.espci.fr
```

Relancez le service. Votre machine doit se mettre à l'heure.

Vous pouvez forcer une synchronisation manuelle avec la commande **ntpdate**. Celle-ci prend pour paramètre un nom de serveur ntp.

```
# ntpdate chronos.espci.fr
10 May 14:09:21 ntpdate[6551]: adjust time server 193.54.82.20 off-
set 0.154057 sec
```

Si vous ne souhaitez pas utiliser le service ntpd, vous pouvez placer cette commande en crontab tous les jours, ou toutes les heures.

Les paramètres régionaux

1. i18n et l10n

Les distributions s'installent et fonctionnent dans de nombreuses langues. Vous aurez probablement remarqué que les pages des manuels s'affichent aussi dans votre langue, pour peu que ces pages aient été traduites. De même, un grand nombre de logiciels affichent leur menu dans la langue dans laquelle le système a été installé ou paramétré.

Contrairement à de nombreux éditeurs de logiciels qui fournissent des versions localisées différenciées de leurs logiciels (par exemple MS Office en français et MS Office en anglais sont deux versions distinctes), les éditeurs de logiciels libres intégrent généralement directement un support pour de nombreux langages, ou fournissent des packages additionnels.

Il faut faire une différence entre la régionalisation (en anglais localization) et l'internationalisation :

- La **régionalisation** consiste à fournir une traduction d'un produit correspondant à la culture locale (langue principalement, mais aussi monnaie ou représentation des nombres). À titre d'exemple, si le français est utilisé en France, en Belgique, en Suisse, au Canada ou dans de nombreux pays d'Afrique, chaque culture dispose de syntaxes pouvant varier (il suffit d'entendre parler un Québequois, on entend bien que c'est du français, mais on ne comprend pas tout) et des monnaies différentes.

- L'**internationalisation** prépare la régionalisation en amont, au niveau du développement du logiciel. Un seul logiciel fera appel à des fonctions d'une API d'internationalisation. Une fonction chargée de récupérer une chaîne de texte dans n'importe quelle langue au sein d'une base vérifiera quel langage est utilisé et sortira la chaîne correspondante. La bibliothèque de fonctions `gettext` est un bon exemple.

Au lieu d'utiliser un modèle figé comme ceci :

```
printf("Hello");
```

Un programmeur pourra procéder comme cela :

```
printf(gettext("Hello"));
```

La fonction `gettext` va rechercher la chaîne correspondant à "Hello" dans les fichiers de régionalisation du programme dans la langue actuelle et en sortira une version traduite. En France la seconde ligne sortira donc :

```
Bonjour
```

Par souci de concision, internationalisation s'écrit généralement i18n (i suivi de 18 lettres puis n) et régionalisation l10n (l, pour localization, suivi de 10 lettres puis n).

2. Réglages locaux

a. Outils de la distribution

Il y a plusieurs moyens de modifier la régionalisation du système, voire d'un programme donné. Chaque distribution fournit un module de configuration. OpenSUSE fournit par exemple au sein de YaST un module de gestion de la langue et la même chose peut être faite avec Debian ainsi :

```
dpkg-reconfigure locales
```

Ces méthodes ne font que modifier des paramètres systèmes facilement accessibles. Il est parfaitement possible de les modifier à la main via les fichiers de configuration texte. De même, chaque environnement de bureau fournit des outils de régionalisation via leur centre de configuration. Il est possible, pour un logiciel donné, de passer de l'un à l'autre.

b. Variables d'environnement

Sous la console, ou dans l'environnement de bureau si la régionalisation est laissée par défaut, les informations sur le pays, la langue ou la monnaie utilisés sont récupérées grâce à des variables d'environnement du shell. La principale se nomme **LANG**. Elle est généralement positionnée dans /etc/profile ou d'autres fichiers (**/etc/sysconfig/language** sur openSUSE par exemple) et son contenu, sur le poste de l'auteur, est le suivant :

```
LANG=fr_FR.UTF-8
```

Son formalisme est le suivant :

```
langue_Pays[.norme][@variante]
```

- **langue** est la langue utilisée (ici fr pour france).
- **Pays** est le pays, en majuscules (ici FR).
- **.norme** indique la norme utilisée pour le codage des caractères (ici en UTF-8).
- **@variante** précise une variante de la langue. Sur certains système qui utisent le codage ISO-8859-15 on peut préciser @euro pour indiquer une variante supportant l'euro de ce codage.

Pour passer en anglais américain :

```
export LANG=en_US.UTF-8
```

La commande locale permet de récupérer des informations sur les éléments de régionalisation supportés par votre système :

```
seb@slyserver:/etc/sysconfig> locale -a|grep fr
fr_BE
fr_BE@euro
fr_BE.utf8
fr_CA
fr_CA.utf8
fr_CH
fr_CH.utf8
fr_FR
fr_FR@euro
fr_FR.utf8
fr_LU
fr_LU@euro
fr_LU.utf8
```

Les éléments de régionalisation supportés par le système sont dans **/usr/share/locale**.

La variable LANG surclasse par défaut les autres variables d'environnement de régionalisation, sauf si elles existent. Sur de nombreux Unix dont Linux, il existe un grand nombre de variables commençant par LC_. Si LANG est la seule variable, elle remplace toutes ces variables.

Si vous appelez la commande locale sans paramètre, elle affiche ceci :

```
seb@slyserver:/etc/sysconfig> locale

LANG=fr_FR.UTF-8
LC_CTYPE="fr_FR.UTF-8"
LC_NUMERIC="fr_FR.UTF-8"
LC_TIME="fr_FR.UTF-8"
LC_COLLATE="fr_FR.UTF-8"
LC_MONETARY="fr_FR.UTF-8"
LC_MESSAGES="fr_FR.UTF-8"
LC_PAPER="fr_FR.UTF-8"
LC_NAME="fr_FR.UTF-8"
LC_ADDRESS="fr_FR.UTF-8"
LC_TELEPHONE="fr_FR.UTF-8"
LC_ALL=
```

Chacune des variables LC peut être modifiée et adaptée. Voici leur signification :

- LC_CTYPE : classe des caractères et conversion.
- LC_NUMERIC : format numérique par défaut, autre que pour la monnaie.
- LC_TIME : format par défaut de la date et de l'heure.
- LC_COLLATE : règles de comparaison et de tri (par exemple pour les caractères accentués).
- LC_MONETARY : format monétaire.
- LC_MESSAGES : format des messages informatifs, interactifs et de diagnostic.
- LC_PAPER : format de papier par défaut (par exemple A4).
- LC_NAME : format du nom d'une personne.
- LC_ADDRESS : idem pour une adresse.
- LC_TELEPHONE : idem pour le téléphone.
- LC_ALL : règles pour toutes les autres variables LC.

Voici un exemple avec une modification de LC_TIME :

```
seb@slyserver:~> date
ven. mai  8 11:28:36 CEST 2009
seb@slyserver:~> export LC_TIME=en_US
seb@slyserver:~> date
Fri May  8 11:28:52 CEST 2009
```

Dans certains cas, il est nécessaire d'exporter la variable LANG=C au sein d'un script par exemple. Ceci a pour effet de rétablir la régionalisation native des commandes, généralement en anglais, ou de type POSIX, et ce afin de garantir un fonctionnement optimal des commandes concernées.

3. Codage des caractères

Tout le monde n'utilise pas l'alphabet occidental. Et même dans les pays occidentaux, de nombreuses variantes d'un même alphabet existent. Les anglais n'utilisent pas les accents, les allemands ont des caractères en plus, d'autres pays ont des ponctuations supplémentaires. C'est encore plus compliqué avec les pays n'utilisant pas le même alphabet ou utilisant des idéogrammes.

Les noms des fichiers peuvent être illisibles dans un autre pays. Le contenu texte d'un fichier aussi.

Chaque variante nécessite l'application d'une variante pour le codage des caractères. Pendant longtemps la table était dite ASCII, il s'agit d'une table de caractères codée sur 1 octet. Cette table peut donc contenir 256 caractères. Les 127 premiers étant invariables ils contiennent tous les caractères américains, c'est-à-dire aucun caractères accentué, ainsi que les caractères de contrôle et de formatage (suppression, retour chariot, etc.). Les 128 suivants sont libres. En France, on y place des caractères semi-graphiques et les accents. C'est le cas pour d'autres pays. Ceci signifie que pour chaque pays, il faut changer de table ASCII et que surtout, si vous accédez à un mot ou un nom de fichier créé dans un pays et contenant des caractères propres à ce pays, il y a un risque important que les noms ne correspondent plus du tout.

Ces tables sont normalisées par l'ISO. Par exemple pour la France, la table ISO 8859-1 correspond aux caractères de l'europe de l'ouest sur Unix, et ISO 8859-15 après le passage à l'euro. Sous Windows c'est la table Windows-1252, sous DOS CP850, etc. Chaque table est incompatible avec les autres. Les polices de caractères doivent être adaptées pour chaque table. Pire, les 256 caractères ne suffisent pas à coder tous les idéogrammes.

Pour remédier à ce problème, il existe une norme informatique appelée unicode, qui est compatible avec le standard ISO 10646. Elle donne à tout caractère de n'importe quel système d'écriture de langue un nom et un identifiant numérique, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel (définition inspirée de Wikipédia). Si un système prend en charge unicode et dispose des polices de caractères associées, alors les textes et noms de fichiers seront correctement affichés, quels que soient la langue et les caractères utilisés. Un nom de fichier

en chinois sera correctement affiché sur un système en français.

Linux, ainsi que de nombreuses pages Internet, utilisent le format unicode UTF-8 par défaut. Ce format est maintenant correctement géré par les systèmes les plus courants.

Vous pouvez convertir un fichier codé dans une table donnée vers une autre table avec le programme **iconv**. Le paramètre -l vous donne toutes les tables supportées. La sortie est tronquée car elle occuperait 4 pages.

```
slyserver:/usr/share/zoneinfo # iconv  -l
...
 437, 500, 500V1, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863,
864, 865,  866, 866NAV, 869, 874, 904, 1026, 1046, 1047, 8859_1,
8859_2, 8859_3, 8859_4,  8859_5, 8859_6, 8859_7, 8859_8, 8859_9,
10646-1:1993, 10646-1:1993/UCS4,
...
  ISO646-ES2, ISO646-FI, ISO646-FR, ISO646-FR1, ISO646-GB, ISO646-HU,
ISO646-IT, ISO646-JP-OCR-B, ISO646-JP, ISO646-KR, ISO646-NO,ISO646-NO2,
...
  WINDOWS-31J, WINDOWS-874, WINDOWS-936, WINDOWS-1250, WINDOWS-1251,
WINDOWS-1252, WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256,
WINDOWS-1257, WINDOWS-1258, WINSAMI2, WS2, YU
```

Pour convertir un fichier, utilisez la syntaxe suivante :

```
iconv-fWINDOWS-1252-fUTF8nom_fichier
```

4. Fuseaux horaires

Linux gère les fuseaux horaires. Par défaut, sans réglage de fuseau, c'est l'heure universelle UTC, temps universel coordonné, invariable dans le monde, qui est choisie. Le décalage des fuseaux horaires s'effectue par rapport au temps UTC.

La définition du fuseau horaire actuel est placée sous Debian dans **/etc/timezone**, et sur Redhat (ou dérivé) dans **/etc/localtime**. Ce sont souvent des liens (raccourcis) ou une copie de l'entrée correspondante dans /usr/share/zoneinfo.

Bien qu'encore une fois les outils de la distribution permettent de choisir son fuseau, vous pouvez modifier votre fuseau à la main avec la commande **tzselect**. Linux gère automatiquement le passage à l'heure d'été ou à l'heure d'hiver.

TCP/IP

1. Bases

L'origine de **TCP/IP** provient des recherches du **DARPA** *(Defense Advanced Research Project Agency)* qui débutent en 1970 et débouchent sur **ARPANET**. Dans les faits, le DARPA a financé l'université de Berkeley qui a intégré les protocoles de base de TCP/IP au sein de son système **UNIX BSD 4**.

TCP/IP s'est popularisé grâce à son interface générique de programmation d'échanges de données entre les machines d'un réseau, les primitives **sockets**, et l'intégration de protocoles applicatifs. Les protocoles de TCP/IP sont supervisés par l'**IAB** *(Internet Activities Board)* lui-même supervisant deux autres organismes :

- L'**IRTF** (*Internet Reseach Task Force*) qui est responsable du développement des protocoles.
- L'**IETF** (*Internet Engineering Task Force*) qui est responsable du réseau Internet.

Les adresses réseau sont distribuées par le **NIC** (*Network Information Center*) et en France l'**INRIA**. L'ensemble des protocoles de TCP/IP est décrit dans les documents **RFC** (*Request For Comments*) (voir le RFC 793).

- La couche inférieure est **IP** (*Internet Protocol*).
- La couche de transport est **TCP** (*Transmission Control Protocol*) ou **UDP** (*User Datagram Protocol*).
- Les couches supérieures sont les couches des protocoles applicatifs, par exemple :
  - **NFS** (*Network File System*) : partage de fichiers à distance.
  - **DNS** (*Domain Name System*) : association hôte<->IP.
  - **FTP** (*File Transfer Protocol*) : transfert de fichiers.
  - **TELNET** : émulation d'un terminal de type texte...

La version du protocole IP représenté est la V4. Le futur, déjà présent, est le protocole IPV6. Compatible IPV4, il propose un adressage sur 128 bits (16 octets) permettant d'étendre les capacités du réseau notamment en matière de taille et d'adressage.

2. Adressage

a. Classes

Il est important de savoir avant l'installation dans quel type de réseau doit s'intégrer le nouveau serveur, TCP/IP bien sûr, mais il faut déjà lui réserver une adresse IP, un hostname (nom de machine réseau), connaître les diverses passerelles, le nom de domaine, la classe utilisée et le masque de sous-réseau netmask.

Voici un bref rappel sur les classes IP. Une adresse IP est définie sur 32 bits et représentée par quatre nombres séparés par des points : **n1.n2.n3.n4**. Cette adresse est constituée de deux parties qui définissent l'adresse réseau et l'hôte dans le réseau.

Distinguez, suivant les cas, quatre ou cinq classes d'adresses : A, B, C, D et E, mais seules les trois premières nous intéressent.

Légende : N et h sont des bits, N identifiant du réseau h identifiant de la machine.

```
Classe A : 0NNNNNNN hhhhhhhh hhhhhhhh hhhhhhhh soit 1.x.x.x à
126.x.x.x.
n1 est compris entre 1 et 126.
16777214 hôtes, 127 réseaux.

Classe B : 10NNNNNN NNNNNNNN hhhhhhhh hhhhhhhh soit de 128.0.x.x à
```

```
191.255.x.x.
n1 est compris entre 128 et 191.
65534 hôtes, 16382 réseaux.

Classe C : 110NNNNN NNNNNNNN NNNNNNNN hhhhhhhh soit de 192.0.0.x à
223.255.255.x.
n1 est compris entre 192 et 223.
254 hôtes, 2097150 réseaux.

Classe D : Commence par 1110, pour la multidiffusion IP.
Classe E : Commence par 1111, pour expérimentation.
```

Il existe des adresses d'hôtes qui ne peuvent pas être exploitées. Par exemple dans la classe C on ne peut avoir que 254 hôtes, alors que l'identifiant de la machine est codé sur 8 bits (donc 256 valeurs). C'est que l'adresse 0 représente l'adresse du réseau, et l'adresse 255 celle du **broadcast** (multidiffusion).

Notez que les adresses suivantes ne doivent pas être routées sur Internet et sont réservées aux réseaux locaux.

- 10.0.0.0 - 10.255.255.255 (10/8)
- 172.16.0.0 - 172.31.255.255 (172.16/12)
- 192.168.0.0 - 192.168.255.255 (192.168/16)

L'adresse 127.0.0.1 est l'adresse de loopback ou bouclage : elle représente la machine elle-même, ainsi que le sous-réseau127.0.0.0/8.

b. Sous-réseaux

De plus, il est possible de découper ces réseaux en sous-réseaux à l'aide de masques permettant un découpage plus fin des adresses. Un **netmask** est un masque binaire qui permet de séparer immédiatement l'adresse du réseau et du sous-réseau de l'adresse de l'hôte dans l'adresse IP globale. Les masques prédéfinis sont :

- **Classe A** : 255.0.0.0
- **Classe B** : 255.255.0.0
- **Classe C** : 255.255.255.0

Pour communiquer directement entre eux les hôtes doivent appartenir à un même réseau ou sous-réseau. Calculer un sous-réseau est assez simple. Voici un exemple pour un réseau de classe C.

- Réseau : 192.168.1.0
- Adresse de réseau : 192.168.1.255
- Masque de réseau : 255.255.255.0

Calculer un masque de sous-réseau :

- Pour calculer le masque de sous-réseau, vous devez tout d'abord déterminer combien de machines vous souhaitez intégrer dans celui-ci. Un réseau de classe C permet d'intégrer 254 machines (0 et 255 étant réservés). Vous souhaitez créer des réseaux contenant 60 machines. Ajoutez 2 à cette valeur pour les adresses réservées (adresse du sous-réseau et adresse de broadcast) ce qui donne **62**.
- Une fois le nombre de machines déterminé, trouvez la puissance de deux exacte ou juste supérieure au nombre trouvé. 2 puissance 6 donne **64**.
- Écrivez le masque en binaire, placez tous les bits du masque de réseau de classe C à 1 et placez à 0 les 6 premiers bits du masque correspondant à la partie machine : **11111111 11111111 11111111 11000000**
- Convertissez ce masque en décimal : **255.255.255.192**, et calculez l'ensemble des sous-réseaux possibles.

Comme vous êtes dans un réseau de classe C, vous pouvez encore faire varier les deux derniers bits de la partie machine :

- 00xxxxxx : 255.255.255.0
- 01xxxxxx : 255.255.255.64
- 10xxxxxx : 255.255.255.128
- 11xxxxxx : 255.255.255.192

- Au final, vous obtenez quatre sous-réseaux de 62 machines, soit 248 machines. Vous tombez bien sur 256 si vous rajoutez les quatre adresses de broadcast et les quatre adresses de réseau.

c. Routage

Le masque de réseau permet de déterminer si une machine destinataire est sur le même réseau que vous ou non. Il faut indiquer le chemin que doivent prendre les paquets IP pour rejoindre leur destination. Si votre machine est un poste client disposant d'une seule carte réseau et que ce réseau ne comporte qu'un seul routeur (cas classique d'une connexion vers Internet) alors vous devez créer deux routes. La première est celle indiquant quelle carte réseau doivent emprunter les paquets pour accéder au reste du réseau (au sous-réseau), la seconde quelle route doivent emprunter les paquets pour sortir du réseau. Généralement, on parle de route par défaut quand un seul routeur est présent.

- Vers réseau1 -> utiliser interface réseau gauche.
- Vers réseau2 -> utiliser interface réseau droite.
- Vers autres -> utiliser interface réseau droite vers routeur1.

Exemple : Réseau1 de classe C 192.168.1.0 sur eth0, Réseau2 de classe B 172.16.0.0 sur eth1, adresse de routeur 192.168.1.254.

```
Réseau          Masque          Interface    Passerelle
192.168.1.0     255.255.255.0   eth0         eth0
172.16.0.0      255.255.0.0     eth1         eth1
0.0.0.0         0.0.0.0         eth0         192.168.1.254
```

Tous les paquets réseaux vers 192.168.1.0 transiteront par eth0. Tous les paquets à destination de 172.16.0.0 transiteront par eth1. Par défaut, tous les autres paquets pour les réseaux non spécifiés transiteront par eth0 et seront traités par la passerelle 192.168.1.254 qui routera les paquets.

3. Configuration

a. Cas des distributions de type Red Hat/Fedora

Red Hat propose des outils pour configurer le réseau de base sans passer par la manipulation des fichiers de configuration. Le programme **netconfig** est en mode texte et permet de mettre en place la configuration TCP/IP de base (IP statique ou dynamique, routeur, nom d'hôte, serveur de noms).

```
# netconfig -device eth0
```

La commande graphique **system-config-network** lance une interface plus complète.

Les distributions Mandriva et openSUSE reprennent le même principe que ce qui est exposé ci-après, mais la syntaxe et la position des fichiers peuvent varier. Vous vous reporterez à la documentation de votre distribution pour plus de détails.

Interfaces réseau

La configuration de base d'une interface réseau se fait à l'aide de la commande **ifconfig**. Cependant la plupart des distributions utilisent des scripts d'administration et des fichiers de configuration qui simplifient énormément les choses car ils configurent à la fois l'interface réseau et les routes.

Configuration de eth0 pour l'adresse de classe C 192.168.1.2

```
ifconfig eth0 inet 192.168.1.2 netmask 255.255.255.0
ifconfig eth0 192.168.1.2
```

Activation de l'interface réseau eth0 :

```
ifconfig eth0 up
```

Arrêt de l'interface réseau eth0 :

```
ifconfig eth0 down
```

Affichage des informations de eth0 :

```
# ifconfig eth0
eth0      Lien encap:Ethernet  HWaddr 00:XX:XX:XX:XX:XX
          inet adr:192.168.1.60  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::21b:fcff:fec9:f81d/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16522 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13631 errors:0 dropped:0 overruns:0 carrier:2
          collisions:0 lg file transmission:1000
          RX bytes:17732221 (16.9 Mb)  TX bytes:1648879 (1.5 Mb)
```

Affichage de toutes les interfaces réseaux activées :

```
ifconfig
```

Affichage de toutes les interfaces réseaux activées ou non :

```
ifconfig -a
```

Le mieux reste d'utiliser les scripts **ifup** et **ifdown**. Ceux-ci se basent sur les fichiers présents dans /etc/sysconfig/network-scripts/. Ces fichiers de configuration d'interface se nomment ifcfg-xxx où xxx est le nom de l'interface réseau, comme eth0 :

```
DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=static
```

Les paramètres parlent d'eux-mêmes. Les valeurs **NETWORK** et **BROADCAST** sont optionnelles si **IPADDR** et **NETMASK** sont renseignés (dans ce cas le calcul est automatique) ou si **DHCP** est utilisé. **BOOTPROTO** indique comment monter l'interface, soit **static**, soit **dhcp**. La valeur **bootp** peut aussi être utilisée. Dans le cas de DHCP, le fichier peut ressembler à ceci :

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

Dans le cas d'une configuration statique, **IPADDR** et **NETMASK** sont obligatoires :

```
DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=static
```

ONBOOT détermine si l'interface doit être automatiquement activée au démarrage de la machine. Une fois le fichier correctement renseigné, on utilise les commandes **ifup**/**ifdown** :

Activation de l'interface eth0 :

```
ifup eth0
```

Arrêt de l'interface eth0 :

```
ifdown eth0
```

Paramètres généraux

Le fichier **/etc/sysconfig/network** contient les paramètres généraux du réseau.

```
NETWORKING=yes
HOSTNAME=poste1.monreseau.org # nom complet
GATEWAY=0.0.0.0 # passerelle par défaut
NISDOMAIN= # nom du domaine NIS
```

- **NETWORKING** : activation ou non du réseau.
- **HOSTNAME** : nom de domaine complet FQDN.
- **GATEWAY** : adresse IP de la passerelle.
- **GATEWAYDEV** : interface réseau permettant d'accéder à la passerelle.
- **NISDOMAIN** : cas d'un domaine NIS.

b. Machines de type Debian

Le fichier de configuration des interfaces réseaux sous Debian (et Ubuntu) est situé dans /etc/network/interfaces. Il n'a pas du tout le même format que précédemment :

```
# cat interfaces
auto lo eth0 eth1
iface lo inet loopback

iface eth0 inet static
        address 192.161.1.60
        netmask 255.255.255.0
        broadcast 192.168.1.255
        gateway 192.168.1.1

iface eth1 inet dhcp
```

Cet exemple montre trois types d'interfaces :

- l'interface lo de loopback,
- l'interface eth1 en dhcp, ne nécessitant pas de configuration plus avancée,
- l'interface eth0 configurée statiquement.

La syntaxe générale d'une déclaration est la suivante :

```
interface nom type mode
```

Avec une configuration statique, précisez les différents paramètres avec les mots clés suivants :

- **address** : l'adresse IP.
- **netmask** : le masque de sous-réseau.
- **broadcast** : l'adresse de broadcast.
- **gateway** : la passerelle par défaut.

Le fichier **/etc/hostname** contient le nom de la machine :

```
# cat /etc/hostname
slyserver
```

c. Routage

Avec l'utilisation des fichiers et des commandes précédentes, il n'y a pas besoin de créer un routage spécifique car la passerelle par défaut est déjà présente (cf. paramètres généraux) et les routes pour les interfaces réseaux sont automatiquement mises en place par **ifup**. Cependant on peut utiliser la commande **route**.

Affiche les routes actuelles :

```
route
netstat -nr
```

Dans l'exemple suivant, les interfaces vmnet1 et vmnet8 sont des interfaces virtuelles issues de la configuration réseau de VMWare.

```
# netstat -rn
Table de routage IP du noyau
Destination     Passerelle    Genmask         Indic   MSS Fenêtre irtt
Iface
192.168.211.0   0.0.0.0       255.255.255.0   U         0 0          0
vmnet8
192.168.1.0     0.0.0.0       255.255.255.0   U         0 0          0
eth0
172.16.248.0    0.0.0.0       255.255.255.0   U         0 0          0
vmnet1
169.254.0.0     0.0.0.0       255.255.0.0     U         0 0          0
eth0
127.0.0.0       0.0.0.0       255.0.0.0       U         0 0          0
lo
0.0.0.0         192.168.1.1   0.0.0.0         UG        0 0          0
eth0
```

Ajout de l'entrée loopback :

```
route add -net 127.0.0.0
```

Ajoute la route vers le réseau 192.168.1.0 passant par eth0. Netmask peut être omis.

```
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

Ajoute la passerelle par défaut vers le routeur :

```
route add default gw 192.168.1.254
```

Supprime la route vers le réseau 172.16.0.0 :

```
route del -net 172.16.0.0 eth0
```

4. Outils réseaux

a. FTP

Il est utile de connaître la commande **ftp** (*file transfer protocol*). Elle permet le transfert de fichiers entre deux machines. Elle prend comme paramètre le nom de la machine distante. Pour que la commande **ftp** fonctionne, il faut que le service ftp fonctionne sur la machine distante et sur le port 21.

Voici un exemple (peu pratique) de connexion avec erreur et nouvel essai.

```
ftp> open
(to) machine
Connected to machine.
220 machine FTP server (Digital UNIX Version 5.60) ready.
Name (machine:root): root
331 Password required for root.
Password:
530 Login incorrect.
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> user
(username) root
331 Password required for root.
Password:
230 User root logged in.
ftp> pwd
257 "/" is current directory.
```

Le plus simple est tout de même :

```
$ ftp machine
Connected to machine.
220 machine FTP server (Digital UNIX Version 5.60) ready.
Name (machine:root): root
331 Password required for root.
Password:
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Voici une liste de commandes ftp.

Commande	Action
open	Suivi d'un nom de machine, ouvre une connexion sur la machine spécifiée.
user	Saisie de l'utilisateur distant pour une connexion.
quit	Fin de la connexion et fin de la commande ftp.
ascii	Transfert des fichiers en mode ASCII (conversion des caractères spéciaux et fin de ligne en MS et Unix par exemple).
binary	Transfert des fichiers en mode binaire.
glob	Supprime l'interprétation des caractères spéciaux.
help	Affiche l'aide.
prompt	Suivi de on ou off, active ou désactive la confirmation individuelle de transfert pour chaque fichier (mget ou mput).
pwd	Affiche le répertoire distant courant.
cd	Suivi du chemin, déplacement dans l'arborescence distante.

ls	Liste les fichiers de la machine distante.
delete	Suivi d'un nom de fichier, supprime le fichier distant.
mdelete	Multiple. Supprime les fichiers distants.
get	Récupère le fichier distants.
mget	Multiple. Récupère les fichiers distants (liste ou modèle).
put	Envoie le fichier local vers la machine distante.
mput	Multiple. Envoie les fichiers locaux sur la machine distante (liste ou modèle).
close/disconnect	Ferme la session actuelle.
lcd	Change de répertoire sur la machine locale.
hash	Durant les transferts, écrit un « # » sur écran pour chaque buffer transféré.
system	Informations sur le système distant.
recv	Réception d'un fichier.
send	Envoi d'un fichier.
rename	Renomme un fichier distant.
mkdir	Crée un répertoire sur la machine distante.
rmdir	Supprime un répertoire sur la machine distante.
!commande	Exécute la commande locale.

b. Telnet

Telnet est un client léger permettant d'ouvrir une connexion et une session sur une machine distante proposant un serveur telnet. Ce serveur est souvent lancé depuis xinetd ou inetd. Sa syntaxe est très simple :

```
$ telnet -l user machine port
Exemple :
# telnet 192.168.1.60
Trying 192.168.1.60...
Connected to 192.168.1.60.
Escape character is '^]'.
Welcome to openSUSE 10.3 (i586) - Kernel 2.6.24.4-default (3).

slyserver login: seb
Mot de passe :
Dernière connexion : jeudi 15 mai 2008 à 06:26:30 CEST de console
sur :0
Vous avez un nouveau message.
Have a lot of fun...
seb@slyserver:~>
```

Attention ! Le service (et le client) telnet n'est absolument pas sécurisé : les connexions transitent en clair sur le réseau, et n'importe quel renifleur IP (wireshark par exemple) peut intercepter et voir tout ce qui est fait. Même le mot de passe est transmis en clair (en texte). Évitez d'utiliser ce service et concentrez-vous sur OpenSSH.

c. Ping

La commande **ping** est une commande centrale, voire incontournable. La première chose que l'on fait généralement pour savoir si une machine est accessible ou non, c'est d'essayer de la « pinguer » (sous réserve que la configuration du firewall autorise les requêtes ICMP).

Ping émet un « écho » réseau, un peu comme un sonar, et attend une réponse, le retour de l'écho. Il utilise pour cela le protocole ICMP. Interrompez la commande **ping** avec [Ctrl] **C**.

```
$ ping www.kde.org
PING www.kde.org (62.70.27.118) 56(84) bytes of data.
64 bytes from 62.70.27.118: icmp_seq=1 ttl=57 time=10.5 ms
64 bytes from 62.70.27.118: icmp_seq=2 ttl=57 time=11.3 ms
64 bytes from 62.70.27.118: icmp_seq=3 ttl=57 time=10.4 ms
64 bytes from 62.70.27.118: icmp_seq=4 ttl=57 time=11.5 ms
...
```

Trois paramètres doivent attirer votre attention :

- `-c` permet de préciser le nombre d'échos à émettre.
- `-b` permet d'émettre un écho sur une adresse de broadcast.
- `-I` permet de spécifier l'interface réseau.

Dans le premier cas le paramètre peut être utile pour tester dans un script si un serveur répond :

```
# ping -c 1 10.9.238.170 >/dev/null 2>&1 && echo "Le serveur répond"
Le serveur répond
```

Dans le second cas, toutes les adresses du sous-réseau concerné par l'adresse de broadcast doivent répondre.

```
# ping -b 192.168.1.255
WARNING: pinging broadcast address
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.232 ms
64 bytes from 192.168.1.60: icmp_seq=1 ttl=64 time=0.240 ms
64 bytes from 192.168.1.130: icmp_seq=1 ttl=255 time=0.285 ms
64 bytes from 192.168.1.139: icmp_seq=1 ttl=255 time=0.292 ms
...
```

Dans le dernier cas, vous pouvez spécifier une carte de sortie. Cette option est très utile pour vérifier une résolution DNS ou une route.

```
# ping -I eth0 192.168.1.60
PING 192.168.1.60 (192.168.1.60) from 192.168.1.10:eth0: 56(84) bytes
of data.
64 bytes from 192.168.1.60: icmp_seq=1 ttl=62 time=0.478 ms
64 bytes from 192.168.1.60: icmp_seq=2 ttl=62 time=0.408 ms
...
```

d. Traceroute

Quand vous tentez d'accéder à un hôte distant depuis votre machine, les paquets IP passent souvent par de nombreuses routes, parfois différentes selon le point de départ et de destination, l'engorgement, etc. Le trajet passe par de nombreuses passerelles (gateways), qui dépendent des routes par défaut ou prédéfinies de chacune d'elles.

La commande **traceroute** permet de visualiser chacun des points de passage de vos paquets IP à destination d'un hôte donné. Dans l'exemple suivant, l'hôte situé en région parisienne sur le réseau du fournisseur Free tente de déterminer la route empruntée pour se rendre sur le serveur www.kde.org. L'adresse IP source (hors réseau local) est volontairement masquée.

```
$ traceroute www.kde.org
traceroute to www.kde.org (62.70.27.118), 30 hops max, 40 byte packets
 1 DD-WRT (192.168.1.1)  0.558 ms   0.533 ms   0.585 ms
```

```
 2 82.xxx.yyy.zzz (82.xxx.yyy.zzz)  6.339 ms   6.404 ms   6.901 ms
 3 * * *
 4 * * *
 5 212.73.205.5 (212.73.205.5)  39.267 ms   35.499 ms   31.736 ms
 6 ae-12-55.car2.Paris1.Level3.net (4.68.109.144)  6.485 ms ae-22-
52.car2.Paris1.Level3.net (4.68.109.48)  6.401 ms   6.338 ms
 7 UUnet-Level3.Level3.net (212.73.240.206)  6.113 ms   6.152 ms
5.866 ms
 8 so-3-2-0.TL2.PAR2.ALTER.NET (146.188.8.121)  6.107 ms   6.410 ms
6.365 ms
 9 so-2-2-0.TL2.STK2.ALTER.NET (146.188.7.33)  87.323 ms   86.840 ms
87.010 ms
10 so-7-1-0.XR2.OSL2.ALTER.NET (146.188.15.62)  96.491 ms   97.148 ms
96.488 ms
11 ge-0-1-0.GW6.OSL2.ALTER.NET (146.188.3.242)  95.972 ms   95.934 ms
96.108 ms
12 213.203.63.74 (213.203.63.74)  95.320 ms   94.321 ms   96.188 ms
13 leeloo.troll.no (62.70.27.10)  94.064 ms   94.052 ms   92.374 ms
14 jamaica.kde.org (62.70.27.118)  97.064 ms   96.182 ms   97.853 ms
```

e. Whois

Savez-vous que vous pouvez obtenir toutes les informations voulues sur un domaine (toto.fr) à l'aide de la commande **whois** ? Par exemple, pour obtenir toutes les informations sur le domaine kde.org :

```
> whois kde.org
...
Domain ID:D1479623-LROR
Domain Name:KDE.ORG
Created On:14-Dec-1996 05:00:00 UTC
Last Updated On:12-Oct-2007 13:10:18 UTC
Expiration Date:13-Dec-2012 05:00:00 UTC
Sponsoring Registrar:easyDNS Technologies Inc. (R1247-LROR)
Status:CLIENT TRANSFER PROHIBITED
Status:CLIENT UPDATE PROHIBITED
Registrant ID:tu2YDGaiunEvz5QA
Registrant Name:Trolltech AS
Registrant Organization:Trolltech AS
Registrant Street1:Sandakerveien 116, PO Box 4332 Nydalen
Registrant Street2:
Registrant Street3:
Registrant City:Oslo
Registrant State/Province:N/A
Registrant Postal Code:N-0402
Registrant Country:NO
Registrant Phone:+1.4721604800
Registrant Phone Ext.:
Registrant FAX:+1.4721604801
Registrant FAX Ext.:
Registrant Email:hostmaster@trolltech.com
Admin ID:tubEUVkFfutkJZMD
Admin Name:Trolltech AS
Admin Organization:Trolltech AS
Admin Street1:Sandakerveien 116, PO Box 4332 Nydalen
Admin Street2:
Admin Street3:
Admin City:Oslo
Admin State/Province:N/A
Admin Postal Code:N-0402
Admin Country:NO
Admin Phone:+1.4721604800
Admin Phone Ext.:
Admin FAX:+1.4721604801
Admin FAX Ext.:
Admin Email:hostmaster@trolltech.com
Tech ID:tuSfXVVJtggMdKWm
Tech Name:Trolltech AS
Tech Organization:Trolltech AS
```

```
Tech Street1:Sandakerveien 116, PO Box 4332 Nydalen
Tech Street2:
Tech Street3:
Tech City:Oslo
Tech State/Province:N/A
Tech Postal Code:N-0402
Tech Country:NO
...
```

f. Netstat

La commande **netstat** permet d'obtenir une foule d'informations sur le réseau et les protocoles.

Le paramètre `-i` permet d'obtenir l'état des cartes réseaux, afin de déterminer une éventuelle panne ou un problème de câble :

```
# netstat -i
Table d'interfaces noyau
Iface  MTU Met   RX-OK RX-ERR RX-DRP RX-OVR   TX-OK TX-ERR TX-DRP
TX-OVR Flg
eth0  1500   0 2332007      0      0      0  677842      0      0
0 BMRU
lo   16436   0    1109      0      0      0    1109      0      0
0 LRU
```

Si vous rajoutez le paramètre `-e`, vous obtenez le même résultat qu'avec ifconfig `-a`.

```
# netstat -ei
Table d'interfaces noyau
eth0   Lien encap:Ethernet  HWaddr 00:XX:D3:XX:AA:XX
       inet adr:12.168.1.60  Bcast:192.168.1.255  Masque:255.255.255.0
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:2335314 errors:0 dropped:0 overruns:0 frame:0
       TX packets:678095 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 lg file transmission:1000
       RX bytes:1055212145 (1006.3 Mb)  TX bytes:61264196 (58.4 Mb)
       Interruption:20 Adresse de base:0x8c00

lo     Lien encap:Boucle locale
       inet adr:127.0.0.1  Masque:255.0.0.0
       UP LOOPBACK RUNNING  MTU:16436  Metric:1
       RX packets:1109 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1109 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 lg file transmission:0
       RX bytes:60423 (59.0 Kb)  TX bytes:60423 (59.0 Kb)
```

Le paramètre `-r` permet d'obtenir, comme route, les tables de routage. Ajoutez le paramètre `-n` pour indiquer les IPs à la place des noms.

```
# netstat -rn
Table de routage IP du noyau
Destination     Passerelle    Genmask         Indic   MSS Fenêtre irtt
Iface
192.168.211.0   0.0.0.0       255.255.255.0   U         0 0          0
vmnet8
192.168.1.0     0.0.0.0       255.255.255.0   U         0 0          0
eth0
172.16.248.0    0.0.0.0       255.255.255.0   U         0 0          0
vmnet1
169.254.0.0     0.0.0.0       255.255.0.0     U         0 0          0
eth0
127.0.0.0       0.0.0.0       255.0.0.0       U         0 0          0
lo
0.0.0.0         192.168.1.1   0.0.0.0         UG        0 0          0
eth0
```

Le paramètre `-a` permet de visualiser toutes les connexions, pour tous les protocoles, y compris les ports en écoute de la machine. La sortie est trop longue pour être placée dans ces pages.

```
# netstat -a | wc -l
495
```

Le paramètre `-A` permet de spécifier le protocole visible : inet, unix, ipx, ax25, netrom et ddp.

```
# netstat -a -A inet
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante
Etat
tcp        0      0 localhost:716           *:*
LISTEN
tcp        0      0 *:sunrpc                *:*
LISTEN
tcp        0      0 localhost:ipp           *:*
LISTEN
tcp        0      0 localhost:smtp          *:*
LISTEN
tcp        0      0 localhost:hpssd         *:*
LISTEN
tcp        0      0 slyserver:41851         imap.free.fr:imap
ESTABLISHED
tcp        0      0 slyserver:41850         imap.free.fr:imap
ESTABLISHED
tcp        0      0 slyserver:54220         by1msg4176111.gate:msnp
ESTABLISHED
tcp        0      0 slyserver:34267         by2msg2105007.phx.:msnp
ESTABLISHED
tcp        0      0 slyserver:47990         by1msg4082314.phx.:msnp
ESTABLISHED
udp        0      0 *:filenet-tms           *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:ipp                   *:*
udp        0      0 172.16.248.1:ntp        *:*
udp        0      0 192.168.211.1:ntp       *:*
udp        0      0 slyserver:ntp           *:*
udp        0      0 localhost:ntp           *:*
udp        0      0 *:ntp                   *:*
raw        0      0 *:icmp                  *:*                     7
```

Enfin le paramètre `-p` permet d'indiquer, quand c'est possible, le PID et le nom du processus

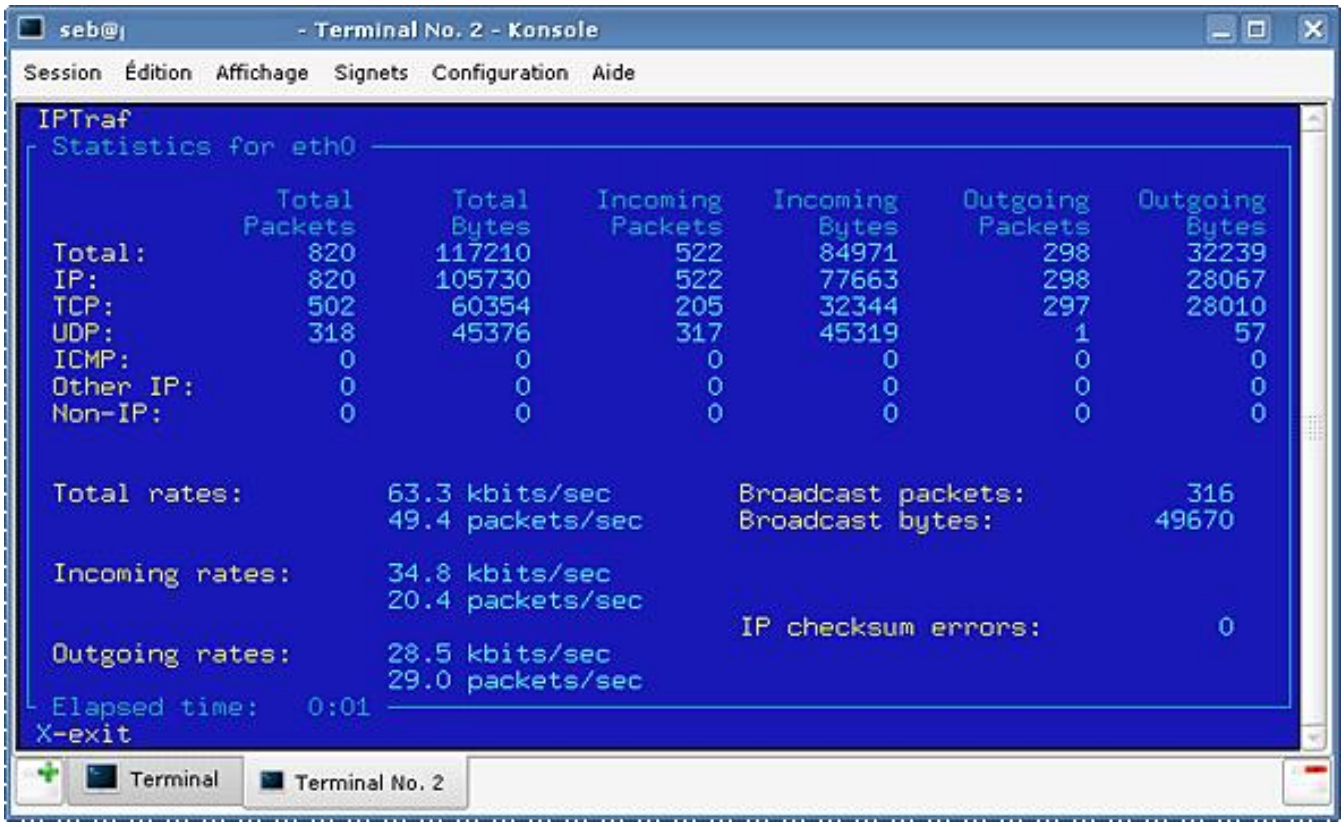
```
# netstat -A inet -p
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante
Etat        PID/Program name
...
tcp        0      0 slyserver:54220         by1msg4176111.gate:msnp
ESTABLISHED 4041/kopete
tcp        0      0 slyserver:34267         by2msg2105007.phx.:msnp
ESTABLISHED 4041/kopete
tcp        0      0 slyserver:47990         by1msg4082314.phx.:msnp
ESTABLISHED 4041/kopete
```

g. IPTraf

La commande **iptraf** permet de visualiser en temps réel l'activité du réseau via un outil texte interactif ou non (ligne de commande). Les menus sont clairs. Vous vous y déplacez avec les touches fléchées et les divers raccourcis précisés.

La capture suivante montre l'affichage détaillé des statistiques de trafic de la carte eth0. Cet écran est accessible via la ligne de commande avec :

```
# iptraf -d eth0
```



IPTraf analyse le trafic eur eth0.

5. Fichiers généraux

a. /etc/resolv.conf

Le fichier **/etc/resolv.conf** est utilisé pour indiquer au système quels serveurs de noms et quels domaines interroger pour résoudre les requêtes DNS clientes. Les API sont incluses dans la bibliothèque et les API standards de Linux (il n'y a pas besoin d'ajouter des outils supplémentaires). On appelle cette bibliothèque le **resolver**.

En configurant DHCP ce fichier est en principe mis automatiquement à jour et ne devrait pas être modifié sauf si vous avez interdit la configuration DNS sur votre client.

```
$ cat /etc/resolv.conf
domain mondomaine.org
search mondomaine.org
nameserver 192.168.1.1
nameserver 192.168.1.2
```

- **domain** : nom du domaine local. Les requêtes sont généralement réduites à des raccourcis relatifs au domaine local. S'il est absent le nom du domaine doit être déterminé à partir du nom d'hôte complet : c'est la partie située après le premier « . ».
- **search** : liste des domaines de recherche. Par défaut lors de l'utilisation de raccourcis (noms d'hôtes courts) le resolver lance une recherche sur le domaine défini par la ligne domain, mais on peut spécifier ici une liste de domaines séparés par des espaces ou des virgules.
- **nameserver** : adresse IP du serveur de noms (le serveur DNS). On peut en placer au maximum trois. Le resolver essaie d'utiliser le premier. En cas d'échec (timeout), il passe au second, et ainsi de suite.
- **options** : des options peuvent être précisées. Par exemple **timeout:n** où n (en secondes) indique le délai d'attente de réponse d'un serveur de noms avant de passer au suivant.

b. /etc/hosts et /etc/networks

Sans même utiliser de serveur de noms, vous pouvez établir une correspondance entre les adresses IP et les noms des machines au sein du fichier **/etc/hosts**.

```
192.168.1.1     server1 www1 ftp
192.168.1.11    poste1
192.168.1.12    poste2
```

Vous pouvez faire de même pour nommer les réseaux (ce qui peut être utile pour les tcp_wrappers ou la commande route) dans le fichier **/etc/networks**.

```
loopnet    127.0.0.0
localnet   192.168.1.0
```

c. /etc/nsswitch.conf

Le fichier **/etc/nsswitch.conf** permet de déterminer l'ordre dans lequel le resolver (ou d'autres services) récupère ses informations. Les deux lignes en gras de l'exemple indiquent que lors d'une requête de résolution de nom (ou de réseau) les fichiers sont prioritaires. Le fichier **/etc/hosts** est d'abord lu, puis, si le resolver ne trouve pas l'information il passe par une résolution DNS.

```
passwd: compat
group:  compat

hosts:          files dns
networks:       files dns

services:       files
protocols:      files
rpc:            files
ethers:         files
netmasks:       files
netgroup:       files nis
publickey:      files

bootparams:     files
automount:      files nis
aliases:        files
```

Il peut arriver que certains produits mal programmés n'utilisent pas le resolver mais directement le DNS, ou le fichier /etc/hosts, ou inversent l'ordre établi dans /etc/nsswitch.conf. Dans ce cas, il n'est pas possible de prévoir (et de prédire) le bon fonctionnement de ce genre de produits...

d. /etc/services

Le fichier **/etc/services** contient la liste des services réseaux connus de Unix ainsi que les ports et protocoles associés. Il est utilisé par de nombreux services (dont xinetd) et sous-systèmes comme le firewall de Linux.

Ce fichier est indicatif : c'est un fichier de description et de définition : tous les services de ce fichier ne tournent pas forcément sur votre machine (et heureusement vu le nombre) ! Et un service peut être configuré pour écouter un autre port. Par contre il est conseillé, lorsque vous rajoutez un service qui n'est pas présent dans ce fichier, de le rajouter à la fin.

```
tcpmux          1/tcp    # TCP Port Service Multiplexer
tcpmux          1/udp    # TCP Port Service Multiplexer
compressnet     2/tcp    # Management Utility
compressnet     2/udp    # Management Utility
compressnet     3/tcp    # Compression Process
compressnet     3/udp    # Compression Process
rje             5/tcp    # Remote Job Entry
rje             5/udp    # Remote Job Entry
echo            7/tcp    Echo
```

```
echo              7/udp    Echo
discard           9/tcp    # Discard
discard           9/udp    # Discard
systat           11/tcp    users        # Active Users
systat           11/udp    users        # Active Users
daytime          13/tcp    # Daytime (RFC 867)
daytime          13/udp    # Daytime (RFC 867)
netstat          15/tcp    # Unassigned [was netstat]
qotd             17/tcp    quote        # Quote of the Day
qotd             17/udp    quote        # Quote of the Day
msp              18/tcp    # Message Send Protocol
msp              18/udp    # Message Send Protocol
chargen          19/tcp    # Character Generator
chargen          19/udp    # Character Generator
ftp-data         20/tcp    # File Transfer [Default Data]
ftp-data         20/udp    # File Transfer [Default Data]
ftp              21/tcp    # File Transfer [Control]
fsp              21/udp    # File Transfer [Control]
ssh              22/tcp    # SSH Remote Login Protocol
ssh              22/udp    # SSH Remote Login Protocol
telnet           23/tcp    # Telnet
telnet           23/udp    # Telnet
...
```

e. /etc/protocols

Le fichier **/etc/protocols** contient la liste des protocoles connus par Unix.

```
# Assigned Internet Protocol Numbers
#
# Decimal   Keyword     Protocol                          References
# -------   -------     --------                          ----------
# protocol  num aliases      # comments
hopopt        0 HOPOPT       # IPv6 Hop-by-Hop Option         [RFC1883]
icmp          1 ICMP         # Internet Control Message        [RFC792]
igmp          2 IGMP         # Internet Group Management      [RFC1112]
ggp           3 GGP          # Gateway-to-Gateway              [RFC823]
ip            4 IP           # IP in IP (encapsulation)       [RFC2003]
st            5 ST           # Stream                 [RFC1190,RFC1819]
tcp           6 TCP          # Transmission Control            [RFC793]
cbt           7 CBT          # CBT                           [Ballardie]
egp           8 EGP          # Exterior Gateway Protocol  [RFC888,DLM1]
igp           9 IGP          # any private interior gateway      [IANA]
bbn-rcc-mon  10 BBN-RCC-MON  # BBN RCC Monitoring                 [SGC]
nvp-ii       11 NVP-II       # Network Voice Protocol      [RFC741,SC3]
pup          12 PUP          # PUP                          [PUP,XEROX]
argus        13 ARGUS        # ARGUS                             [RWS4]
emcon        14 EMCON        # EMCON                              [BN7]
xnet         15 XNET         # Cross Net Debugger         [IEN158,JFH2]
chaos        16 CHAOS        # Chaos                              [NC3]
udp          17 UDP          # User Datagram               [RFC768,JBP]
...
```

Services réseaux xinetd

1. Présentation

Le démon **xinetd** est un « super-service » permettant de contrôler l'accès à un ensemble de services, **telnet** par exemple. Beaucoup de services réseaux peuvent être configurés pour fonctionner avec xinetd, comme les services ftp, ssh, samba, rcp, http, etc. Des options de configuration spécifiques peuvent être appliquées pour chaque service géré.

Lorsqu'un hôte client se connecte à un service réseau contrôlé par xinetd, xinetd reçoit la requête et vérifie tout d'abord les autorisations d'accès TCP (voir **tcp_wrappers** au prochain chapitre) puis les règles définies pour ce service (autorisations spécifiques, ressources allouées, etc.). Une instance du service est alors démarrée et lui cède la connexion. À partir de ce moment **xinetd** n'interfère plus dans la connexion entre le client et le serveur.

2. Configuration

Les fichiers de configuration sont :

- **/etc/xinetd.conf** : configuration globale
- **/etc/xinetd.d/*** : répertoire contenant les fichiers spécifiques aux services. Il existe un fichier par service, du même nom que celui précisé dans /etc/services.

```
$ ls -l /etc/xinetd.d
total 92
-rw-r--r-- 1 root root  313 sep 22  2007 chargen
-rw-r--r-- 1 root root  333 sep 22  2007 chargen-udp
-rw-r--r-- 1 root root  256 mar 20 22:11 cups-lpd
-rw-r--r-- 1 root root  409 nov  4  2005 cvs
-rw-r--r-- 1 root root  313 sep 22  2007 daytime
-rw-r--r-- 1 root root  333 sep 22  2007 daytime-udp
-rw-r--r-- 1 root root  313 sep 22  2007 discard
-rw-r--r-- 1 root root  332 sep 22  2007 discard-udp
-rw-r--r-- 1 root root  305 sep 22  2007 echo
-rw-r--r-- 1 root root  324 sep 22  2007 echo-udp
-rw-r--r-- 1 root root  492 sep 22  2007 netstat
-rw-r--r-- 1 root root  207 avr 23 19:04 rsync
-rw-r--r-- 1 root root  337 fév 17 14:22 sane-port
-rw-r--r-- 1 root root  332 sep 22  2007 servers
-rw-r--r-- 1 root root  334 sep 22  2007 services
-rw-r--r-- 1 root root  351 jun 21  2007 svnserve
-rw-r--r-- 1 root root  277 nov  8  2007 swat
-rw-r--r-- 1 root root  536 sep 21  2007 systat
-rw-r--r-- 1 root root  387 fév  4 10:11 tftp.rpmsave
-rw-r--r-- 1 root root  339 sep 22  2007 time
-rw-r--r-- 1 root root  333 sep 22  2007 time-udp
-rw-r--r-- 1 root root 2304 avr  4 11:39 vnc
-rw------- 1 root root  768 sep 22  2007 vsftpd
```

Contenu de xinetd.conf :

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}
includedir /etc/xinetd.d
```

- **instances** : nombre maximal de requêtes qu'un service xinetd peut gérer à un instant donné.

- **log_type** : dans notre cas, les traces sont gérées par le démon **syslog** via **authpriv** et les traces sont placées dans /var/log/secure. **FILE /var/log/xinetd** aurait placé les traces dans /var/log/xinetd.

- **log_on_success** : xinetd va journaliser l'événement si la connexion au service réussit. Les informations tracées sont l'hôte (**HOST**) et le **PID** du processus serveur traitant la connexion.

- **log_on_failure** : idem mais pour les échecs. Il devient simple de savoir quels hôtes ont tenté de se connecter si par exemple la connexion n'est pas autorisée.

- **cps** : xinetd n'autorise que 25 connexions par secondes à un service. Si la limite est atteinte, xinetd attendra 30 secondes avant d'autoriser à nouveau les connexions.

- **includedir** : inclut les options des fichiers présents dans le répertoire indiqué.

Exemple /etc/xinetd.d/telnet :

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        disable = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
}
```

La première ligne en commentaire, **default**, a une importance particulière. Elle n'est pas interprétée par xinetd mais par **ntsysv** ou **chkconfig** pour déterminer si le service est actif.

- **service** : nom du service qui correspond à un service défini dans /etc/services.

- **flags** : attributs pour la connexion. REUSE indique que la socket sera réutilisée pour une connexion telnet.

- **socket_type** : spécifie le type de socket. Généralement **stream** (tcp) ou **dgram** (udp). Une connexion directe IP se fait par **raw**.

- **wait** : indique si le serveur est single-threaded (yes) ou multi-threaded (no).

- **user** : sous quel compte utilisateur le service sera lancé.

- **server** : chemin de l'exécutable devant être lancé.

- **log_on_failure** : le **+=** indique qu'on rajoute l'option associée au fichier de trace en plus de celles par défaut. Ici : le login.

- **disable** : indique si le service est actif ou non.

Certaines options peuvent améliorer les conditions d'accès et la sécurité :

- **only_from** : permet l'accès uniquement aux hôtes spécifiés.

- **no_access** : empêche l'accès aux hôtes spécifiés (ex : 172.16.17.0/24).

- **access_times** : autorise l'accès uniquement sur une plage horaire donnée (ex :09:00-18:30).

3. Démarrage et arrêt des services

On distingue deux cas.

Premier cas, le service **xinetd** est un service comme un autre dont le démarrage ou l'arrêt peut s'effectuer avec la commande **service** ou directement via l'exécution de /etc/init.d/xinetd.

```
# service xinetd start
```

Dans ce cas, la commande **chkconfig** (Red Hat, openSUSE) autorise ou non le lancement du service au démarrage pour chaque niveau d'exécution (runlevel).

```
# chkconfig --level 345 xinetd on
```

Second cas, comme xinetd gère plusieurs services, l'arrêt de xinetd arrête tous les services associés, et le démarrage de xinetd lance tous les services associés. Il n'est pas possible de choisir quels services de xinetd seront lancés dans tel ou tel niveau d'exécution. Mais vous pouvez choisir d'activer ou de désactiver simplement un service avec **chkconfig**.

```
# chkconfig telnet on
```

Connexion PPP

1. Choix et réglage du modem

a. Le cas des Winmodems

Tous les modems RTC (analogiques) se connectant sur un port série (externe), ou émulant un vrai port série (carte PCI ou via le port USB) sont entièrement supportés sous Linux.

Cependant, il existe une catégorie particulière de modems appelés les **winmodems**. Ils se présentent parfois comme des « vrais » modems (ils leur ressemblent parfois). D'une manière générale, fuyez ce type de modems. Cependant quelques modèles sont connus pour fonctionner sous Linux. Rendez-vous sur le site http://linmodems.org/ pour obtenir des informations à ce sujet.

D'autres adaptateurs que les modems RTC sont reconnus par Linux comme des modems ; c'est le cas de quelques adaptateurs ADSL, mais aussi des téléphones portables reliés par une câble USB ou via une connexion Bluetooth.

b. Les fichiers périphériques

Les ports série de type RS232 se nomment **ttySn** :

- **/dev/ttyS0** : premier port série.
- **/dev/ttyS1** : second port série.
- etc.

Les ports série de type USB se nomment **ttyUSBn : /dev/ttyUSB0**, et ainsi de suite.

Les ports de communication série via bluetooth se nomment **rfcommn** (pour radio frequency communication) : **/dev/rfcomm0**, et ainsi de suite.

Pour utiliser les ports série et établir une communication, vous devez pouvoir écrire sur les périphériques (pour envoyer les ordres) et donc soit avoir les droits correspondants, soit utiliser un programme SUID, ou encore disposer de règles udev adaptées.

c. Régler le port série

Les ports série se gèrent via la commande **setserial**.

La commande **setserial** permet d'interroger la configuration d'un port série avec le paramètre `-g`. Le port série ttyS0 est de type 16550A (le plus rapide), utilise l'IRQ 4 et le port d'adresse 0x03f8.

```
# setserial -g /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Les informations n'étant pas suffisantes, rajoutez le paramètre `-a`. Vous obtenez entre autres la vitesse de la ligne qui est de 115200 bits par seconde (le mot baud est à proscrire).

```
# setserial -a -g /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
        Baud_base: 115200, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test
```

Setserial permet aussi de configurer le port. En interrogeant le port, vous pouvez savoir quels paramètres ont permis son réglage avec le `-G` :

```
# setserial -G /dev/ttyS0
/dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_normal
skip_test
```

De là, il est possible d'extrapoler de nouvelles valeurs. Par exemple passez ainsi le port série à une vitesse de 57600 bps :

```
# setserial /dev/ttyS0 baud_base 57600
# setserial -a -g /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
        Baud_base: 57600, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test
```

d. Les commandes AT

Les modems utilisent tous un jeu de commandes standard appelées **commandes AT**. Leur vrai nom est **Commandes Hayes**, du nom de la société les ayant inventées. AT signifie Attention. Le modem attend après ces premières lettres une suite permettant de le configurer, de numéroter, de raccrocher, etc.

Le jeu est en principe standard mais la configuration varie d'un modèle à un autre. Si vous ne rentrez pas dans les détails, une configuration générique suffit et fonctionne pour la quasi-totalité des modems. Comme il n'est pas possible de décrire les commandes AT ici, vous en trouverez une liste sur le site 3com (qui a racheté US Robotics, le meilleur fabricant de modems) : http://www.usr.com/support/3cp3056/3cp3056-french-ug/atcoms.htm

En voici tout de même quelques-unes :

- Numéroter : ATDT0102030405
- Répondre : ATA
- Raccrocher : ATH

2. PPP

Le protocole **PPP** (*Point to Point Protocol*) permet de vous relier à une autre machine afin d'y accéder, ou à son réseau, ce qui est souvent le cas d'Internet. Aujourd'hui encore et malgré les nombreuses solutions proposées par le câble ou l'ADSL, certaines connexions se font encore via un modem RTC (modem classique connecté sur port USB, série ou interne) relié à une prise téléphonique classique.

L'établissement d'une liaison PPP nécessite :

- Un client disposant des outils ppp (pppd) et chat pour dialoguer avec le serveur.
- Un serveur disposant de pppd et des moyens de fournir une adresse IP (dhcp).
- Un modem.

La suite ne prend en considération que la partie cliente.

Vous devez connaître :

- Le port série sur lequel est branché votre modem : ttySX (série ou USB), ttyACMX (usb), rfcommX (bluetooth), etc.
- Le numéro d'appel de votre fournisseur d'accès Internet (FAI).
- Le nom d'utilisateur et le mot de passe chez votre FAI.
- L'adresse du serveur DNS de votre FAI.

Le modem n'est pas forcément RTC. Un téléphone portable reconnu comme modem via le câble de connexion au PC ou depuis le protocole Bluetooth fait un excellent modem. Pour peu qu'il soit aux normes 3G ou Edge, les débits peuvent être très impressionnants. Un grand nombre de connexions ADSL sont aussi effectuées via le protocole PPP. Dans ce cas la suite s'applique mais des modifications sont à prévoir.

3. Connexion via la console

a. À la main

Dans l'exemple qui suit :

- Le numéro de téléphone est 0102030405.
- Le login est « login ».
- Le mot de passe est « password ».
- Le périphérique est /dev/modem.

Il peut être nécessaire, bien que cela puisse être géré par DHCP au moment de la connexion, de modifier le fichier /etc/resolv.conf pour indiquer les serveurs de noms (DNS) de votre fournisseur.

La connexion PPP nécessite que le service **pppd** (généralement /usr/sbin/pppd) soit exécuté en tant que root. Pour cela, le droit SUID est souvent positionné :

```
-rwsr-xr-t 1 root root 316392 2008-04-04 19:03 pppd*
```

Une autre solution est de donner ces droits à l'outil de connexion (chat, kppp, etc.) ou de modifier en conséquence les droits des périphériques (cas de plusieurs distributions).

Les fichiers de configuration sont situés dans /etc/ppp :

```
# ls -l /etc/ppp
total 48
-rw------- 1 root root  690 sep 21  2007 chap-secrets
-rw-r--r-- 1 root root  449 sep 21  2007 filters
lrwxrwxrwx 1 root root    5 mai  9 20:47 ip-down -> ip-up
drwxr-xr-x 2 root root 4096 sep 21  2007 ip-down.d
-rwxr-xr-x 1 root root 6175 avr 24 00:26 ip-up
drwxr-xr-x 2 root root 4096 sep 21  2007 ip-up.d
-rw-r--r-- 1 root root 7943 sep 21  2007 options
-rw------- 1 root root  340 sep 21  2007 options.pptp
-rw------- 1 root root 1219 sep 21  2007 pap-secrets
drwxr-xr-x 2 root root 4096 fév 23 23:18 peers
-rwxr-xr-x 1 root root 3778 avr 24 00:26 poll.tcpip
```

Voici un exemple de connexion à un serveur PPP :

```
#/bin/sh
/usr/sbin/pppd connect '/usr/sbin/chat -v ABORT ERROR ABORT "NO
CARRIER" \
    ABORT BUSY  "" ATZ OK ATDT0102030405 CONNECT "" ogin: "login" \
    word: "password"' \
    /dev/modem 38400 noipdefault debug crtscts modem defaultroute &
```

b. Par les fichiers

Vous allez avoir besoin de deux fichiers. Le premier va contenir les commandes du service pppd, le second la séquence de communication avec le FAI. Les deux sont placés dans /etc/ppp/peers.

Soit le premier fichier /etc/ppp/peers/cnx1 :

```
# cat /etc/ppp/peers/cnx1
/dev/modem
connect '/usr/sbin/chat -v -f /etc/ppp/peers/cnx1-chat'

defaultroute
```

```
noipdefault
usepeerdns
115200

debug
noauth

maxfail 10
lcp-echo-interval 5
lcp-echo-failure 12
holdoff 3
noaccomp noccp nobsdcomp nodeflate nopcomp novj novjccomp
lock
crtscts
```

Chaque ligne contient au moins une instruction dont voici les plus pertinentes :

- **/dev/modem** : le périphérique de connexion (le modem) ;
- **connect** : la chaîne de connexion envoyée au FAI ;
- **defaultroute** : la route par défaut est remplacée par celle fournie par le FAI ;
- **noipdefault** : le FAI fournit l'IP par son DHCP ;
- **usepeerdns** : récupère les informations DNS du FAI ;
- **115200** : la vitesse de communication du périphérique (elle sera négociée) ;
- **debug** : fournit le détail complet de la connexion ;
- **noauth** : ce n'est pas le script ppp qui établit l'authentification (voir ligne connect) ;
- **maxfail** : n tentatives de connexion avant d'abandonner ;
- **holdoff** : attente de n secondes entre deux connexions ;
- **lock** : permet l'accès exclusif au fichier périphérique ;
- **crtscts** : active le contrôle de flux matériel.

Soit le second fichier /etc/ppp/peers/cnx1-chat utilisé par la ligne **connex** du premier fichier /etc/ppp/peers/cnx1 :

```
# cat /etc/ppp/peers/cnx1-chat
ABORT ERROR
ABORT "NO CARRIER"
ABORT BUSY  "" ATZ
OK ATDT0102030405
CONNECT ""
ogin: "login"
word: "password"
```

Il ne s'agit que d'un exemple. Vous devez vérifier tant du côté de votre FAI que du côté de la documentation de votre modem quelles sont les bonnes commandes AT à passer (elles sont généralement standard).

c. Connexion

Initialisez la connexion :

```
# pppd call cnx1
```

Vous devriez voir les leds de votre modem clignoter, et si le haut-parleur est activé le bruit caractéristique se fait entendre. Si la connexion est établie, une nouvelle interface réseau apparaît : **ppp0**.

```
# ifconfig ppp0
ppp0      Link encap:Point-Point Protocol
          inet addr:10.xx.yy.zz  P-t-P:10.xx.yy.zz Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

d. Connexion par front-end

Plutôt que d'établir une connexion depuis les commandes en ligne et des fichiers de configuration, il existe plusieurs outils graphiques, front-ends à PPP, qui permettent à la fois de configurer le modem et d'établir une connexion PPP avec tous les réglages réseaux possibles.

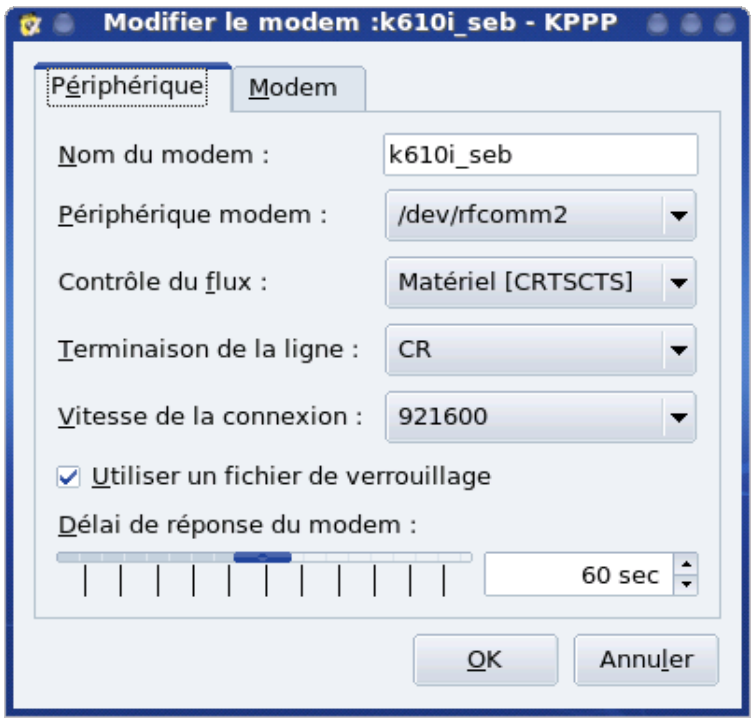
KDE est un très bon exemple avec l'outil **kppp**. L'interface principale ressemble à ce que vous avez peut-être pu déjà connaître sous Windows :

- le choix d'un profil de connexion,
- un nom d'utilisateur,
- un mot de passe,
- et les boutons, dont celui permettant de se connecter.

kppp permet d'établir une connexion PPP

La configuration se fait en trois étapes :

- configuration du modem (la capture représente un téléphone portable 3G via une connexion bluetooth),
- configuration des informations de connexion : profil, numéro de téléphone, sauvegarde du mot de passe, coût, etc.,
- configuration des paramètres réseaux : réglages du DNS, du client DHCP, de la passerelle, etc.



Configuration du modem avec kppp

Enfin, une fois connecté une fenêtre d'état vous permet d'obtenir des informations détaillées sur la connexion, sa durée et vous offre la possibilité de vous déconnecter.

La liaison ppp est établie.

Les distributions sont souvent accompagnées d'outils de type NetworkManager qui permettent une connexion ethernet, Wi-Fi ou ppp à la volée en quelques clics. La distribution Mandriva mérite une mention spéciale : la configuration d'une connexion Internet via un mobile 3G au travers du protocole Bluetooth s'est fait en quelques secondes, l'outil Drakconf ayant tout détecté seul.

OpenSSH

1. Présentation

OpenSSH est un protocole de shell sécurisé, un mécanisme qui permet l'authentification sécurisée, l'exécution à distance et la connexion à distance. Il permet aussi le transport sécurisé du protocole X Window. En fait, il est capable d'encapsuler des protocoles non sécurisés en redirigeant les ports.

Les packages à utiliser pour un serveur sont **openssh**, **openssl** et **openssh-clients**. Pour X on rajoute les packages **openssh-askpass*** (il peut y en avoir plusieurs suivant l'environnement de bureau). La liste des packages à installer dépend de chaque distribution.

L'utilisation la plus commune reste l'accès distant sécurisé à une machine via le client ssh.

2. Configuration

La configuration est /etc/ssh/sshd_config. Quelques options sont éventuellement à modifier :

- **Port** : le numéro de port, par défaut 22 ;
- **Protocol** : fixé à 2,1 il autorise SSH1 et SSH2. On préférera SSH2 et donc on laissera la valeur 2 seule ;
- **ListenAddress** : par défaut ssh écoute sur toutes les IP du serveur. On peut autoriser uniquement l'écoute sur une interface donnée ;
- **PermitRootLogin** : ssh autorise les connexions de root. On peut placer la valeur à « **no** ». Dans ce cas, il faudra se connecter en simple utilisateur et passer par **su** ;
- **Banner** : chemin d'un fichier dont le contenu sera affiché aux utilisateurs lors de la connexion.

Ssh est un service System V à lancer avec service ou directement par /etc/init.d/sshd.

```
# service sshd start
```

3. Utilisation

La commande **ssh** permet d'établir une connexion.

```
$ ssh -l login host
$ ssh login@host
```

L'option -X permet d'activer la redirection (forwarding) du protocole X Window.

```
$ ssh -X login@host
```

4. Clés et connexion automatique

Il est possible d'établir une connexion automatique vers une autre machine sans saisir de mot de passe. Pour cela, il est nécessaire depuis le compte utilisateur du client (la machine qui va se connecter) de générer une paire de clés, privée et publique. Aucune passphrase ne doit être saisie.

Du côté du serveur ssh, la clé publique du client doit être placée dans un fichier contenant les clés autorisées à se connecter dans le compte de destination.

a. Côté client

- Générez une clé au format RSA avec la commande **ssh-keygen** :

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bean/.ssh/id_rsa):
Created directory '/home/bean/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bean/.ssh/id_rsa.
Your public key has been saved in /home/bean/.ssh/id_rsa.pub.
The key fingerprint is:
f6:39:23:4e:fa:53:d0:4e:65:7f:3f:fd:a3:f4:8e:2a bean@p64p17bicb3
```

- Le répertoire de l'utilisateur contient maintenant un répertoire .ssh :

```
$ cd .ssh
$ ls
id_rsa  id_rsa.pub
```

- Le fichier id_rsa.pub contient la clé publique :

```
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArB/VskR9v708J2EDG1LM1Q6HmKJc
P2UenurnSr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQE
hhPLwJVLXbGNc8lMf742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2q
zKKx7ZuNQEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzfk
vu7zQbOGDGGPiF3aOvd0oSBNgiJtZ+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3T
Q4xU6WxkxqTBcsjEm1gIymFAyxDo+zzf63jxLGO8Pp50DKf7DUqBx7+rjw==
bean@slyserver
```

b. Côté serveur

- Allez dans le répertoire .ssh du compte auquel vous souhaitez accéder sur le serveur (créez-le s'il n'existe pas) :

```
$ cd /home/seb/.ssh
```

- Éditez le fichier authorized_keys2 (créez-le s'il n'existe pas) et copiez-y sur une nouvelle ligne le contenu du fichier id_rsa.pub du client. Sauvez.

```
$ echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEArB/VskR9v708J2EDG1LM1Q6HmKJcP2Uenurn
Sr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQEhhPLwJVL
XbGNc8lMf742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2qzKKx7ZuN
QEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzfkvu7zQbOG
DGGPiF3aOvd0oSBNgiJtZ+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3TQ4xU6Wxk
xqTBcsjEm1gIymFAyxDo+zzf63jxLGO8Pp50DKf7DUqBx7+rjw== bean@slyser
ver" >> autorized_keys2
```

- Tentez une connexion, le mot de passe n'est pas demandé :

```
$ ssh seb@slyserver
```

Monter un serveur DHCP

1. Présentation

Le service **DHCP** (*Dynamic Host Configuration Protocol*), protocole de configuration dynamique des hôtes, permet aux hôtes d'un réseau de demander et recevoir des informations de configuration (adresse, routage, DNS, etc.). Il y a en général un seul serveur DHCP par segment de réseau même si plusieurs sont possibles. Si le serveur est sur un autre segment, on peut utiliser un agent de retransmission DHCP.

Autrement dit, un client DHCP recherche tout seul un serveur DHCP qui lui communiquera son adresse IP. L'adresse IP est assignée soit dynamiquement à partir de plages d'adresses prédéfinies, soit statiquement en fonction de l'adresse MAC du demandeur. Les informations sont valables un laps de temps donné (un bail) qui peut être renouvelé et configurable.

DHCP est un sur-ensemble de **BOOTP** (*Bootstrap Protocol*). Quand le client cherche à contacter un serveur, c'est BOOTP qui fournit les informations d'adressage. DHCP gère les renouvellements. BOOTP se base sur le protocole de transport UDP.

Un hôte n'a aucune information réseau disponible au démarrage. Il doit trouver seul un serveur DHCP. Pour cela, BOOTP effectue un broadcast sur l'IP 255.255.255.255 avec une trame contenant ses informations (comme son adresse MAC) et les informations souhaitées (type de requête, ici DHCPDISCOVER, port de connexion, etc.). Le broadcast est envoyé par définition à tous les hôtes du réseau local. Quand le serveur DHCP détecte la trame, il effectue lui aussi un broadcast (l'hôte client n'a pas encore d'IP) avec les informations de base souhaitées par l'hôte (DHCPOFFER, premiers paramètres). L'hôte établit une première configuration puis demande confirmation de l'IP (DHCPREQUEST). Le serveur DHCP confirme (DHCPHACK). Le bail est confirmé et le client dispose dès lors de toutes les informations valides.

2. Serveur dhcpd

a. Démarrage

Le serveur **dhcpd** est un service (daemon) lancé à l'aide d'un script (/etc/init.d/dhcpd). Il est configuré à l'aide du fichier /etc/dhcpd.conf. Les adresses IP allouées sont placées dans /var/lib/dhcp/dhcpd.leases.

```
# service dhcpd start
```

Ou :

```
# /etc/init.d/dhcpd start
```

3. Informations de base

Le fichier de configuration d'un serveur est, si vous restez dans des réglages de base, assez simple.

```
ddns-update-style none; # pas de mise à jour du DNS par DHCP
option domain-name "toto.fr"; # nom de domaine transmis au client
option domain-name-servers 192.168.1.254; # liste des DNS séparés
par des virgules
default-lease-time 21600; # durée du bail par défaut en secondes
sans demande explicite
max-lease-time 43200; # durée max du bail si la demande du client
est plus élevée
```

Comme dhcpd peut gérer plusieurs sous-réseaux, on doit lui préciser les règles à appliquer pour chaque sous-réseau. Généralement dans le cadre d'un petit réseau un seul bloc sera présent mais tous les cas sont envisageables. Si vous êtes certain de n'avoir qu'un seul réseau, vous pouvez omettre la déclaration du subnet (sous-réseau).

```
# Gestion du sous-réseau 192.168.1.0
subnet 192.168.1.0 netmask 255.255.255.0
{
  option routers 192.168.1.254; # passerelle pour ce réseau
  option subnet-mask 255.255.255.0; # masque de sous-réseau
```

```
  range 192.168.1.2 192.168.1.250; # Configuration de l'intervalle DHCP

  # Cas d'attributions d'IP statiques
  host station1
  {
    hardware ethernet 00:A0:ad:41:5c:b1; # Adresse MAC
    fixed-address 192.168.1.1; # cette machine aura l'IP 192.168.1.1
  }
```

Certains clients DHCP ignorent totalement le fait qu'un serveur DHCP peut dynamiquement allouer un nom (hostname) à l'hôte. Dans l'exemple précédent, la machine avec l'IP 192.168.1.1 devrait obtenir le nom station1. Voici un exemple :

```
  # les hôtes se verront attribués les noms des host déclarés
  use-host-decl-names on;
  host station1
  {
    hardware ethernet 00:A0:ad:41:5c:b1; # Adresse MAC
    fixed-address 192.168.1.1; # cette machine aura l'IP 192.168.1.1
et le nom station1
  }
```

Vous pouvez aussi travailler au cas par cas :

```
  host station2
  {
    hardware ethernet 00:A0:ad:41:5c:b2; # Adresse MAC
    fixed-address 192.168.1.251; # ce host aura l'IP 192.168.1.251
    option host-name "station2"; # ce host aura comme nom station2
  }
```

4. Côté client

Sous Linux et les distributions de type Red Hat, Fedora, Mandriva, openSUSE, etc., modifiez le fichier **`/etc/sysconfig/network-script/ifcfg-xxx`** en spécifiant dhcp pour **BOOTPROTO** et relancez la connexion réseau (**ifdown** puis **ifup**).

Le client dhcpcd permet d'activer dhcp sur une interface réseau. La méthode la plus simple est, par exemple pour eth0 :

```
# dhcpcd eth0 &
```

Vous pouvez transmettre des options à dhcpcd pour prendre en charge diverses possibilités. Parmi ces options :

- `-D` : autorise la modification du nom de domaine ;
- `-H` : autorise la modification du nom d'hôte ;
- `-R` : évite l'écrasement du fichier resolv.conf ;
- `-l` : permet de modifier le leasetime (en secondes).

```
# dhcpcd -D -H -l 86400 eth0
```

Serveur DNS

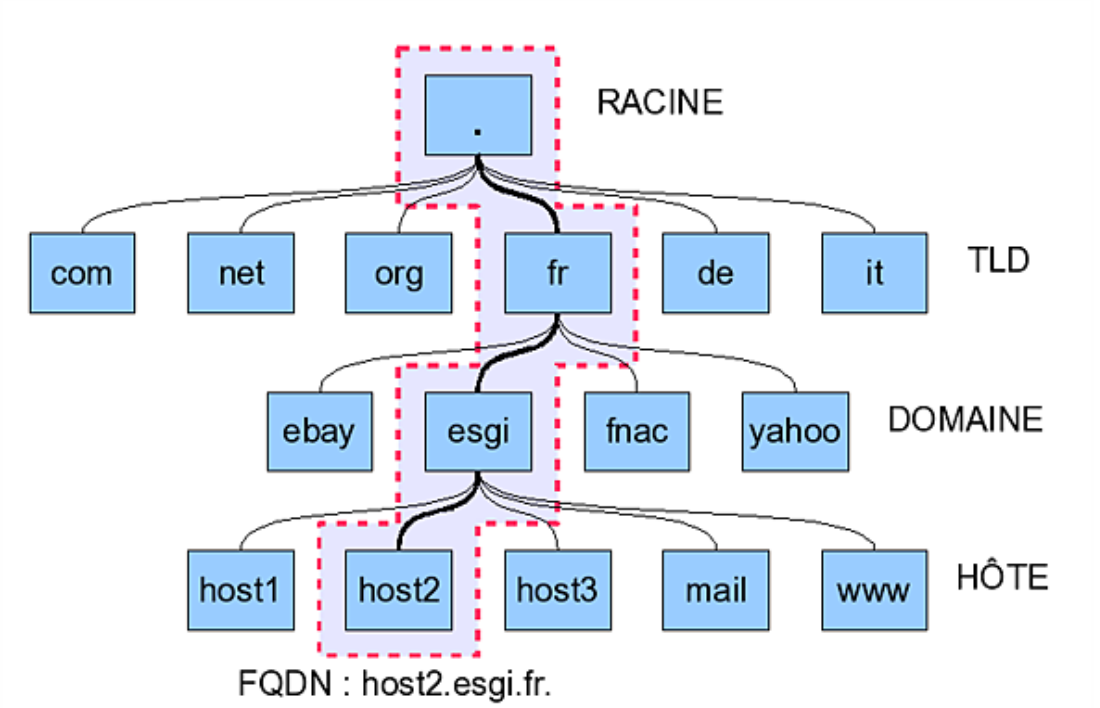
1. Présentation

Le Système de Noms de Domaines **DNS** (*Domain Name System*) transforme les noms d'hôte en adresses IP : c'est la **résolution de nom**. Il transforme les adresses IP en noms d'hôte : c'est la **résolution inverse**. Il permet de regrouper les machines par domaines de nom. Il fournit des informations de routage et de courrier électronique.

Le DNS permet de faire référence à des systèmes basés sur IP (les *hôtes*) à l'aide de noms conviviaux (les *noms de domaines*). L'intérêt d'un DNS est évident. Les noms de domaine sont plus simples à retenir, et si son adresse IP change l'utilisateur ne s'en rend même pas compte. On comprend que le DNS est un service clé critique pour Internet.

Les noms de domaine sont séparés par des points, chaque élément pouvant être composé de 63 caractères ; il ne peut y avoir qu'un maximum de 127 éléments et le nom complet ne doit pas dépasser 255 caractères. Le nom complet non abrégé est appelé **FQDN** (*Fully Qualified Domain Name*). Dans un FQDN, l'élément le plus à droite est appelé **TLD** *(Top Level Domain)*, celui le plus à gauche représente l'hôte et donc l'adresse IP.

Le DNS contient une configuration spéciale pour les routeurs de courrier électronique (définitions MX) permettant une résolution inverse, un facteur de priorité et une tolérance de panne.



Représentation d'une arborescence DNS

Une zone est une partie d'un domaine gérée par un serveur particulier. Une zone peut gérer un ou plusieurs sous-domaines, et un sous-domaine peut être réparti en plusieurs zones. Une zone représente l'unité d'administration dont une personne peut être responsable.

2. Lancement

Le service s'appelle **named**.

```
# service named start
```

Ou :

```
# /etc/init.d/named start
```

3. Configuration de Bind

Bind (*Berkeley Internet Name Daemon*) est le serveur de noms le plus utilisé sur Internet. Bind 9 supporte l'IPv6, les noms de domaine unicode, le multithread et de nombreuses améliorations de sécurité.

a. Configuration générale

La configuration globale de Bind est placée dans le fichier **`/etc/named.conf`**. La configuration détaillée des zones est placée dans /var/lib/named. /etc/named.conf est composé de deux parties. La première concerne la configuration globale des options de Bind. La seconde est la déclaration des zones pour les domaines individuels. Les commentaires commencent par un # ou //.

Attention il arrive parfois (notamment sur RHEL 4.x) que la configuration de Bind soit « chrootée » (déplacée dans une arborescence spécifique d'où le service ne peut sortir, le reste de l'arborescence lui étant inaccessible). Sur Centos et RHEL 4.x et supérieurs named.conf est dans /var/named/chroot/etc/.On peut modifier ce mode en modifiant le fichier de configuration /etc/sysconfig/named.

```
# cat /etc/sysconfig/named
...
CHROOT=/var/named/chroot
...
```

Dans ce cas, tous les fichiers de configuration, y compris les zones, sont relatifs à ce chemin. Voici un fichier named.conf de base.

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};
zone "localhost" in {
    type master;
    file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};
zone "." in {
    type hint;
    file "root.hint";
};
```

b. Section globale

La configuration globale est placée dans la section **options**. Voici un détail de quelques options importantes (le point-virgule doit être précisé) :

- **directory "filename";** : emplacement des fichiers contenant les données des zones.
- **forwarders { adresse-ip; };** : si le serveur bind ne peut résoudre lui-même la requête, elle est renvoyée à un serveur DNS extérieur, par exemple celui du fournisseur d'accès.
- **listen-on-port 53 {127.0.0.1; adresse-ip; };** : port d'écoute du DNS suivi des adresses d'écoute. On indique ici les adresses IP des interfaces réseau de la machine. Il ne faut pas oublier 127.0.0.1.
- **allow-query { 127.0.0.1; réseau; };** : machine(s) ou réseau(x) autorisés à utiliser le service DNS. Par exemple 192.168.1/24. Si la directive est absente, tout est autorisé.
- **allow-transfer { 192.168.1.2; };** : machine(s) ou réseau(x) autorisés à copier la base de données dans le cas d'une relation maître et esclave. Par défaut aucune copie n'est autorisée.

- **notify no;** : on notifie ou non les autres serveurs DNS d'un changement dans les zones ou d'un redémarrage du serveur.

c. Section de zones

Pour chaque domaine ou sous-domaine, on définit deux sections **zone**. La première contient les informations de résolution de nom (nom vers IP) et la seconde les informations de résolution inverse (IP vers Nom). Dans chacun des cas, la zone peut être maître **Master** ou esclave **Slave** :

- **Master** : le serveur contient la totalité des enregistrements de la zone dans ses fichiers de zone. Lorsqu'il reçoit une requête, il cherche dans ses fichiers (ou dans son cache) la résolution de celle-ci.

- **Slave** : le serveur ne contient par défaut aucun enregistrement. Il se synchronise avec un serveur maître duquel il récupère toutes les informations de zone. Ces informations peuvent être placées dans un fichier. Dans ce cas l'esclave stocke une copie locale de la base. Lors de la synchronisation, le numéro de série de cette copie est comparé à celui du maître. Si les numéros sont différents, une nouvelle copie a lieu, sinon la précédente continue à être utilisée.

d. Zone de résolution

Elle est généralement appelée **zone**. Pour chaque domaine ou sous-domaine, elle indique dans quel fichier sont placées les informations de la zone (c'est-à-dire et entre autres les adresses IP associées à chaque hôte), son type (maître ou esclave), si on autorise ou non la notification, l'adresse IP du serveur DNS maître dans le cas d'un esclave, etc.

Le nom de la zone est très important puisque c'est lui qui détermine le domaine de recherche. Quand le DNS reçoit une requête, il recherche dans toutes les zones une correspondance.

```
zone "domaine.org" {
    type     "master";
    file    "domaine.org.zone";
};
```

- **type** : master ou slave ;

- **file** : nom du fichier qui contient les informations de la zone. Il n'y a pas de règles précises de nommage mais pour des raisons de lisibilité il est conseillé de lui donner le même nom que la zone tant pour une zone master que pour une slave. Pour un master, c'est l'original éventuellement rempli par vos soins. Pour un slave, ce n'est pas obligatoire. S'il est présent, ce sera une copie du master, synchronisée.

- Dans le cas d'un Master, on peut rajouter **allow-transfer** (serveurs autorisés à dupliquer la zone) et **notify yes** (indique une mise à jour ou une relance pour les slaves).

En cas de Slave : on rajoute la directive **masters** pour indiquer à partir de quel serveur Master dupliquer.

e. Zone de résolution inverse

Pour chaque réseau ou sous-réseau IP (ou plage d'adresses) on définit une zone de résolution inverse dont le fichier contient une association IP vers nom de machine. C'est en fait presque la même chose que la zone de résolution sauf que l'on doit respecter une convention de nommage :

- Le nom de la zone se termine toujours par une domaine spécial **.in-addr.arpa**.

- On doit tout d'abord déterminer quel réseau la zone doit couvrir (cas des sous-réseaux). Pour nous : un réseau de classe C 192.168.1.0 soit **192.168.1/24**.

- On inverse l'ordre des octets dans l'adresse : **1.168.192**.

- On ajoute **in-addr.arpa**. Notre nom de zone sera donc **1.168.192.in-addr.arpa**.

- Pour le reste, les mêmes remarques que pour la zone de résolution s'appliquent.

```
Zone "1.168.192.in-addr.arpa" {
    type         master;
    file        "192.168.1.zone";
};
```

f. Exemple

Soit un domaine **domaine.org** sur un réseau de classe C 192.168.1.0. Soit deux serveurs DNS 192.168.1.1 Master et 192.168.1.2 Slave.

Sur le Master

```
zone "domaine.org" {
    type    master;
    file    "domaine.org.zone";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
zone "1.168.192.in-addr.arpa" {
    type    master;
    file    "192.168.1.zone";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
```

Sur le Slave

```
zone "domaine.org" {
    type        slave;
    file        "domaine.org.zone";
    masters     { 192.168.1.1; };
};
zone "1.168.192.in-addr.arpa" {
    type        slave;
    file        "192.168.1.zone";
    masters     { 192.168.1.1; } ;
};
```

g. Zones spéciales

La zone racine « . » permet de spécifier les serveurs racines. Quand aucune des zones n'arrive à résoudre une requête, c'est la zone racine qui est utilisée par défaut et qui renvoie sur les serveurs racines.

La zone de loopback n'est pas nécessaire bien que utile. Elle fait office de **cache DNS**. Quand une requête arrive sur le serveur et qu'il ne possède pas l'information de résolution, il va la demander aux serveurs DNS racines qui redescendront l'information. Celle-ci est alors placée en cache. Du coup les accès suivants seront bien plus rapides !

4. Fichiers de zones

a. Définitions

Les fichiers de zones utilisent plusieurs termes, caractères et abréviations spécifiques.

- **RR** : *Ressource Record*. Nom d'un enregistrement DNS (les données du DNS).
- **SOA** : *Star Of Authority*. Permet de décrire la zone.
- **IN** : *the Internet*. Définit une classe d'enregistrement qui correspond aux données Internet (IP). C'est celle par défaut si elle n'est pas précisée pour les enregistrements.

- **A** : *Address*. Permet d'associer une adresse IP à un nom d'hôte. Pour Ipv6 c'est AAAA.
- **NS** : *Name Server*. Désigne un serveur DNS de la zone.
- **MX** : *Mail eXchanger*. Désigne un serveur de courrier électronique, avec un indicateur de priorité. Plus la valeur est faible, plus la priorité est élevée.
- **CNAME** : *Canonical Name*. Permet de rajouter des alias : lier un nom à un autre. On peut créer des alias sur des noms d'hôte et aussi sur des alias.
- **PTR** : *Pointer*. Dans une zone de résolution inverse, fait pointer une IP sur un nom d'hôte.
- **TTL** : *Time To Live*. Durée de vie des enregistrements de la zone.
- **@** : dans les déclarations de la zone, c'est un alias (caractère de remplacement) pour le nom de la zone déclarée dans /etc/named.conf. Ainsi si la zone s'appelle domaine.org, @ vaut domaine.org. Dans la déclaration de l'administrateur de la SOA, il remplace ponctuellement le point dans l'adresse de courrier électronique.
- Le point « **.** »: Si l'on omet le point en fin de déclaration d'hôte, le nom de la zone est concaténé à la fin du nom. Par exemple pour la zone domaine.org, si on écrit **poste1**, cela équivaut à **poste1.domaine.org**. Si on écrit **poste1.domaine.org** (sans le point à la fin) alors on obtient comme résultat **poste1.domaine.org.domaine.org** ! Pour éviter cela, vous devez écrire **poste1.domaine.org.** (notez le point à la fin).
- Certains enregistrements nécessitent une notion de durée, qui est généralement exprimée en secondes, mais aussi parfois avec des abréviations :
  - **1M** : une minute, soit 60 secondes (1M, 10M, 30M, etc.) ;
  - **1H** : une heure, 3600 secondes ;
  - **1D** : un jour, 86400 secondes ;
  - **1W** : une semaine, 604800 secondes ;
  - **365D** : un an, 31536000 secondes.


Attention, et ceci est très important : dans les fichiers de zones, IL NE FAUT JAMAIS COMMENCER UNE LIGNE PAR DES ESPACES OU TABULATIONS. Ça ne marche absolument pas : les espaces ou tabulations seraient interprétés comme faisant partie du nom indiqué, de l'adresse ou de l'option.

b. Zone

Commencez tout d'abord par une directive **TTL** qui indique le temps de vie de la zone en secondes. Cela signifie que chaque enregistrement de la zone sera valable durant le temps indiqué par **TTL** (note : il est possible de modifier cette valeur pour chaque enregistrement). Durant ce temps, les données peuvent être placées en cache par les autres serveurs de noms distants. Une valeur élevée permet de réduire le nombre de requêtes effectuées et de rallonger les délais entre les synchronisations.

```
$TTL 86400
```

Après les directives TTL, placez un enregistrement de ressources **SOA** :

```
<domain>    IN    SOA    <primary-name-server> <hostmaster-email> (
                <serial-number>
                <time-to-refresh>
                <time-to-retry>
                <time-to-expire>
                <minimum-TTL> )
```

- **domain** : c'est le nom de la zone, le même nom que celui utilisé dans /etc/named.conf. On peut le remplacer par @ sinon il ne faut pas oublier de le terminer par un point (pour éviter une concaténation).

- **primary-name-server** : le nom sur le serveur DNS maître sur cette zone. Il ne faudra pas oublier de le déclarer dans la liste des hôtes (enregistrements PTR ou A).

- **hostmaster-email** : adresse de courrier électronique de l'administrateur du serveur de nom. Le caractère @ étant déjà réservé à un autre usage, on utilise un point pour le remplacer. Ainsi « admin@domaine.org » devra s'écrire « **admin.domaine.org.** » .

- **serial-number** : c'est un numéro de série que l'on doit incrémenter manuellement à chaque modification du fichier zone pour que le serveur de nom sache qu'il doit recharger cette zone. Elle est utilisée pour la synchronisation avec les serveurs esclaves. Si le numéro de série est le même qu'à la dernière synchronisation les données ne sont pas rafraîchies. Par convention on place **YYYYMMDDNN** (année-mois-jour-numéro) sur dix chiffres.

- **time-to-refresh** : indique à tout serveur esclave combien de temps il doit attendre avant de demander au serveur de noms maître si des changements ont été effectués dans la zone.

- **time-to-retry** : indique au serveur esclave combien de temps attendre avant d'émettre à nouveau une demande de rafraîchissement si le serveur maître n'a pas répondu. La demande aura lieu toutes les time-to-retry secondes.

- **time-to-expire** : si malgré les tentatives de contacts toutes les time-to-retry secondes le serveur n'a pas répondu au bout de la durée indiquée dans time-to-expire, le serveur esclave cesse de répondre aux requêtes pour cette zone.

- **Minimum-TTL** : le serveur de nom demande aux autres serveurs de noms de mettre en cache les informations pour cette zone pendant au moins la durée indiquée.

```
@    IN    SOA    dns1.domaine.org.    hostmaster.domaine.org. (
             2005122701    ; serial
             21600         ; refresh de 6 heures
             3600          ; tenter toutes les 1 heures
             604800        ; tentatives expirent après une semaine
             86400    )    ; TTL mini d'un jour
```

Passez ensuite au enregistrements **NS** (*Name Server*) où vous spécifiez les serveurs de noms de cette zone.

```
    IN    NS    dns1
    IN    NS    dns2
```

Quand on ne spécifie pas en début de ligne un nom d'hôte ou de zone (complet ou @), cela veut dire qu'on utilise le même que la ligne du dessus. Tant qu'on n'en précise pas de nouveau, c'est le dernier indiqué qui est utilisé. Ainsi ci-dessus les lignes pourraient être :

```
@    IN    NS    dns1
@    IN    NS    dns2
```

ou :

```
domaine.org.    IN    NS    dns1
domaine.org.    IN    NS    dns2
```

Notez l'absence de point après le nom de l'hôte et donc domaine.org est concaténé pour obtenir dns1.domaine.org.

```
IN NS dns1
```

équivaut à :

```
IN NS dns1.domaine.org.
```

Passez ensuite à l'énumération des serveurs de courrier électronique de la zone. La valeur numérique située après MX indique la priorité. Plus la valeur est basse plus le serveur est prioritaire et susceptible d'être contacté en premier. Si les valeurs sont identiques, le courrier est redistribué de manière homogène entre les serveurs. Si un serveur ne répond pas (chargé, en panne) la bascule vers une autre machine est automatique.

```
    IN    MX    10    mail
    IN    MX    15    mail2
```

Si vous souhaitez qu'une machine réponde en passant par le FQDN domaine.org sans préciser d'hôte (par exemple http://domaine.org sans utiliser http://www.domaine.org) alors vous pouvez maintenant déclarer une adresse IP pour ce serveur. Ainsi la commande **ping domaine.org** répondra 192.168.1.3 !

```
IN A 192.168.1.3
```

Vous pouvez maintenant déclarer les autres hôtes dont les serveurs de noms, de mails, les postes, etc.

```
dns1        IN    A    192.168.1.1
dns2        IN    A    192.168.1.2
server1     IN    A    192.168.1.3
server2     IN    A    192.168.1.4
poste1      IN    A    192.168.1.11
poste2      IN    A    192.168.1.12
poste3      IN    A    192.168.1.13
```

On remarque que nos serveurs mail et mail2 ne sont pas déclarés, et que l'on n'a pas indiqué de serveur Web et ftp. Nous allons utiliser les alias, en faisant pointer ces noms d'hôtes sur d'autres hôtes.

```
mail        IN    CNAME     server1
mail2       IN    CNAME     server2
www         IN    CNAME     server1
ftp         IN    CNAME     server1
```

La configuration de la zone est terminée, il faut maintenant s'occuper de la zone de résolution inverse.

c. Zone de résolution inverse

La zone de révolution inverse est presque identique à la précédente, si ce n'est que les enregistrements A sont remplacés par des enregistrements PTR destinés à traduire une IP en hôte. Le TTL et la déclaration SOA doivent être si possible identiques (sauf le nom de la zone). Vous placez aussi les enregistrements NS.

```
    IN    NS    dns1.domaine.org.
    IN    NS    dns2.domaine.org.
```

Vous n'êtes pas obligé de placer dans la zone de résolution inverse la traduction des adresses IP du DNS, étant donné que c'est le DNS lui-même qui résout son propre nom ! Cependant le faire peut accélérer la démarche, le DNS n'ayant pas à exécuter une requête sur lui-même. Passez aux enregistrements PTR traduisant l'adresse IP pour chaque hôte.

```
1     IN    PTR    dns1.domaine.org.
2     IN    PTR    dns2.domaine.org.
3     IN    PTR    server1.domaine.org.
4     IN    PTR    server2.domaine.org.
11    IN    PTR    poste1.domaine.org.
12    IN    PTR    poste2.domaine.org.
13    IN    PTR    poste3.domaine.org.
```

Il est théoriquement possible pour la même IP d'attribuer plusieurs hôtes ; les RFC ne sont pas très explicites sur cette possibilité qui, au final, peut créer des problèmes.

5. Diagnostic des problèmes de configuration

La commande **named-checkconf** vérifie la syntaxe du fichier **named.conf**. Vous lui fournissez en paramètre le fichier. La sortie indiquera les lignes posant problème.

La commande **named-checkzone** vérifie la syntaxe d'un fichier de zone (y compris de résolution inverse). Vous lui spécifiez en paramètre le nom du fichier zone.

a. Interrogation dig et host

Le programme **dig** est un outil d'interrogation avancé de serveur de noms, capable de restituer toutes les informations des zones.

```
> dig free.fr

; <<>> DiG 9.4.1-P1 <<>> free.fr
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63972
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;free.fr.                       IN      A

;; ANSWER SECTION:
free.fr.                86363   IN      A       212.27.48.10

;; Query time: 1 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:36:09 2008
;; MSG SIZE  rcvd: 41
```

Par défaut dig ne restitue que l'adresse de l'hôte passé en paramètre. En cas de réussite, le statut vaut **NOERROR**, le nombre de réponses est indiqué par **ANSWER** et la réponse se situe en dessous de la section **ANSWER**. Pour obtenir une résolution inverse il existe deux solutions.

```
$ dig 10.48.27.212.in-addr.arpa ptr
```

ou plus simplement :

```
$ dig -x 212.27.48.10

; <<>> DiG 9.4.1-P1 <<>> -x 212.27.48.10
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60222
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;10.48.27.212.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
10.48.27.212.in-addr.arpa. 86400 IN     PTR     www.free.fr.

;; Query time: 31 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:36:51 2008
;; MSG SIZE  rcvd: 68
```

Dans la première syntaxe, remarquez que vous pouvez rajouter un paramètre d'interrogation. Voici les principaux.

- **a** : uniquement l'adresse ;
- **any** : toutes les informations concernant le domaine ;
- **mx** : les serveurs de messagerie ;
- **ns** : les serveurs de noms ;
- **soa** : la zone Start of Authority ;

- **hinfo** : infos sur l'hôte ;
- **txt** : texte de description ;
- **ptr** : zone reverse de l'hôte ;
- **axfr** : liste de tous les hôtes de la zone.

```
$ dig free.fr any
; <<>> DiG 9.4.1-P1 <<>> free.fr any
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28893
;; flags: qr aa; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 8

;; QUESTION SECTION:
;free.fr.                       IN      ANY

;; ANSWER SECTION:
free.fr.                86400   IN      NS      freens2-g20.free.fr.
free.fr.                86400   IN      A       212.27.48.10
free.fr.                86400   IN      NS      freens1-g20.free.fr.
free.fr.                86400   IN      MX      20 mx2.free.fr.
free.fr.                86400   IN      SOA     freens1-g20.free.fr.
hostmaster.proxad.net. 2008051001 10800 3600 604800 86400
free.fr.                86400   IN      MX      10 mx1.free.fr.

;; ADDITIONAL SECTION:
freens2-g20.free.fr.    86400   IN      A       212.27.60.20
mx1.free.fr.            86400   IN      A       212.27.48.6
mx2.free.fr.            86400   IN      A       212.27.42.56
freens1-g20.free.fr.    86400   IN      A       212.27.60.19
mx2.free.fr.            86400   IN      A       212.27.42.58
mx1.free.fr.            86400   IN      A       212.27.48.7
mx2.free.fr.            86400   IN      A       212.27.42.57
mx2.free.fr.            86400   IN      A       212.27.42.59

;; Query time: 9 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:35:32 2008
;; MSG SIZE  rcvd: 318
```

L'outil **host** fournit le même résultat de manière peut-être un peu plus simple.

```
$ host free.fr
free.fr has address 212.27.48.10
free.fr mail is handled by 10 mx1.free.fr.

$ host -t any free.fr
free.fr has address 212.27.48.10
free.fr name server freens1-g20.free.fr.
free.fr has SOA record freens1-g20.free.fr. hostmaster.proxad.net.
2008051001 10800 3600 604800 86400
free.fr mail is handled by 10 mx1.free.fr.

$ host -a free.fr
Trying "free.fr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64513
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;free.fr.                       IN      ANY

;; ANSWER SECTION:
free.fr.                86140   IN      A       212.27.48.10
free.fr.                86140   IN      NS      freens1-g20.free.fr.
free.fr.                86140   IN      SOA     freens1-g20.free.fr.
```

```
hostmaster.proxad.net. 2008051001 10800 3600 604800 86400
free.fr.               86140   IN      MX      10 mx1.free.fr.

;; ADDITIONAL SECTION:
freens1-g20.free.fr.    86140   IN      A       212.27.60.19
mx1.free.fr.            86140   IN      A       212.27.48.7

Received 176 bytes from 10.23.254.240#53 in 4 ms
```

Courrier électronique

1. Principe

- Quand un client (un utilisateur) envoie un message, il utilise un **MUA** (*Mail User Agent*), par exemple Outlook Express, Thunderbird, Evolution, Kmail, Mutt, etc.
- Le **MUA** envoie le message au **MTA** (*Mail Transport Agent*). Le MTA étudie l'adresse électronique pour isoler l'utilisateur et le domaine de destination. Puis il vérifie les informations DNS de type **MX** (*Mail exchanger*) pour le domaine choisi, pour savoir à quel serveur transmettre le courrier. Si aucun MTA n'est disponible, le message est placé en file d'attente et relance la distribution plus tard (le délai dépend de la configuration du MTA).
- Le MX peut être soit un autre MTA, qui jouera le rôle de routeur (cas d'une redirection vers un sous-domaine par exemple), soit un **MDA** (*Mail Delivery Agent*). Le MDA place le message dans un fichier temporaire, peut le filtrer, etc.
- À ce niveau, le destinataire reçoit le message : soit il le récupère en lisant directement le fichier temporaire (cas de la commande mail par exemple) soit il passe par un protocole de type **POP** ou **IMAP**.
- Le protocole de transport de messages est le **SMTP** (*Simple Mail Transfer Protocol*) sur le port 25.
- Les protocoles de réception de messages soit **POP** (*Post Office Protocol*) sur le port 110 (POP3), soit **IMAP** (*Internet Message Access Protocol*).

Deux suites de courrier électronique se partagent l'essentiel du marché sur Unix : **sendmail** et **postfix**.

La suite libre **sendmail** est la plus connue et la plus utilisée. Sendmail a été créé en 1981 par Eric Allman et a été intégré à BSD 4.2 en 1983. On estimait en 2000 son utilisation à plus de 100 millions de serveurs de courrier électronique. Tant qu'il ne faut pas modifier fortement sa configuration de base, sendmail est idéal. Si vous souhaitez aller plus loin, l'achat d'un livre complet s'avère plus que nécessaire. La configuration de sendmail est si complexe qu'un langage de macros appelé **m4** a été inventé rien que pour lui. Aussi, vous n'éditez pas (ou très rarement) le fichier de configuration de sendmail : vous éditez le fichier source des macros et vous le « recompilez » : m4 va créer le fichier de configuration de sendmail. Sendmail est devenu un monstre de puissance et de configuration.

Le produit **postfix** tend à être de plus en plus utilisé non pas par les déçus de sendmail mais par ceux qui craignent de devoir le configurer. C'est une alternative à sendmail. Les buts de ses développeurs (dont certains sont ceux de sendmail) sont :

- la compatibilité avec sendmail ;
- la rapidité (plus de un million de messages par jour sur un simple Pentium 4) ;
- la simplicité d'administration (fichier de configuration simple et lisible) ;
- la sécurité (peut être chrootée) ;
- la modularité (décomposition des traitements).

On retiendra la simplicité. En effet, on peut configurer postfix avec une seule commande sans avoir besoin d'éditer de fichier. C'est ce serveur que vous allez utiliser.

2. postfix

a. Configuration simple

La configuration de **postfix** est située dans /etc/postfix/main.cf. On modifie ses valeurs soit à la main, soit à l'aide de la commande **postconf**.

Postfix lance tout d'abord un service maître, **master**, qui sera chargé des processus secondaires **smtpd**, **pickup** et **nqmgr**.

Sur certaines distributions il faut modifier la configuration par défaut qui utilise la suite sendmail. Par exemple sur Red Hat vous devez indiquer d'utiliser postfix à la place de sendmail avec la commande **alternatives**.

```
#alternatives --set mta /usr/sbin/sendmail.postfix
```

Appliquez une configuration de base avec la commande **postconf**.

Domaine d'origine des messages

```
#postconf -e "myorigin = mondomaine.org"
```

De quels domaines recevoir le courrier

```
#postconf -e "mydestination = mondomaine.org"
```

De quels clients relayer le courrier

```
#postconf -e "mynetworks = 192.168.1.0/24, 127.0.0.1"
```

Sur quelles interfaces écouter

```
#postconf -e "inet_interface = all"
```

Lancez le service.

```
#service postfix start
```

ou :

```
#/etc/init.d/postfix start
```

b. Alias d'utilisateurs

Vous pouvez placer dans le fichier **/etc/aliases** des alias pour les utilisateurs locaux. Par exemple, si les messages de webmaster, admin et root doivent être redirigés vers regis :

```
regis: webmaster, admin, root
```

c. Test

Les traces sont placées dans /var/log/maillog. Testez le serveur de cette manière (par exemple) :

```
mail -s `echo $USER` root@server1 < /etc/passwd
```

Si tout fonctionne, vous obtiendrez des traces :

```
Fri 26 11:38:18 station1 postfix/pickup[12357] : F145040154: uid=0
from <root>
Fri 26 11:38:18 station1 postfix/cleanup[12318] : F145040154: mes-
sage-id=<20060126113017.F145040154:@station1.mondomaine.org>
Fri 26 11:38:26 station1 postfix/nqmgr[3469] : F145040154:
from=<root@station1.example.com>, size=314, nrcpt=1 (queue active)
Fri 26 11:38:32 station1 postfix/smtp[12468] : F145040154:
to=<root@server1>, relay=server1.mondomaine.org[192.168.1.1], de-
lay=17, status=sent (250 ok dirdel)
```

3. POP et IMAP

Il existe plusieurs suites pour gérer POP et IMAP. Une suite se nomme **cyrus-imap** et est en principe réservée aux grosses structures et aux serveurs 100% dédiés au courrier, c'est-à-dire où les utilisateurs ne se connectent pas.

Une solution se nomme **dovecot**. Après son installation, il suffit juste de le démarrer en tant que service pour que tout fonctionne, ou presque.

Éditez le fichier **/etc/dovecot.conf** pour vérifier les protocoles supportés.

```
protocols = imap pop3
```

Lancez le service :

```
service dovecot start
```

ou :

```
/etc/init.d/dovecot start
```

Testez en envoyant un message. Configurez un client de messagerie pour vérifier si le serveur POP fonctionne :

```
# telnet localhost 110
trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
+OK POP3 localhost.localdomain server ready
USER regis
+OK user name accepted, password please
PASS password
+OK Mailbox open, 1 messages
STAT
+OK 1 384
TOP 1 99999
<message ici>
...
DELE 1
+OK Message deleted
QUIT
+OK bye
```

Service HTTP Apache

1. Présentation

Apache 2 est le serveur HTTP le plus utilisé actuellement sur les serveurs Web. Sa configuration et sa flexibilité en font un serveur incontournable.

Lorsqu'un serveur Apache reçoit des requêtes, il peut les redistribuer à des processus fils. La configuration permet de lancer des processus de manière anticipée et d'adapter dynamiquement ce nombre en fonction de la charge.

Apache est modulaire. Chaque module permet d'ajouter des fonctionnalités au serveur. Le module le plus connu est probablement celui gérant le langage PHP, « mod_php ». Chaque module s'ajoute via les fichiers de configuration, et il n'y a même pas besoin de relancer le serveur Apache : on lui donne juste l'ordre de relire sa configuration.

Apache peut gérer plusieurs sites Web en même temps, ayant chacun leur nom, à l'aide des hôtes virtuels.

2. Arrêt/Relance

Le nom du service dépend de la distribution. Il est souvent intitulé **apache** ou **httpd**. Suivant la distribution lancez le service via la commande **service** ou directement par son nom /etc/init.d/apache.

- **/etc/init.d/httpd start** : démarre ;
- **/etc/init.d/httpd stop** : stoppe ;
- **/etc/init.d/httpd restart** : redémarre ;
- **/etc/init.d/httpd reload** : demande à Apache de relire sa configuration sans redémarrer.

Apache est fourni avec l'outil **apachectl** qui reprend les paramètres (liste non exhaustive) start, stop, status, reload, et surtout **configtest** qui valide ou non le contenu du fichier de configuration de Apache.

3. Configuration

La configuration principale est stockée dans /etc/httpd/conf/httpd.conf. Elle contrôle les paramètres généraux du serveur Web, les hôtes virtuels et les accès. La configuration des différents modules est placée dans /etc/httpd/conf.d. Les modules sont présents dans /etc/httpd/modules/. Par défaut la racine du serveur, celle où sont placées les pages du site, est dans /var/www ou /srv/www. Cette position dépend de la directive **DocumentRoot** dans les fichiers de configuration.

4. Directives générales

Il n'est pas possible de lister toutes les directives du fichier httpd.conf mais quelques-unes sont importantes.

- **ServerRoot** : répertoire contenant les fichiers du serveur (configuration et modules). C'est généralement /etc/httpd.
- **Listen** : ports sur lesquels le serveur Apache écoute. Par défaut 80 (443 en https). On peut en spécifier plusieurs avec plusieurs directives Listen. Si le serveur dispose de plusieurs adresses IP, on peut rajouter l'IP au port associé : **Listen 192.168.1.3:80**.
- **User** : utilisateur des processus Apache. On n'utilise jamais root, mais un compte créé pour l'occasion, généralement Apache.
- **Group** : idem mais pour le groupe.
- **ServerAdmin** : adresse de courrier électronique de l'administrateur.

- **ServerName** : nom d'hôte (et port) du serveur. Il ne correspond pas forcément au nom d'hôte de la machine. Par contre il doit être valide. **ServerName www.mondomaine.org**

- **UseCanonicalName** : si elle vaut **on**, Apache va répondre en utilisant les informations de ServerName et Port, et pas les informations envoyées par le client. Par exemple, un http//192.168.1.3 se transforme en http://www.mondomaine.org.

- **UserDir** : nom d'un sous-répertoire ou chaque utilisateur peut placer ses fichiers HTML personnels. Généralement public_html. On y accède avec **http://www.mondomaine.org/~login/page.html**.

- **ErrorLog** : fichier où sont placées les logs d'erreur du serveur. **`/var/log/httpd/error_log`**.

- **CustomLog** : fichier journal de Apache. **`/var/log/httpd/access_log`**.

- **Timeout** : durée pendant laquelle le serveur attend des émissions/réceptions au cours d'une communication. Elle est réglée sur 300 secondes.

- **KeepAlive** : définit si le serveur peut exécuter plus d'une requête par connexion. C'est à off par défaut mais si on passe à on Apache peut générer rapidement des processus enfants pour le soulager s'il est très chargé.

- **MaxKeepAliveRequests** : nombre maximum de requêtes par connexion persistante. Une valeur élevée peut augmenter les performances du serveur. 100 par défaut.

- **KeepAliveTimeout** : durée pendant laquelle le serveur (généralement un processus fils) attend après avoir servi une requête. Par défaut 15 secondes. Après les 15 secondes, la demande sera réceptionnée par le serveur avec un Timeout.

- **StartServers** : nombre de serveurs créés au démarrage. Par défaut 8. Apache gérant ensuite dynamiquement le nombre de serveurs fils, ce paramètre a peu d'importance car le nombre va baisser ou augmenter rapidement.

5. Gestion des performances

- **MaxRequestPerChild** : nombre de requêtes pouvant être exécutées par un processus fils avant de s'arrêter. Par défaut 4000. Ça permet une occupation mémoire plus réduite en la libérant plus rapidement.

- **MaxClients** : limite du nombre total de requêtes pouvant être traitées simultanément. Par défaut 150. La valeur limite le risque de saturation du serveur.

- **MinSpareServers / MaxSpareServers** : suivant la charge de la machine, Apache peut lancer d'autres processus serveurs pour s'adapter à la charge actuelle. Par défaut, elles sont de 5 et 20. Ces deux valeurs déterminent les nombres limites autorisés. Si un serveur est peu chargé avec 15 processus, Apache en supprimera mais en gardera toujours au moins 5. Si le serveur devient chargé avec 10 processus, Apache en créera des supplémentaires à concurrence de 20 maximum.

- **MinSpareThreads / MaxSpareThreads** : chaque serveur fils peut accepter un certain nombre de requêtes simultanément. Pour cela il utilise les threads. Ces deux valeurs sont fixées par défaut à 20 et 75 et le mécanisme fonctionne comme ci-dessus.

- **ThreadsPerChild** : nombre de threads par défaut au lancement d'un serveur fils. Par défaut 25.

6. Les répertoires, alias et emplacements

a. Directory

Les balises **<Directory chemin>** et **</Directory>** permettent de regrouper des directives qui ne s'appliqueront

qu'au chemin (et à ses sous-répertoires) donnés. La directive **Options** est fortement conseillée.

```
<Directory /var/www/html/images>
    Options +Indexes +FollowSymLinks
    DirectoryIndex index.php index.html
    Order allow, deny
    Allow from All
</Directory>

<Directory /var/www/html/cgi-bin>
    Options +ExecCGI
</Directory>
```

La directive **Options** accepte les valeurs suivantes précédées de + ou - et séparées par des espaces :

- **All** : toutes les options sauf MultiViews ;
- **Indexes** : si jamais le répertoire ne contient pas de fichier HTML par défaut (cf DirectoryIndex), le contenu du répertoire est affiché sous forme de listing ;
- **ExecCGI** : l'exécution de scripts CGI est autorisée ;
- **FollowSymLinks** : le serveur suit les liens symboliques.

La directive **DirectoryIndex** précise les fichiers html ou cgi par défaut lors du chargement d'une URL.

```
DirectoryIndex index.php index.html
```

Au chargement, sans préciser le nom du fichier html, le serveur tentera de charger index.php et, s'il est absent, index.html. Dans le cas contraire, c'est l'option Indexes qui détermine si le contenu doit être visible sous forme de répertoire.

La directive **Allow** indique quels clients seront autorisés à accéder au répertoire. Ce peut être **`all`**, un domaine, une IP, une IP tronquée (sous-réseau), une paire réseau/sous-réseau, etc. La directive **`Deny`** interdit l'accès et s'utilise de la même manière. L'ordre est déterminé par la directive **`Order`**.

b. Alias

La directive **Alias** permet de créer un raccourci entre l'arborescence logique du site Web et un chemin du système de fichiers.

```
Alias /help "/usr/share/doc/html"
```

Dans ce cas, l'url http://www.monsite.org/help ne cherchera pas dans le répertoire /var/www/html/help mais dans /usr/share/doc/html.

Contrairement aux balises **<Directory>**, les balises **<Location>** et **</Location>** permettent d'appliquer des directives basées sur l'URL (et pas les répertoires).

```
<Location /help>
    Options +All -FollowSymLinks
    Order deny, allow
    Deny from all
    Allow from .mondomaine.org
</Location>
```

7. Hôtes virtuels

Un serveur Apache est capable de gérer plusieurs sites Web sur un même serveur. Il existe plusieurs méthodes. La première se base sur les noms (plusieurs sites Web pour un serveur) l'autre sur les adresses ip (une adresse IP pour chaque site Web). Vous allez aborder la première version.

La directive **NameVirtualHost** spécifie l'adresse IP sur laquelle le serveur va recevoir les requêtes d'accès aux hôtes virtuels.

Les balises **<VirtualHost>** et **</Virtualhost>** permettent de définir un hôte virtuel.

```
NameVirtualHost 192.168.1.3

<VirtualHost 192.168.1.3>
    ServerName          www2.mondomaine.org
    ServerAdmin         webmaster@www2.mondomaine.org
    DocumentRoot        /var/www/www2.mondomaine.org/
    ErrorLog        logs/www2_error_log
    CustomLog       logs/www2_access_log
</VirtualHost>
```

Relancez Apache. Avec un navigateur (ex : Firefox) on vérifie si notre hôte virtuel répond avec l'URL http://www2.mondomaine.org (cette adresse doit être déclarée dans /etc/hosts ou connue du serveur de noms et pointer sur le bon serveur).

Remarquez cependant que si vous passez par http://www.mondomaine.org vous n'obtenez plus le site par défaut ! En effet quand on déclare des hôtes virtuels, le premier de la liste devient l'hôte par défaut et prioritaire.

Quand on accède au serveur, Apache recherche d'abord une correspondance entre le nom d'hôte spécifié par l'URL et chaque ServerName des hôtes virtuels. Si aucune correspondance exacte n'est trouvée, c'est le premier hôte virtuel qui est choisi par défaut en faisant abstraction des paramètres globaux.

Rajoutez un hôte virtuel pour le site principal.

```
<VirtualHost 192.168.1.3>
    ServerName          www.mondomaine.org
    ServerAdmin         webmaster@www.mondomaine.org
    DocumentRoot        /var/www/html
    ErrorLog        logs/error_log
    CustomLog       logs/access_log
</VirtualHost>
```

Attention, la règle ci-dessus s'applique aussi avec un nom court. Si vous inscrivez http://www ou http://www2 vous tomberez toujours sur l'hôte virtuel par défaut. Il faut demander http://www.mondomaine.org et http://www2.mondomaine.org.

On peut placer dans un hôte virtuel toutes les directives souhaitées (ou presque).

Partage de fichiers

1. NFS

a. Lancement

Le partage de fichier **NFS** (*Network File System*) ou système de fichiers réseau permet de partager tout ou partie de son système de fichiers à destination de clients NFS, bien souvent d'autres Unix. Dans sa version de base c'est un système simple et efficace. Nous allons étudier la version 2.

NFS s'appuie sur le **portmapper** (**portmap**), le support **nfs** du noyau et les services **`rpc.nfsd`** et **`rpc.mountd`**.

Pour lancer le service NFS, portmap et nfs doivent être lancés (en vérifier le statut avant).

```
# service portmap status # /etc/init.d/portmap status ou rpcinfo -p
# service nfs status
# service portmap start
# service nfs start
# service nfslock start
```

Pour savoir si le service est disponible sur un hôte distant :

```
# rpcinfo -p hote
```

b. Partage côté serveur

La liste des systèmes de fichiers à exporter se trouve dans **`/etc/exports`**. Il contient un partage par ligne.

```
# Rep exportes  Autorisations d'acces
/               poste1(rw) poste2(rw,no_root_squash)
/projects       *.mondomaine.org(rw)
/home/joe       poste*.mondomaine.org(rw)
/pub            192.168.1.0/255.255.255.0(ro)
```

Chaque ligne est composée de deux parties. La première est le chemin du répertoire exporté. La seconde contient les autorisations d'accès.

L'autorisation d'accès est composée de paires hôtes/permissions selon le format suivant :

```
host(permissions)
```

Si l'hôte n'est pas défini, c'est tout le réseau (portée dite mondiale) qui sera concerné par les permissions. Si les permissions ne sont pas définies, l'export sera en lecture seule. Il ne faut surtout pas mettre d'espaces entre l'hôte et les permissions. L'hôte peut être :

- un nom d'hôte unique ;
- un domaine ;
- un réseau ou un sous-réseau ;
- une combinaison de l'ensemble avec des caractères de substitution (*, ?).

Les permissions peuvent être :

- **ro** : lecture seule ;
- **rw** : lecture écriture ;
- **no_root_squash** : le root distant équivaut au root local ;

- **root_squash** : si root se connecte au partage, son uid sera remplacé par celui d'un utilisateur anonyme. Ainsi il n'y a pas de risques que l'utilisateur root d'un poste local puisse être root sur un partage distant ;
- **all_squash** : étend la règle précédente à tous les utilisateurs ;
- **anonuid / anongid** : uid et gid pour l'utilisateur anonyme.

Pour une gestion correcte des droits et des permissions, **les utilisateurs de même nom (login) doivent avoir les mêmes UID et GID sur le serveur et le client**. NFS se base en effet sur ces valeurs pour la sécurité des données du partage. Le nom de login seul ne suffit pas. Dans l'exemple ci-dessus, l'utilisateur **joe** est autorisé à accéder au partage **/home/joe** (on suppose que c'est son répertoire personnel) sur tous les postes du domaine. L'utilisateur joe doit être déclaré de la même manière (même UID) sur le serveur et sur tous les postes. C'est pour cela que l'on utilise souvent NIS avec NFS.

La commande **exportfs** permet de contrôler les partages.

- **`exportfs -r`** : rafraîchit la liste des partages après modification de /etc/exports ;
- **`exportfs -v`** : liste des partages ;
- **`exportfs -a`** : exporte (ou recharge) tous les partages de /etc/exports ou un partage donné ;
- **`exportfs -u`** : stoppe le partage donné. `-a` pour tous.

La commande **`showmount`** montre les partages d'un hôte donné.

```
showmount -e host
```

c. Montage côté client

Le support NFS est inclus sous forme de module du noyau. Il est automatiquement chargé à l'utilisation d'un accès NFS.

Dans **`/etc/fstab`** notez les modifications :

```
server1:/pub /mnt/pub nfs defaults 0 0
```

Le périphérique est remplacé par le chemin du partage sous la forme **serveur:chemin**. Le système de fichiers est **nfs**. C'est identique avec la commande **mount** :

```
mount -t nfs serveur1:/pub /mnt/pub
```


Si les montages NFS sont définis dans /etc/fstab, mount `-a` ne va pas les monter. Un service est généralement présent dans chaque distribution pour les monter et les démonter aux arrêts relances. Sur Red Hat, le service /etc/rc.d/init.d/netfs les montera automatiquement au démarrage.

NFS dispose d'options de montage spécifiques :

- **nolock** : option de compatibilité avec d'anciens serveurs NFS ;
- **intr** : interrompt une requête NFS si le serveur ne répond pas ;
- **hard** : bloque les processus qui tentent d'accéder à un partage inaccessible ;
- **soft** : un processus retournera une erreur en cas d'accès infructueux ;
- **rsize=8192, wsize=8192** : taille des blocs de lecture/écriture sur le serveur. Une écriture de 8 ko est plus rapide que 8 écritures de 1 ko.

FTP

Le serveur **FTP** (*File Transfer Protocol*) le plus courant est **vsftpd** (*Very Secure FTP Daemon*). Il a l'avantage d'être très petit, performant et rapide tout en étant tout de même très configurable (moins toutefois que Proftpd ou d'autres). Il convient dans la quasi-totalité des situations. C'est un service qui peut aussi bien être lancé par **xinetd** que (dans les dernières versions des distributions) en tant que service seul.

Deux niveaux de sécurité sont utilisables :

- **Anonyme** : tout le monde peut se connecter au serveur FTP en tant que utilisateur **ftp** ou **anonymous**. l'environnement FTP est chrooté.

- **Utilisateur** : les utilisateurs qui existent sur le serveur peuvent se connecter avec leur mot de passe et ont un accès complet à leurs données dans leur répertoire personnel.

Les utilisateurs anonymes étant considérés comme l'utilisateur ftp, c'est le répertoire personnel de ce compte qui est la racine du ftp.

Le fichier de configuration est présent dans /etc/vsftpd/vsftpd.conf.

La racine du ftp par défaut est dans /var/ftp.

Le script de lancement est /etc/init.d/vsftpd (service vsftpd start).

Pour activer ou non l'accès anonyme on modifie le fichier de configuration. Dans ce cas, l'utilisateur peut se connecter en tant que anonymous ou ftp. Dans tous les cas, il sera reconnu comme utilisateur « ftp » du serveur une fois connecté :

```
anonymous_enable=YES/NO
```

Pour activer ou non l'envoi de fichiers sur le serveur par des anonymes. Dans ce cas, l'autorisation d'écriture dans un répertoire est fonction des droits du répertoire sur le serveur (notamment si l'utilisateur ftp a le droit d'écrire ou non dans un répertoire) :

```
anon_upload_enable=YES/NO
```

Vous pouvez interdire à des utilisateurs de se connecter en plaçant leur noms dans /etc/vsftpd.ftpusers.

Vous pouvez ajouter des utilisateurs dans /etc/vsftpd.user_list si **`userlist_enable=YES`**. Dans ce cas, c'est la valeur de **`userlist_deny (YES/NO)`** qui déterminera si le fichier contient les utilisateurs interdits ou autorisés.

On peut créer dans chaque répertoire du serveur un fichier **`.message`**. Dans ce cas, son contenu sera affiché lors de l'accès au répertoire.

Partages Windows avec Samba

1. Présentation

Samba est un ensemble de serveurs implémentant les protocoles SMB/CIFS et NetBIOS/WINS pour Unix. Son utilisation la plus connue est le partage de ressources entre Windows et Unix, mais il fonctionne parfaitement bien entre deux Unix. Un **partage** est aussi appelé **service**. Samba est composé de deux services :

- **smbd** : serveur SMB/CIFS.
    - Authentification et autorisation.
    - Partages de fichiers et d'imprimantes.
- **nmbd** : serveur de noms NetBIOS.
    - Parcours des ressources.
    - Serveur WINS.

Un troisième service, **winbindd**, permet d'utiliser les comptes utilisateur d'un domaine Microsoft. Les dernières versions de Samba (3 et suivantes) permettent aussi de se raccorder à Active Directory.

Les fonctions principales de Samba sont :

- Authentification des utilisateurs.
- Partage de fichiers et d'imprimantes.
- Parcours des ressources partagées du réseau.
- Résolution de noms (indépendante de DNS) Nom Netbios IP ou vice versa.

2. Configuration

La configuration de Samba se trouve dans /etc/samba/smb.conf. Vous pouvez tester sa syntaxe avec l'outil **testparm**.

Le fichier smb.conf reprend la syntaxe des fichiers de configuration de Windows de type **ini** avec des sections délimitées par des crochets **[ ]**.

Par défaut trois sections sont présentes :

- **[global]** : réglages génériques et globaux du serveur, nom, commentaires, méthode d'authentification, réglages par défaut, etc.
- **[homes]** : partage des répertoires personnels des utilisateurs.
- **[printers]** : partage des imprimantes.

Les paramètres sont de la forme :

```
nom = valeur
```

Les commentaires commencent par un point-virgule **;** ou un dièse **#**.

```
[global]
workgroup = MYGROUP
```

```
netbios name = posteN
security = share
```

- **workgroup** : nom de groupe / domaine de travail ;
- **netbios name** : nom netbios de la machine ;
- **security** : méthode d'authentification (cf. plus bas).

3. Partage de fichiers

Chaque partage doit disposer de sa propre section dans smb.conf. Voici comment partager le répertoire **/opt/partage1** sous le nom de service **partage1** :

```
[partage1]
comment = Répertoire partagé 1
path = /opt/partage1
browseable = yes
public = no
writable = yes
printable = no
group = partage
```

Voici une description des valeurs et de quelques autres possibles.

- **partage1** : le nom du partage tel qu'il apparaît dans le « voisinage ».
- **comment** : le commentaire tel qu'il apparaît à côté du nom de partage.
- **path** : le chemin du partage.
- **public** : le partage est accessible à l'utilisateur par défaut (guest).
- **browseable** : le partage apparaît dans le « voisinage ».
- **writable** : le partage est accessible en lecture et écriture.
- **printable** : le partage est une imprimante.
- **group** : nom du groupe par défaut pour la connexion.
- **valid users** : nom des utilisateurs autorisés à accéder à ce partage.
- **read only** : le partage est en lecture seule pour tout le monde.
- **guest ok** : aucun mot de passe n'est nécessaire pour accéder au partage. Dans ce cas le compte invité par défaut sera utilisé.
- **guest only** : le partage est accessible uniquement aux invités.

L'accès aux partages est autorisé par défaut (voir partie sur les méthodes d'authentification) en fonction des droits Unix. À l'accès au partage, un nom d'utilisateur et un mot de passe sont demandés. Les droits du répertoire partagé et de son contenu déterminent les droits de l'utilisateur.

4. Partage des imprimantes

En plus de la section **[printers]**, vous pouvez ajouter des sections pour des imprimantes spécifiques. Le paramètre

printing de la section **[global]** permet de modifier le sous-système d'impression basé par défaut sur CUPS.

```
[Bureau150]
comment = Laserjet 2100
printer = lj2100
valid users = riri fifi loulou
path = /var/spool/lj2100
public = no
writable = no
printable = yes
browseable = yes
```

- **printer** : nom de l'imprimante sous Linux.
- **valid users** : nom des utilisateurs autorisés à accéder à ce partage. Un @ devant le nom indique un groupe d'utilisateurs.
- **path** : chemin du spool d'impression.

5. Méthodes d'authentification

Samba propose plusieurs méthodes d'authentification définies dans la section **[global]** :

- **user** : méthode par défaut ; l'accès à l'ensemble des partages d'un serveur se fait par la validation d'un nom d'utilisateur et d'un mot de passe uniques.
- **share** : méthode de validation des identifiants partage par partage. Dans ce cas, tous les accès aux partages, même publics, nécessitent des identifiants.
- **domain** : utilisation d'un groupe de travail avec authentification.
- **ads** : utilisation de Active Directory.

D'autres types d'authentification sont possibles comme un annuaire LDAP.

```
[global]
        workgroup = MYGROUP
        netbios name = posteN
        security = share
```

6. Correspondance des noms et des mots de passe

Les mots de passe du protocole SMB n'ont pas la même forme que les mots de passe Unix/Linux. Il faut recréer les mots de passe pour chaque utilisateur devant utiliser SMB avec la commande **smbpasswd**. Les utilisateurs doivent déjà exister sous Unix.

```
smbpasswd -a toto
```

Les utilisateurs SMB sont présents dans **/etc/samba/smbpasswd**. La commande **mksmbpasswd** peut réaliser cela en batch :

```
cat /etc/passwd | mksmbpasswd > /etc/samba/smbpasswd
```

Vous pouvez établir une table de correspondance entre les noms d'utilisateurs Windows et ceux de Unix dans **/etc/samba/smbusers**.

```
# Unix_name = SMB_name1 SMB_name2 ...
root = administrator admin administrateur
```

7. Clients SAMBA

a. En ligne

Toute machine sous Microsoft Windows peut accéder aux partages Samba. Les navigateurs des environnements de bureau **KDE** et **GNOME** acceptent la navigation dans les partages grâce au protocole **smb:/** dans les URL. KDE propose même l'équivalent d'un voisinage réseau.

L'outil **smbclient** est une sorte de client FTP pour le protocole SMB. Les chemins d'accès aux ressources sont de la forme :

```
//machine/partage
```

Par exemple pour se connecter au service (ici un partage) d'une machine :

```
smbclient //machine/service -U login%passwd
```

Pour lister les services proposés par une machine:

```
smbclient -L hostname -U login%passwd # liste des ressources
```

b. Montage

Montez un système de fichiers SMB avec la commande **smbmount**.

```
smbmount //machine/partage /mnt/mountpoint -o username=login
```

On peut aussi réaliser le montage dans /etc/fstab. Tout comme avec nfs, c'est un service spécialisé qui montera et démontera les partages. Les derniers noyaux Linux ont remplacé **smbfs** par **cifs** :

```
//machine/partage /mnt/mountpoint cifs defaults,username=nobody 0 0
```

Bases de sécurité

1. Sécurité informatique

Les objectifs principaux de la sécurité informatique concernent :

- **La sécurité de la connexion** : il s'agit de contrôler que les utilisateurs qui se connectent sont bien autorisés à le faire et de leur interdire l'accès au système dans le cas contraire.
- **L'intégrité des données** : il s'agit de faire en sorte que les fichiers et les bases de données ne soient pas corrompus et de maintenir la cohérence entre les données.
- **La confidentialité des données** : l'accès aux données en consultation et en modification doit être limité aux seuls utilisateurs autorisés.

Vous disposez de plusieurs moyens :

- L'authentification des utilisateurs par un mot de passe.
- Le cryptage des données.
- La sécurité physique en contrôlant l'accès des personnes aux salles informatiques, en utilisant des circuits inviolables matériellement.
- L'information sur les risques pénaux encourus en cas d'infraction. Un « braquage » informatique est un délit, pas un jeu.
- Le contrôle fréquent des droits d'accès aux fichiers et aux bases de données.
- Le contrôle des « checksum » des fichiers pour s'assurer de leur intégrité.
- La sauvegarde régulière des données.
- L'audit des principaux événements du système.
- L'installation de murs de feu « firewall » qui contrôlent les accès au système informatique depuis l'extérieur et limitent l'accès à des services externes par des utilisateurs non avertis ou qui n'en ont pas besoin pour, par exemple, limiter le risque de rapatriement de virus.
- L'installation d'un antivirus, même sous Linux, si le serveur traite des données depuis et vers des systèmes d'exploitation concernés par les virus.
- L'installation d'outils anti-spams et anti-spywares, selon le même principe, afin d'éviter une intrusion et la saturation des serveurs de courrier électronique.
- Le démarrage uniquement des services réellement utiles sur le serveur et sur le client.

Quelques méthodes simples permettent de limiter les risques :

- Vous pouvez définir une valeur de umask restrictive (ex : 077) quitte à étendre ensuite les droits d'accès de certains fichiers.
- Il ne faut pas quitter son terminal sans se déconnecter ou le verrouiller.
- Il faut prêter attention aux dates de dernière connexion réussies et infructueuses qui sont affichées à chaque connexion.

- Ne jamais autoriser l'accès, même en lecture, au fichier .profile.
- Ne jamais mettre le . en première position du PATH, et contrôler ses chemins.

2. Contrôler les droits d'endossement

Les droits d'endossement (bits SUID et SGID) sont souvent une cause d'insécurité du système. En effet un utilisateur mal intentionné profitant de l'inattention ou de l'absence d'un collègue ou d'un administrateur n'étant pas déconnecté de sa console peut modifier les droits de certaines commandes à son avantage. L'exemple le plus courant est de recopier un shell en tant que programme peu utilisé (par exemple sx) et de lui donner les droits SUID. En lançant cette commande vous pouvez devenir root.

Obtenir le droit de lister tous les fichiers

```
# chmod u+s cat
```

Obtenir un shell root

```
# cp /bin/sh /bin/sx
# chmod u+s /bin/sx
...
$ sx
# ...
```

La commande suivante permet de rechercher tous les fichiers disposant des bits SUID et/ou SGID :

```
# find / -type f \( -perm -4000 -o -perm -2000 \)
# find / -type f \( -perm -4000 -o -perm -2000 \)
/bin/su
/bin/umount
/bin/eject
/bin/mount
/bin/ping
/bin/ping6
/sbin/unix2_chkpwd
/sbin/unix_chkpwd
/usr/bin/expiry
/usr/bin/write
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gnotski
/usr/bin/mahjongg
/usr/bin/chfn
/usr/bin/yset
/usr/bin/wall
/usr/bin/crontab
/usr/bin/v4l-conf
/usr/bin/gnomine
/usr/bin/same-gnome
/usr/bin/gnotravex
/usr/bin/gnobots2
...
```

Dans la liste précédente il y a un intrus : **/usr/bin/yset** qui permet de modifier les réglages d'un serveur de son et qui n'a nul besoin d'avoir le droit SUID. Il y a un problème.

```
# ls -l /usr/bin/yset
-rwsr-sr-x 1 root root 604040 mai 19 21:28 /usr/bin/yset
# md5sum /usr/bin/yset
04ff72010ff1cf1c14d7706159cdf8bf  /usr/bin/yset
# ls -l /bin/bash
-rwxr-xr-x 1 root root 604040 sep 22  2007 /bin/bash
# md5sum /bin/bash
04ff72010ff1cf1c14d7706159cdf8bf  /bin/bash
```

Quelqu'un a recopié un shell avec un autre nom.

3. Vérifier les packages

Le chapitre Installation de Linux et de logiciels a abordé toute la gestion des packages logiciels. Parmi les diverses options, certaines permettent de contrôler la validité d'un package. La base RPM contient, outre le nom du fichier, son type (configuration, binaire, etc.) et dans certains cas (binaire) la somme de contrôle (checksum) MD5 du fichier.

Suivant l'exemple précédent, comment restaurer le fichier yset ? En trois étapes :

- Trouvez le package d'origine :

```
# rpm -qf /usr/bin/yset
yiff-2.14.5-0.pm.1
```

- Contrôlez l'état du package installé :

```
# rpm -V yiff
SM5....T    /usr/bin/yset
```

  - S : la taille n'est pas la bonne
  - M : les permissions ont été modifiées
  - 5 : la somme de contrôle MD5 diffère
  - T : la date de modification n'est pas correcte.

- Réinstallez le package d'origine, selon les modalités propres à votre distribution.

4. Politique de mot de passe

Les mots de passe sont à la base de l'authentification d'un utilisateur. Ils doivent être sûrs. C'est pourtant généralement une grosse lacune, tant au travail qu'à la maison, et même sur Internet :

- mot de passe présent sur un post-it ;
- emploi d'un gestionnaire de mot de passe automatique lui-même sans mot de passe ;
- même mot de passe pour tous les sites Web et logiciels ;
- mot de passe jamais changé ;
- mot de passe et/ou compte communs à toute la famille/service ;
- mot de passe pas assez complexe ;
- etc.

Il ne sert à rien de sombrer dans la paranoïa. Vous devez trouver un compromis. Si vous demandez aux utilisateurs de modifier leur mot de passe trop souvent, ou s'il est trop difficile, ils auront tendance à le noter. S'il est trop simple et que vous attendez trop longtemps, il n'est plus sûr.

Les utilisateurs doivent choisir un bon mot de passe, en évitant la simplicité ou plutôt les évidences : noms des enfants, de sa femme, de lieux, dates de naissance et généralement tout ce qui tient à cœur et qui est connu du milieu professionnel ou personnel.

Un compromis peut être de modifier les règles de changement de mot de passe avec chage (ou passwd) de manière à placer une durée de validité des mots de passe de 40 jours. Les commandes de modification de la politique de gestion des mots de passe ont été présentées au chapitre Tâches administratives - Administration des utilisateurs.

```
# chage -l bean
Minimum:        7
Maximum:        40
Warning:        10
Inactive:       5
Last Change:            avr 10, 2008
Password Expires:       mai 20, 2008
Password Inactive:      mai 25, 2008
Account Expires:        jan 01, 2010
```

Les modules PAM influent sur la politique de gestion des mots de passe, forçant éventuellement à en choisir un plus ou moins complexe. De manière surprenante, un mot de passe doit être assez facile à retenir par un utilisateur, ce qui n'implique pas forcément qu'il soit facile à pirater (par John the Riper par exemple). Il existe des mots de passe qui peuvent se retenir par des moyens mnémotechniques. Vous pouvez aussi générer des mots de passe automatiquement avec l'outil **pwgen**.

```
$ pwgen
uash6She lohJo7ae Ohphab3i ouRik9ie uM4va3im Neer7Eit eib3Hauy xo9Iuy5p
ahSiW0uf AhG6wail Yai6neeh phae4ioV deeL3aip Uz5ahzaa aiV5phee Aegaiy7x
ioPh1ahn Ong6Baib Eish4rip eik9Gie1 ien3Iepe xohduj7U aiP2keov So5ovaht
Voh9oxoe ahs2Meeg Ooch5xix Phe3yiuz eeCa5ohv aig9Ai3o Go4Ateeh Hee6thei
Rai6Daeh aid8ieNg Thah6ien daphaiG0 Iefai5oh Pheife6i Poora8ah Coh5Aida
ViC7ieth hohG5sei Aa9Jeilu eopoX8Si jooh3Eif dooPhai1 chohqu1G ieNgae3o
wiCeiSi3 aej6Piev eoTha1Fu ieR2yeeb Eireili6 saiGhie2 XohRoo1a cahb2Yah
Guungah0 ube3vo0D oshol3Op Pui6agh5 Ao7baeN1 foTek9Ei aeM3lala Ene2baol
geloV9ai Weeyu2ie Uvae2Vie dei0euL7 Xee9uaza ed8Eeghu eebiu2Ka zey0Liuh
be6Ailoi eiph8Ohb Yahpahr4 aij4dahG oQu2chae Fe5eeg9c Hoosh6oh Iip8eiwe
AuPie0um Ahxai9eo Dae5oquu Ie7Viek8 pa2aew8B fohham7A fah1Oogi ieH9vee8
saeC8sha Aejeey6i Eithoow1 yi9vei0L ohC7eegh IaTh4ohn ti6Fooxe Oiche7oh
Tah9uos7 Paej2Iec chuiD8ei aicoGh5l saiKeiw2 mae9mieY Ais9oanu Mah9xej3
Zi2nacai gaiM4thi sapa1Fah kie8oZo7 Po5uuho8 thae3Aim Ohjahgh9 Weike8ra
Cah4weiZ teoji0Oo vi0hei6O Zieha3ai Keip2bie bahR7bah ahSai0Ei afoh3Thi
eeNieTh8 Zei7eth8 uV5eichi kue1Eedi sueThe0V wohChe2u Ohl1zicu Loo1soo3
yahb9uSi EelieGh1 aeMiThi1 OoFoh8wu Ieyei5ka Roph7ape uem5quuK ahQu7eec
NahSha6A kooMou0y gu1chaiJ hae2ku0Z uC3oeNgo xuSha7qu Iucai0fu uK4icewe
eP7aetig ahYai0ee Eetahfu8 yeep0oPi Veimaij3 Oht0aiPh buTh9oob ood4nieC
sah7Ahj1 koozah0J Vieb9Bit eeP9nee1 ea1SohCe Afei4ohS eikahk0W rachoG4c
```

Ces mots de passe sont pseudo-aléatoires. Si vous parlez l'anglais (et que vous parlez geek/leet), ils s'écrivent comme ils se prononcent. Par exemple :

`dooPhai1 : Do you fail ?`

Vous pouvez demander à générer des mots de passe totalement aléatoires d'une longueur donnée, ici de 10 caractères :

```
$ pwgen -s -1 10
ER9BAgHsZH
```

5. Interdire les connexions

a. /bin/false

Certains comptes ne doivent pas être interactifs : les connexions depuis une console devraient leur être interdites. Ces comptes peuvent être dédiés à une application, à un service, à une connexion FTP, etc., mais la connexion devrait être refusée : pas de shell !

Dans la liste des shells autorisés, un attire l'attention :

```
$ cat /etc/shells
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
```

```
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
/bin/zsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/passwd
/usr/bin/bash
/usr/bin/tcsh
/usr/bin/zsh
```

Le shell **/bin/false** n'en est pas vraiment un. Il interdit les connexions interactives. Vous avez déjà rencontré la commande **false** dans le chapitre Le shell et les commandes GNU. Elle retourne toujours faux. Dès que **login** tente d'exécuter le shell de connexion l'utilisateur est refoulé.

```
$ cat /etc/passwd|grep false
avahi:x:104:106:User for Avahi:/var/run/avahi-daemon:/bin/false
haldaemon:x:101:102:User for haldaemon:/var/run/hal:/bin/false
icecream:x:102:103:Icecream Daemon:/var/cache/icecream:/bin/false
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
messagebus:x:100:101:User for D-Bus:/var/run/dbus:/bin/false
ntp:x:74:104:NTP daemon:/var/lib/ntp:/bin/false
polkituser:x:103:105:PolicyKit:/var/run/PolicyKit:/bin/false
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
vscan:x:65:110:Vscan account:/var/spool/amavis:/bin/false
```

b. /etc/nologin

Avant même de passer par un pseudo-shell de connexion, les modules PAM autorisent de nombreuses limitations. Parmi eux le module « **pam_nologin** » qui vous permet d'interdire la connexion des utilisateurs sauf root. Si un utilisateur tente de se connecter le contenu du fichier **/etc/nologin** est affiché. C'est utile en cas de maintenance : seul root peut alors se connecter.

```
$ pwd
/etc/pam.d
$ grep nologin *
login:auth       requisite      pam_nologin.so
ppp:auth     required       pam_nologin.so
sshd:auth        requisite      pam_nologin.so
```

Pensez aussi que vous pouvez interdire l'accès d'un compte donné via le module PAM « **pam_listfile** ».

c. /etc/securetty

Dans le même genre, le fichier **/etc/securetty** contient la liste des terminaux considérés comme sûrs. Pour le service donné, la connexion sera interdite si le terminal de la personne tentant de s'y connecter n'apparaît pas. Dans l'exemple suivant les logins (via la commande login) sont autorisés uniquement depuis les pseudo-terminaux locaux, c'est-à-dire seulement depuis les consoles directement accessibles sur l'ordinateur via les combinaisons [Alt][F1] à [Alt][F7].

```
$ grep securetty *
login:auth       [user_unknown=ignore success=ok ignore=ignore
auth_err=die default=bad]        pam_securetty.so
$ cat /etc/securetty
#
# This file contains the device names of tty lines (one per line,
# without leading /dev/) on which root is allowed to login.
#
tty1
tty2
tty3
tty4
tty5
tty6
```

Ne confondez pas securetty (Secure tty) avec security !

6. Tester les mots de passe

Les outils crack et « John the ripper » tentent de décrypter vos mots de passe, tant depuis un dictionnaire que par la force brute (les uns après les autres). Dans le pire des cas, ils les trouvent en quelques secondes ; dans le meilleur, en plusieurs jours, voire semaines. Dans ce cas, le mot de passe peut être considéré comme sûr.

"John the ripper" s'utilise très simplement. La commande est **john**. Pour tester l'intégralité de votre fichier **/etc/shadow** :

```
# john /etc/shadow
Loaded 5 password hashes with 5 different salts (OpenBSD Blowfish
[32/64])
```

Dans le mode par défaut, john :

- tente une détection simple via des combinaisons courantes liées au compte,
- passe au mode dictionnaire avec application de règles,
- puis tente une recherche incrémentale.

Une recherche peut prendre de quelques secondes à quelques semaines !

Pour tester un seul utilisateur :

```
# john -user:seb /etc/shadow
```

Pour tester les utilisateurs avec un dictionnaire :

```
# john -users:seb -wordlist:/var/lib/john/wordlists/all /etc/shadow
```

La même chose mais en testant plusieurs règles par mot (inversion, majuscules, minuscules, etc.) :

```
# john -users:seb -wordlist:/var/lib/john/wordlists/all -rules
/etc/shadow
```

Pour reprendre une recherche interrompue ([Ctrl][C]) :

```
# john -restore
```

John place ses résultats dans le répertoire ~/.john, généralement chez root qui seul, en principe, devrait pouvoir lancer cet outil :

```
# ls -l .john
total 4
-rw------- 1 root root 70 mai 23 21:58 john.pot
-rw------- 1 root root 124 mai 23 21:54 john.rec
```

Le fichier **john.pot** contient les résultats trouvés par John. Ici le fichier n'est pas vide. C'est problématique : john a trouvé un mot de passe. Le fichier **john.rec** contient l'état actuel de la recherche, utilisé en interne et en cas de reprise après interruption.

Sur une machine basée sur un Intel Core 2 duo 64 bits à 3.4 Ghz, le mot de passe de seb (celui de l'auteur) a été cracké en 4 minutes et 22 secondes. Il était (volontairement cette fois) basé sur un mot du dictionnaire.

Si vous appuyez sur une touche durant la recherche, john affiche son état.

Soit un compte henri dont le propriétaire a pour mot de passe le même mot que son nom de compte, à savoir « henri » :

```
# john -users:henri /etc/shadow
Created directory: /root/.john
```

```
Loaded 1 password hash (OpenBSD Blowfish [32/64])
henri            (henri)
guesses: 1  time: 0:00:00:00 100% (1)  c/s: 4.34  trying: henri
```

Voici de quoi faire grandement réfléchir ceux qui pensent que personne ne viendra les déranger, et les encourager à modifier leur mot de passe avec des règles précises (majuscules, minuscules, chiffres, etc.).

7. Rechercher des rootkits

a. Principe du rootkit

Une fois qu'un pirate quelconque aura réussi, via une faille ou un mot de passe trop simple, à pénétrer votre machine, il cherchera probablement à s'aménager une porte d'entrée plus importante, ou plutôt une porte de derrière, une **backdoor**, afin de pouvoir revenir se servir de votre machine à des fins douteuses ou y puiser ou stocker des données (certains PCs se sont retrouvés comme cela à contenir des milliers de fichiers mp3 ou divx et à servir de serveur pour des personnes peu scrupuleuses).

L'idéal pour le pirate est de pouvoir s'accaparer les droits de root. C'est son but : devenir intégralement le maître de votre machine. Pour cela il n'a pas forcément besoin de « taper » directement sur ce compte. Sous Linux, il est courant (et voir conseillé) de se connecter en simple utilisateur puis de passer root le temps d'effectuer les manipulations nécessaires. Or pour passer root, vous utilisez généralement **su**.

Si, via le service ftp mal configuré, ou via ssh (et scp ensuite), le pirate tente de se connecter à un compte dont le mot de passe est évident (exemple du compte henri), qu'il y place un script appelé su de son cru et qu'il modifie le PATH par défaut pour y placer son chemin en premier, alors c'est gagné :

```
$ pwd
/home/seb
$ cat su
#!/bin/bash
echo -e "Mot de passe :\c"
read -s password
echo "$@ $password" > /tmp/fic
echo
echo "su: Mot de passe incorrect."
/bin/su $@
$ chmod +x su
$ export PATH=$HOME:$PATH
$ su - root
Mot de passe : ===> FAUX SU
su: Mot de passe incorrect. ===> FAUX SU
Mot de passe : ===> VRAI SU
```

Il n'y a plus qu'à afficher le contenu du fichier pour obtenir le mot de passe :

```
$ cat /tmp/fic
- root azerty
```

La méthode, simpliste mais efficace, est loin d'être imparable : su ne demande le mot de passe qu'une seule fois, et sauf faute d'inattention le subterfuge est ici volontairement visible. Le fait de placer des scripts, de modifier l'environnement, de remplacer un fichier par un autre de manière à obtenir un accès privilégié à une machine s'appelle installer un **rootkit**. Une fois celui-ci en place, il garantit, tant qu'il n'a pas été détecté, un accès root à la machine.

b. chkrootkit

L'outil **chkrootkit** est un outil simple permettant de rechercher la présence des rootkits les plus connus et les plus courants. Il est efficace seulement s'il est mis à jour régulièrement, lancé régulièrement, et ne se substitue pas aux contrôles déjà cités précédemment.

```
# chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
```

```
Checking `cron'... not infected
Checking `crontab'... not infected
...
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs...
nothing found
Searching for suspicious files and dirs, it may take a while...
...
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmind/IIS Worm... nothing found
...
Searching for ENYELKM rootkit default files... nothing found
Searching for anomalies in shell history files... Warning: `' is
linked to another file
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... chkproc: nothing detected
Checking `rexedcs'... not found
Checking `sniffer'... eth0: not promisc and no PF_PACKET sockets
vmnet8: not promisc and no PF_PACKET sockets
vmnet1: not promisc and no PF_PACKET sockets
Checking `w55808'... not infected
...
```

8. Les virus

Les premiers virus sont apparus sous Unix. Si le système est généralement sécurisé, et que les virus sur les plates-formes Unix et Linux (dont MacOS X) sont presque inexistants, dans le cas plus ou moins probable où un virus serait présent et risque de compromettre la sécurité de votre machine, celle des autres ou de tout un réseau, il est de votre devoir de l'éradiquer.

Si votre machine sert de serveur, notamment de courrier électronique ou de fichiers sur un réseau contenant des machines sous Windows fortement exposées, vous ne devez pas servir de vecteur indirect de propagation. Vous devez éliminer la menace.

Il existe plusieurs antivirus sous Linux, certains commerciaux gratuits ou non, certains libres. L'antivirus Clam (ClamAV) est libre et gratuit. Il est disponible à l'adresse http://www.clamav.net/. Ses bases de signatures sont mises à jour quotidiennement.

Freshclam permet de mettre à jour les bases placées dans **`/var/lib/clamav`** :

```
# pwd
/var/lib/clamav
# freshclam
ClamAV update process started at Sun May 25 16:29:37 2008
connect_error: getsockopt(SO_ERROR): fd=4 error=113: No route to host
Can't connect to port 80 of host database.clamav.net (IP: 194.116.142.73)
Trying host database.clamav.net (213.251.187.177)...
Downloading main.cvd [100%]
main.cvd updated (version: 46, sigs: 231834, f-level: 26, builder: sven)
Downloading daily.cvd [100%]
daily.cvd updated (version: 7233, sigs: 69750, f-level: 26, builder: ccordes)
Database updated (301584 signatures) from database.clamav.net
(IP: 213.251.187.177)
# ll
total 14628
```

```
-rw-r--r-- 1 vscan vscan  1894380 mai 25 16:29 daily.cvd
-rw-r--r-- 1 vscan vscan 13050207 mai 25 16:29 main.cvd
-rw------- 1 vscan vscan       52 mai 25 16:29 mirrors.dat
```

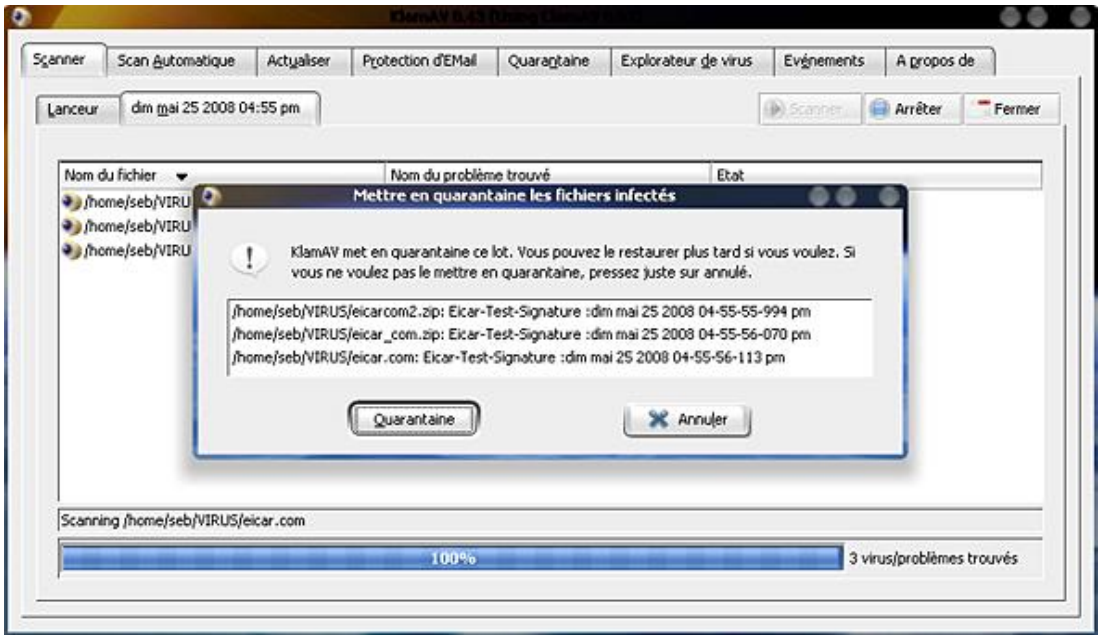
Scanclam permet de rechercher les éventuels virus :

```
# clamscan -v -r /usr/local/bin
Scanning /usr/local/bin/avidemux2_qt4
/usr/local/bin/avidemux2_qt4: OK
Scanning /usr/local/bin/avidemux2_gtk
/usr/local/bin/avidemux2_gtk: OK
Scanning /usr/local/bin/avidemux2_cli
/usr/local/bin/avidemux2_cli: OK
Scanning /usr/local/bin/i18n/qt_it.qm
/usr/local/bin/i18n/qt_it.qm: OK
Scanning /usr/local/bin/i18n/avidemux_it.qm
/usr/local/bin/i18n/avidemux_it.qm: OK
----------- SCAN SUMMARY -----------
Known viruses: 300812
Engine version: 0.93
Scanned directories: 2
Scanned files: 5
Infected files: 0
Data scanned: 45.64 MB
Time: 3.362 sec (0 m 3 s)
```

Voici le même test effectué avec un faux virus sous trois formes : binaire, compressé gzip et compressé bzip2. Si un virus est détecté le fichier correspondant est déplacé dans **/home/seb/VIRUS** :

```
$ clamscan -v -r --move=/home/seb/VIRUS /home/seb/bin
Scanning /home/seb/bin/eicarcom2.zip
/home/seb/bin/eicarcom2.zip: Eicar-Test-Signature FOUND
/home/seb/bin/eicarcom2.zip: moved to '/home/seb/VIRUS//eicarcom2.zip'
Scanning /home/seb/bin/eicar_com.zip
/home/seb/bin/eicar_com.zip: Eicar-Test-Signature FOUND
/home/seb/bin/eicar_com.zip: moved to '/home/seb/VIRUS//eicar_com.zip'
Scanning /home/seb/bin/eicar.com
/home/seb/bin/eicar.com: Eicar-Test-Signature FOUND
/home/seb/bin/eicar.com: moved to '/home/seb/VIRUS//eicar.com'

----------- SCAN SUMMARY -----------
Known viruses: 300812
Engine version: 0.93
Scanned directories: 1
Scanned files: 3
Infected files: 3
Data scanned: 0.00 MB
Time: 1.216 sec (0 m 1 s)
```

Clamav peut être lancé en tant que service. Dans ce cas, il peut accepter des demandes de recherche de virus dans des arborescences précisées ce qui permet à des outils comme klamav de pouvoir exister : un frontend à clamav sous KDE.



Klamav a bien détecté le virus de test.

9. Les limites de l'utilisateur

Le champ d'action des PAM est plus vaste que la simple connexion puisqu'il gère aussi l'environnement de l'utilisateur. Avant même de voir le module concerné, la commande **ulimit** permet d'agir sur l'environnement du shell et des processus qu'il contrôle. Le paramètre `-a` affiche toutes les options contrôlées par **ulimit** :

```
$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 16380
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) 1753125
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 16380
virtual memory          (kbytes, -v) 3333600
file locks                      (-x) unlimited
```

Les quelques lignes en gras méritent votre attention.

- **max memory size** : la taille mémoire max que l'utilisateur peut occuper ;
- **open files** : le nombre maximum de descripteurs de fichiers, donc le nombre maximum de fichiers pouvant être ouverts ;
- **max user processes** : le nombre maximum de processus qu'un utilisateur peut lancer.

Ces valeurs peuvent être changées selon certaines limites imposées par l'administrateur. Il existe des limites soft (douces ou basses) qui sont les valeurs par défaut retournées par ulimit, et des limites hard (dures, hautes) qui ne peuvent être dépassées.

Pour passer le nombre maximum de fichiers ouverts à 2048 :

```
$ ulimit -n 2048
```

```
$ ulimit -n
2048
```

L'administrateur root peut contrôler les valeurs par défaut grâce au fichier **/etc/security/limits.conf**.

```
$ grep seb /etc/security/limits.conf
seb             hard    nproc        32768
seb             soft    nofile         1024
seb             hard    nofile         4096
```

Dans cet exemple l'utilisateur seb est limité à un maximum de 32768 processus, peut ouvrir par défaut 1024 fichiers mais peut monter, via une action ulimit de sa part, à 4096.

10. Les droits SUDO

a. Donner des privilèges étendus

La commande **sudo** permet d'attribuer le droit d'exécuter des commandes données à un ou plusieurs utilisateurs, sur une ou plusieurs machines. En pratique, si un utilisateur doit exécuter une commande que seul root peut en principe exécuter, vous pouvez ajouter un droit sudo à cet utilisateur pour cette commande.

Le fichier de configuration de sudo est **/etc/sudoers**. Il est possible de l'éditer à la main, ou par la commande **visudo**. Cette dernière commande vérifie la syntaxe du fichier à la sauvegarde.

La syntaxe classique d'une ligne sudo est la suivante :

```
user        machine = (user2) commande
```

- **user** : l'utilisateur (ou l'alias) à qui s'applique la règle.
- **machine** : la machine (ou l'alias) où s'applique la règle.
- **user2** : en tant que quel utilisateur exécuter la commande.
- **commande** : la commande à exécuter.

Par exemple, la ligne suivante va autoriser l'utilisateur seb à exécuter la commande `fsck` et ses paramètres avec les droits root sur n'importe quelle machine (où cette règle est présente) :

```
seb               ALL = /sbin/fsck
```

Pour utiliser fsck, seb doit utiliser la commande sudo comme ceci :

```
seb@slyserver:~/handbrake> sudo /sbin/fsck
seb's password:
fsck 1.41.1 (01-Sep-2008)
e2fsck 1.41.1 (01-Sep-2008)
...
```

Par défaut, le mot de passe de l'utilisateur seb est demandé avant de continuer. L'utilisateur peut obtenir la liste de ses droits sudo :

```
seb@slyserver:~/handbrake> sudo -l
User seb may run the following commands on this host:
    (root) /sbin/fsck
```

Par défaut, l'utilisateur est authentifié une première fois, puis son mot de passe n'est plus demandé tant qu'il reste dans la même session (tant qu'il ne ferme pas sa console ou son environnement).

L'avantage de sudo, outre d'attribuer des droits ponctuels à un groupe de personnes données, est la traçabilité. Les messages de sudo sont transmis à syslog qui peut les rediriger dans un fichier. Ce peut être /var/log/messages :

```
May  8 14:39:08 slyserver sudo:      seb
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=/sbin/fsck
May  8 14:40:14 slyserver sudo:      seb
```

```
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=list
```

La destination peut être réglée via syslog.conf ou syslog_ng.conf.

b. Syntaxe de /etc/sudoers

L'exemple précédent est volontairement limité. Il est possible de :

- créer des groupes d'utilisateurs,
- créer des groupes de machines,
- créer des groupes de commandes,
- forcer ou non l'utilisation d'un mot de passe,
- forcer l'exécution d'une commande sous un utilisateur autre que root.

Les groupes sont appelés des alias.

Pour créer des alias d'utilisateurs, utilisez cette syntaxe :

```
User_Alias     ADMINS = seb, steph, henri
```

Si tous les admins doivent pouvoir utiliser la commande fsck, la ligne devient :

```
ADMINS    ALL= /sbin/fsck
```

Si les administrateurs ne peuvent exécuter ces commandes que sur des machines données, créez un alias de machines :

```
Host_Alias    SERVERS= slyserver, eeepc
```

La ligne sudo devient :

```
ADMINS    SERVERS= /sbin/fsck
```

Il est possible d'ajouter plusieurs commandes les unes après les autres, avec ou sans paramètres. Si les ADMINS doivent aussi exécuter la commande /sbin/dumpe2fs, la ligne devient la suivante :

```
ADMINS  SERVERS=/sbin/fsck, /sbin/dumpe2fs
```

ou, pour passer à la ligne, mettez un antislash « \ » en fin de ligne :

```
ADMINS  SERVERS=/sbin/fsck, \
            /sbin/dumpe2fs
```

Vous pouvez créer des alias de commandes afin des les regrouper :

```
Cmnd_Alias ADMCMD=/sbin/fsck, /sbin/dume2fs
```

La ligne sudo devient :

```
ADMINS  SERVERS=ADMCMD
```

Vous pouvez empêcher la saisie du mot de passe de l'utilisateur en ajoutant NOPASSWD comme ceci :

```
ADMINS  SERVERS=NOPASSWD: ADMCMD
```

Les utilisateurs de l'alias ADMINS n'auront plus à saisir de mot de passe pour taper les commandes de l'alias ADMCMD.

Vous pouvez aussi forcer l'utilisation d'un mot de passe avec PASSWD. Ajoutez ainsi la commande mkfs :

```
ADMINS  SERVERS=NOPASSWD: ADMCMD, PASSWD:/sbin/mkfs
```

Pour que les ADMINS lancent une commande sous un autre utilisateur que root, placez le nom de l'utilisateur (ou l'alias) entre parenthèses comme ceci :

```
ADMINS  ALL=(steph) PASSWD: /sbin/service
```

ADMINS pourra exécuter /sbin/service en tant que steph, avec saisie de son mot de passe.

Le résultat du `sudo -l` associé est le suivant :

```
seb@slyserver:~> sudo -l
User seb may run the following commands on this host:
    (root) NOPASSWD: /sbin/fsck, /sbin/dumpe2fs
    (root) /sbin/mkfs
    (steph) /sbin/service
```

Pour lancer la commande sous un autre utilisateur, utilisez le paramètre -u de sudo :

```
seb@slyserver:~> sudo -u steph /sbin/service
```

Vous pouvez préciser plusieurs utilisateurs, mais aussi créer des alias comme pour le reste :

```
Runas_Alias    LISTUSR=seb,steph
```

Puis vous utilisez cet alias dans la ligne sudo :

```
ADMINS  ALL=(LISTUSR) PASSWD: /sbin/service
```

Vous pouvez utiliser l'alias ALL : il définit l'ensemble des utilisateurs, machines et commandes, selon la position où il est écrit. La ligne suivante signifie que les ADMINS ont le droit d'exécuter toutes les commandes, sur toutes les machines, sans mot de passe :

```
ADMINS ALL=NOPASSWD: ALL
```

Le point d'exclamation permet d'exclure certains utilisateurs, commandes, machines, d'un alias ou d'une liste. La ligne suivante donne tous les droits sur tous les programmes sans mot de passe aux ADMINS, sauf sur `/sbin/mkfs` qui nécessite un mot de passe et `/sbin/resize2fs` est interdit :

```
ADMINS  SERVERS=NOPASSWD: ALL, !/sbin/resize2fs, PASSWD:/sbin/mkfs
```

Si un ADMINS tente de lancer resize2fs, il obtient une erreur :

```
seb@slyserver:~> sudo /sbin/resize2fs
Sorry, user seb is not allowed to execute '/sbin/resize2fs' as root
on slyserver.
```

La règle suivante, très dangereuse, autorise tout le monde à exécuter tout ce qu'il veut, n'importe où et sans mot de passe, en tant que n'importe quel utilisateur :

```
ALL         ALL=(ALL) NOPASSWD: ALL
```

Ci-après une ligne fréquemment utilisée pour autoriser un utilisateur à tout faire sauf à lancer des shells ou la commande su, à condition que les alias SU et SHELLS soient complets :

```
seb         ALL = ALL, !SU, !SHELLS
```

Cette ligne ne garantit pas que seb puisse obtenir des droits par des moyens détournés, par exemple si les commandes ont été renommées, ou par rebonds de commandes successifs.

11. Audit plus complet

Pour effectuer l'audit d'un système, vous pouvez utiliser, en plus de l'accès aux traces et aux historiques, et des commandes déjà répertoriées, des produits libres ou gratuits comme tripwire qui vérifie l'intégrité du système, COPS qui surveille la sécurité du système, Crack qui détecte les mauvais mots de passe...

12. Les bulletins de sécurité

a. CERT : Computer Emergency Response Team

Historique

C'est à un étudiant de l'université de Cornell que vous devez le premier virus capable de se répliquer sur le réseau Internet (qui s'appelait Arpanet à l'époque). Développé et lâché fin 1998 sans intention de nuire, ce programme se propageait et se répliquait tout seul en exploitant des failles de sécurité de Unix et de ses services. Ce programme a rapidement saturé le réseau Internet et les machines qu'il avait atteintes, paralysant le réseau Internet, alors composé de 60000 ordinateurs, pour plusieurs jours. Seulement 4% des machines avaient été infectées.

L'éradication de ce virus a nécessité du temps et beaucoup de moyens, fournis par le MIT, Berkeley, etc. Il a fallu étudier son fonctionnement pour comprendre comment il avait procédé et partant de là les trous de sécurité des systèmes et des serveurs ont pu être corrigés. Des patches correctifs furent diffusés. Le DARPA, initiateur du projet Arpanet (puis Internet) mit alors en place une nouvelle structure appelée le CERT (CERT/CC, *CERT Coordination Center*) chargée d'analyser les menaces futures et la vulnérabilité des systèmes.

Internet représente des dizaines de millions de machines interconnectées, avec des systèmes d'exploitation et des services différents. Le CERT (http://www.cert.org/) dont le siège est aujourd'hui à l'université de Canergie Mellon continue d'étudier les éventuelles vulnérabilités sur Internet, et ce même à long terme, afin de tenter d'obtenir la meilleure sécurité possible.

Rôle du CERT

Les missions du CERT sont les suivantes :

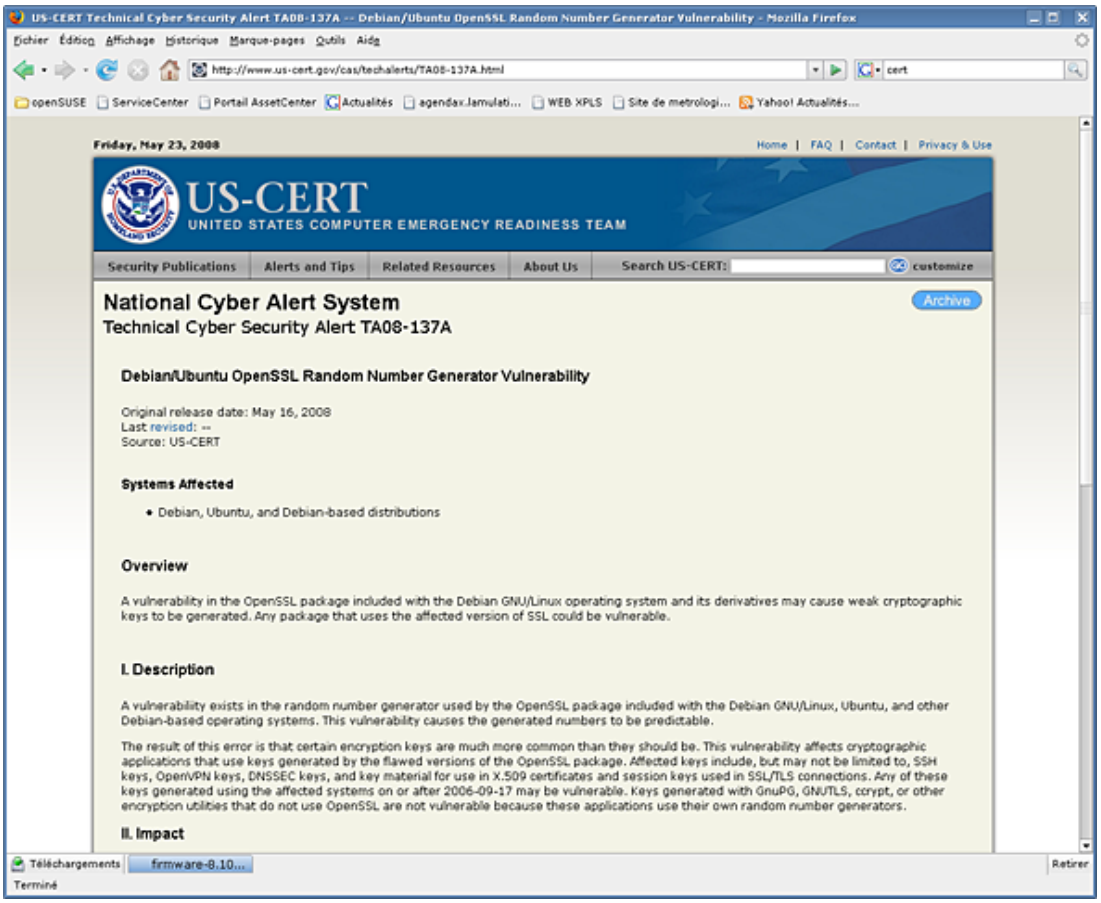
- centralisation et analyse des demandes d'assistance suite aux incidents de sécurité (attaques) sur les réseaux et les systèmes d'information : réception des demandes, analyse des symptômes et éventuelle corrélation des incidents ;
- traitement des alertes et réaction aux attaques informatiques : analyse technique, échange d'informations avec d'autres CERTs, contribution à des études techniques spécifiques ;
- établissement et maintenance d'une base de donnée des vulnérabilités ;
- prévention par diffusion d'informations sur les précautions à prendre pour minimiser les risques d'incident ou au pire leurs conséquences ;
- coordination éventuelle avec les autres entités : centres de compétence réseaux, opérateurs et fournisseurs d'accès à Internet, CERTs nationaux et internationaux.

Bulletins du CERT

Il existe plusieurs comités CERT : par pays, pour l'industrie, etc. Les bulletins sont émis indépendamment mais peuvent être repris d'un CERT à l'autre. Voici deux sites où trouver des alertes :

http://www.certa.ssi.gouv.fr/site/index.html

Et surtout, http://www.us-cert.gov/



Une alerte de sécurité sur US-CERT

Un exemple d'alerte est une faille de sécurité de la bibliothèque OpenSSL sous Debian « *Debian/Ubuntu OpenSSL Random Number Generator Vulnerability* », Référence TA08-137A. Les clés aléatoires générées depuis la version OpenSSL de Debian et Ubuntu ne le sont pas vraiment, ce qui limite fortement la sécurité de ces clés et qui pose un gros problème puisque la mise à jour de la bibliothèque ne suffit pas ; il faut aussi regénérer les clés existantes...

b. Bugtraq

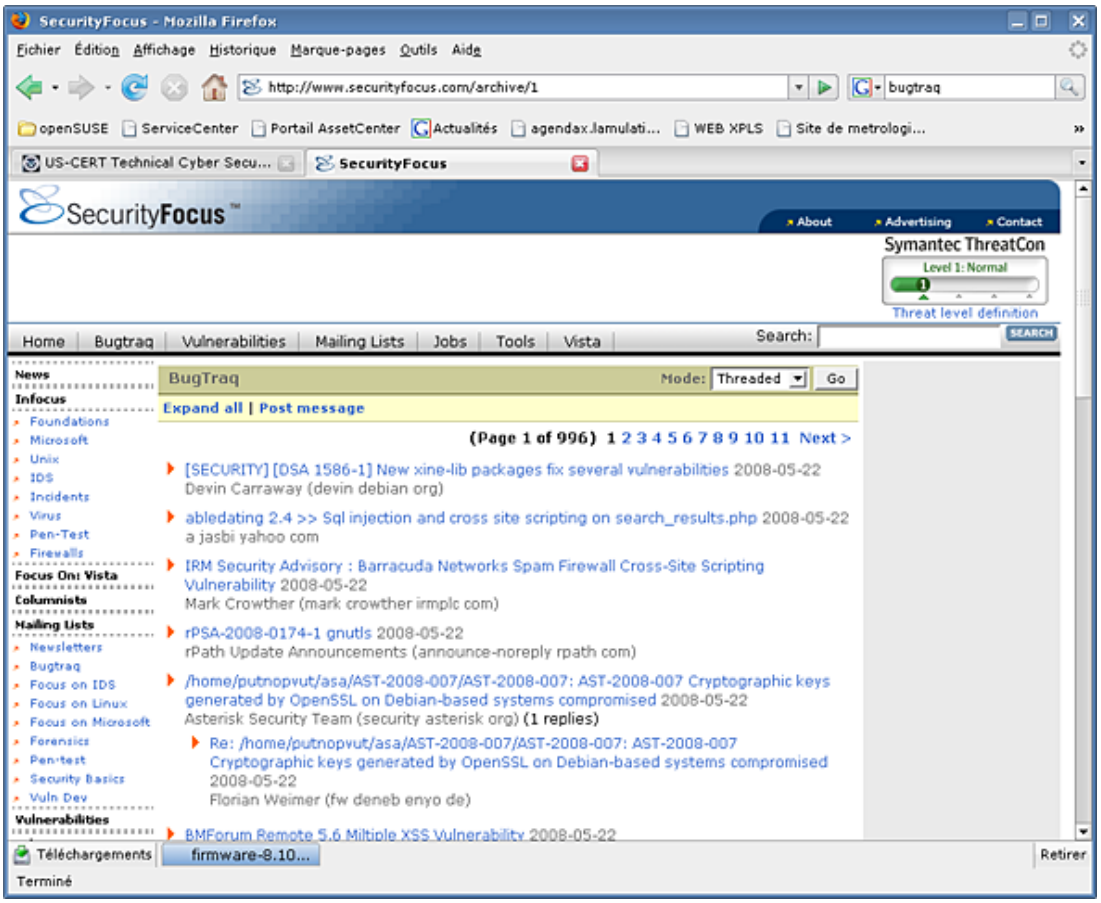
Bugtraq est une liste de diffusion électronique (e-mailing) qui existe depuis 1993 et qui regroupe des discussions sur les vulnérabilités, la sécurité, les annonces, les moyens d'exploiter les failles et comment les corriger. La liste a été créée à l'origine à cause de deux problèmes :

- les nombreux échecs du CERT à prévenir les problèmes ;
- le laisser-aller de nombreux éditeurs et constructeurs qui ne fournissaient pas de mises à jours de sécurité malgré les failles découvertes.

Bugtraq appartenait à SecurityFocus, lui-même appartenant aujourd'hui à l'éditeur de suites de sécurité Symantec.

Vous pouvez vous inscrire à la mailing-list depuis le site http://www.securityfocus.com/archive.

Dans cette même page vous avez accès (premier cadre en haut à gauche) à l'historique (archives) de la liste.



Liste des bulletins de sécurité sur SecurityFocus

c. Les bulletins des distributions

Chaque éditeur des plus grandes distributions fournit aussi des bulletins de sécurité. Ils reprennent eux-mêmes les alertes de sécurité d'autres organismes comme le CERT ou des listes Bugtraq, mais comme chaque éditeur est responsable des packages qu'il diffuse, il doit diffuser lui-même un correctif d'autant plus que certains produits sont allégrement patchés par l'éditeur et diffèrent fortement de l'original. D'où l'émission d'une alerte pour prévenir ses clients et utilisateurs. Voici les endroits où vous pouvez obtenir des informations de sécurité pour les principales distributions :

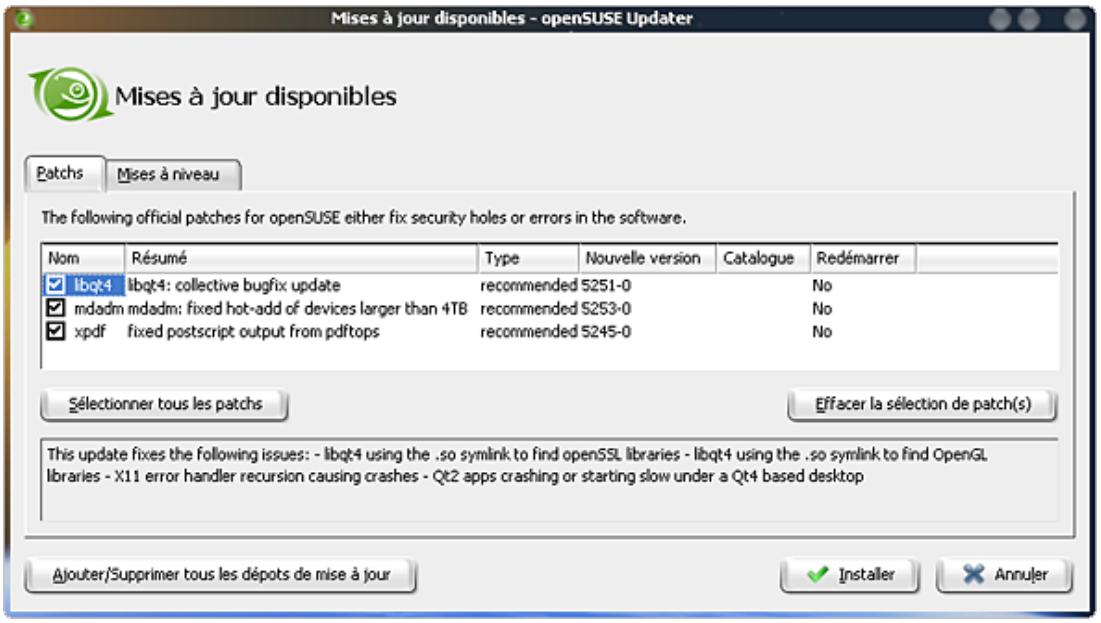
- Debian : http://www.debian.org/security/
- openSUSE : http://lists.opensuse.org/opensuse-security-announce/
- Fedora : https://www.redhat.com/archives/fedora-security-list/
- Ubuntu : http://www.ubuntu.com/usn
- Mandriva : http://www.mandriva.com/security/advisories
- Red Hat Enterprise : http://www.redhat.com/security/updates/errata/

d. Les correctifs

Il ne suffit pas de fournir une alerte lorsqu'un trou de sécurité est détecté, il faut aussi le réparer. Les éditeurs fournissent pour cela soit des packages corrigés (le lien est souvent dans le bulletin d'alerte), soit des patchs correctifs. Généralement chaque distribution est fournie avec un composant logiciel permettant de récupérer ces patches, mais aussi de vous informer de leur disponibilité.

Sur openSUSE, le produit openSUSE Updater se place dans la barre des tâches (ou plutôt la boîte à miniatures de

KDE ou son équivalent sur d'autres environnements) et vous informe de la présence de mises à jour. Vert : pas de mises à jours ou mises à jour mineures ; orange : mises à jour recommandées (pas de sécurité) ; rouge : mises à jour critiques.



Les mises à jour de bug et de sécurité par openSUSE Updater.

Chaque distribution est différente de ce point de vue : vous devez vous renseigner.

Sécurité des services et du réseau

1. Vérifier les ports ouverts

a. Les sockets

Les connexions réseau entre deux machines s'effectuent par des **sockets**. Une socket est une connexion entre deux points via un réseau. Une machine dispose d'une adresse IP et de ports (virtuels) de connexion numérotés, auxquels sont rattachés des services (voir pour cela le chapitre Le réseau). Un client établit une connexion depuis un port de sa machine (port > 1024, généralement choisi aléatoirement parmi les ports libres) vers un port donné d'une autre machine, par exemple un serveur Web sur le port 80. La communication établie entre les deux passe par une socket.

Le système peut être configuré pour accepter ou rejeter des connexions depuis ou vers certains ports locaux ou distants, idem pour les adresses IP. C'est le rôle du firewall (mur de feu) comme **Netfilter**.

Sur une installation Linux de base, sauf si l'option était présente lors de l'installation, le firewall n'est pas toujours activé par défaut. Les ports ne sont pas filtrés et toute machine extérieure peut tenter d'établir une connexion sur un port de votre machine : ce qui s'appelle ouvrir une socket.

Cela ne signifie pas forcément qu'il y a un trou de sécurité : si aucun service n'est à l'écoute, la connexion est impossible. C'est cependant rarement le cas. De nombreux services sont présents et démarrés par défaut. Si certains souffrent de vulnérabilités, ou sont configurés de manière trop laxiste, il existe un risque réel d'intrusion.

b. Informations depuis netstat

Le chapitre Le réseau vous a présenté l'outil **netstat** qui permet d'obtenir des informations et des statistiques réseau sur une machine locale. Notamment vous pouvez vérifier quels sont les ports à l'écoute sur votre machine, qui a établi une connexion, et quels sont les processus (services) locaux à l'écoute :

```
# netstat -a -A inet -p
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale        Adresse distante
Etat        PID/Program name
tcp        0      0 *:5800                *:*
LISTEN      31232/kded [kdeinit
tcp        0      0 *:5801                *:*
LISTEN      23224/xinetd
tcp        0      0 *:vnc-server          *:*
LISTEN      31232/kded [kdeinit
tcp        0      0 *:5901                *:*
LISTEN      23224/xinetd
tcp        0      0 *:sunrpc              *:*
LISTEN      3076/portmap
tcp        0      0 *:ndmp                *:*
LISTEN      26746/sshd
tcp        0      0 *:6000                *:*
LISTEN      31088/Xorg
tcp        0      0 *:7634                *:*
LISTEN      3006/hddtemp
tcp        0      0 *:ftp                 *:*
LISTEN      26772/vsftpd
tcp        0      0 localhost:ipp         *:*
LISTEN      3092/cupsd
tcp        0      0 localhost:smtp        *:*
LISTEN      3159/master
tcp        0      0 slyserver:38144       eeepc:ssh
ESTABLISHED 25622/ssh
tcp        0      0 slyserver:45046       eeepc:ms-wbt-server
ESTABLISHED 7803/rdesktop
tcp        0      0 slyserver:ftp         eeepc:49502
FIN_WAIT2   -
udp        0      0 *:xdmcp               *:*             27850/kdm
udp        0      0 *:59742               *:*             2984/avahi-daemon:
udp        0      0 *:mdns                *:*             2984/avahi-daemon:
```

```
udp        0      0 *:sunrpc                *:*                     3076/portmap
udp        0      0 *:ipp                   *:*                     3092/cupsd
```

Les lignes en gras montrent trois connexions établies ou terminées entre les machines slyserver et eeepc.

- l'eeepc était connecté sur le port ftp de slyserver,
- slyserver est connecté sur le port ssh de l'eeepc,
- slyserver est connecté sur le port ms-wbt-server de l'eeepc (protocole terminal server).

c. L'outil nmap

Il existe une armada d'outils de sécurité, de vérification, de tests, etc. L'outil **nmap** en fait partie. Il se définit comme un outil d'exploration réseau et d'audit de sécurité. Il permet de tester les connexions réseaux d'une machine donnée et de retourner un grand nombre d'informations. Notamment, en analysant les trames, il arrive souvent à déterminer le type et la version du système d'exploitation distant.

Le mode d'emploi de nmap fait plus de 2000 lignes, il est donc impossible d'étudier l'ensemble du produit. Mais voici quelques possibilités.

Examinez les ports à l'écoute sur la machine de test ayant servi à la rédaction de ce livre. Plusieurs ports sont ouvert (des services sont à l'écoute). Certains d'entre eux peuvent présenter des risques : telnet, netbios-ssn et microsoft-ds (partages windows), ftp, vnc, etc.

```
# nmap localhost

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 14:41 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 1684 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
25/tcp    open  smtp
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
5800/tcp  open  vnc-http
5801/tcp  open  vnc-http-1
5900/tcp  open  vnc
5901/tcp  open  vnc-1
6000/tcp  open  X11
10000/tcp open  snet-sensor-mgmt

Nmap finished: 1 IP address (1 host up) scanned in 0.170 seconds
```

En quelques commandes il est possible de désactiver les services non nécessaires :

```
# chkconfig telnet off
# service vsftpd stop
# service smb stop
# service nmb stop
# service xinetd restart
# nmap localhost
Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 14:48 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 1688 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
111/tcp   open  rpcbind
631/tcp   open  ipp
5800/tcp  open  vnc-http
5801/tcp  open  vnc-http-1
5900/tcp  open  vnc
5901/tcp  open  vnc-1
6000/tcp  open  X11
```

```
10000/tcp open  snet-sensor-mgmt
```

Dès qu'un service réseau est arrêté, le port n'est plus accessible.

Le paramètre `-A` permet de détecter en plus le système d'exploitation distant et sa version. Pour cela un ou plusieurs ports doivent être ouverts. Mieux : nmap interroge chaque service associé aux ports trouvés quand c'est possible pour récupérer des informations. Notez en gras les valeurs remarquables :

```
# nmap -A machine

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 19:54 CEST
Interesting ports on machine.mondomaine.com (192.168.1.25):
Not shown: 1676 closed ports
PORT     STATE SERVICE     VERSION
9/tcp    open  discard
13/tcp   open  daytime
21/tcp   open  ftp         vsftpd 2.0.5
22/tcp   open  ssh         OpenSSH 4.3p2 Debian 9 (protocol 2.0)
25/tcp   open  smtp        Postfix smtpd
37/tcp   open  time         (32 bits)
53/tcp   open  domain
80/tcp   open  http        Apache httpd 2.2.3 ((Debian) PHP/5.2.0-8+etch10)
111/tcp  open  rpcbind      2 (rpc #100000)
113/tcp  open  ident       OpenBSD identd
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: SLYNET)
199/tcp  open  smux        Linux SNMP multiplexer
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: SLYNET)
606/tcp  open  mountd       1-2 (rpc #100005)
631/tcp  open  ipp         CUPS 1.2
901/tcp  open  http        Samba SWAT administration server
1389/tcp open  ldap        OpenLDAP 2.2.X
2049/tcp open  nfs          2 (rpc #100003)
3128/tcp open  squid-http
7937/tcp open  nsrexec      1 (rpc #390113)
7938/tcp open  rpcbind      2 (rpc #100000)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.14 - 2.6.17
Uptime: 112.678 days (since Wed Jan 30 21:39:53 2008)
Network Distance: 2 hops
Service Info: Host:  machine.mondomaine.org; OSs: Unix, Linux, OpenBSD

OS and Service detection performed. Please report any incorrect results
at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 115.360 seconds
```

Ainsi, une machine dispose au final de bien peu de secrets si elle est mal protégée. Le serveur est sous Linux Debian Etch disposant d'un noyau compris entre 2.6.14 et 2.6.17 (une erreur de détection ici, le noyau est un 2.6.18). Les services :

- OpenSSH 4.3p2
- Apache 2.2.3 et PHP 5.2.0
- Postfix
- Samba 3.x
- Cups 1.2
- OpenLDAP 2.2.X
- Vsftpd 2.0.5

Il suffit d'aller consulter les bulletins de sécurité pour voir si l'une de ces versions est connue pour avoir des

problèmes de sécurité, et si c'est le cas le serveur présente un risque.

2. Supprimer les services inutiles

a. Généralités

Si vous vous faites « hacker » votre système, c'est que la personne mal intentionnée a trouvé le moyen de rentrer. Contrairement à l'idée répandue, l'installation d'un firewall ne résout pas tous les problèmes, d'autant plus que sur un poste de travail la tendance est d'ouvrir plusieurs ports réseaux vers Internet (ou plutôt depuis Internet) : ftp, http, p2p (réseaux eDonkey, c'est-à-dire eMule, Bittorrent, etc.), ssh et ainsi de suite. Or il suffit qu'un seul des services associés présente un risque pour que votre machine soit attaquée avec les problèmes qui en découlent.

Même si le service en tant que tel n'a pas de trou connu, le paramétrage que vous avez appliqué peut être trop simple ou laxiste. Il ne serait pas très malin de laisser votre serveur ssh accepter les connexions depuis l'extérieur si votre mot de passe ou celui de root est toto, password ou quelque chose de ressemblant. Lors d'une attaque l'auteur de ce livre a eu l'occasion de constater que l'attaquant tentait en boucle de se connecter, via ssh, en utilisant une série de logins/mots de passe prédéfinis parmi certaines combinaisons classiques préconfigurées par défaut dans certains programmes. Un cas simple est une installation de MySQL par défaut où le compte d'administration n'a pas de mot de passe.

Donc, pensez à désactiver tous les services dont vous n'avez pas besoin. S'il s'avère que certains vous sont nécessaires à certains moments et pas à d'autres, n'hésitez pas à les démarrer seulement à ce moment, et à les stopper ensuite. De même, sur votre firewall (netfilter ou autre) ne laissez ouverts que les ports strictement nécessaires.

b. Services standalone

Les services standalone, c'est-à-dire lancés de manière indépendantes, sont contrôlés via la commande **service** ou **`/etc/init.d/service`**. Pour contrôler les arrêts et relances de ces services de manière pérenne, utilisez les commandes **chkconfig** (distributions RPM) ou **rcudpate.d** (Debian).

c. Services xinetd

Les services contrôlés par xinetd peuvent être activés et désactivés par la ligne « **disable** » de leur fichier de configuration.

```
$ pwd
/etc/xinetd.d
$ grep disable *
chargen:         disable        = yes
chargen-udp:     disable        = yes
cups-lpd:        disable        = yes
...
vmware-authd:    disable        = no
vnc:             disable        = yes
...
```

Pour prendre en compte les changements, forcez xinetd à recharger sa configuration.

```
# /etc/init.d/xinetd reload
```

3. Les tcp_wrappers

Les **enveloppeurs TCP** ou tout simplement **tcp_wrappers** permettent la vérification des accès à un service réseau donné (service, xinetd, portmapper). Chaque programme utilisant les tcp_wrappers est compilé avec la bibliothèque **libwrap** de manière statique (la commande ldd ne permet pas de voir la bibliothèque).

Pour savoir si un service réseau est compilé avec **libwrap**, on saisit la commande suivante :

```
strings -f <binaire> | grep hosts_access
```

Voici un exemple avec xinetd qui utilise les tcp_wrappers :

```
# strings -f /usr/sbin/xinetd |grep hosts_access
/usr/sbin/xinetd: hosts_access
```

Si aucune ligne n'est retournée, le programme n'utilise pas les tcp_wrappers.

Parmi les services utilisant les tcp_wrappers, on trouve :

- **sendmail** (y compris postfix),
- **sshd** (ssh),
- **xinetd** (et donc indirectement tous les services associés),
- **vsftpd** (ftp),
- **portmap** (et donc nis, nfs),
- **in.telnetd** (telnet) ainsi que la plupart des services supportés par xinetd,
- **dovecot** (imap, pop).

La vérification d'accès à un service enveloppé TCP a lieu en trois étapes :

- l'accès est-il explicitement autorisé ?
- sinon, l'accès est-il explicitement interdit ?
- sinon, par défaut, l'accès est autorisé.

Les fichiers de configuration sont **`/etc/hosts.allow`** et **`/etc/hosts.deny`**. La syntaxe est commune :

```
daemon_list : client_list [:options]
```

- **daemon_list** : liste des **exécutables (PAS DES SERVICES)** séparés par des virgules. Vous pouvez mettre **ALL** pour spécifier tous les services. Si vous disposez de plusieurs interfaces réseau on peut utiliser la syntaxe avec @ : service@ip.

```
in.telnetd: ...
sshd, portmap: ...
sshd@192.168.1.7 : ...
```

- **client_list** : clients autorisés ou interdits pour ce service. On peut spécifier l'adresse IP, le nom, le masque de réseau, le nom du réseau, etc.

```
... : 192.168.1.7, 192.168.1.8
... : 192.168.1.
... : poste1, poste2
... : 192.168.1.0/255.255.255.0
... : .mondomaine.org
```

La liste des clients admet une syntaxe avancée :

- **ALL** : correspondance systématique.
- **LOCAL** : tous les hôtes dont le nom ne contient pas de point (poste1, poste2, etc.).
- **UNKNOWN** : hôtes dont le nom ne peut pas être résolu.

- **KNOWN** : hôtes dont le nom peut être résolu.
- **PARANOID** : hôtes dont le nom ne peut être résolu ou dont l'IP n'a pas de résolution inverse.
- **EXCEPT** : permet d'exclure certains hôtes.

```
ALL EXCEPT poste10
```

/etc/hosts.allow est lu en premier, puis **/etc/hosts.deny**. La recherche s'arrête à la première correspondance trouvée. Une ligne dans hosts.allow autorise la connexion. Une ligne dans hosts.deny interdit la connexion. Si l'accès n'est pas explicitement refusé, la connexion est autorisée : la requête ne correspond à aucun critère.

Dans l'exemple suivant :

- Seuls les membres du sous-réseau 192.168.1.0 ont le droit de se connecter au serveur ftp (interdit pour tous les autres).
- Les hôtes poste1 et poste2 ont accès à telnet et portmap.
- Les hôtes de baddomaine.org, sauf trusted, n'ont aucune connexion possible.
- Le serveur pop/imap est interdit à tous ceux du réseau 192.168.0.0 sauf 192.168.1.5.

```
# /etc/hosts.allow
vsftpd:            192.168.1.
in.telnetd, portmap:    poste1, poste2

# /etc/hosts.deny
ALL : .baddomaine.org except trusted.baddomaine.org
vsftpd,in.telnetd,portmap : ALL
dovecot : 192.168.0. EXCEPT 192.168.1.5
```

4. Netfilter

a. Présentation

Netfilter est une architecture de filtrage des paquets pour les noyaux Linux 2.4 et 2.6. Le filtrage se fait au sein même du noyau au niveau des couches 2, 3 et 4 du modèle OSI, c'est-à-dire les liaisons données, réseau et transport. Il est par exemple capable d'agir à bas niveau sur les interfaces ethernet (2), sur la pile IP (3) et sur les protocoles de transport comme TCP (4). Le filtrage est **stateless** : comme Netfilter n'inspecte que les en-têtes des paquets, il est extrêmement rapide et n'entraîne pas de temps de latence.

L'inspection des contenus des paquets (protocoles applicatifs) peut se faire à l'aide d'extensions mais devrait cependant être réservée à des outils de l'espace utilisateur.

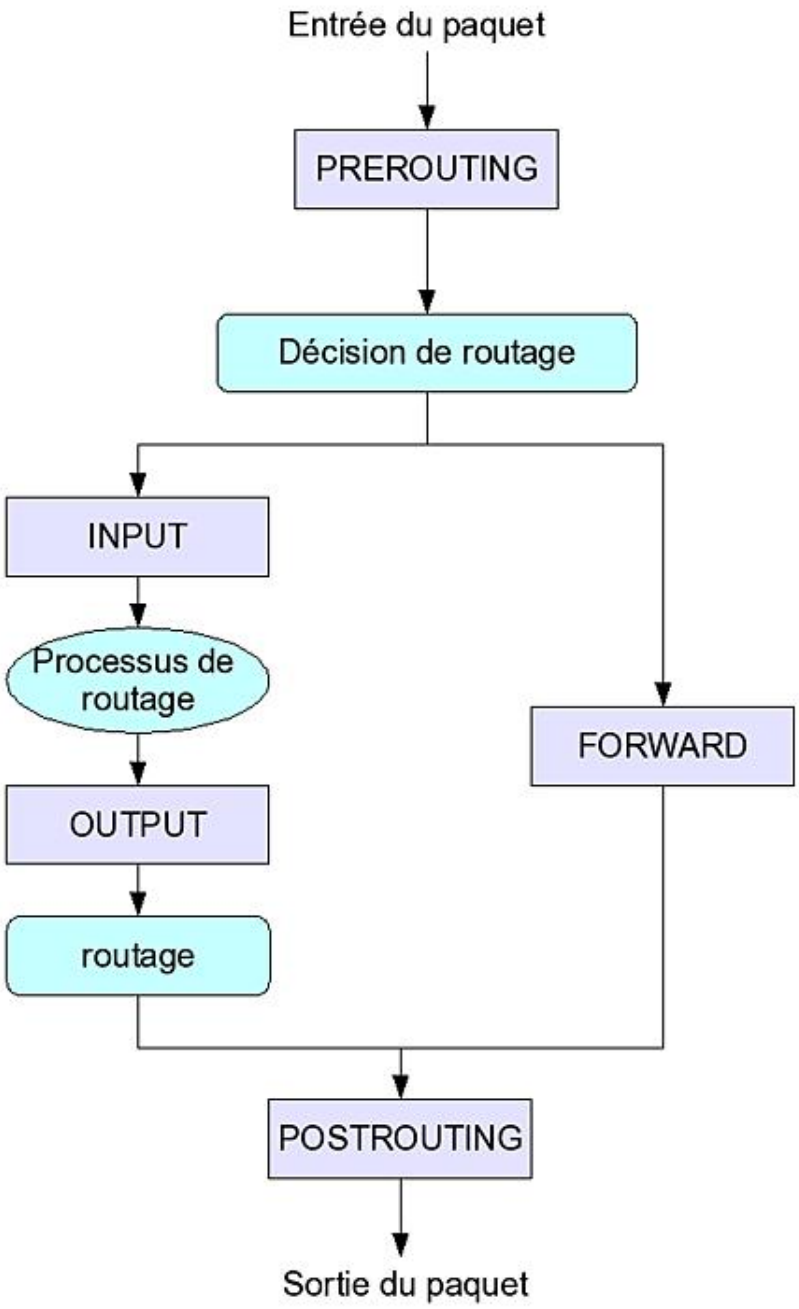
Autrement dit, Netfilter est un firewall agissant au niveau du protocole.

Le programme utilisateur permettant d'agir sur les règles de filtrage est **iptables**.

L'implémentation au niveau du noyau est réalisée par des modules.

b. Vie d'un paquet

Plutôt qu'un long discours, voici un schéma. Le paquet arrive par le haut et ressort par le bas. Entre les deux, il passe par différents états au niveau de netfilter.



Les étapes de la vie d'un paquet réseau avec netfilter.

Chaque état (rectangle) correspond à un point de filtrage possible par la commande **iptables**.

- **PREROUTING** : traite les paquets à leur arrivée. Si un paquet est à destination du système local, il sera traité par un processus local (INPUT, OUTPUT). Sinon, et si le forwarding est activé, les règles FORWARD et POST_ROUTING seront appliquées.
- **FORWARD** : les paquets ne font que traverser le système local. Traite les paquets routés à travers le système local.
- **INPUT** : traite les paquets destinés au système local, en entrée (après le routage).
- **OUTPUT** : traite les paquets quittant le système local, avant POSTROUTING.
- **POSTROUTING** : traite les paquets juste avant leur sortie du système.

c. Principe des règles

Lorsqu'un paquet est traité par netfilter, il l'est par rapport à un certain nombre de règles qui déterminent ce qu'il faut en faire.

- Les règles sont ordonnées : la position d'une règle dans une liste indique quand et si la règle sera utilisée.
- Les paquets sont testés avec chacune des règles, l'une après l'autre.
- Netfilter fonctionne selon le mécanisme de la première correspondance. Si une règle correspond, les autres règles sont négligées et la vérification s'arrête.
- Une règle peut spécifier plusieurs critères.
- Pour qu'une règle corresponde à un paquet, tous les critères doivent correspondre.
- Si malgré toutes les règles, le paquet passe, une règle par défaut peut être appliquée.

d. Cibles de règles

Une cible de règle détermine quelle est l'action à entreprendre lorsqu'un paquet correspond aux critères d'une règle. On utilise l'option **-j** de **iptables** pour spécifier la cible.

Les deux cibles de base sont DROP et ACCEPT. Des extensions de netfilter ajoutent d'autres cibles comme LOG ou REJECT.

- **DROP** : le paquet est rejeté. Aucune notification n'est envoyée à la source.
- **REJECT** : le paquet est rejeté, retournant une erreur à la source.
- **ACCEPT** : le paquet est accepté.
- **LOG** : une information est envoyée à syslog pour les traces.

Vous pouvez créer des règles sans cible. Dans ce cas, la règle incrémentera un compteur de paquets et un compteur d'octets associés à la règle afin d'établir une collecte de statistiques.

e. Premier exemple

Voici une règle simple qui interdit tous les paquets en provenance de 192.168.1.11.

```
# iptables -A INPUT -s 192.168.1.11 -j DROP
```

- **-A** : point de filtrage (INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING) qu'on appelle aussi **chaîne**.
- **-s** : source, peut être une adresse IP, un nom d'hôte, un réseau, etc.
- **-j** : jump, cible de la règle (ACCEPT, DROP, REJECT...)

Résultat : vous interdisez l'entrée des paquets dont la source est 192.168.1.11.

f. Opérations de base

Les règles sont numérotées à partir de 1.

- Ajoutez une règle avec **-A**.

```
# iptables -A INPUT -s 192.168.1.2 -j DROP
```

- Insérez une règle avec **-I** à la position souhaitée.

```
# iptables -I OUTPOUT -d 192.168.1.25 -j DROP 3 # inserer à la
3eme position
```

- Supprimez une règle avec **-D**.

```
# iptables -D INPUT 1 # supprime la règle 1
```

- Listez les règles avec **-L**.

```
# iptables -L OUTPUT
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- Utilisez **-F** (flush) pour supprimer l'ensemble des règles.

```
# iptables -F
```

- Utilisez **-P** (policy) pour modifier les règles par défaut d'une chaîne.

```
# iptables -P INPUT DROP
# iptables -L INPUT
Chain INPUT (policy DROP)
target     prot opt source               destination
```

g. Critères de correspondance

Général

Les critères de correspondance déterminent la validité d'une règle. Tous les critères doivent être vérifiés pour qu'un paquet soit bloqué. Les critères de base sont :

- **-i** : interface entrante (filtrage couche 2) ;
- **-o** : interface sortante (filtrage couche 2) ;
- **-p** : protocole de couche 4. Pour les noms, voir le fichier **/etc/protocols** ;
- **-s** : adresse IP de la source (ou réseau) ;
- **-d** : adresse IP de destination (ou réseau).

- Interdire les entrées par eth0.

```
iptables -A INPUT -i eth0 -j DROP
```

- Interdire le forward entre eth1 et eth2.

```
iptables -A FORWARD -i eth1 -o eth0 -j DROP
```

- Interdire le protocole ICMP en entrée (le ping !).

```
iptables -A input -p icmp -j DROP
```

TCP, UDP et ICMP

Suivant le protocole (couche 4) certaines options sont possibles. C'est le cas de tcp, udp ou icmp (notamment utilisé par ping). Le filtrage au niveau des protocoles est généralement effectué par des extensions à netfilter.

- **-p** : protocole (tcp, udp, icmp, etc.)
- **--sport** : port source
- **--dport** : port destination

Si vous souhaitez par exemple interdire les connexion entrantes à destination du port 80 (serveur httpd) procédez ainsi :

```
iptables -A INPUT -p tcp -dport 80 -j DROP
```

Arguments des critères

Pour les adresses, vous pouvez spécifier :

- un hôte par son nom ou son adresse IP ;
- un réseau par son nom ou son masque (192.168.1.0/24, 192.168.1.0/255.255.255.0).

Pour les ports :

- un numéro ;
- un nom (voir /etc/services) ;
- une gamme de ports : **123:1024**.

Dans tous les cas, la négation avec ! (point d'exclamation) est possible.

Pour interdire en entrée toutes les connexions sauf celles de 10.0.0.1 :

```
# iptables -A INPUT -s ! 10.0.0.1 -j DROP
```

h. Sauver ses réglages

Les règles définies avec iptables sont chargées en mémoire et transmises dynamiquement au noyau. En redémarrant la machine, elles sont perdues. **Red Hat** permet de sauver l'ensemble des règles de manière à les rendre persistantes.

```
# service iptables save
```

Les règles sont sauvées dans le fichier **/etc/sysconfig/iptables**.

Pour les autres distributions, consultez la documentation. Dans certains cas, il peut être utile de créer son propre script.

Les règles iptables devraient être chargées **AVANT** l'activation du réseau. Ainsi il n'y a aucun risque au niveau de la sécurité car les règles seront directement valides à l'activation du réseau.

5. GPG

a. Un clone de PGP

GPG (*Gnu Privacy Guard*) est un clone libre de **PGP** (*Pretty Good Privacy*). Il implémente l'algorithme de chiffrage RSA. PGP fait l'objet d'une norme que GPG respecte. Ceci signifie que les deux implémentations sont compatibles : les clés générées par l'un ou l'autre sont interchangeables.

Le but de GPG est de chiffrer une communication grâce à un chiffrement par clés asymétriques. Une clé permet de signer le texte, une autre clé sert à crypter le texte.

Le chiffrement par clés asymétriques utilise deux clés, une clé publique et une clé privée :

- Votre **clé publique**, diffusée à votre entourage, permet à celui-ci de chiffrer un message qui vous est destiné et que vous ne pourrez déchiffrer qu'avec votre clé privée.

- Votre **clé privée** vous permet de signer un message. La personne qui le reçoit vérifie votre signature à l'aide de la clé publique que vous lui avez fournie, attestant ainsi que vous êtes bien l'auteur du message.

La suite présente tout le nécessaire pour créer et gérer un trousseau de clés (appelé aussi porte-clés) : vos clés publiques et privées, les clés publiques de vos amis, les signatures, etc.

Les commandes sont présentées en mode console. Il existe des outils graphiques, comme kgpg, pour gérer vos clés. De même, de nombreux clients de messagerie intègrent GPG (Kmail l'intègre par défaut). Via le plugin Enigmail, GPG peut être facilement incorporé à Thunderbird. Les signatures GPG sont aussi utilisées pour vérifier l'origine de packages RPM, par exemple.

b. Générer les clés

Utilisez `gpg` avec le paramètre `--gen-key`. GPG commence par créer, s'ils n'existent pas, le répertoire racine de GPG propre à chaque utilisateur et les fichiers qui contiendront les éléments gérés par gpg.

```
seb@slyserver:~> gpg --gen-key
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: répertoire '/home/seb/.gnupg' créé
gpg:  nouveau fichier de configuration '/home/seb/.gnupg/gpg.conf'
créé
gpg: AVERTISSEMENT: les options de '/home/seb/.gnupg/gpg.conf' ne
sont pas encore actives cette fois
gpg: le porte-clés '/home/seb/.gnupg/secring.gpg' a été créé
gpg: le porte-clés '/home/seb/.gnupg/pubring.gpg' a été créé
```

Sélectionnez le type de clé souhaité. Le premier choix inclut les deux autres, c'est donc le choix à effectuer :

```
Sélectionnez le type de clé désiré:
   (1) DSA et Elgamal (par défaut)
   (2) DSA (signature seule)
   (5) RSA (signature seule)
Votre choix ? 1
```

Vous devez ensuite choisir la taille, en bits, de la clé à générer. Plus la clé est longue, plus le chiffrement est complexe. Mais sur des machines plus anciennes le décryptage peut alors prendre plus de temps. Vous pouvez choisir une clé de 1024 bits par exemple :

```
La paire de clés DSA fera 1024 bits.
les clés ELG peuvent faire entre 1024 et 4096 bits de longueur.
Quelle taille de clé désirez-vous ? (2048) 1024
La taille demandée est 1024 bits
```

Indiquez ensuite la durée de validité des clés. Par défaut, si vous prévoyez de l'utiliser le plus longtemps possible, choisissez zéro pour une durée infinie. Sinon, précisez la durée souhaitée selon le format indiqué. Validez votre choix. Sachez qu'il est possible d'invalider une clé générée par erreur avec une clé de révocation que vous rencontrerez un peu plus loin.

```
Spécifiez combien de temps cette clé devrait être valide.
         0 = la clé n'expire pas
```

```
      <n>  = la clé expire dans n jours
      <n>w = la clé expire dans n semaines
      <n>m = la clé expire dans n mois
      <n>y = la clé expire dans n années
La clé est valide pour ? (0)
La clé n'expire pas du tout
Est-ce correct ? (o/N) o
```

Afin de générer la clé, GPG vous demande de saisir votre nom, votre adresse de courrier électronique (e-mail) et un commentaire. Dans le nom, n'hésitez pas à placer vos noms et prénoms. Le commentaire pourra ensuite être utilisé comme alias pour certaines commandes GPG. Il ne sera plus possible par la suite de changer ces éléments. Il vous sera par contre possible d'ajouter des adresses e-mail afin d'en associer plusieurs à une seule clé.

```
Vous avez besoin d'un nom d'utilisateur pour identifier votre clé;
le programme le construit à partir du nom réel, d'un commentaire et
d'une adresse e-mail de cette manière:
   « Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de> »

Nom réel: Sébastien ROHAUT
Adresse e-mail: sebastien.rohaut@domaine.fr
Commentaire: slyce
Vous utilisez le jeu de caractères 'utf-8'.
Vous avez sélectionné ce nom d'utilisateur:
    "Sébastien ROHAUT (slyce) <sebastien.rohaut@domaine.fr>"

Changer le (N)om, le (C)ommentaire, l'(E)-mail ou (O)K/(Q)uitter ? O
```

La **passphrase** ou phrase de passe est complémentaire à la clé privée. Elle sera demandée à chaque envoi ou réception de message privé crypté. Elle garantit aussi, tout au moins tant qu'elle n'est pas trouvée, une certaine sécurité de la clé privée même si elle est volée car cette clé ne peut pas être utilisée sans cette phrase. Choisissez une phrase (ou un mot) assez complexe, dont vous vous souviendrez tout le temps, mais pas trop longue car n'oubliez pas que vous devrez la saisir régulièrement.

```
Vous avez besoin d'une phrase de passe pour protéger votre clé secrète.
Entrez la phrase de passe:
Répétez la phrase de passe :
```

La dernière étape est effectuée par GPG, la génération des clés elles-mêmes. GPG va se servir des informations saisies et d'un générateur de nombres aléatoires. Un conseil vous est donné : durant le calcul, faites travailler votre ordinateur afin de « secouer » le générateur de nombres aléatoires.

```
Un grand nombre d'octets aléatoires doit être généré. Vous devriez
faire autre-chose (taper au clavier, déplacer la souris, utiliser
les disques) pendant la génération de nombres premiers; cela donne
au générateur de nombres aléatoires une meilleure chance d'avoir
assez d'entropie.
gpg: clé 13E021A8 marquée comme ayant une confiance ultime.
les clés publique et secrète ont été créées et signées.

gpg: vérifier la base de confiance
gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle
de confiance PGP
gpg: profondeur: 0  valide:   1  signé:   0
confiance: 0-. 0g. 0n. 0m. 0f. 1u
pub   1024D/13E021A8 2009-05-04
Empreinte de la clé = 6115 DE46 2678 40AB 0AB2  B8B9 1F12 D427 13E0 21A8
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@domaine.fr>
sub   1024g/BA311C5C 2009-05-04
```

Les quatre dernières lignes récapitulent toutes les informations sur le travail effectué par GPG :

- **pub** : elle donne la longueur de la clé (1024 bits), de numéro 13E021A8, générée le 04/05/2009.
- **empreinte** : c'est le « fingerprint » permettant de déterminer la validité de la clé publique, un peu comme une clé de numéro de sécurité sociale, ou plus proche de vous la somme md5 d'un fichier quelconque. Il est quasiment impossible (mais pas tout à fait), sur des clés de grande taille, d'avoir une empreinte identique.
- **uid** : les informations nominatives (nom, commentaire ou alias/pseudo, adresse e-mail).

- **sub** : c'est la taille (1024 bits), le numéro et la date de génération de la clé privée.

c. Générer une clé de révocation

Générer une clé de révocation vous permettra de révoquer votre clé, c'est-à-dire d'annuler sa validité. Elle est signée par votre clé privée. Pensez à deux cas de figure : si votre clé privée a été volée, il faut l'annuler pour qu'elle ne soit plus valide. Si vous avez perdu votre passphrase, il faudra aussi révoquer votre clé. Dans les deux cas, vous devrez ensuite régénérer une clé. La clé de révocation permet à toutes les personnes disposant de votre clé publique de vérifier qu'elle l'annule. Une fois la clé de révocation générée, conservez-la dans un endroit sûr, ne la diffusez pas et ne la perdez pas.

Il est donc très important de créer une clé de révocation juste après avoir créé le couple de clés publique/privée, et de la stocker dans un endroit sûr. Ceci est vrai pour plusieurs raisons : si vous perdez par exemple votre clé privée, vous n'aurez plus la possibilité de générer la clé de révocation. Si vous oubliez votre passphrase, vous ne pourrez plus utiliser votre clé privée, donc vous ne pourrez pas créer de clé de révocation.

Par contre, faites attention à ne pas divulguer votre clé de révocation : si celle ci était connue n'importe qui pourrait révoquer une clé que vous utilisez.

Lancez la commande **gpg** avec `--gen-revoke` et le nom ou le commentaire (alias, pseudo) associé à la clé. Saisissez la raison pour laquelle vous voulez générer une clé de révocation. Mettez la raison que vous voulez et le commentaire associé au choix et confirmez.

```
seb@slyserver:~> gpg --gen-revoke Slyce

sec  1024D/13E021A8 2009-05-04 Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr>

Générer un certificat de révocation pour cette clé ? (o/N) o
choisissez la cause de la révocation:
  0 = Aucune raison spécifiée
  1 = La clé a été compromise
  2 = La clé a été remplacée
  3 = La clé n'est plus utilisée
  Q = Annuler
(Vous devriez sûrement sélectionner 1 ici)
Votre décision ? 1
```

Entrez une description optionnelle ; terminez-la par une ligne vide :

```
> Exemple public pour livre
>
Cause de révocation: La clé a été compromise
Exemple public pour livre
Est-ce d'accord ? (o/N) o
```

Vous devez saisir ensuite, comme à chaque fois qu'il s'agit d'utiliser la clé privée, votre passphrase :

```
Vous avez besoin d'une phrase de passe pour déverrouiller la clé
secrète pour l'utilisateur: « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> »
clé de 1024 bits DSA, ID 13E021A8, créée le 2009-05-04
```

La clé est maintenant générée. Les consignes d'usages s'affichent : protégez votre clé.

```
sortie avec armure ASCII forcée.
Certificat de révocation créé.
Déplacez-le dans un support que vous pouvez cacher ; si Mallory a
accès à ce certificat il peut l'utiliser pour rendre votre clé
inutilisable.
Une bonne idée consiste à imprimer ce certificat puis à le stocker
ailleurs, au cas où le support devient illisible. Mais attention :
le système d'impression de votre machine pourrait stocker ces données
et les rendre accessibles à d'autres personnes !
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)
Comment: A revocation certificate should follow

iGIEIBECACIFAkoAQC8bHQJFeGVtcGxlIHB1YmxpYyBwb3VyIGxpdnJlAAoJEB8S
```

```
...
jlkHZg===9xGd
-----END PGP PUBLIC KEY BLOCK-----
```

Placez votre clé GPG sur un support sûr : imprimez-la, mettez-la sur un cd gravé (même toute seule), sur une clé USB (ne faites pas confiance à ce support), dans un coffre, ou apprenez-la par coeur (si vous avez une bonne mémoire).

d. Gérer le trousseau

Listez les clés gérées par GPG avec le paramètre `--list-key`. Pour chaque clé, vous voyez les informations sur les clés publiques et privées, leur propriétaire et l'éventuelle date d'expiration.

```
seb@slyserver:~> gpg --list-key
/home/seb/.gnupg/pubring.gpg
----------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@domaine.fr>
sub   1024g/BA311C5C 2009-05-04
...
```

Les signatures, ou plutôt la liste des signatures de chacune de vos clés est affichée, par clé publique et clé privée avec le `--list-sigs` qui inclut le `--list-key`. Les signatures seront expliquées par la suite.

```
seb@slyserver:~> gpg --list-sigs
/home/seb/.gnupg/pubring.gpg
----------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@domaine.fr>
sig 3        13E021A8 2009-05-04  Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr>
sub   1024g/BA311C5C 2009-05-04
sig          13E021A8 2009-05-04  Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr>
```

e. Exporter la clé publique

Pour que des personnes puissent utiliser votre clé publique pour vous envoyer des messages, vous devez leur fournir la clé et donc l'exporter dans un format lisible. Ceci se fait avec `gpg --export --armor` :

```
seb@slyserver:/usr/share/man> gpg --export --armor
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)

mQGiBEn/SbsRBADYR9n4VE2k+z7Ec9uT71+2KKlFgEgajQoYUSchyqZnIov74sCe
TSDyNMFI6SAJ/FrsM1SMJgtYUMedkfjl1hTl4i2T+YzIatJC00zAuwvhzyD23xpr
Q0dqtzBlb0pUbFNOgx5gF2McbiwknYcup5dbCVLjr6kqSC8iTACseZuIPwCg1Rfd
HcsKri8zWmeAp6u1nEpu7xED/j5gFeLhBhceac42O/Wy3dNrq6omro47RL7Jldp2
...
rr0RMO6vMWxo0oqpNaWAA8igXzUKexlOGfe2VmBmiEkEGBECAAkFAkn/SbsCGwwA
CgkQHxLUJxPgIajcwwCeJKyF88yfjGQuVdtelnE3SYIs7/EAoMpfNEkgmPQyAX+C
tU/c3EHZLlxk
=4v68
-----END PGP PUBLIC KEY BLOCK-----
```

La sortie est tronquée car elle est trop longue. Il suffit de copier ce bloc complet et de l'envoyer, comme vous le souhaitez, à la personne à qui vous voulez la fournir (pièce jointe de mail, serveur de clés, diffusée sur un site, etc.).

Vous pouvez exporter votre clé publique vers un serveur de clés GPG, par défaut keys.gnupg.net. Ainsi une personne désirant obtenir votre clé publique, pour la signer par exemple comme nous le verrons ultérieurement, pourra la récupérer sur le serveur concerné.

```
seb@slyserver:~> gpg -k
/home/seb/.gnupg/pubring.gpg
----------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce)
```

```
<sebastien.rohaut@domaine.fr>
sub   1024g/BA311C5C 2009-05-04

seb@slyserver:/usr/share/man> gpg --send-keys 13E021A8
gpg: envoi de la clé 13E021A8 au serveur hkp keys.gnupg.net
```

Si vous vous rendez ensuite sur le serveur http://keys.gnupg.net/ et que vous tapez **Slyce** (par exemple, l'e-mail ou l'identifiant de clé marche aussi), vous obtiendrez la liste des clés correspondantes, dont celle ci-dessus.

f. Importer une clé

Vous pouvez importer une clé dans votre trousseau avec le paramètre `--import`. Le second paramètre peut être un fichier qui contient la clé. Si vous ne passez pas de fichier, effectuez un copié-collé de la clé dans la console, la dernière ligne devant être `-----END PGP PUBLIC KEY BLOCK-----` puis tapez [Ctrl] **D**.

```
seb@slyserver:~> gpg --import
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.0

mQGiBEn/SbsRBADYR9n4VE2k+z7Ec9uT71+2KKlFgEgajQoYUSchyqZnIov74sCeTSDy
NMFI6SAJ/FrsM1SMJgtYUMedkfjl1hTl4i2T+YzIatJC00zAuwvhzyD23xprQ0dqtzBl
b0pUbFNO
...
ZykVTkKarr0RMO6vMWxo0oqpNaWAA8igXzUKexlOGfe2VmBmiEkEGECAAkFAkn/SbsCG
wwACgkQHxLUJxPgIajcwwCe-JKyF88yfjGQuVdtelnE3SYIs7/EAoMpfNEkgmPQyAX+CtU/
c3EHZLlxk=4v68
-----END PGP PUBLIC KEY BLOCK-----
gpg: clé 13E021A8: « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> » n'a pas changé
gpg:        Quantité totale traitée: 1
gpg:                      inchangée: 1
```

Vous pouvez aussi recevoir une clé d'un serveur :

```
seb@slyserver:~> gpg --recv-keys D0FE7AFB
gpg: requête de la clé D0FE7AFB du serveur hkp keys.gnupg.net
gpg: clé D0FE7AFB: clé publique « Josh Triplett
<josh@joshtriplett.org> » importée
gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle
de confiance PGP
gpg: profondeur: 0  valide:   1  signé:   0
confiance: 0-. 0g. 0n. 0m. 0f. 1u
gpg:        Quantité totale traitée: 1
gpg:                       importée: 1
```

Vous pouvez supprimer une clé du trousseau ainsi :

```
seb@slyserver:~> gpg --delete-keys D0FE7AFB
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub  1024D/D0FE7AFB 2004-06-24 Josh Triplett <josh@joshtriplett.org>

Enlever cette clé du porte-clés ? (o/N) o
```

Enfin, vous pouvez régénérer le fingerprint d'une clé, c'est-à-dire son empreinte. Comme vu lors de la génération des clés, vous n'aurez la preuve qu'une clé publique appartient réellement à une personne donnée que si son empreinte est identique à celle qu'il vous a fournie.

```
seb@slyserver:~> gpg --fingerprint slyce@slyunix.org
pub   1024D/5D022685 2009-05-06
    Empreinte de la clé = 57BF 3ABF F392 1616 068B  7C73 779D 6BCF
5D02 2685
uid                  Sebastien ROHAUT (Slyce) <slyce@slyunix.org>
sub   2048g/4CA7E7CD 2009-05-06
```

g. Signer une clé

Signer une clé publique, c'est certifier que cette clé est bien celle de la personne indiquée par l'identifiant. C'est pour cela qu'il faut faire la vérification d'empreinte avant de signer une clé. Pour signer une clé qui vous a été remise, par exemple celle de votre ami Steph, vous devez d'abord l'importer dans votre trousseau :

```
seb@slyserver:~> gpg --import
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)

mQGiBEoB63QRBACrfwOSo2EKJRYF0lh0Y06X2ZNNEq8+50IJmXYTXLx4b+niYzjf
V1pW5/0Of1/iM80RFU8NjkIKuxKnK0jji/WQbwAyWcTVMQ8CNrH2Fgbahi+8mn/Z
...
dAIbDAUJAA0vAAAKCRD1IZKWFJB27BVSAJoCQ1i49aTNi3NopE+zgiZYwYSXaACe
LFxBG3QJUkmLpSX8L/9vzjAZ+6Y==QMLk
-----END PGP PUBLIC KEY BLOCK-----
gpg: clé 149076EC: clé publique « Steph (Cle de test)
<steph@free.fr> » importée
gpg:        Quantité totale traitée: 1
gpg:                       importée: 1
```

Vérifiez que tout s'est bien passé en listant les clés :

```
seb@slyserver:~> gpg -k Steph
pub   1024D/149076EC 2009-05-06 [expire: 2009-05-16]
uid                  Steph (Cle de test) <steph@free.fr>
sub   2048g/ABC8FBE2 2009-05-06 [expire: 2009-05-16]
```

Récupérez l'empreinte de la clé de Steph, pour la comparer avec celle qu'il vous a fournie :

```
seb@slyserver:~> gpg --fingerprint Steph
pub   1024D/149076EC 2009-05-06 [expire: 2009-05-16]
    Empreinte de la clé = B50E 5C76 C0A6 4BDC F83E  FB8F F521 9296
1490 76EC
uid                  Steph (Cle de test) <steph@free.fr>
sub   2048g/ABC8FBE2 2009-05-06 [expire: 2009-05-16]
```

Si l'empreinte est correcte, éditez la clé, ce qui ouvre un petit interpréteur de commandes pour travailler sur le trousseau ou la clé concernée. Notez l'état de confiance et de validité : il est à « inconnu » car la clé n'est pas signée :

```
seb@slyserver:~> gpg --edit-key Steph
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


pub  1024D/149076EC  créé: 2009-05-06  expire: 2009-05-16  utilisation: SC
                     confiance: inconnu       validité: inconnu
sub  2048g/ABC8FBE2  créé: 2009-05-06  expire: 2009-05-16  utilisation: E
[ inconnue] (1). Steph (Cle de test) <steph@free.fr>

Commande>
```

Tapez **trust** pour donner votre niveau de confiance :

```
Commande> trust
pub  1024D/149076EC  créé: 2009-05-06  expire: 2009-05-16  utilisation: SC
                     confiance: inconnu       validité: inconnu
sub  2048g/ABC8FBE2  créé: 2009-05-06  expire: 2009-05-16  utilisation: E
[ inconnue] (1). Steph (Cle de test) <steph@free.fr>

Décidez maintenant à quel point vous avez confiance en cet utilisateur
pour qu'il vérifie les clés des autres utilisateurs (vous pouvez
vérifier son passeport, vérifier les empreintes de plusieurs
sources différentes, etc.)

  1 = ne sais pas ou ne dirai pas
  2 = je ne fais PAS confiance
```

```
  3 = je crois marginalement
  4 = je fais entièrement confiance
  5 = je donne une confiance ultime
  m = retour au menu principal

Votre décision ? 4

pub  1024D/149076EC  créé: 2009-05-06  expire: 2009-05-16  utilisation: SC
                     confiance: entière      validité: inconnu
sub  2048g/ABC8FBE2  créé: 2009-05-06  expire: 2009-05-16  utilisation: E
[ inconnue] (1). Steph (Cle de test) <steph@free.fr>
Notez que la validité affichée pour la clé n'est pas nécessairement
correcte tant que vous n'avez pas relancé le programme.
```

Notez les niveaux de confiance : de 1 à 5, par ordre croissant. La traduction française est un peu surprenante. Le choix 3 par exemple correspond à « Je fais un peu confiance ». Si vous choisissez 5, une confirmation est demandée.

Tapez maintenant **sign** pour signer la clé avec votre clé privée. Vous aurez besoin de saisir votre passphrase :

```
Commande> sign

pub  1024D/149076EC  créé: 2009-05-06  expire: 2009-05-16  utilisation: SC
                     confiance: entière      validité: inconnu
 Empreinte de la clé principale: B50E 5C76 C0A6 4BDC F83E  FB8F F521
9296 1490 76EC

     Steph (Cle de test) <steph@free.fr>

Cette clé va expirer le 2009-05-16.
Etes-vous vraiment sûr(e) que vous voulez signer cette clé
avec votre clé « Sébastien ROHAUT (Slyce) <sebas-
tien.rohaut@domaine.fr> » (13E021A8)

Signer réellement ? (o/N) o

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Sébastien ROHAUT (Slyce) <sebas-
tien.rohaut@domaine.fr> »
clé de 1024 bits DSA, ID 13E021A8, créée le 2009-05-04
```

La clé de Steph est maintenant signée.

h. Signer et chiffrer

Maintenant que vous disposez de la clé publique de vos amis, vous pouvez signer un message (de vous) à destination de ceux-ci.

```
seb@slyserver:~> gpg --clearsign -u sebastien.rohaut@fdomaine.fr -a

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> »
clé de 1024 bits DSA, ID 13E021A8, créée le 2009-05-04

Salut les amis
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Salut les amis
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.9 (GNU/Linux)

iEYEARECAAYFAkoB8f8ACgkQHxLUJxPgIagxlACgoB0NKICcEX5D13CCWnYz3tXt
S4oAoKxFR0AiKFb4krF/XScBWfg4Xcmo=zmms
-----END PGP SIGNATURE-----
```

Envoyez votre message en clair, avec la signature, au destinataire. Celui-ci, recevant le message, va faire

l'opération inverse : vérifier si la signature correspond bien à vous et au message (en supposant que le message ait été placé dans un fichier message.asc). C'est le cas ici : gpg confirme que le message a été signé par Sébastien ROHAUT.

```
steph@slyserver:~> gpg --verify message.asc
gpg: Signature faite le mer. 06 mai 2009 22:24:31 CEST avec la clé
DSA ID 13E021A8
gpg: Bonne signature de « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> »
```

L'étape suivante consiste à chiffrer (crypter) le message complet, de manière à ce que seul le destinataire puisse le déchiffrer. Pour que Sébastien puisse envoyer un message chiffré à Steph dont il détient la clé publique, faites comme ceci :

```
seb@slyserver:~> gpg -r Steph -e --armor
Salut Steph, je teste l'envoi d'un message crypté
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.9 (GNU/Linux)

hQINAzFP80qryPviEAf/eZpDHfd4KeSQ2ENECdlrgRAsDIULeDeI/ZPPbdiHJ1Wv
...
iSmi8LMJ1MixdV0PuGp/yA===0caY
-----END PGP MESSAGE-----
```

Envoyez le message crypté (du BEGIN au END) à Steph. Celui-ci de son côté va tenter de le décrypter : lui seul doit pouvoir le faire avec sa clé privée.

```
steph@slyserver:~> gpg --decrypt message.asc
Vous avez besoin d'une phrase de passe pour déverrouiller la clé
secrète pour l'utilisateur: « Steph (Cle de test) <steph@free.fr> »
clé de 2048 bits ELG, ID ABC8FBE2, créée le 2009-05-06 (ID clé
principale 149076EC)

gpg: chiffré avec une clé de 2048 bits ELG, ID ABC8FBE2, créée le 2009-05-06
      « Steph (Cle de test) <steph@free.fr> »
Salut Steph, je teste l'envoi d'un message crypté
```

Le message est apparu sans erreur.

Dernière étape : chiffrer et signer en même temps. Si votre message est dans un fichier **test.msg**, tapez ceci :

```
seb@slyserver:~> gpg -u sebastien.rohaut@free.fr -r Steph -a -e -s
test.msg

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> »
clé de 1024 bits DSA, ID 13E021A8, créée le 2009-05-04
```

Un fichier test.msg.asc a été créé, contenant le message chiffré et crypté.

```
seb@slyserver:~> cat test.msg.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.9 (GNU/Linux)

hQIOAzFP80qryPviEAf/dxwdIBhRYhx/j0nxddcb8Ryw4lrZm8msj+vffEZUYyrW
...
ibpz9InSJoW3F7UcZzQvODt6EIuNlHDtzpljqLcr7iNIhpyR1J6nFo55WHQVDxLV
B39McBmmEdI==hI/9
-----END PGP MESSAGE-----
```

De son côté, Steph va déchiffrer et vérifier le message ainsi :

```
steph@slyserver:~> gpg -d -v test.msg.asc
Version: GnuPG v2.0.9 (GNU/Linux)
gpg: en-tête d'armure:
gpg: la clé publique est ABC8FBE2
gpg: utilisation de la sous-clé ABC8FBE2 à la place de la clé
principale 149076EC
```

```
Vous avez besoin d’une phrase de passe pour déverrouiller la
clé secrète pour l’utilisateur: « Steph (Cle de test) <steph@free.fr> »
gpg: utilisation de la sous-clé ABC8FBE2 à la place de la clé
principale 149076EC
clé de 2048 bits ELG, ID ABC8FBE2, créée le 2009-05-06 (ID clé
principale 149076EC)

gpg: no running gpg-agent - starting one
gpg: chiffré avec une clé de 2048 bits ELG, ID ABC8FBE2, créée le 2009-05-06
      « Steph (Cle de test) <steph@free.fr> »
gpg: données chiffrées avec AES256
gpg: nom de fichier original: ’test.msg’
Salut, ceci est un message chiffré et crypté
gpg: Signature faite le mer. 06 mai 2009 22:33:47 CEST avec la clé DSA ID 13E021A8
gpg: utilisation du modèle de confiance PGP
gpg: Bonne signature de « Sébastien ROHAUT (Slyce)
<sebastien.rohaut@domaine.fr> »
gpg: signature binaire, algorithme de hachage SHA1
```

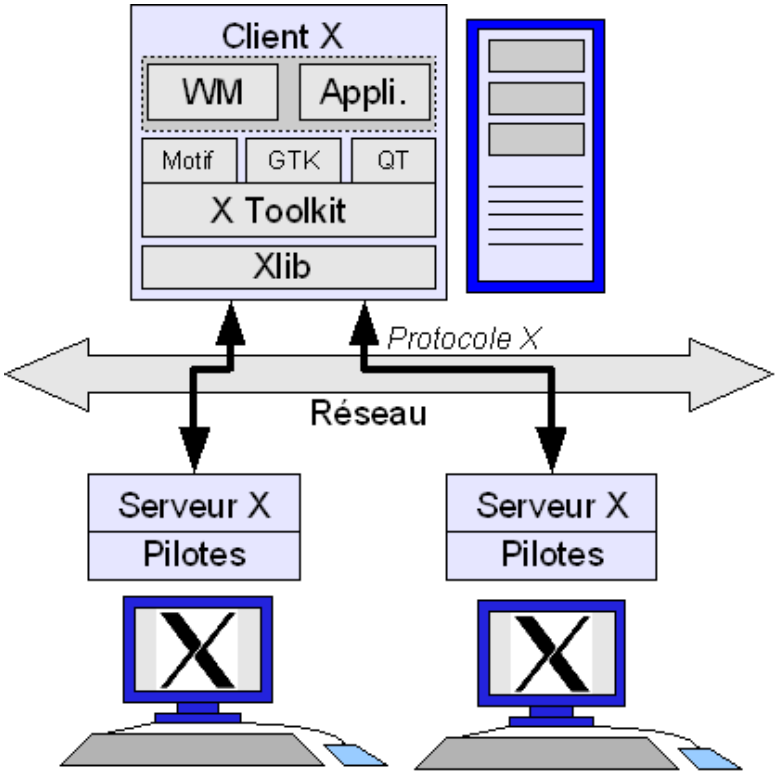
Comment fonctionne un environnement graphique ?

1. X Window System

a. Un modèle client/serveur

Alors que dans d'autres systèmes d'exploitation l'interface graphique est intégrée au plus profond du système, Unix et Linux disposent d'une architecture graphique totalement différente. Le système graphique de base s'appelle **X Window System** ou plus couramment **X Window**, **X11** ou tout simplement **X**.

X n'est pas qu'un simple programme. C'est un système graphique complet chargé de dessiner et de gérer les événements des composants habituels d'un environnement graphique utilisateur **GUI** *(Graphical User Interface)* : fenêtres, boutons, menus, listes, ascenseurs, cases à cocher, curseur de souris, etc. Notez la subtilité : X peut afficher, gérer et afficher ces composants graphiques mais n'est pas chargé de les mettre en place. X ne gère que les interactions entre l'homme et la machine.



Architecture X Window

X a une a particularité : il est client/serveur. Le serveur X est souvent un composant logiciel sur un ordinateur disposant d'un clavier, d'une souris et d'un écran. Il reçoit et répond à des ordres d'affichage, ou issus du clavier et de la souris. Le client X se connecte au serveur et lui envoie des ordres d'affichage, des demandes de saisie au clavier ou l'état de la souris. Autrement dit, un client X est un programme qui est capable de dialoguer avec le serveur X. Dans les faits un client X est un logiciel graphique. Pour pouvoir communiquer avec le serveur il utilise un composant appelé Xlib. Le client et le serveur ne sont pas toujours sur la même machine. Le serveur qui gère l'affichage peut être sur un premier ordinateur et le logiciel graphique sur un autre ordinateur. Les ordres, appelés requêtes, entre le client et le serveur passent par le réseau.

Ne confondez pas X Window avec Windows. Le premier est un système d'affichage client/serveur, le second est un système d'exploitation. Le « Window » de X Window ne prend pas de « s ». De plus Windows et X Window ne sont pas compatibles entre eux, même s'il est possible d'installer un serveur X sous Windows.

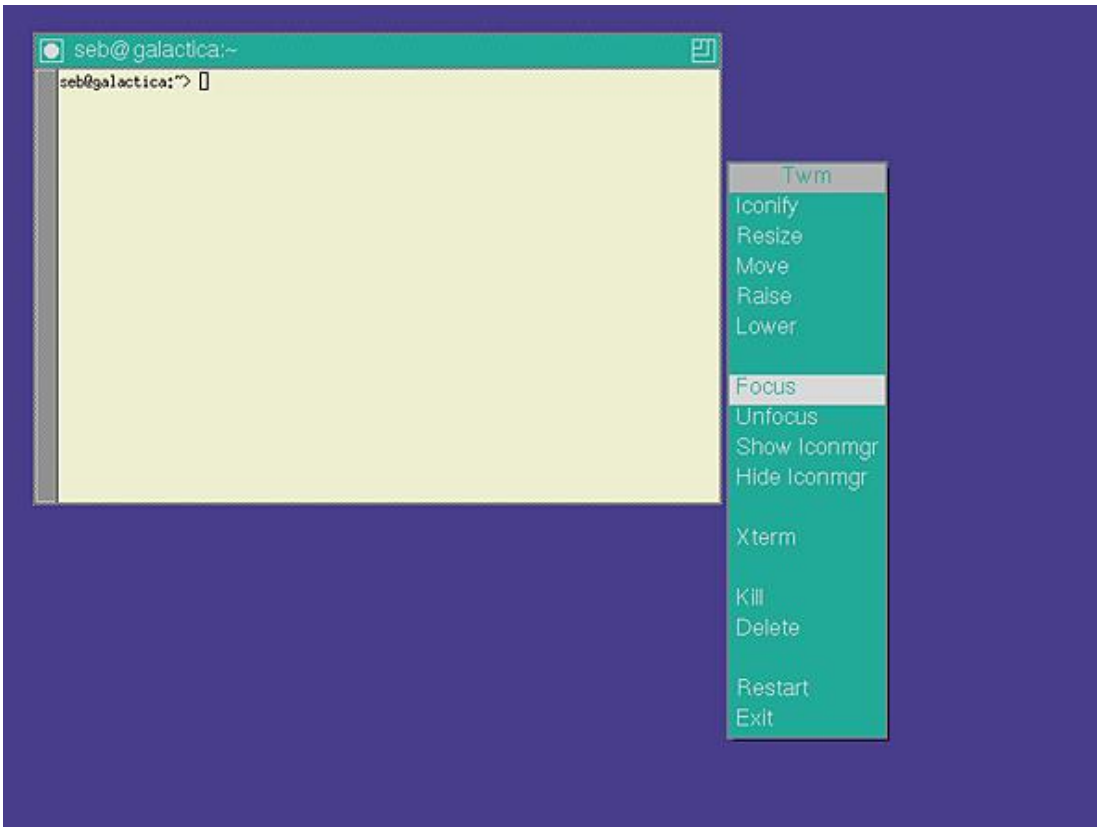
Ainsi vous pouvez lancer un programme (client) sur une première machine qui est affiché sur une autre machine (serveur). Pour l'utiliser vous devez aller sur le serveur et utiliser son écran, son clavier et sa souris.

Si vous lancez X Window seul vous lancez uniquement le serveur. Le résultat peut vous surprendre : un écran gris avec une croix en guise de curseur de souris. Vous aurez beau tout essayer, nulle fenêtre ne vient égayer l'affichage et les boutons de la souris n'ont aucun effet. Même si vous lanciez un client, vous remarqueriez vite un problème : il

n'y a pas de contours de fenêtres.

b. Le gestionnaire de fenêtres

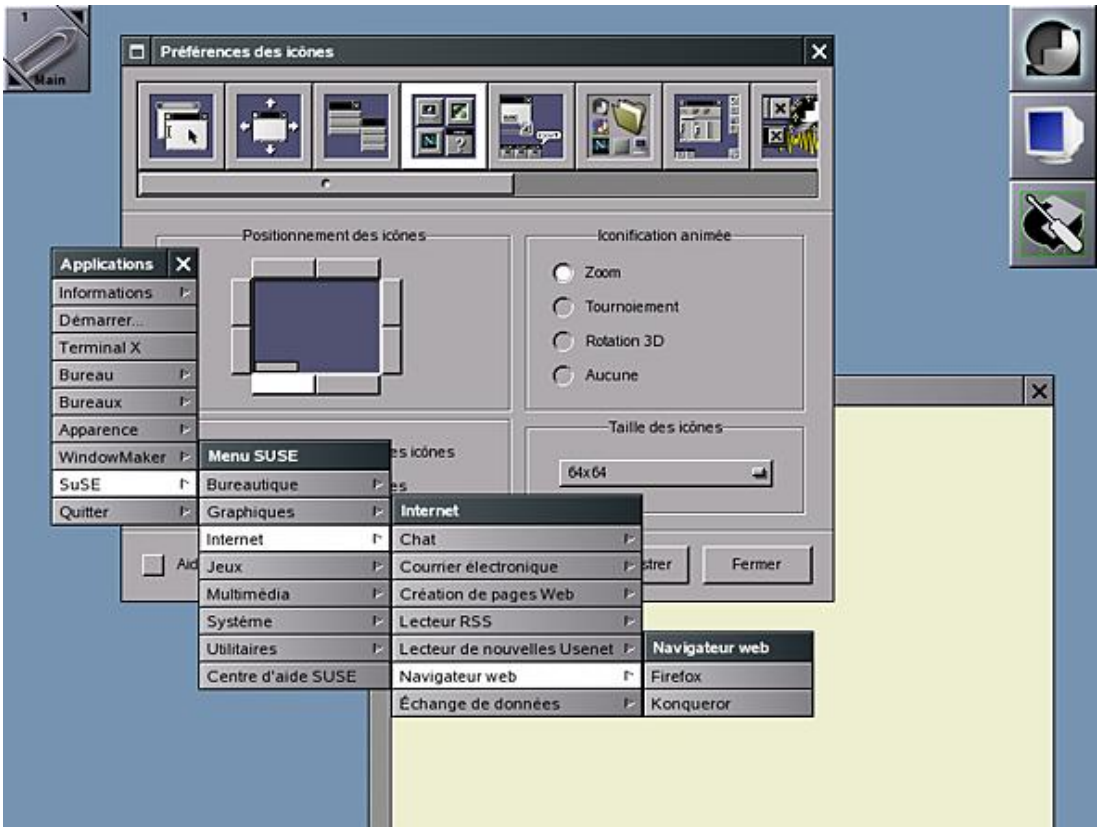
Parmi les requêtes possibles, certaines indiquent de créer une fenêtre et de la décorer en dessinant les divers éléments de celle-ci : la barre de titre, le cadre, les divers boutons. Comme X ne fournit que le nécessaire de base, il dessine la fenêtre mais ce n'est pas lui qui détermine comment doivent être dessinés ces éléments. Un autre programme client X doit dire au serveur comment dessiner la fenêtre : c'est le **gestionnaire de fenêtres** ou *Window Manager*. Le serveur X affiche le résultat dessiné par ce gestionnaire : fenêtres, sélections, déplacements et décorations (styles, couleurs, etc.).



Un gestionnaire simple : twm

Cela veut aussi dire qu'il n'y a pas qu'un seul gestionnaire de fenêtres mais plusieurs. Certains sont très simples et basiques et se limitent au strict minimum, par exemple TWM ici affiché avec une fenêtre et un menu. Notez l'extrême dépouillement. Ce n'est pas le meilleur moyen de faire une démonstration des qualités de Linux auprès de vos amis.

D'autres sont très complets et permettent de travailler dans des conditions très agréables car outre des fenêtres de base ils proposent des thèmes visuels agréables et personnalisables, des menus contextuels et même parfois des panneaux de configuration, comme par exemple WindowMaker. C'est beaucoup mieux, d'autant plus que dans cet exemple précis vous pouvez changer les menus, les thèmes visuels, rajouter des boutons dans la barre de droite, etc. Pendant longtemps ce style de gestionnaire de fenêtre a été le plus utilisé : rapide, performant, peu gourmand en ressources.



WindowMaker, un window manager évolué

c. Les widgets et les toolkits

Les composants et leur bibliothèque

Les gestionnaires de fenêtres les plus développés manquent encore de quelque chose : une intégration plus poussée des logiciels (et de leurs styles) dans l'environnement. C'est que sauf dans des cas spéciaux qui s'amusent à tout redessiner eux-mêmes, chaque **WM** *(Windows Manager)* et logiciel utilisent des bibliothèques (au sens informatique : un ensemble de fonctions) graphiques qui proposent des fonctions toutes faites pour créer des éléments d'interfaces graphiques. Un élément d'interface graphique (bouton, menu, champ de saisie, etc.) s'appelle un **Widget** *(Window gadget)* soit un gadget pour fenêtres. Ce mot signifie aussi *machin*. Une bibliothèque graphique en contient plusieurs.

La bibliothèque contient à la fois les fonctions pour dessiner les widgets et les fonctions pour les gérer. Quand la bibliothèque contient un kit complet et étendu de widget, on la nomme **Widget Toolkit**, ou tout simplement **toolkit** ou boîte à outils pour interface graphique. Il en existe plusieurs que l'on différencie selon leur complexité, leur usage, leur beauté (c'est relatif), le langage de programmation pour les utiliser, etc. X Window dispose d'un toolkit par défaut appelé **Xt** *(X Toolkit)*. Le plus connu et utilisé a longtemps été **MOTIF** d'autant plus qu'il est un standard POSIX IEEE. Cependant, comme il a longtemps été considéré comme propriétaire (non libre jusqu'en 2000) les programmeurs en ont créés d'autres. L'effet a été des plus intéressants : une pléthore de toolkits différents, pas compatibles entre eux, dessinant de manière totalement différente les widgets. Vous pouvez encore trouver aujourd'hui sur Linux des exemples où un programme n'est pas du tout dessiné de la même manière qu'un autre, donnant ainsi une impression de bazar visuel due à des styles très hétérogènes.

GTK, Qt

Ne pouvant pas utiliser Motif (et son équivalent libre Lesstif n'était pas encore très au point), les développeurs d'interfaces graphiques sous Linux ont concentré leurs développements sur deux toolkits différents. Bien programmés, ils permettent de faire totalement abstraction de X Window. Il en existe des versions qui n'ont pas besoin de X Window (pour Windows, Macintosh, les assistants personnels, les téléphones, etc.).

- Le premier se nomme **GTK***(GIMP toolkit)*. Il a été créé pour dessiner l'interface graphique du logiciel de retouche d'images GIMP. Devenant de plus en plus performant, de nombreux programmeurs l'ont repris pour leurs propres programmes et GTK est devenu indépendant. C'est aujourd'hui la bibliothèque par défaut de l'environnement bureautique GNOME. La bibliothèque GTK est programmée en langage C.

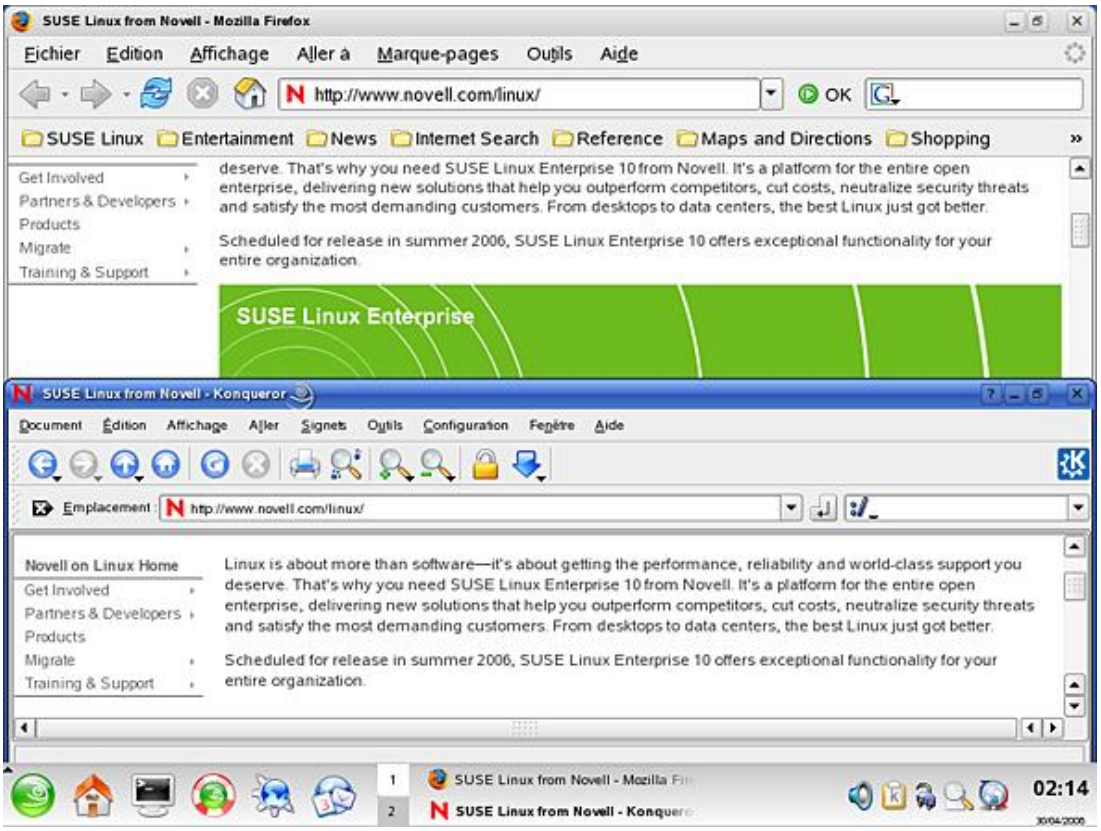
- Le second se nomme **QT***(cute, mignon)*. Le Q ne veut rien dire : les développeurs de ce toolkit trouvaient que

la lettre Q était très jolie sur l'éditeur de textes qu'ils utilisaient… La bibliothèque est développée en langage C++ mais peut être exploitée dans une dizaine de langages. En plus des fonctions de dessin et de gestion des widgets, QT propose un environnement complet de développement d'applications graphiques et non-graphiques : bases de données SQL, XML, multithreading, gestion de fichiers, internationalisation, etc. La bibliothèque QT est celle utilisée par l'environnement bureautique KDE.

- D'autres toolkits existent, dont **MOTIF**, **AWT**, **ATHENA**, et ainsi de suite. Ils sont moins utilisés mais on peut encore rencontrer des applications qui les utilisent.

GTK et QT ne sont pas compatibles, tout au moins nativement. Si vous lancez un programme développé avec GTK sous KDE, le contour de la fenêtre sera dans le style de KDE (QT) mais son contenu sera dessiné par GTK. Regardez le résultat sur la capture suivante. La fenêtre du haut représente le navigateur Internet Firefox qui utilise la bibliothèque GTK tandis que la fenêtre du bas représente le navigateur Konqueror qui utilise la bibliothèque QT. Les menus, les boutons et le contenu ne sont pas identiques. Heureusement SUSE Linux fait bien les choses et propose une homogénéisation des thèmes graphiques. La fin de ce chapitre vous explique d'ailleurs comment rendre votre affichage optimal en terme de thèmes visuels.

Quelques programmes utilisent leurs propres toolkits. C'est le cas de OpenOffice.org et c'est la principale raison de sa lenteur au premier chargement. Quand vous lancez ce programme, il doit aussi charger son toolkit qui fait double emploi avec celui de votre gestionnaire de fenêtres ou de votre environnement bureautique. Heureusement les dernières versions s'adaptent à votre environnement, mais certains programmes (ou plutôt leurs programmeurs) résistent encore.



Deux toolkits, deux styles

d. Les bureaux virtuels

S'il y a une possibilité qui manque chez certains des concurrents de Linux c'est bien le bureau virtuel. Que faire lorsqu'il y a tellement de fenêtres affichées qu'il faut sans arrêt les réduire pour trouver celles masquées, ou les redimensionner pour afficher celles cachées ? Utilisez les bureaux virtuels.

X Window permet l'utilisation de plusieurs espaces de travail dans une même session. Au lieu d'avoir un bureau, vous pouvez en avoir deux, trois, quatre… En fait, vous pouvez en théorie en obtenir autant que vous voulez, si vous disposez d'assez de mémoire. C'est le gestionnaire de fenêtres, ou l'environnement de bureau, qui choisit (et donc vous). KDE permet par exemple d'en disposer de vingt.

L'intérêt est évident. Plutôt que de concentrer toutes vos fenêtres sur un seul bureau, étalez-les sur les bureaux disponibles. D'autant plus qu'il suffit d'un simple clic pour passer d'un bureau à l'autre, ou même pour déplacer une fenêtre d'un bureau à un autre. En plus, il est possible de régler une fenêtre afin qu'elle reste affichée sur l'ensemble

des bureaux (de l'épingler sur l'écran).

2. Les environnements de bureau

Si Unix, et Linux en particulier, ont longtemps conservé une réputation de système d'exploitation pour professionnels, universitaires, informaticiens ou bidouilleurs, c'est qu'il manquait d'un environnement graphique orienté bureautique et convivial. Comme indiqué dans les premiers chapitres, les habitudes que les utilisateurs ont prises avec les interfaces Windows ou MacOS ne peuvent pas se balayer d'un revers de manche. Pendant des années, les gestionnaires de fenêtres sont restés difficiles à configurer. Peut-on demander à un habitué de la souris de modifier un fichier de configuration texte dont la syntaxe n'est pas simple à appréhender à la main ?

En 1996 Linux est parfaitement mûr pour les entreprises mais il reste un marché à conquérir. Depuis l'automne 1995, Windows 95 a mis brutalement fin à la carrière de la plupart des systèmes d'exploitation pour PC de bureau. Les DR-DOS, PC-DOS et surtout OS/2, bien supérieurs, ont végété puis disparu. Les environnements graphiques Unix et Linux ne peuvent absolument pas rivaliser en termes de convivialité et d'offre logicielle grand public avec les systèmes de Microsoft et de Apple. Tout est à faire : il faut rattraper les concurrents, faire mieux puis innover.

Ce travail de titan ne décourage pas certains groupes de programmeurs. Puisque Linux n'est pas assez convivial pour le grand public, il faut créer un environnement bureautique graphique adapté aux besoins. Deux équipes démarrent des projets à quelques mois d'intervalle.

KDE

Logo KDE

En octobre 1996 une première équipe menée par Matthias Ettrich annonce le lancement du projet **KDE***(K Desktop Environment)*. Le K n'a plus de signification particulière (sauf au tout début : *kool*) et est seulement la lettre la plus proche du L de Linux dans l'alphabet. KDE offre une interface graphique et des applications unifiées autour d'un toolkit appelé QT et développées en C++ ; KDE 1 était déjà impressionnant en 1997, permettant d'utiliser Linux comme un mélange de Windows 3.1 et 95. Mais c'est à partir de KDE 2, en 2000, et des choix architecturaux mis en place que KDE a révélé sa puissance, égalant les environnements graphiques concurrents. La sortie de la version 3 (la version 3.5.9 date de février 2008) va encore plus loin et devance tous ses concurrents tant l'intégration des divers composants est poussée. KDE est l'environnement graphique préféré de la majorité des utilisateurs de Linux. KDE se veut le plus complet en termes d'intégration, de configuration et d'offre logicielle (quitte à avoir de nombreux doublons et à laisser le choix à l'utilisateur). KDE est aussi l'environnement de bureau utilisé par Linus Torvalds, le créateur de Linux.

La version 4.0 sortie début 2008 est une refonte totale de l'environnement bureautique KDE. Contrairement à l'évolution constante et continue entre les versions 1.0 et 3.x, KDE4 est une rupture totale par rapport à ses prédécesseurs. L'environnement est basé sur des composants appelés les plasmoïdes. La rupture est tellement totale d'ailleurs que la version 4.0 considérée stable ne l'est pas vraiment à l'écriture de ce chapitre. La version 4.1 annoncée en juillet 2008 est la première pouvant être utilisée dans un environnement de production.

GNOME

Logo Gnome

En août 1997 une équipe menée par Miguel de Icaza et Federico Mena décide de créer un environnement de bureau entièrement libre pour concurrencer KDE (dont QT n'était pas libre à l'époque). **GNOME** *(Gnu Network Object Model Environment)* est un environnement de bureau basé sur le toolkit GTK et programmé en C. C'est l'environnement graphique de bureau officiel du projet GNU. La philosophie de GNOME est radicalement différente de celle de KDE. GNOME est volontairement épuré, se concentrant sur les fonctionnalités essentielles d'un environnement et privilégiant quelques applications au détriment d'autres. L'intégration est moins poussée car les applications développées en GTK ne sont pas toutes des applications GNOME.

Le projet Freedesktop

Logo Freedesktop

Les développeurs de KDE et de GNOME ont décidé de travailler sur une base commune de configuration pour une meilleure intégration des applications GNOME sous KDE, et réciproquement, afin d'harmoniser l'infrastructure de l'ensemble. Le but n'est pas de concevoir un bureau unique car chacun a ses avantages. Le résultat est la création de **Freedesktop**, une zone de communication et de collaboration informelles destinée à travailler à l'interopérabilité des divers environnements graphiques pour Linux et Unix. Les choix de Freedesktop sont technologiquement neutres, aucun environnement n'est privilégié. Mais ce qui sort de Freedesktop est bien souvent intégré dans GNOME et KDE. Du menu principal commun au programme de détection du nouveau matériel, en passant par les copier-coller et les raccourcis-clavier identiques, c'est une foule d'améliorations que Freedesktop a apporté.

Xorg

1. Présentation

Depuis son apparition jusqu'en 2004, Linux (ainsi que les distributions BSD) était accompagné de l'environnement X Window libre XFree86. Un changement de licence rendant XFree86 un peu moins libre et notamment incompatible avec la GPL de la Free Software Foundation eut pour conséquence un "fork" (création d'une nouvelle branche de développement) à partir de la dernière version libre sous licence MIT.

X.org finit par regrouper la plupart des anciens développeurs de XFree86. X.org était un fork de XFree86 datant d'avant le changement de licence, mais avec des idées novatrices largement rejetées par les mainteneurs de XFree86. X.org, appelé de manière commune Xorg, devint, et est toujours, l'environnement X Window principal de Linux.

X.org est rapide, modulaire, disponible avec de nombreux pilotes, compatible. Les dernières versions sont capables d'autodétection du matériel sans disposer de fichier de configuration complexe.

Logo Xorg

La dernière version de Xorg est la 7.3, et la 7.4 est en attente.

2. Installation

Xorg est livré avec toutes les distributions Linux. Les distributions d'avant 2008 sont fournies généralement avec Xorg 7.2. C'est que le développement et les innovations sont si rapides que parfois les outils associés (pour la configuration notamment) ont du mal à suivre.

Vous pouvez récupérer le code source de X.org mais attendez-vous à des heures et des heures de compilation. Vous préférerez installer Xorg depuis les CDs, DVDs ou dépôts d'installation de votre distribution.

Xorg est installé par défaut dans /usr/X11R6. Cependant de nombreuses distributions placent maintenant les binaires, les bibliothèques, les fichiers partagés, etc., dans l'arborescence classique /usr et notamment les modules et les pilotes dans /usr/lib/xorg (ou /usr/lib64/xorg).

La configuration est par contre toujours au même endroit : /etc/X11.

Voici un exemple de packages Xorg installés sur une distribution openSUSE 10.3 en 64 bits (exception faite des packages de développement) :

```
$ rpm -qa|grep -i xorg | grep -v devel
xorg-x11-libXv-7.2-61
xorg-x11-libXrender-32bit-7.2-65
xorg-x11-libXdmcp-7.2-53
xorg-x11-libSM-7.2-58
xorg-x11-libXau-32bit-7.2-54
xorg-x11-libxkbfile-7.2-63
xorg-x11-libXmu-7.2-65
xorg-x11-libXt-32bit-7.2-65
xorg-x11-libXp-7.2-60
xorg-x11-libxcb-7.2-51.2
xorg-x11-libs-7.2-103.4
xorg-x11-libICE-7.2-61
xorg-x11-libSM-32bit-7.2-58
xorg-x11-fonts-core-7.2-85
xorg-x11-libX11-7.2-75
xorg-x11-libICE-32bit-7.2-61
```

```
xorg-x11-libXext-32bit-7.2-65
xorg-x11-fonts-7.2-85
xorg-x11-libXv-32bit-7.2-61
xorg-x11-driver-video-7.2-189.2
xorg-x11-libXrender-7.2-65
xorg-x11-libfontenc-32bit-7.2-59
xorg-x11-libX11-32bit-7.2-75
xorg-x11-server-7.2-143.11
xorg-x11-libXau-7.2-54
xorg-x11-libXt-7.2-65
xorg-x11-libXfixes-32bit-7.2-64
xorg-x11-libxkbfile-32bit-7.2-63
xorg-x11-libXmu-32bit-7.2-65
xorg-x11-Xvnc-7.1-91.2
xorg-x11-libX11-ccache-7.2-88
xorg-x11-libfontenc-7.2-59
xorg-x11-libXext-7.2-65
xorg-x11-libXpm-7.2-65
xorg-x11-libXprintUtil-7.2-60
xorg-x11-libXpm-32bit-7.2-65
xorg-x11-libXp-32bit-7.2-60
xorg-x11-libxcb-32bit-7.2-51.2
xorg-x11-libs-32bit-7.2-103.4
xorg-x11-driver-input-7.2-110
xorg-x11-libXfixes-7.2-64
xorg-x11-libXprintUtil-32bit-7.2-60
xorg-x11-7.2-135.4
```

Notez ici la présence de nombreux packages 32 bits. Il est en effet possible d'avoir à la fois des bibliothèques en 32 et 64 bits, celles-ci étant généralement placées dans des endroits différents. Un programme 32 bits ne fonctionne qu'avec des bibliothèques 32 bits, un programme 64 bits qu'avec des bibliothèques 64 bits.

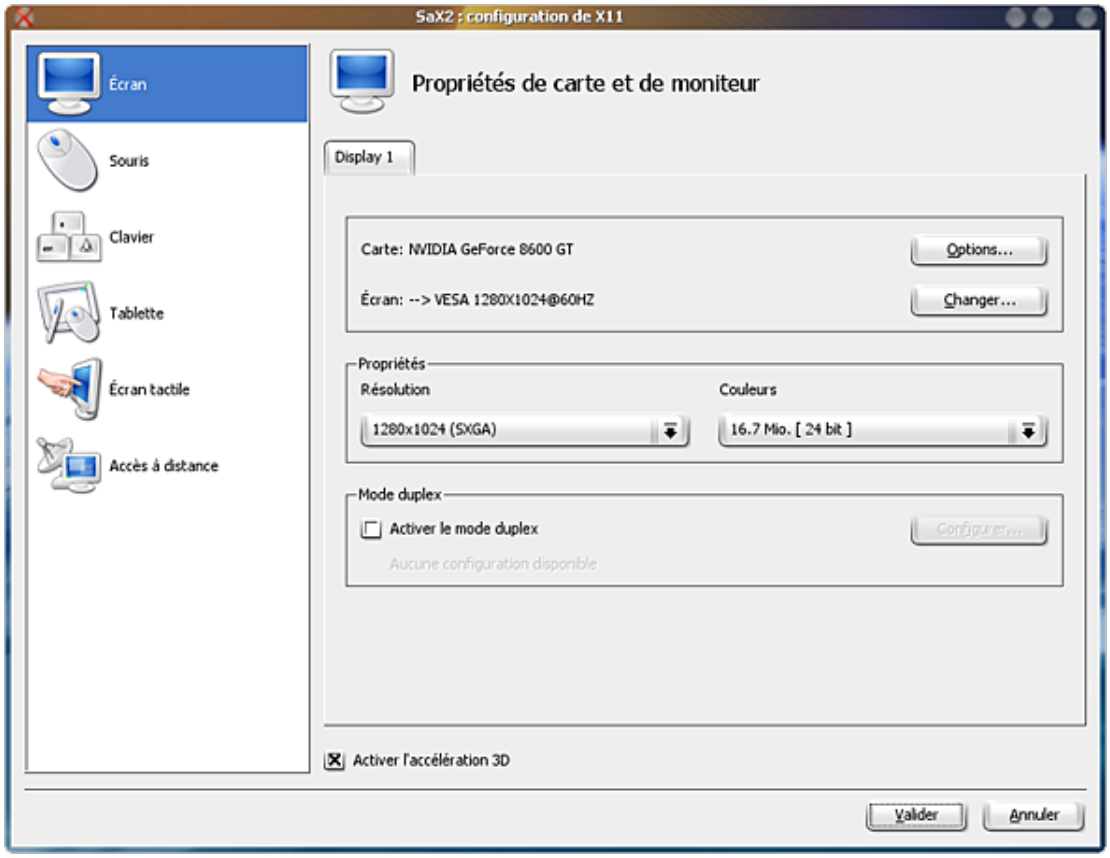
```
$ rpm -ql xorg-x11-libXv-32bit-7.2-61
/usr/lib/libXv.so.1
/usr/lib/libXv.so.1.0.0
$ rpm -ql xorg-x11-libXv-7.2-61
/usr/lib64/libXv.so.1
/usr/lib64/libXv.so.1.0.0
```

3. Configuration

a. Via la distribution

Le fichier de configuration de Xorg est situé dans /etc/X11 et se nomme xorg.conf : /etc/X11/xorg.conf. Il est rare, mais tout à fait possible, de créer un fichier xorg.conf entièrement à la main. Dans la pratique, la force des distributions Linux réside en partie dans leur capacité à détecter votre matériel et à configurer l'environnement graphique en conséquence. De ce fait ce sont ces outils qui construisent, selon vos indications et le matériel détecté, le fichier xorg.conf.

Sur openSUSE, l'outil SaX permet de configurer l'environnement graphique.

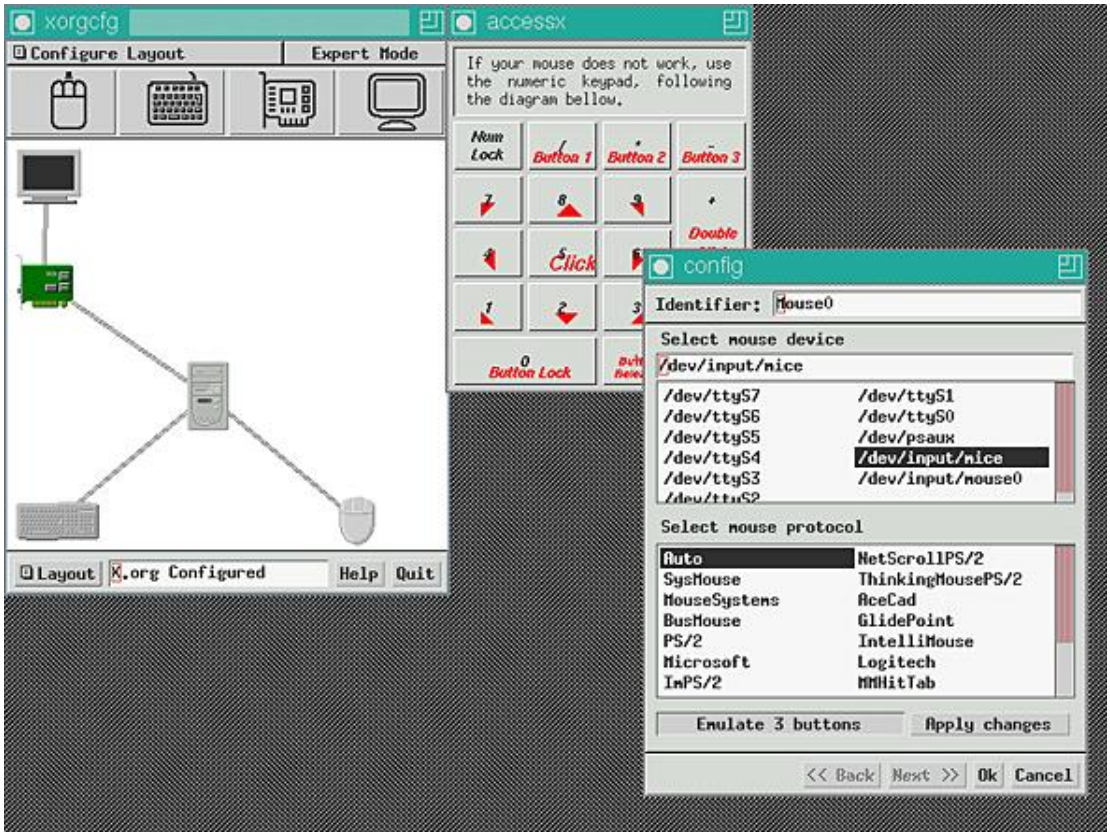


L'outil de configuration X Sax2 de openSUSE

b. Xorgcfg

La commande **xorgcfg** tente de détecter à votre place tous les paramètres de configuration du serveur X et crée un fichier xorg.conf. Dans la pratique, une fois le serveur X disposant d'une configuration de base, X est lancé avec un gestionnaire de fenêtres de base (twm) et un outil graphique permettant de modifier, ou plutôt d'affiner la configuration.

Cependant gardez à l'esprit que c'est un outil de base, qui nécessitera ensuite d'aller modifier la configuration à la main dans le fichier résultant. De plus, les divers tests effectués durant l'écriture de cette partie de l'ouvrage montrent une instabilité de xorgcfg : si vous « tuez » Xorg à ce moment (avec [Alt][Ctrl][Suppr]), l'ordinateur semble mal réagir, et il vous faudra peut-être redémarrer celui-ci via le bouton reset.



Xorgcfg, outil de configuration basique

c. Xorgconfig

L'outil **xorgconfig** est comme xorgcfg mais est basé sur des questions en mode texte. Il est équivalent et remplace l'outil **xf86config** de l'ancien serveur X de XFree86. Vous devez répondre, dans l'ordre, aux questions suivantes :

- Quel est le protocole de communication avec la souris ?
- Voulez-vous une émulation des trois boutons avec les deux autres ?
- Quel est le périphérique de la souris ?
- Quel est votre type de clavier ?
- En quelle langue ?
- A-t-il des options particulières ?
- Quel est votre modèle de moniteur (type générique) ?
- Quelles sont les fréquences de balayage de votre moniteur ?
- De combien de mémoire dispose votre carte graphique ?
- Quelle est la profondeur des couleurs (ex : 24 bits) ?
- Où sauver le fichier ?

Pendant longtemps seul ce genre d'outil a été disponible pour configurer le serveur X. Aujourd'hui les outils des distributions le font très bien. Cependant vous devez tout de même connaître les bases de l'écriture du fichier de

configuration xorg.conf : les modifications manuelles peuvent y être courantes.

4. Structure de xorg.conf

a. Découpage

Le fichier **`/etc/X11/xorg.conf`** est ordonné sous forme de sections et de sous-sections. Chacune correspond soit à une fonctionnalité du serveur X, soit à un périphérique d'entrée ou de sortie.

Pour faire fonctionner X il faut :

- une sortie ; un écran et la carte graphique associée ;
- une entrée ; clavier, souris, etc.

Dans ce dernier cas, l'absence de la souris ne permet pas forcément au système de démarrer. La suite détaille le contenu des différentes sections.

b. Valeurs booléennes

Certaines options de xorg.conf acceptent des valeurs texte ou numérique, mais beaucoup ne prennent que deux valeurs. Dans ce cas plusieurs choix sont possibles :

- Les valeurs 1, on, true et yes sont considérées comme VRAI et toutes identiques.
- Les valeurs 0, off, false, no sont considérées comme FAUX et toutes identiques.

c. Section InputDevice

Une section **InputDevice** décrit un périphérique d'entrée, parmi essentiellement :

- les claviers,
- les souris,
- les touchpads,
- les tablettes graphiques,
- etc.

Il faut une section pour chacun des périphériques d'entrée.

Une section InputDevice est composée de :

- Un identifiant unique, **Identifier**.
- Un pilote, **Driver** (ex : kbd, mouse, evdev, etc.).
- Les diverses options, **Option**, liées au pilote.

Les chemins des fichiers de périphériques sont généralement placés dans la section **Files** sous les entrées **InputDevice**.

Voici par exemple la section qui décrit un clavier. Le pilote se nomme **kbd**. Diverses options sont positionnées, qui servent à décrire le modèle, le type (azerty, qwerty, etc.), le nombre de touches, des options supplémentaires, etc. Le clavier décrit ici est un clavier français de 102 touches.

```
Section "InputDevice"
  Identifier   "Keyboard[0]"
  Driver       "kbd"
  Option       "Protocol" "Standard"
  Option       "XkbLayout" "fr"
  Option       "XkbModel" "pc102"
  Option       "XkbOptions" "caps:shiftlock"
  Option       "XkbRules" "xfree86"
EndSection
```

Vous trouverez de l'aide dans les manpages sous le nom kbd :

- Sous Linux le protocole est toujours **Standard**.
- **XkbModel** définit le modèle de clavier. Les claviers récents prennent **pc105** comme valeur y compris pour les claviers à touches multimédia.
- **XkbLayout** définit la langue du clavier, **fr** pour un clavier français, **us** pour un américain, **be** pour un belge, etc.
- **XkbOptions** fournir des options spécifiques. Par exemple **caps:shiftlock** permet d'obtenir le même fonctionnement qu'un clavier sous Windows : la touche CapsLock sous Linux permet d'accéder en principe aux majuscules accentuées, avec cette option elle accède aux chiffres.

Voici maintenant une section qui correspond à la souris (un modèle Logitech) :

```
Section "InputDevice"
  Identifier   "Mouse[1]"
  Driver       "evdev"
  Option       "Buttons" "10"
  Option       "InputFashion" "Mouse"
  Option       "Name" "Logitech USB R*"
  Option       "Protocol" "event"
  Option       "SendCoreEvents" "on"
  Option       "Vendor" "Sysp"
  Option       "ZAxisMapping" "4 5"
EndSection
```

Vous pouvez remarquer que la structure est identique, seul le pilote et évidemment les options changent. Si X reconnaît la plupart des souris, il faut souvent modifier les options **Buttons** et **ZaxisMapping**. De plus l'interprétation des événements des boutons dépend des applications, environnements de bureaux, etc. et pas de X Window.

Le pilote se nomme ici **evdev**. C'est un pilote générique chargé de gérer tout type d'événements en entrée, tant pour les claviers que les souris, et surtout pour ces dernières car il est un peu plus compliqué à mettre en place pour les claviers. Mais il existe des pilotes spécifiques : **kbd** en est un, **mouse** en est un autre. Pour les portables, le pilote **synpatics** pour les touchpads est souvent utilisé, et même le pilote **wacom** pour les tabletPC et les tablettes graphiques du même nom. Voici une entrée trouvée sur un eeePC de Asus :

```
Section "InputDevice"
    Identifier "Touchpad"
    Driver "synaptics"
    Option "SHMConfig" "on"
EndSection
```

d. Section Monitor

Le section **Monitor** décrit l'écran de votre ordinateur. Les moniteurs sont aujourd'hui tous de type DCC, c'est-à-dire qu'ils sont capables de retourner au pilote graphique, et donc au serveur X, les fréquences et les résolutions qu'ils supportent. Ainsi une section Monitor est parfois réduite à sa plus simple expression.

Cependant quand le modèle n'est pas déterminé (ce qui peut arriver selon le pilote graphique) ou si le moniteur retourne des informations hasardeuses, vous pouvez être amené à choisir un modèle générique, ou à définir vous-même les options de votre moniteur. Pour cela munissez-vous de la documentation de celui-ci et notamment recherchez :

- La taille réelle de l'affichage, largeur et hauteur en millimètres.

- La plage de fréquences horizontales.
- La plage de fréquences verticales.
- Éventuellement, la liste des résolutions supportées.

La section utilisée en exemple décrit un moniteur générique de type Vesa acceptant une résolution de 1280x1024 à 60Hz. Elle fonctionne parfaitement avec l'écran LCD 19 pouces 4/3 de l'auteur.

```
Section "Monitor"
    Identifier    "Monitor[0]"
    VendorName    "--> VESA"
    ModelName     "1280X1024@60HZ"
    UseModes      "Modes[0]"
    DisplaySize   340 270
    HorizSync     31.0 - 64.0
    VertRefresh   50.0 - 60.0
    Option        "DPMS"
EndSection
```

Si votre modèle retourne ses fréquences et les résolutions supportées et que le pilote associé sait les interpréter, vous n'avez pas besoin de toute cette description. La section suivante fonctionne aussi très bien chez l'auteur car le pilote et le moniteur sont en phase.

```
Section "Monitor"
    Identifier   "L1915S"
    ModelName    "L1915S"
    VendorName   "LG"
EndSection
```

La plupart des options parlent d'elles-mêmes. Cependant notez la présence de UseModes qui fait l'objet du prochain point abordé.

e. Section Modes

La section **Modes** est associée à la section Monitor qui y fait parfois appel via la ligne **UseModes**. Certains moniteurs, et pas toujours anciens ou sans marques, nécessitent des réglages spécifiques de fréquences et de timings pour accéder à certaines résolutions d'affichage. Sous Windows, ces réglages sont totalement invisibles : soit le pilote est générique (comme les réglages du premier moniteur du point précédent), soit ces informations sont dans le fichier de description « .inf » fourni avec le moniteur.

Les moniteurs récents (depuis quelques années) n'ont généralement pas besoin de modelines.

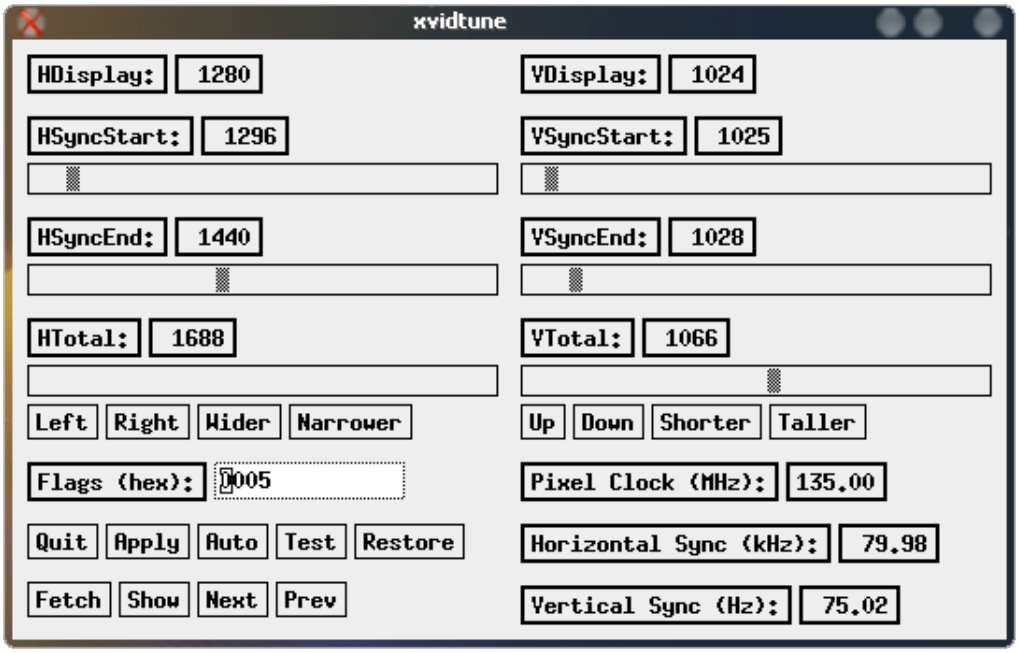
Xorg et la plupart des outils de configuration disposent d'une base de données de moniteurs assez complète, mais il est possible que le vôtre n'y soit pas présent.

```
Section "Modes"
  Identifier   "Modes[0]"
  ModeLine   "1280x1024" 102.6 1280 1312 1472 1632 1024 1028
1032 1048 -hsync -vsync
EndSection
```

Il ne vous sera pas demandé de créer des modelines. Vous pouvez trouver de l'aide aux adresses suivantes :

- http://www.x.org/wiki/FAQVideoModes vous explique notamment comment, via les traces de xorg, créer éventuellement vos propres modelines selon la résolution souhaitée. Encore faut-il que le pilote fournisse ces informations.
- http://xtiming.sourceforge.net/cgi-bin/xtiming.pl vous permet de calculer des modelines selon les spécifications de votre moniteur. Il peut faire des miracles, mais il peut être parfois risqué de l'utiliser.
- http://john.fremlin.de/programs/linux/read-edid/ vous fournit des programmes permettant de décoder les informations EDID de votre moniteur.

- http://gentoo-wiki.com/TIP_Getting_modelines vous explique comment interpréter et utiliser les résultats de la précédente commande.



xvidtune permet de régler finement les résolutions

Si votre moniteur ne semble pas fonctionner à une résolution qu'il est censé accepter (écran noir, mise en veille, X qui quitte avec un message vous disant « no screen found », pas de résolution applicable, etc.) pensez à vérifier au sein du fichier xorg.conf si la section Monitor fait appel à des modelines, isolez la ligne correspondante et commentez-la. Relancez X ([Alt][Ctrl][Suppr]) : il se peut que ce soit l'origine du problème.

Après la configuration de X, vous pourrez tenter d'optimiser l'affichage de votre écran s'il vous pose des problèmes. Les cas classiques (pour des écrans CRT, à tube), sont un affichage excentré, pas assez haut ou large, ou au contraire. Si les réglages de votre moniteur ne permettent pas d'obtenir une bonne image, vous pouvez vous aider de l'outil **xvidtune** qui permet d'influer sur les réglages des modes vidéos.

Attention ! xvidtune peut faire des miracles, mais il peut aussi endommager irrémédiablement votre écran si vous précisez des valeurs de timings et de fréquences farfelues. L'utilisation de cet outil est à vos risques et périls.

---

f. Section Device

La section **Device** décrit le pilote et les options de la carte graphique. Comme il peut y avoir plusieurs cartes graphiques, il peut y avoir plusieurs sections de ce type, avec des identifiants différents. Notez qu'avec certains pilotes, certaines options se placent dans une autre section, **Screen**. L'exemple suivant présente la configuration d'une carte graphique NVidia avec le pilote propriétaire.

```
Section "Device"
  Identifier    "Device[0]"
  Driver        "nvidia"
  VendorName    "NVidia"
  BoardName     "GeForce 8600 GT"
  Option        "NoLogo" "0"
  Option        "DPI" "86 x 86"
  Option        "RenderAccel" "True"
  Option        "AddARGBGLXVisuals" "True"
EndSection
```

Comme pour les autres pilotes, les options dépendent du pilote et ne sont pas toujours les mêmes d'un pilote à un autre. Les options ci-dessus sont propres au pilote Nvidia. La première permet d'éviter ou non l'affichage du logo du constructeur au démarrage, et la seconde force le nombre de DPI de l'affichage. Les deux dernières activent

l'accélération matérielle du rendu (notamment composite) et certaines optimisations pour les extensions OpenGL.

Voici un second exemple avec une carte ATI et le pilote fglrx propriétaire :

```
Section "Device"
  Identifier   "aticonfig-Device[0]"
  Driver       "fglrx"
  Option       "XAANoOffscreenPixmaps" "true"
  Option       "TexturedVideo" "On"
  Option       "UseFastTLS" "1"
  Option       "Textured2D" "on"
  Option       "TexturedXRender" "on"
  Option       "BackingStore" "on"
  Option       "VideoOverlay" "Off"
  Option       "OpenGLOverlay" "Off"
  BusID        "PCI:1:0:0"
EndSection
```

Pour chacune des options, vous devez vous reporter au mode d'emploi du pilote.

g. Section Screen

La section **Screen** est une sorte de métasection : elle fait appel aux sections Monitor et Device pour regrouper tous les paramètres d'affichage : connaissant les capacités tant de la carte graphique que du moniteur, il est alors possible de choisir quelles résolutions graphiques doivent être accessibles, pour quels nombres de couleurs, et quels sont les modes par défaut.

Une section Screen contient une ou plusieurs sous-sections appelées **Display** qui déterminent, pour un type d'affichage en n bits (8 bits : 256 couleurs, 16 bits : 65536 couleurs, 24 bits : 16 millions de couleurs) quelles sont les résolutions qui devraient être accessibles.

- **Depth** définit en nombres de bits la profondeur des couleurs.
- **Modes** les résolutions supportées.
- La première résolution de la liste est la résolution par défaut.
- Vous passez d'une résolution à une autre avec [Ctrl][Alt] + ou - ou via les possibilités offertes par votre environnement de bureau.
- Si une résolution n'est pas supportée elle est automatiquement désactivée.

Dans l'exemple suivant, standard, les précédentes sections Device et Monitor sont regroupées. La section Screen se nomme Screen[0]. Quatre sous-sections Display configurent l'affichage en 8, 15, 16 et 24 bits. Quatre résolutions devraient être accessibles pour chacune, la 1280x1024 l'étant par défaut.

L'entrée **DefaultDepth** précise la profondeur des couleurs par défaut. Dans ce cas, l'affichage sera par défaut en 1280x1024 et 16 millions de couleurs.

```
Section "Screen"
  Identifier    "Screen[0]"
  Device        "Device[0]"
  Monitor       "Monitor[0]"
  DefaultDepth   24
  SubSection    "Display"
    Depth        15
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
    Depth        16
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
    Depth        24
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
```

```
    Depth        8
    Modes        "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
EndSection
```

h. Section ServerLayout

Un fichier xorg.conf dispose d'au moins une section ServerLayout, et généralement d'une seule, qui est la section de base regroupant les sections d'affichage et d'entrée nécessaires pour le lancement de X Window et le fonctionnement de la session. Elle est composée :

- d'un identifiant unique, Identifier ;
- d'une ou plusieurs entrées **Screen** (en cas de multi-head, double écran ou plus) ;
- d'une ou plusieurs entrées **InputDevice** (pour le clavier, la souris, les tablettes, etc.).

X gère très bien les affichages multi-écran et multi-carte via une extension appelée **Xinerama**. Voici un cas simple :

```
Section "ServerLayout"
  Identifier   "Layout[all]"
  Screen       "Screen[0]" 0 0
  InputDevice  "Keyboard[0]" "CoreKeyboard"
  InputDevice  "Mouse[1]" "CorePointer"
EndSection
```

Notez ce qui suit le nommage des sections Screen et InputDevice :

- Les **0 0** après Screen fournissent les informations de position de l'écran en cas de dual ou de multi-head. Il s'agit des positions X et Y en partant des coins en haut à gauche.
- Le **CoreKeyboard** indique que ce périphérique d'entrée est le clavier principal. Il ne peut y en avoir qu'un. S'il est absent, X recherche tout type de périphérique d'entrée pouvant en faire office.
- Le **CorePointer** est identique, mais pour le pointeur, c'est-à-dire la souris.

Si vous disposez de plusieurs souris (ex : souris USB et touchpad) ou claviers (ex : un clavier USB branché sur la base d'un portable, et le clavier du portable), les autres périphériques d'entrée peuvent être déclarés dans la section ServerLayout, mais dans ce cas écrivez "**SendCoreEvents**" à la suite : cela permet au périphérique d'envoyer des événements au serveur X comme les autres. X pourra tous les gérer :

```
Section "ServerLayout"
  Identifier    "Layout[all]"
  Screen        "Screen[0]" 0 0
  InputDevice   "Keyboard[0]" "CoreKeyboard"
  InputDevice   "Mouse[1]" "CorePointer"
  InputDevice   "Touchpad" "SendCoreEvents"
EndSection
```

i. Section Files

La section Files indique au serveur X les chemins vers certains fichiers ou répertoires qui peuvent lui être nécessaires :

- les périphériques d'entrée,
- les polices de caractères,
- les modules complémentaires,
- la base des codes couleurs RVB (un peu comme les codes HTML).

Certains chemins n'ont pas besoin d'être précisés car ils sont définis par défaut. En fait, en théorie depuis la version 7.3 aucun chemin n'est nécessaire, mais il peut être utile d'en préciser d'autres méconnus. Les entrées sont les suivantes :

- **InputDevices** : le chemin des périphériques d'entrée ;
- **FontPath** : le chemin des polices de caractères ;
- **RGBPath** : le chemin du fichier des codes RGB ;
- **ModulePath** : le chemin des modules complémentaires de Xorg.

L'exemple suivant, issu d'un fichier de configuration Xorg 7.2, précise les chemins des périphériques et des polices.

```
Section "Files"
  InputDevices    "/dev/gpmdata"
  InputDevices    "/dev/input/mice"
  FontPath      "/usr/share/fonts/misc:unscaled"
  FontPath      "/usr/share/fonts/75dpi:unscaled"
  FontPath      "/usr/share/fonts/100dpi:unscaled"
  FontPath      "/usr/share/fonts/Type1"
  FontPath      "/usr/share/fonts/URW"
  FontPath      "/usr/share/fonts/Speedo"
  FontPath      "/usr/share/fonts/cyrillic"
  FontPath      "/usr/share/fonts/truetype"
  FontPath      "/usr/share/fonts/uni:unscaled"
  FontPath      "/opt/kde3/share/fonts"
  FontPath      "/usr/local/share/fonts"
EndSection
```

La section Files est facultative. Des chemins par défaut, codés en dur, sont généralement respectés par les éditeurs. Notamment les polices de caractères sont automatiquement recherchées dans /usr/lib/X11/fonts. Cependant, les scripts de lancement de X sont souvent modifiés par l'éditeur de la distribution pour fournir d'autres chemins. Sur les dernières Mandriva par exemple, les polices sont toutes dans /usr/share/fonts.

Les versions récentes de Xorg vont automatiquement chercher les chemins dans le répertoire /etc/X11/fontpath.d. Les chemins sont des liens symboliques vers les répertoires correspondants de /usr/share/fonts et reprennent le format des entrées **FontPath**.

j. Section Modules

La section Modules fournit au serveur X une liste de modules complémentaires et optionnels à charger pour lui ajouter des nouvelles fonctionnalités. Un module est déclaré avec une ligne **Load**.

```
Section "Module"
  Load     "dbe"
  Load     "type1"
  Load     "freetype"
  Load     "extmod"
  Load     "glx"
EndSection
```

Voici une liste de modules courants. Ils sont pour la plupart optionnels mais certains composants (périphériques d'entrée, applications, cartes graphiques) peuvent nécessiter certaines extensions pour fonctionner. Comme faire fonctionner OpenOffice.org sans le support des polices de caractère par exemple ?

- **dbe** *(Double Buffer Extension)* deux tampons d'affichage. Un principal qui correspond à l'affichage actuel, et un autre pour préparer l'affichage en arrière-plan. Une fois qu'il est prêt les deux tampons sont commutés. La vitesse est accélérée et aucun artéfact n'est présent.
- **extmod** : module d'extension du protocole X, que tout le monde ou presque utilise. À terme il devrait être incorporé au sein même de X.
- **freetype** : permet d'utiliser les polices de caractères TrueType (ttf).

- **type1** : permet d'utiliser les polices Type1.
- **bitmap** : permet d'utiliser les polices bitmap (inutile pour les dernières versions).
- **GLcore** : mode de base de rajout des extensions OpenGL.
- **glx** : extensions GLX (extension à OpenGL).
- **dri** *(Direct Rendering Infrastructure)* : OpenGL fait appel aux fonctions matérielles de la carte graphique, l'affichage 3D est grandement accéléré.
- **i2c** : mise en place du bus série i2c, pour communiquer entre autres avec le moniteur.
- **ddc** : prise en charge du protocole DDC pour les moniteurs (*Display Data Channel*), qui passe par le bus i2c.
- **int10** : couche d'émulation/accès en mode réel à l'interruption 10 de la carte graphique, notamment pour accéder aux fonctionnalités VESA de la carte et du moniteur
- **vbe** *(Vesa Bios Extension)* : extensions Vesa pour les accès à certains modes et résolutions de la carte.

k. Section ServerFlags

La section **ServerFlags** permet de définir les options globales du serveur X Window. Les options sont multiples mais en voici quelques-unes pertinentes :

- **DontZap** : si vrai, désactive la séquence Alt-Ctrl-Backspace (qui tue le serveur X).
- **DontVTSwitch** : si vrai, désactive l'accès aux consoles via Alt-Ctrl-Fn.
- **DontZoom** : si vrai, interdit de changer de résolution par Alt-Ctrl-+/-.
- **AllowMouseOpenFail** : si vrai, X se charge même en l'absence de souris.
- **XkbDisable** : si vrai, désactive le clavier.
- **Xinerama** : active ou non le support dual/multi-head.
- **AIGLX** : active ou non le support AIGLX (dépend du pilote).
- **BlankTime** : durée en minutes par défaut de l'activité de l'économiseur d'écran avant de passer dans le mode Stand by de l'économie d'énergie.
- **StandbyTime** : en minutes, durée de la phase Stand by de l'économie d'énergie DPMS.
- **SuspendTime** : idem mais pour Suspend.
- **Offtime** : idem mais pour l'extinction.

L'exemple suivant montre une section qui permet de démarrer X sans souris (rien n'empêche d'utiliser ensuite une souris USB ou sans fil), avec interdiction de tuer le serveur, de changer de résolution et de passer sur une console. C'est la configuration par défaut d'une borne Internet sous Linux.

```
Section "ServerFlags"
  Option    "AllowMouseOpenFail" "on"
  Option    "DontZap" "on"
  Option    "DontZoom" "on"
  Option    "DontVTSwitch" "on"
```

```
EndSection
```

I. Section Extensions

Section optionnelle, Extensions permet d'activer ou de désactiver des extensions de Xorg. Les plus connues sont Damage et Composite. Elles vont de pair.

L'extension Damage permet de signaler aux fenêtres qu'une partie de leur affichage doit être redessiné. Si les extensions composites sont activées (effet de transparence par exemple) une zone de l'écran peut devoir être redessinée alors qu'elle est couverte par une fenêtre dont le contenu n'a pas changé mais qui peut être transparente : c'est celle d'en dessous qui a changé. L'extension Damage signale le changement. Elle est en principe activée par défaut.

L'option composite est celle qui permet d'avoir les effets du même nom : ombres, transparence des fenêtres, alpha-blending, etc. Associée à la 3D, notamment AIGLX, vous obtenez avec les effets proposés par Compiz-Fusion : cubes 3D, effets de bureau. Il n'est pas toujours nécessaire d'avoir une machine et une carte puissantes : d'anciennes cartes fonctionnent très bien.

```
Section "Extensions"
  Option     "Composite" "Enable"
EndSection
```

5. xorg.conf complet

Voici le fichier xorg.conf complet :

- Carte Nvidia.
- Écran LG 19 pouces.
- Résolution de 1280x1024.
- 16 millions de couleurs.
- Clavier français 105 touches.
- Souris 10 boutons.
- Extensions composites activées.

```
Section "ServerLayout"
  Identifier     "Layout[all]"
  Screen         "Screen[0]" 0 0
  InputDevice    "Keyboard[0]" "CoreKeyboard"
  InputDevice    "Mouse[1]" "CorePointer"
EndSection

Section "Files"
  InputDevices   "/dev/gpmdata"
  InputDevices   "/dev/input/mice"
  FontPath     "/usr/share/fonts/misc:unscaled"
  FontPath     "/usr/share/fonts/75dpi:unscaled"
  FontPath     "/usr/share/fonts/100dpi:unscaled"
  FontPath     "/usr/share/fonts/Type1"
  FontPath     "/usr/share/fonts/URW"
  FontPath     "/usr/share/fonts/Speedo"
  FontPath     "/usr/share/fonts/cyrillic"
  FontPath     "/usr/share/fonts/truetype"
  FontPath     "/usr/share/fonts/uni:unscaled"
  FontPath     "/opt/kde3/share/fonts"
  FontPath     "/usr/local/share/fonts"
EndSection
```

```
Section "Module"
  Load         "dbe"
  Load         "type1"
  Load         "freetype"
  Load         "extmod"
  Load         "glx"
EndSection

Section "ServerFlags"
  Option       "AllowMouseOpenFail" "on"
  Option       "DontZap" "on"
  Option       "DontZoom" "on"
  Option       "DontVTSwitch" "on"
EndSection

Section "InputDevice"
  Identifier   "Keyboard[0]"
  Driver       "kbd"
  Option       "Protocol" "Standard"
  Option       "XkbLayout" "fr"
  Option       "XkbModel" "pc105"
  Option       "XkbOptions" "caps:shiftlock"
  Option       "XkbRules" "xfree86"
EndSection

Section "InputDevice"
  Identifier   "Mouse[1]"
  Driver       "evdev"
  Option       "Buttons" "10"
  Option       "InputFashion" "Mouse"
  Option       "Name" "Logitech USB R*"
  Option       "Protocol" "event"
  Option       "SendCoreEvents" "on"
  Option       "Vendor" "Sysp"
  Option       "ZAxisMapping" "4 5"
EndSection

Section "Modes"
  Identifier     "Modes[0]"
  ModeLine    "1280x1024" 102.6 1280 1312 1472 1632 1024 1028
1032 1048 -hsync -vsync
EndSection

Section "Monitor"
  Identifier   "Monitor[0]"
  VendorName   "--> VESA"
  ModelName    "1280X1024@60HZ"
  UseModes     "Modes[0]"
  DisplaySize   340 270
  HorizSync     31.0 - 64.0
  VertRefresh   50.0 - 60.0
  Option       "DPMS"
EndSection

Section "Device"
  Identifier   "Device[0]"
  Driver       "nvidia"
  VendorName   "NVidia"
  BoardName    "GeForce 8600 GT"
  Option       "RenderAccel" "True"
  Option       "AddARGBGLXVisuals" "True"
EndSection

Section "Screen"
  Identifier   "Screen[0]"
  Device       "Device[0]"
  Monitor      "Monitor[0]"
  DefaultDepth  24
```

```
  SubSection   "Display"
    Depth       15
    Modes      "1280x1024"
  EndSubSection
  SubSection   "Display"
    Depth       16
    Modes      "1280x1024"
  EndSubSection
  SubSection   "Display"
    Depth       24
    Modes      "1280x1024"
  EndSubSection
  SubSection   "Display"
    Depth       8
     Modes     "1280x1024"
  EndSubSection
EndSection

Section "Extensions"
    Option    "Composite" "Enable"
EndSection
```

6. Tester et lancer X

a. Vérifier la configuration

Une fois votre configuration X terminée, il est temps de tester le serveur et de le lancer. Pour cela il faut évidemment que vous ne soyez pas déjà en mode graphique. Passez en mode texte (init 2 ou 3 selon votre distribution) et tapez :

```
$ X -probeonly

X Window System Version 7.2.0
Release Date : TRue Jan 22 17:08:26 UTC 2008-05-31
X Protocol Version 11, Revision 0, Release 7.2
Build Operating System: openSUSE SUSE LINUX
Current Operating System : Linux opensuse 2.6.22.17-0.1-default #1
SMP 2008/02/10 20:01:04 UTC i686
Build Date: 22 Junuary 2008
    Before reporting problems, check http://wiki.x.org
    To make sure that you have the lastest version.
Module Loader present
Markers: (--) probed, (**) from config file, (==) default setting,
    (++) from command line, (!!) notice, (II) informational,
    (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: “/var/log/Xorg.0.log”, Time: Sat May 31 10:16:06 2008
(==) Using config file: ”/etc/X11/xorg.conf”
(II) Module already build-in
```

Aucune erreur n’a été détectée. Souvent, si une erreur survient, vous obtenez ceci :

```
(EE) No drivers available.

Fatal server error:
no screens found
```

Dans ce cas :

- Soit le fichier **`/etc/X11/xorg.conf`** contient une erreur de syntaxe.
- Soit un périphérique est mal configuré : écran absent, aucune résolution supportée, mauvais pilote, etc.

L’analyse des traces est alors d’un grand secours.

b. Les traces

Les traces sont présentes dans le fichier **/var/log/Xorg.0.log**. Elles sont souvent très longues : avec la configuration ci-dessus, les traces contiennent 600 lignes, et d'autres peuvent se rajouter en cours de fonctionnement. Les traces contiennent tous les détails du chargement et de la configuration de X Window.

En cas de problème, les dernières lignes sont généralement très parlantes et indiquent où se situe le problème. Si X fonctionne mais ne réagit pas comme indiqué, alors vous devrez fouiller plus en détails dans le fichier. Voici un exemple de la trace avec quelques sorties intéressantes, notamment sur la carte graphique et le clavier. Notez en gras les détections effectuées. Xorg a détecté une carte Nvidia 8600GT, sa version du BIOS, le bus PCI express, la quantité de mémoire, et la marque et le modèle du moniteur.

```
X Window System Version 7.2.0
Release Date: Tue Jan 22 17:03:42 UTC 2008
X Protocol Version 11, Revision 0, Release 7.2
Build Operating System: openSUSE SUSE LINUX
...
(==) Using config file: "/etc/X11/xorg.conf"
(==) ServerLayout "Layout[all]"
(**) |-->Screen "Screen[0]" (0)
(**) | |-->Monitor "Monitor[0]"
(**) | |-->Device "Device[0]"
(**) |-->Input Device "Keyboard[0]"
(**) |-->Input Device "Mouse[1]"
(II) No default mouse found, adding one
(**) |-->Input Device "<default pointer>"
(**) FontPath set to:
    /usr/share/fonts/misc:unscaled,
    /usr/share/fonts/75dpi:unscaled,
    /usr/share/fonts/100dpi:unscaled,
    /usr/share/fonts/Type1,
    /usr/share/fonts/URW,
    /usr/share/fonts/Speedo,
    /usr/share/fonts/cyrillic,
    /usr/share/fonts/truetype,
    /usr/share/fonts/uni:unscaled,
    /opt/kde3/share/fonts,
    /usr/local/share/fonts
(==) RgbPath set to "/usr/share/X11/rgb"
(**) Input device list set to "/dev/gpmdata,/dev/input/mice"
(==) ModulePath set to
"/usr/lib64/xorg/modules/updates,/usr/lib64/xorg/modules"
(**) Option "AllowMouseOpenFail" "on"
(**) Extension "Composite" is enabled
(II) Open ACPI successful (/var/run/acpid.socket)
(II) Loader magic: 0x7c46e0
(II) Module ABI versions:
    X.Org ANSI C Emulation: 0.3
    X.Org Video Driver: 1.2
    X.Org XInput driver : 0.7
    X.Org Server Extension : 0.3
    X.Org Font Renderer : 0.5
(II) Loader running on linux
(II) LoadModule: "pcidata"
(II) Loading /usr/lib64/xorg/modules//libpcidata.so
(II) Module pcidata: vendor="X.Org Foundation"
    compiled for 7.2.0, module version = 1.0.0
    ABI class: X.Org Video Driver, version 1.2
(++) using VT number 7
...
(II) Setting vga for screen 0.
(**) NVIDIA(0): Depth 24, (--) framebuffer bpp 32
(==) NVIDIA(0): RGB weight 888
(==) NVIDIA(0): Default visual is TrueColor
(==) NVIDIA(0): Using gamma correction (1.0, 1.0, 1.0)
(**) NVIDIA(0): Option "RenderAccel" "True"
(**) NVIDIA(0): Option "AddARGBGLXVisuals" "True"
(**) NVIDIA(0): Enabling RENDER acceleration
(II) NVIDIA(0): Support for GLX with the Damage and Composite X
```

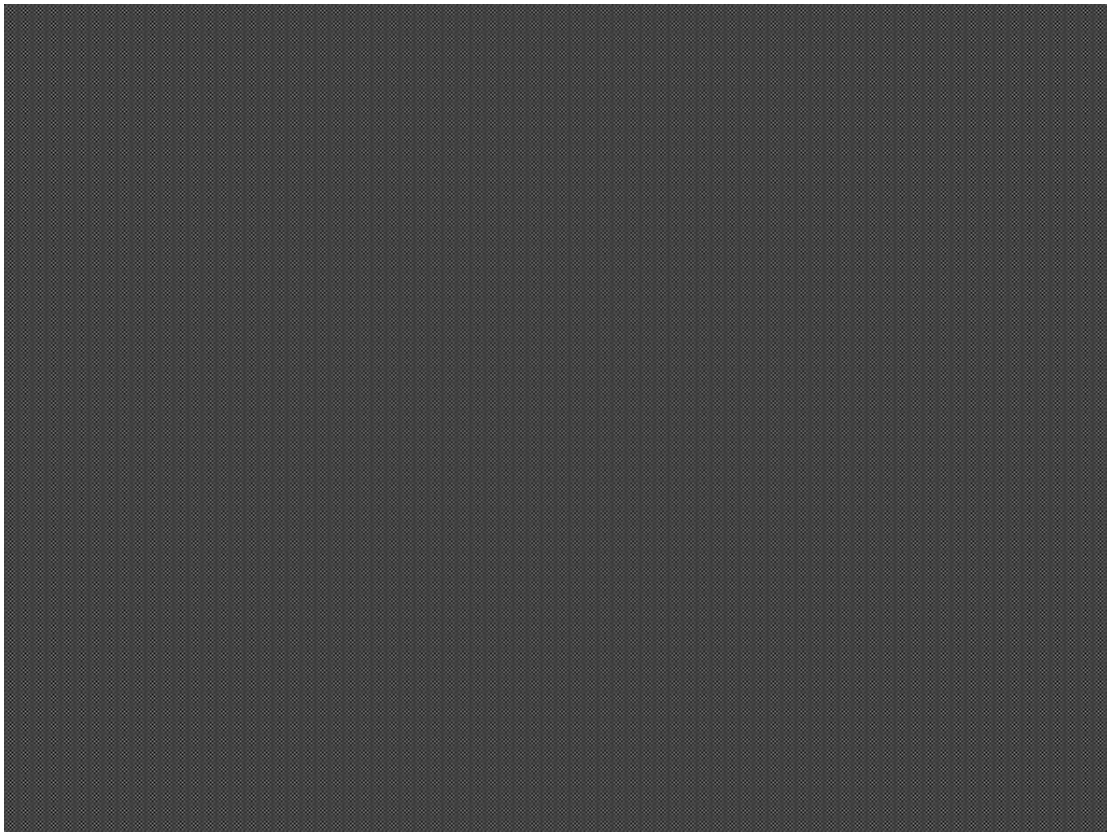
```
extensions is
(II) NVIDIA(0): enabled.
(II) NVIDIA(0): NVIDIA GPU GeForce 8600 GT (G84) at PCI:1:0:0 (GPU-0)
(--) NVIDIA(0): Memory: 262144 kBytes
(--) NVIDIA(0): VideoBIOS: 60.84.35.00.11
(II) NVIDIA(0): Detected PCI Express Link width: 16X
(--) NVIDIA(0): Interlaced video modes are supported on this GPU
(--) NVIDIA(0): Connected display device(s) on GeForce 8600 GT at
PCI:1:0:0:
(--) NVIDIA(0): LG L1915S (CRT-0)
(--) NVIDIA(0): LG L1915S (CRT-0): 400.0 MHz maximum pixel clock
(II) NVIDIA(0): Assigned Display Device: CRT-0
(II) NVIDIA(0): Validated modes:
(II) NVIDIA(0): "1280x1024"
(II) NVIDIA(0): Virtual screen size determined to be 1280 x 1024
(--) NVIDIA(0): DPI set to (85, 86); computed from "UseEdidDpi" X
config
(--) NVIDIA(0): option
(**) NVIDIA(0): Enabling 32-bit ARGB GLX visuals.
(--) Depth 24 pixmap format is 32 bpp
...
(II) NVIDIA(0): Initialized GPU GART.
(II) NVIDIA(0): Setting mode "1280x1024"
(II) Loading extension NV-GLX
(II) NVIDIA(0): NVIDIA 3D Acceleration Architecture Initialized
(II) NVIDIA(0): Using the NVIDIA 2D acceleration architecture
(==) NVIDIA(0): Backing store disabled
(==) NVIDIA(0): Silken mouse enabled
...
**) Option "CoreKeyboard"
(**) Keyboard[0]: Core Keyboard
(**) Option "Protocol" "Standard"
(**) Keyboard[0]: Protocol: Standard
(**) Option "AutoRepeat" "500 30"
(**) Option "XkbRules" "xfree86"
(**) Keyboard[0]: XkbRules: "xfree86"
(**) Option "XkbModel" "pc102"
(**) Keyboard[0]: XkbModel: "pc102"
(**) Option "XkbLayout" "fr"
(**) Keyboard[0]: XkbLayout: "fr"
...
```

c. Tester le serveur

Testez le serveur X en le lançant avec son nom :

```
$ X
```

Si X fonctionne, vous devez voir un écran gris, en fait une succession de points noirs et blancs, et une croix au milieu qui est le curseur de la souris. Bougez la souris. Si elle fonctionne, c'est que votre serveur est bien configuré.



Un écran gris : X fonctionne

Tuez ensuite le serveur X avec [Ctrl][Alt][Retour arrière] car aucun gestionnaire de fenêtres n'étant démarré il est impossible d'en faire quoi que ce soit.

Le Display Manager

1. Principe

Le Display Manager, ou gestionnaire d'affichage, est un élément de X Window qui se charge de la connexion des utilisateurs, locaux ou distants, de leur authentification, puis qui charge leur environnement de travail en leur ouvrant une session. Il gère un ensemble d'affichages X, qui peuvent être locaux ou distants.

Pour les connexions distantes, le Display Manager s'appuie sur le protocole standard XDMCP : X Display Manager Control Protocol.

Le Display Manager est en gros l'équivalent graphique des services proposés par init, getty et login : il demande des identifiants et mots de passe, authentifie les personnes et ouvre une session.

Le gestionnaire par défaut se nomme XDM : X Display Manager. Son style graphique n'est pas très attrayant mais il est léger et fonctionne avec tous les serveurs X.

Une session X peut être vue comme une session de console : c'est la durée de vie du processus d'un utilisateur après la connexion. Sous une console c'est le shell, sous X c'est un « session manager », généralement le gestionnaire de fenêtre (ou un processus de l'environnement bureautique) ou un terminal graphique. Quand ce processus est fermé (déconnexion du terminal, sortie de l'environnement bureautique, etc.) la session se termine et la boîte de connexion de XDM (ou tout autre Display Manager) est réaffichée.

Il est possible d'ouvrir plusieurs sessions X depuis un seul Display Manager, y compris sur une même machine.

Linux dispose de plusieurs Display Manager mais trois sont principalement utilisés :

- XDM : version de base.
- GDM : version proposée par GNOME.
- KDM : version proposée par KDE.

Les versions GNOME et KDE sont évoluées : elles proposent la même chose que XDM avec des fonctionnalités supplémentaires :

- liste des utilisateurs ;
- icônes (avatars) associées ;
- choix d'une session graphique particulière (Gnome, fvwm, KDE, etc.) ;
- possibilité d'autoconnexion ;
- thèmes graphiques attrayants ;
- liste des serveurs X distants (XDMCP) ;
- passage d'un utilisateur à un autre ;
- etc.

2. XDM

a. Configuration générale

La configuration de XDM se situe dans /etc/X11/xdm. Le premier fichier chargé est xdm-config. Toutes ses lignes sont du type :

```
DisplayManager.ressource: valeur.
```

Chaque ligne représente une ressource de XDM. La formation LPI1 ne demande pas de connaître par cœur la configuration de XDM mais quelques éléments de base. Notamment xdm-config charge d'autres fichiers. Dans l'exemple suivant, les lignes en gras montrent quelques autres scripts de configurations appelés, soit interprétés directement par xdm, soit exécutés avant ou après la connexion :

```
!! xdm-config: Configuration of the xdm
!
DisplayManager.errorLogFile:        /var/log/xdm.errors
DisplayManager.pidFile:            /var/run/xdm.pid
DisplayManager.authDir:             /var/lib/xdm
DisplayManager.keyFile:             /etc/X11/xdm/xdm-keys
DisplayManager.servers:/etc/X11/xdm/Xservers
DisplayManager.accessFile:/etc/X11/xdm/Xaccess
DisplayManager.willing:             su nobody -c /etc/X11/xdm/Xwilling
!
! ATTENTION: `authName' should be in general MIT-MAGIC-COOKIE-1
! For XDM-AUTHENTICATION-1 which is default for xterminals see
! manual page of xdm and the manual coming with the xterminal.
!
DisplayManager.*.authName:          MIT-MAGIC-COOKIE-1
DisplayManager.*.authComplain:      false
!
! All displays should use authorization, but we cannot be sure
! X terminals will be configured that way, so by default
! use authorization only for local displays :0, :1, etc.
!
DisplayManager._0.authorize:        true
DisplayManager._1.authorize:        true
DisplayManager._93.authorize:       true
!
! The scripts handling the setup, the startup, the session its self,
! and the reset of an X session.
!
DisplayManager.*.setup:              /etc/X11/xdm/Xsetup
DisplayManager.*.chooser:            /etc/X11/xdm/RunChooser
DisplayManager.*.startup:            /etc/X11/xdm/Xstartup
DisplayManager.*.session:            /etc/X11/xdm/Xsession
DisplayManager.*.reset:              /etc/X11/xdm/Xreset
!
DisplayManager._0.terminateServer:   true
DisplayManager._93.terminateServer:  true
!
DisplayManager*resources: /etc/X11/xdm/Xresources
DisplayManager.*.terminateServer:    false
!
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
!
!DisplayManager.requestPort:         0
```

b. Setup : Xsetup

Le fichier **/etc/X11/xdm/Xsetup** ou tout autre associé à l'entrée **DisplayManager.*.setup** est exécuté avant l'affichage de la boîte de login. C'est dans ce fichier que vous pouvez modifier la présentation de l'affichage et de Xdm, afficher de nouvelles fenêtres (ex : une vue des traces systèmes, une horloge, etc.). Le contenu de ce fichier dépend de l'éditeur de la distribution. L'usage le plus courant consiste à :

- modifier le fond d'écran avec la commande xsetroot ;
- modifier les couleurs, avec xrdb ;
- changer par les ressources X la géométrie (les positions et tailles) des boîtes de dialogue ;

- afficher une console avec xconsole ;
- activer le clavier numérique avec numlockx ;
- modifier le clavier avec xmodmap ;
- etc.

Voici un exemple épuré issu encore une fois de openSUSE. Les lignes en gras montrent les actions pour xdm :

```
#
# Handle background:
#   First kdm/gdm choise, then xdm/user choise and
#   if no choise is given use the system defaults.
# Choix du fond d'écran
if test "$kdm" = "yes" -o "$gdm" = "yes" ; then
    : # $xsetroot -solid '#738dc6'
elif test -s ${background}.gz -a -x $xpmroot ; then
    $xpmroot ${background}.gz
elif test -s ${background} -a -x $xpmroot ; then
    $xpmroot $background
elif test -x $backprg ; then
    $backprg
else
    $xsetroot -gray
fi

#
# Enable Numlock if set
# Activation du clavier numérique
if test -r /var/run/numlock-on && type -p numlockx > /dev/null ;
then
    numlockx on
fi

#
# Xresources
# Modification des couleurs et de la géométrie
if test "$kdm" != "yes" -a "$gdm" != "yes" ; then
    $xrdb -override -retain <<-EOF
      #ifdef COLOR
      *Foreground: black
      *Background: #cdd2b4
      #endif
      #if (WIDTH < 320)
      XConsole*geometry:   125x80+0-0
      #elif (WIDTH < 400)
      XConsole*geometry:   130x85+0-0
      #elif (WIDTH < 640)
      XConsole*geometry:   150x90+0-0
      #elif (WIDTH < 800)
      XConsole*geometry:   240x95+0-0
      #elif (WIDTH < 1024)
      XConsole*geometry:   300x100+0-0
      #elif (WIDTH < 1152)
      XConsole*geometry:   384x110+0-0
      #elif (WIDTH < 1280)
      XConsole*geometry:   432x120+0-0
      #else
      XConsole*geometry:   480x130+0-0
      #endif
      EOF
fi
if test "$kdm" = "yes" -o "$gdm" = "yes" ; then
    $xrdb -override -retain ${ETCDIR}/Xresourcesf
i
```

```
#
# The geometry of xconsole is set in the Xresource file.
# Lancement d'une console
(
    exec setsid $xconsole -notify -nostdin -verbose -exitOnFail
) & echo $! > /var/run/xconsole.pid
```

c. Chooser : RunChooser

Le fichier **/etc/X11/xdm/RunChooser** ou tout autre associé à l'entrée **DisplayManager.*.chooser** permet d'afficher la boîte de dialogue des serveurs X distants pour s'y connecter. Il ressemble au fichier Xsetup mais se limite au paramétrage et au lancement du programme appelé **chooser** (/usr/X11R6/bin/chooser, /usr/lib/X11/chooser ou tout autre programme ayant le même rôle). Ce programme n'a d'intérêt que si le serveur X distant accepte les connexions via XDMCP.

d. Startup : Xstartup

Le fichier **/etc/X11/xdm/Xstartup** ou tout autre associé à l'entrée **DisplayManager.*.startup** est le premier exécuté, s'il est présent, après la réussite de l'authentification. Il est exécuté par root avec les droits de root. Ne surtout pas le confondre avec le prochain fichier (session). Il sert notamment à :

- effacer l'écran ;
- écrire les informations de connexion dans les fichiers /var/log adéquats ;
- vérifier si la connexion est locale ou distante ;
- vérifier si l'utilisateur est autorisé à se connecter ;
- etc.

e. Session : Xsession

Le fichier **/etc/X11/xdm/Xsession** ou tout autre associé à l'entrée **DisplayManager.*.session** est le premier exécuté, s'il est présent, après la réussite de l'authentification, avec les droits de l'utilisateur authentifié. S'il est absent X va exécuter une commande **Xterm**, qui sera donc le processus de session de l'utilisateur. D'ailleurs si la session ne peut être démarrée correctement, ce fichier contient souvent le nécessaire pour lancer une console. Après une configuration de base, Xsession tente de lancer dans cet ordre les fichiers suivants :

- le fichier $HOME/.xsession, propre à l'utilisateur ;
- s'il est absent, le fichier $HOME/.xinitrc, propre à l'utilisateur ;
- s'il est absent, le fichier /etc/X11/xdm/sys.xsession ;
- s'il est absent, le fichier /etc/X11/xinit/xinitrc.

En pratique, un drapeau (flag) peut être présent pour sauter le fonctionnement par défaut et forcer directement le chargement d'un Window Manager (gestionnaire de fenêtres) et notamment KDE ou GNOME, sans passer par les fichiers précédents. Les environnements évolués prennent en charge ensuite leur propre configuration.

```
#
# Forced X session type if the user asked for
# an other session environment.
#
if test "$forced" = "yes" ; then
    unset WINDOW_MANAGER STARTUP
     test -x $syssess && exec_login "$syssess"
     exec_login "/bin/bash $syssess"
fi
```

```
# User login X session
# If the user doesn't have their own xsession, then run
# system xsession or xinitrc script if they exist

if test -f $session ; then
   test -x $session && exec_login "$session"
   exec_login "/bin/bash $session"
elif test -f $xinitrc ; then
   test -x $xinitrc && exec_login "$xinitrc"
   exec_login "/bin/bash $xinitrc"
elif test -f $syssess; then
   test -x $syssess && exec_login "$syssess"
   exec_login "/bin/bash $syssess"
elif test -f $sysinit ; then
   test -x $sysinit && exec_login "$sysinit"
   exec_login "/bin/bash $sysinit"
elif test -n "$WINDOWMANAGER" ; then
   unset WINDOW_MANAGER STARTUP
   exec_login "$WINDOWMANAGER"
fi
```

f. Reset : Xreset

Le fichier **/etc/X11/xdm/Xreset** ou tout autre associé à l'entrée **DisplayManager.*.reset** est exécuté lorsque l'utilisateur ferme sa session. Vous y trouverez notamment l'écriture des traces de déconnexion.

g. Resources : Xresources

Le fichier **/etc/X11/xdm/Xresources** ou tout autre fichier associé à l'entrée **DisplayManager*resources** est interprété par xdm et contient les définitions des ressources visuelles utilisées par xdm comme les polices de caractères, les messages d'accueil ou d'échec, les couleurs, l'adaptation de l'affichage en fonction du nombre de couleurs et de la taille de l'écran, etc.

C'est notamment dans ce fichier que vous pouvez modifier le message d'accueil ou d'échec :

```
xlogin*titleMessage:   Xlogin
xlogin*greetColor:   darkgray
xlogin*promptColor:  darkgray
xlogin*failColor:    red
xlogin*greeting:       Bienvenue sur CLIENTHOST
xlogin*fail:           -- Connexion refusée -
```

Avec une telle configuration l'utilisateur sera accueilli avec le message « Bienvenue sur HOSTNAME » affiché en gris foncé. S'il n'est pas authentifié, le message « -- Connexion refusée » est affiché en rouge.

h. Servers : Xservers

Le fichier **/etc/X11/xdm/Xservers** ou tout autre fichier associé à l'entrée **DisplayManager.servers** fournit la liste des spécifications des serveurs locaux de X, ou plutôt de ceux qui ne nécessitent pas de connexion via XDMCP. La ligne suivante indique que le premier serveur :0 est local, son binaire est /usr/bin/X avec comme option -br vt7 (occupe le terminal virtuel vt7).

```
:0 local /usr/bin/X -br vt7
```

C'est dans cette ligne que vous pouvez, si vous le souhaitez, modifier le nombre de couleurs du serveur X au lancement de xdm en lui passant les bons arguments. Pour basculer en 256 couleurs :

```
:0 local /usr/bin/X -br vt7 -depth 8
```

i. AccessFile : Xaccess et XDMCP

Le fichier **/etc/X11/xdm/Xaccess** ou tout autre fichier associé à l'entrée **DisplayManager.accessFile** fournit la liste des hôtes autorisés à se connecter via XDMCP à votre serveur X. Pour autoriser les connexions vous devez passer

par deux étapes :

- Dans **/etc/X11/xdm/xdm-config**, modifiez comme ceci la ligne suivante qui est en principe commentée (retirez le point d'exclamation devant) :

```
$ grep request /etc/X11/xdm/xdm-config
...
DisplayManager.requestPort: 0
```

- Dans **/etc/X11/xdm/Xaccess** décommentez la ligne qui commence par une étoile

  ```
  * CHOOSER BROADCAST
  ```

  Ou encore tout simplement

  ```
  *
  ```

  Sans rien d'autre. Cela permet à tous les hôtes de se connecter chez vous.

3. Gdm et kdm

Gdm et **kdm** sont respectivement les gestionnaires d'affichage de GNOME et de KDE.

Pourquoi avoir présenté si longuement xdm alors que ces deux produits sont bien plus jolis, pratiques, performants (indiquez ici la liste des superlatifs souhaités) ? C'est que, sauf bien entendu pour toutes les ressources graphiques, gdm et kdm utilisent (bien que ce ne soit pas une obligation) les mêmes fichiers que xdm dont :

- Xsetup
- Xstartup
- Xsession
- Xreset
- Xresources
- Xaccess
- etc.

GDM, le Display Manager de Gnome

La configuration avancée de gdm et kdm passe donc par la configuration des fichiers de XDM. Par contre, pour tout le reste, vous pouvez modifier les fichiers propres à gdm et xdm, ou plutôt si les modifications sont simples, passer par l'interface graphique associée.

- La configuration de KDM se situe dans **/etc/X11/kdm/kdmrc** ou dans **<prefix-kde>/share/config/kdmrc**, par exemple sur openSUSE le fichier se situe dans **/opt/kde3/share/config/kdm/kdmrc**.

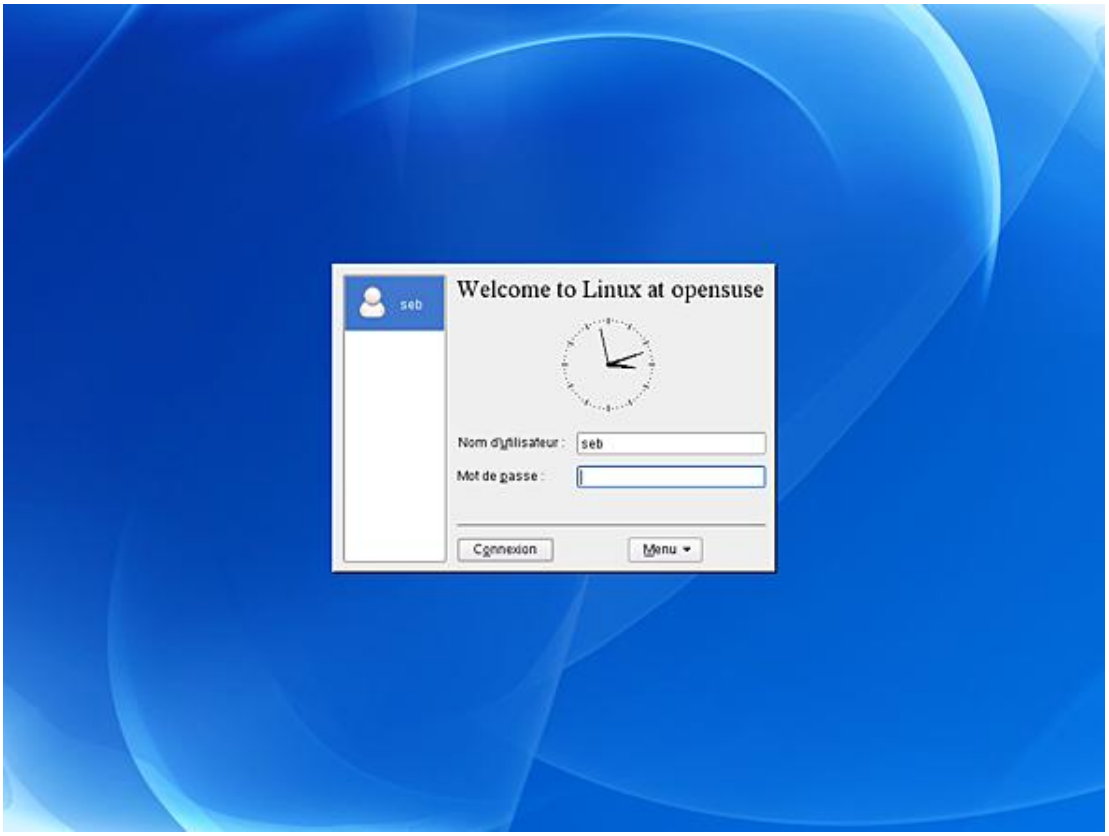
- La configuration de GDM se situe dans **/etc/X11/gdm/gdm.conf**, ou **/etc/gdm/gdm.conf** ou encore dans **/usr/share/gdm/gdm.conf**.

Bien que gdm et kdm utilisent par défaut les options des fichiers de XDM, vous pouvez en modifier la configuration de manière à en faire une totale abstraction. Ainsi par exemple pour kdm vous pouvez activer XDMCP ainsi :

```
[Xdmcp]
Enable=true
Port=177
KeyFile=/opt/kde3/share/config/kdm/kdmkeys
Xaccess=/etc/X11/xdm/Xaccess
```

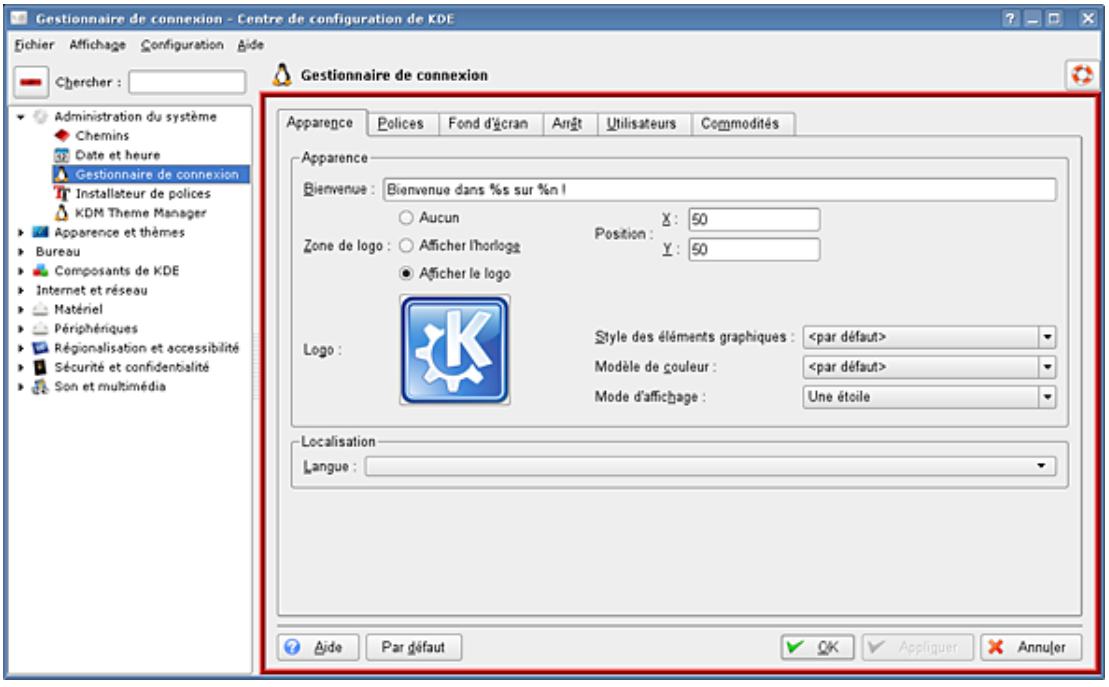
Une autre (fausse) bonne idée est d'autoriser l'export des fenêtres X vers votre serveur X (affichage distant) via la commande **xhost+** en supprimant le paramètre **nolisten** de la configuration. Commentez la ligne suivante :

```
ServerArgsLocal=-nolisten tcp
```



KDM, le Display Manager de KDE

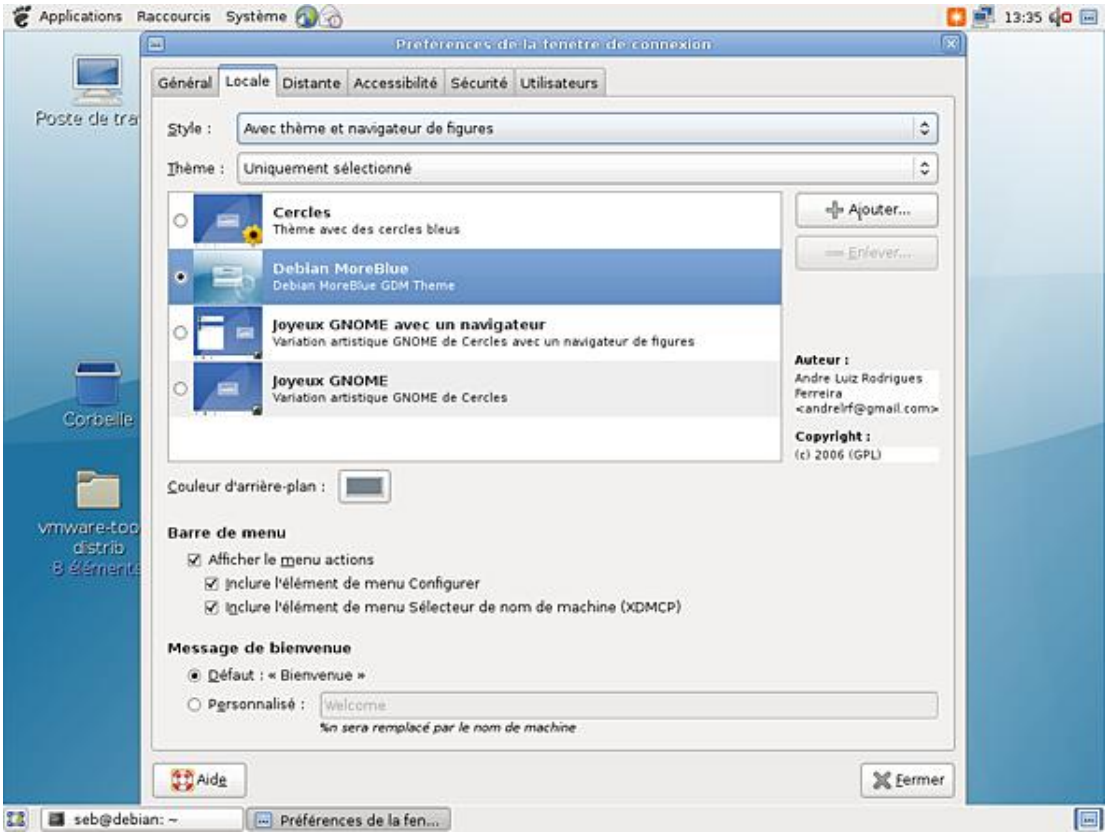
Pour le reste, passez par l'outil de configuration graphique. Lancez le **Centre de configuration de KDE**, et dépliez l'entrée **Administration du système**. Cliquez sur **Gestionnaire de connexion**. En tant que simple utilisateur vous ne pourrez rien faire : cliquez sur le bouton **Mode superutilisateur**, en bas.



Personnalisation de KDM

De là, vous pouvez entièrement configurer kdm.

Pour gdm, exécutez depuis le menu la commande gdmsetup, qui effectue la même chose. L'exemple suivant provient d'une installation Debian Etch de base.



Personnalisation de GDM

4. xdm, gdm ou kdm au boot

a. inittab

Pour démarrer X dès le boot, et donc passer par un display manager pour vous connecter, vous avez deux solutions, les deux pouvant être proposées par diverses distributions :

- passer par **/etc/inittab** ;
- ou lancer xdm comme service.

Dans le premier cas, éditez votre fichier **/etc/inittab** et recherchez une ligne qui ressemble à l'une des lignes suivantes. Vous trouverez aussi bien les lignes once que respawn, la seconde étant préférable :

```
xdm:5:once:/usr/X11R6/bin/xdm
xdm:5:respawn:/usr/X11R6/bin/xdm
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Si une ligne équivalente, ou ressemblante, est présente et que vous continuez à accéder à une connexion en mode texte, c'est que :

- soit X est mal configuré et donc après quelques tentatives il rend la main à la console. Dans ce cas l'écran devrait clignoter plusieurs fois à ce moment ;
- soit vous êtes dans le mauvais niveau d'exécution par défaut.

Dans ce dernier cas, vérifiez et modifiez la ligne suivante pour démarrer en niveau 5 (pour les distributions rpm) par défaut :

```
id:5:initdefault:
```

et passez en tant que root au niveau d'exécution 5 :

```
# telinit 5
```

b. Service

Si aucune ligne n'apparaît, xdm est peut-être lancé en tant que service. Vérifiez ceci en le recherchant dans **/etc/init.d** :

```
$ ls /etc/init.d/xdm
/etc/init.d/xdm
```

Vérifiez la configuration du niveau associé via **rcupdate.d** ou **chkconfig**, ou encore à la main :

```
# chkconfig --list xdm
xdm     0:off 1:off 2:off 3:off 4:off 5:on 6:off
# ls -l /etc/rc.d/rc?.d/S*xdm
lrwxrwxrwx 1 root root 6 mai 14 12:22 /etc/rc.d/rc5.d/S10xdm ->
../xdm
```

c. /etc/sysconfig

Sur plusieurs distributions le choix et les réglages par défaut de plusieurs gestionnaires d'affichage sont placés dans des fichiers de configuration au sein de **/etc/sysconfig**. Sous Mandriva par exemple il s'agit du fichier **/etc/sysconfig/desktop**.

```
$ cat /etc/sysconfig/desktop
DISPLAYMANAGER=kdm
```

Sous openSUSE, les réglages sont plus précis (notamment pour la gestion de la résolution et des thèmes) et le fichier est **/etc/sysconfig/displaymanager**. Vous y retrouvez notamment des réglages qui se substituent à ceux de xdm concernant les messages à afficher, les accès XDMCP, etc.

```
$ cat /etc/sysconfig/displaymanager | grep -v ^#
DISPLAYMANAGER_XSERVER=Xorg
DISPLAYMANAGER_XGL_OPTS="-accel glx:pbuffer -accel xv:pbuffer"
DISPLAYMANAGER="kdm"
DISPLAYMANAGER_REMOTE_ACCESS="no"
DISPLAYMANAGER_ROOT_LOGIN_REMOTE="no"
DISPLAYMANAGER_STARTS_XSERVER="yes"
DISPLAYMANAGER_XSERVER_TCP_PORT_6000_OPEN="no"
DISPLAYMANAGER_AUTOLOGIN=""
DISPLAYMANAGER_PASSWORD_LESS_LOGIN="no"
DISPLAYMANAGER_AD_INTEGRATION="no"
DISPLAYMANAGER_SHUTDOWN="auto"
DISPLAYMANAGER_RANDR_MODE_VGA="auto"
DISPLAYMANAGER_RANDR_MODE_auto="1024x768_60 64.11 1024 1080 1184
1344 768 769 772 795 -HSync +Vsync"
KDM_USERS=""
KDM_GREETSTRING=""
DISPLAYMANAGER_KDM_THEME="SUSE"
```

Window Manager et environnement personnel

1. Via le display manager

Vous déterminez généralement le type d'environnement graphique chargé par la session X depuis gdm ou kdm (il est rare d'utiliser xdm) et le menu associé. Ceux-ci se rappellent de la session précédente comme session par défaut grâce à la présence du fichier **.dmrc** dans votre répertoire personnel :

```
$ pwd
/home/seb
$ cat .dmrc

[Desktop]
Session=kde
```

Vous remarquerez, si vous utilisez l'un ou l'autre des deux gestionnaires d'affichage que la liste des sessions possibles est partagée. C'est l'un des avantages de Freedesktop : les fichiers de description des sessions sont placés dans **/usr/share/xsessions** (la description du fichier kdmrc indique d'autres positions possibles).

```
$ pwd
/usr/share/xsessions
$ ls -ltotal 20
-rw-r--r-- 1 root root 7262 oct 4 2007 fvwm2.desktop
-rw-r--r-- 1 root root 6796 mai 28 13:44 kde.desktop
-rw-r--r-- 1 root root 3305 nov 28 2007 twm.desktop
```

Chaque fichier **desktop** représente une session possible. Les descriptions et les commandes de lancement sont présentes dans chacun des fichiers, et notamment via les lignes Exec et Name :

```
$ grep -E "Exec|Name=" *.desktop
fvwm2.desktop:Exec=fvwm
fvwm2.desktop:TryExec=fvwm
fvwm2.desktop:Name=FVWM
kde.desktop:Exec=/opt/kde3/bin/startkde
kde.desktop:TryExec=/opt/kde3/bin/startkde
kde.desktop:Name=KDE3
twm.desktop:Exec=twm
twm.desktop:TryExec=twm
twm.desktop:Name=TWM
```

Quand vous sélectionnez une session particulière dans gdm ou kdm, celui-ci liste les sessions dans **/usr/share/xsessions** en propose la liste. Quand l'utilisateur se connecte, son choix est écrit comme session par défaut dans son fichier **.dmrc** puis la commande correspondant à la ligne **Exec** est exécutée.

2. Startx

Si vous démarrez votre ordinateur sans que xdm ou un autre display manager ne démarre, la commande **startx** permet de lancer le serveur X et l'environnement associé.

```
$ startx
```

S'il n'y avait aucun fichier de configuration, seul X, un écran gris avec un curseur de souris sans aucune action possible, serait affiché.

Quand vous démarrez X avec startx, il cherche à exécuter dans cet ordre :

- le fichier **$HOME/.xinitrc** ;
- **/etc/X11/xinit/xinitrc**.

Si vous devriez éviter de toucher le fichier **/etc/X11/xinit/xinitrc** pour qu'il reste fidèle au standard défini par votre distribution, vous pouvez modifier à volonté votre fichier personnel **.xinitrc**. Généralement, vous pouvez récupérer

le premier qui peut servir de modèle au votre. Voici un exemple de **.xinitrc** sur Mandriva :

```
$ cat xinitrc
#!/bin/sh

if [ "`whoami`" != root ]; then
  xsetroot -solid "#21449C"
fi

exec /etc/X11/Xsession $*
```

La dernière ligne lance Xsession, qui va rechercher un gestionnaire de fenêtres à lancer. Mais un simple fichier concernant une seule ligne peut suffire à démarrer KDE, toujours sur Mandriva :

```
$ cat $HOME/.xinitrc
/usr/bin/startkde
```

Sur openSUSE :

```
/opt/kde3/bin/startkde
```

Et pour lancer GNOME :

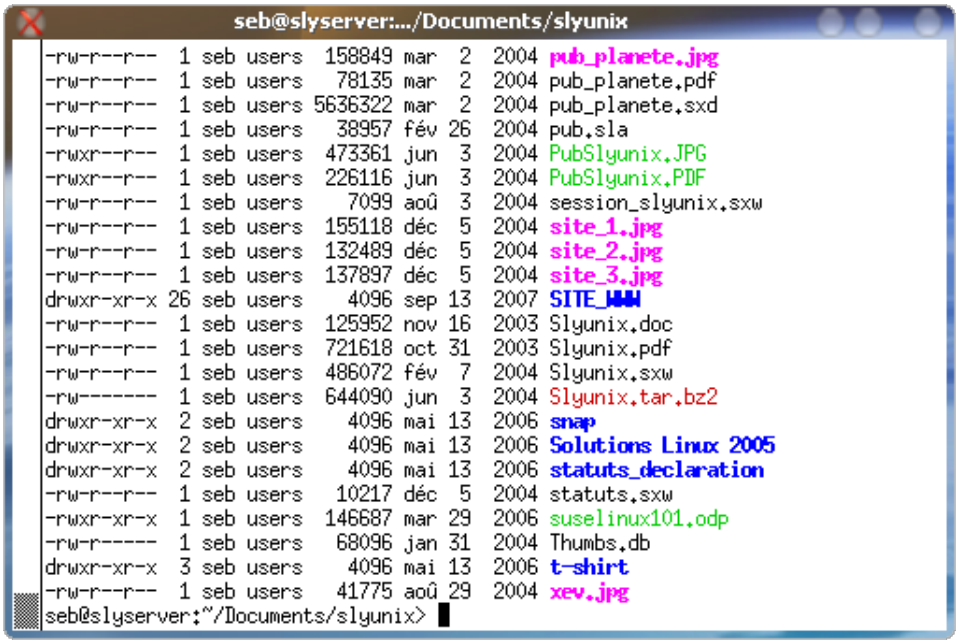
```
exec gnome-session
```

3. Les terminaux

Pour accéder à la ligne de commande Sous X vous devez utiliser un émulateur de terminal. Ils sont nombreux. Lequel choisir ?

Tout d'abord l'émulateur de base du serveur X se nomme **xterm**. Quel que soit votre environnement de travail X il sera toujours présent. C'est pourquoi c'est souvent lui qui représente la session par défaut en cas de problème sous X.

```
$ xterm &
```



Xterm, le terminal virtuel classique

Pour configurer Xterm, vous devez utiliser les boutons de la souris et la touche [Ctrl] :

- [Ctrl] + bouton gauche : options principales ;

- [Ctrl] + bouton droit : choix des polices de caractère ;
- [Ctrl] + bouton du milieu : options du terminal.

Il existe d'autres terminaux qui peuvent vous plaire :

- **rxvt** : un terminal très léger, tout petit, qui ressemble à Xterm mais sans ses menus de configuration. Notez que vous devez utiliser une version spéciale appelée urxvt si votre distribution est unicode.
- **aterm** : un terminal proche de rxvt, prévu pour le gestionnaire Afterstep, qui accepte la pseudo transparence du fond.
- **eterm** : un terminal prévu pour remplacer xterm (comme tous les terminaux d'ailleurs) qui a été l'un des premiers à proposer des menus. Il supporte les onglets dans ses dernières versions.
- **konsole** : le terminal livré avec KDE : support des profils, de divers fonds, styles dont la transparence, et les onglets. Il n'est pas le plus léger en taille mais un de ceux les plus légers en ressources et des plus rapides.
- **gnome-terminal** : le terminal livré avec GNOME. Il propose la même chose que celui de KDE, ou serait-ce l'inverse…
- etc.

Vous pouvez choisir le terminal que vous souhaitez, selon vos goûts et quel que soit votre environnement de travail. L'essentiel est ce que vous en faites : saisir des commandes, exécuter et créer des scripts, etc. Gardez seulement à l'esprit que choisir un terminal, ou tout autre produit associé à l'environnement que vous utilisez peut être un gain de ressources, dont la mémoire. En effet en chargeant konsole depuis GNOME, cela marche très bien mais les composants de base de KDE sont aussi chargés.

4. Les gestionnaires de fenêtres

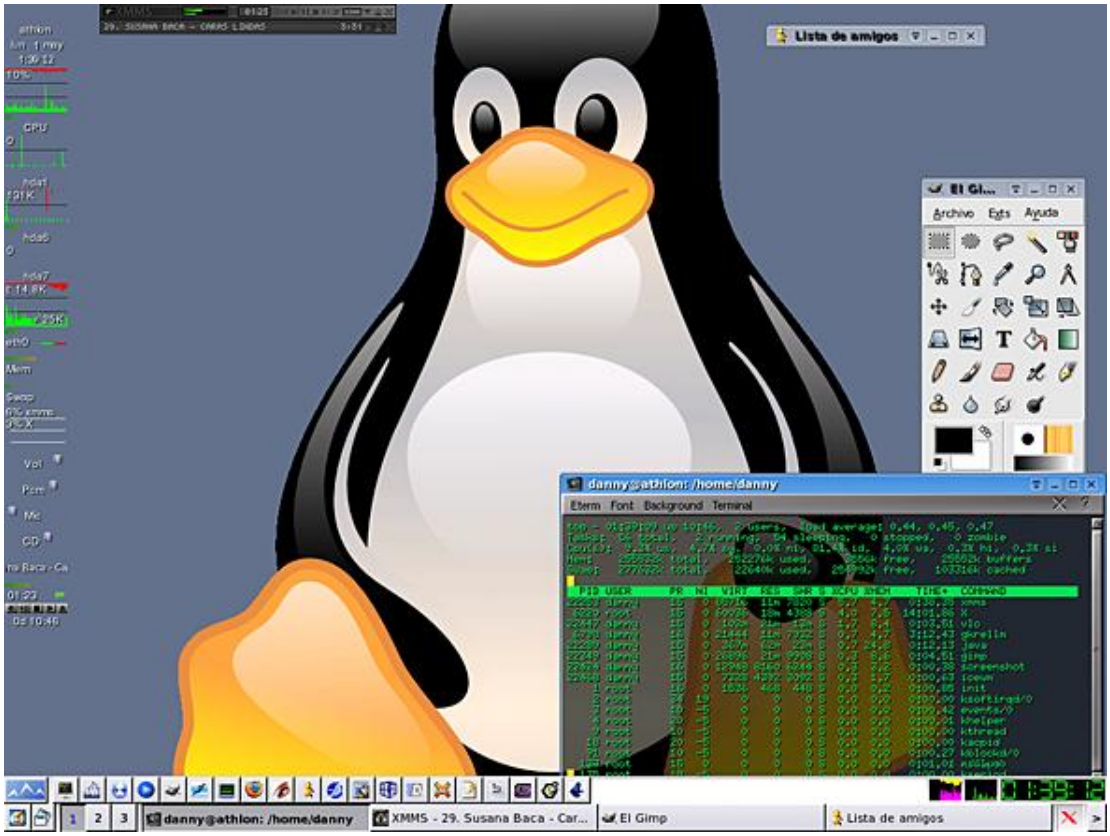
Le principe des gestionnaires de fenêtre a été abordé en début de chapitre. Vous pouvez choisir un environnement selon vos goûts et vos besoins. Voici une liste des gestionnaires de fenêtres et d'environnements de bureau les plus connus.

a. twm

Le seul gestionnaire de fenêtre officiel de X Window depuis la version X11R4 est TWM : Tom's Window Manager. C'est le seul qui soit livré par défaut quelle que soit la version de X. En version de base, telle que vous avez pu le voir dans la capture au début de ce chapitre, twm est minimal, et complètement dépassé par les derniers environnements. Cependant il est extrêmement configurable, fournit des barres de titres, des icônes, etc. Sa configuration est réalisée par des fichiers textes.

b. IceWM

Créé pour être visuellement agréable et très léger, il est possible d'appliquer à IceWM des thèmes et de le faire ressembler à d'autres environnements comme Windows, OS/2 ou celui que vous voulez, si le thème est disponible. IceWM est notamment utilisé par défaut avec les eeePC de Asus (ceux tournant sous Linux). Sa configuration utilise des fichiers textes mais plusieurs panneaux de contrôles existent.



IceWM. Capture sous licence GPL provenant de Wikimedia Commons

c. Fvwm

Fvwm est un dérivé de Twm. Officiellement figé en version 1.24 en 1994, d'autres évolutions sont sorties, et une version fvwm95 en son temps imitait la barre des tâches et les décorations de fenêtres de Windows 95 et 98. Il est très configurable et à l'origine de plusieurs autres gestionnaires dont Afterstep (prévu pour ressembler à NextStep), XFce, et Enlightenment. Bien configuré (fichiers textes, mais des frontends existent) il est superbe et léger, et il continue d'être utilisé encore aujourd'hui, et fourni et supporté en standard par la plupart des éditeurs de distributions.

d. CDE

Common Desktop Environment, CDE, est un environnement de bureau basé sur Motif et provenant à l'origine de openVMS de HP. C'est un environnement proposé à l'origine par The Open Group. Il aurait pu devenir un standard sur tous les Unix (il est livré par défaut sur Solaris, HPUX, True64, etc.), mais étant propriétaire, et Motif ayant été libéré sur le tard, les plates-formes libres se sont tournées vers les autres, dont KDE et GNOME.

e. WindowMaker

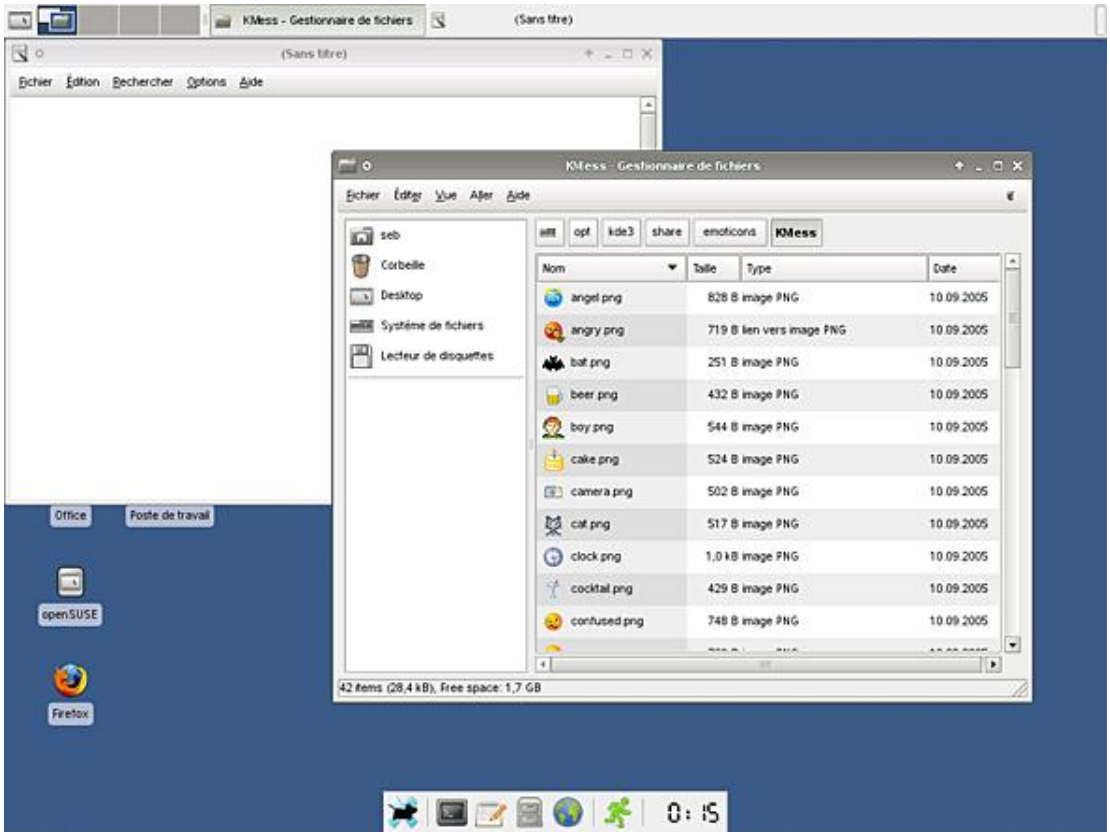
WindowMaker, dont une capture est présente au début du chapitre, est un gestionnaire de fenêtres très évolué développé pour ressembler à NextStep et fonctionner avec GNUstep, l'environnement compatible OpenSTEP des machines NeXT de Steves Jobs. Il est à l'origine une version améliorée de AfterStep. Il est stable, fiable, très bien fini, facile à configurer, et reste relativement léger.

f. Enlightenment

Enlightenment est le gestionnaire de fenêtres de référence pour de nombreux geeks et nerds. Il est entièrement paramétrable, scriptable et modulaire. Il fonctionne sur de petites configurations comme sur de gros serveurs. La version actuelle, la 16 (actuellement 16.999) existe depuis plusieurs années, et la version 17 est un peu l'arlésienne, ou le « vaporware », du logiciel libre, à la différence qu'elle sortira vraiment un jour : quand elle sera prête. Ce sera alors un vrai environnement de bureau (Desktop Shell).

g. Xfce

Devant les mastodontes KDE et GNOME, les machines disposant de peu de ressources n'avaient pas accès à ces environnements à cause du manque de mémoire et de vitesse. Xfce a été développé dans cette optique : c'est un environnement de bureau complet et très léger, basé sur les bibliothèques GTK mais sans consommer de manière excessive des ressources. Il fonctionne ainsi très rapidement avec 64 Mo de mémoire et est livré avec plusieurs applications dont un gestionnaire de fichiers (Thunar), un éditeur de texte, etc. bref le minimum pour travailler. De très nombreuses applications sont disponibles pour cet environnement.



L'environnement de bureau XFCE

h. KDE et GNOME

KDE et GNOME : ils sont présentés en début de chapitre. Le choix est excellent si votre machine dispose d'au moins 256 Mo (512 pour être confortable) et pour avoir un environnement comparable à celui de Windows et MacOS.

i. Les autres

Il existe des dizaines d'autres gestionnaires de fenêtres, dont par exemple Compiz-Fusion, qui représente probablement l'avenir avec ses superbes effets 3D et composites. Parmi les gestionnaires de fenêtres remarquables vous trouverez :

- Afterstep
- Compiz Fusion
- Fluxbox
- Openbox
- Metacity
- Blackbox

- Ion
- Wmii
- etc.

5. Exporter ses fenêtres

Comme vous l'avez vu lors de la présentation de X, celui-ci fonctionne en mode client-serveur. Par défaut le client se connecte aux serveur X local et est affiché sur l'écran principal de l'utilisateur. Cependant l'affichage peut théoriquement être déporté vers tout serveur X.

Soient deux serveurs X :

- l'un sur 192.168.1.60 ;
- l'autre sur 192.168.1.70.

Vous voulez lancer depuis le premier le programme **xcalc** vers le second.

Pour cela, vous devez autoriser le second serveur X à recevoir des requêtes extérieures :

- l'option **-nolisten tcp** ne doit pas être présente via X et/ou XDM ;
- vous devez autoriser la connexion avec la commande **xhost**.

La commande **xhost** permet de contrôler les permissions d'accès au serveur X. Sa syntaxe de base prend un plus « + » ou un moins « - » pour autoriser ou non, de manière globale (tout le monde), les connexions au serveur X.

```
$ xhost +
access control disabled, clients can connect from any host
$ xhost -
access control enabled, only authorized clients can connect
```

Après les signes vous pouvez préciser un nom d'hôte ou un nom de login. Sur le second serveur X, vous devez autoriser le premier :

```
$ xhost +192.168.1.60
192.168.1.60 being added to access control list
```

Depuis la première machine vous avez deux moyens d'exporter l'affichage vers la seconde :

- seulement le programme avec le paramètre `-display` ;
- tous les programmes en exportant une nouvelle variable **DISPLAY**.

Dans les deux cas vous devez préciser, outre l'adresse, le numéro de serveur et éventuellement d'écran. Chaque serveur X se voit affecté un identifiant. S'il est le seul, c'est le zéro (0), si c'est le second : 1, et ainsi de suite. Les « coordonnées » du serveur X, s'il est seul, sont **192.168.1.70:0**. Si plusieurs écrans sont raccordés, ils sont identifiés de manière numérique :0 0,1,2, etc. Si aucun écran n'est précisé, c'est le premier qui est choisi par défaut. Une coordonnée complète est donc **192.168.1.70:0.0**.

Dans le premier cas procédez ainsi :

```
$ xcalc -display 192.168.1.70:0
```

Dans le second cas, exportez la variable **DISPLAY** avec la nouvelle valeur :

```
$ export DISPLAY=192.168.1.70:0$
xcalc
```

La politique de sécurité par défaut est d'empêcher l'accès au serveur X même si xhost+ est précisé. Dans ce cas, en espérant que le serveur ssh le permette, passez par ssh avec le paramètre -X.

6. Les ressources d'une application X

a. Modifier l'apparence d'un programme

Chaque programme X fait appel à des ressources du serveur X Window pour fonctionner, notamment pour les polices de caractères, la forme des boutons, les couleurs, etc. Le fichier Xresources de la configuration de Xdm est un bon exemple.

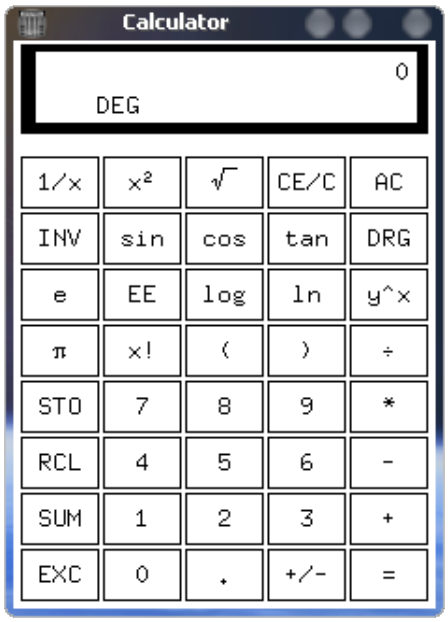
Vous pouvez modifier vous-même les ressources d'un programme X. Cette méthode est moins utilisée, voire ne fonctionne pas, avec les applications issues d'environnements comme KDE ou GNOME.

Chaque programme X admet plusieurs paramètres standards en ligne de commande dont :

- -**bg** : couleur de fond, ressource *background.
- -**bd** : couleur de bordure, ressource *borderColor.
- -**fg** : couleur d'avant plan, ressource *foreground.
- -**fn** : police principale, ressource *font.
- -**geometry** : taille et position de la fenêtre, ressource *TopLevelShell.geometry.
- -**title** : titre de la fenêtre, ressource *title.
- etc.

Xcalc peut se lancer en deux modes : TI-30 et HP (notation polonaise inversée).

L'exemple suivant se propose de modifier l'affichage de l'application **xcalc**. Voici une capture de la calculatrice avant les modifications.



La calculatrice X par défaut

Voici le même résultat avec un changement des paramètres par défaut : titre « Caculette », fond gris, couleur principale marron, bordures jaunes :

```
$ xcalc -title "Caculette" -bg grey -fg brown -bd yellow &
```

La même calculatrice, modifiée via la ligne de commande

b. Éditer les ressources

La commande **xprop** permet de connaître toutes les propriétés d'une fenêtre X.

- Lancez tout d'abord xcalc

  ```
  $ xcalc &
  ```

- Lancez ensuite xprop depuis un terminal.

  ```
  $ xprop
  ```

- Le curseur de la souris se transforme en croix. Cliquez sur la fenêtre de xcalc. Vous devriez obtenir une sortie de ce genre :

```
_NET_WM_ICON_GEOMETRY (CARDINAL) = 649, 984, 133, 20
WM_PROTOCOLS(ATOM): protocols WM_DELETE_WINDOW
WM_STATE(WM_STATE):
              window state: Normal
              icon window: 0x0
_NET_WM_STATE(ATOM) =
_NET_WM_ALLOWED_ACTIONS(ATOM) = _NET_WM_ACTION_MOVE,
_NET_WM_ACTION_RESIZE, _NET_WM_ACTION_MINIMIZE,
_NET_WM_ACTION_SHADE, _NET_WM_ACTION_MAXIMIZE_VERT,
_NET_WM_ACTION_MAXIMIZE_HORZ, _NET_WM_ACTION_FULLSCREEN,
_NET_WM_ACTION_CHANGE_DESKTOP, _NET_WM_ACTION_CLOSE
_KDE_NET_WM_FRAME_STRUT(CARDINAL) = 5, 5, 19, 5
_NET_FRAME_EXTENTS(CARDINAL) = 5, 5, 19, 5
_NET_WM_DESKTOP(CARDINAL) = 0
WM_CLIENT_LEADER(WINDOW): window id # 0x3e00026
WM_LOCALE_NAME(STRING) = "fr_FR.UTF-8"
_KDE_NET_WM_USER_CREATION_TIME(CARDINAL) = 1321633138
WM_CLASS(STRING) = "xcalc", "XCalc
"WM_HINTS(WM_HINTS):
              Client accepts input or input focus: True
              Initial state is Normal State.
              bitmap id # to use for icon: 0x3e00001
```

```
WM_NORMAL_HINTS(WM_SIZE_HINTS):
              program specified size: 226 by 308
              window gravity: NorthWest
WM_CLIENT_MACHINE(STRING) = "slyserver"
WM_COMMAND(STRING) = { "xcalc" }
WM_ICON_NAME(STRING) = "Calc"
WM_NAME(STRING) = "Calculator"
```

Dans les informations retournées la ligne en gras indique le nom (xcalc ou XCalc) qu'il faudra utiliser pour accéder aux ressources de l'application X. C'est généralement le nom réel du programme.

Une ressource est décrite sous la forme :

```
class*resource: value
instance*resource: value
class.resource: value
instance.resource: value
```

- **class** représente toutes les invocations d'un même programme. La classe XCalc représente toutes les instances de XCalc, donc si vous lancez n fois le programme xcalc les ressources de type xcalc.xxxx seront communes à tous les xcalc lancés.

- **instance** est le nom d'une instance d'un programme particulier.

Le manuel de xcalc fournit dans les sections CUSTOMIZATION et WIDGET HIERARCHY les ressources et les valeurs particulières acceptées. Xcalc dispose des modes hp et ti. En mode ti par exemple les ressources sont du type :

```
XCalc.ti.resource : value
```

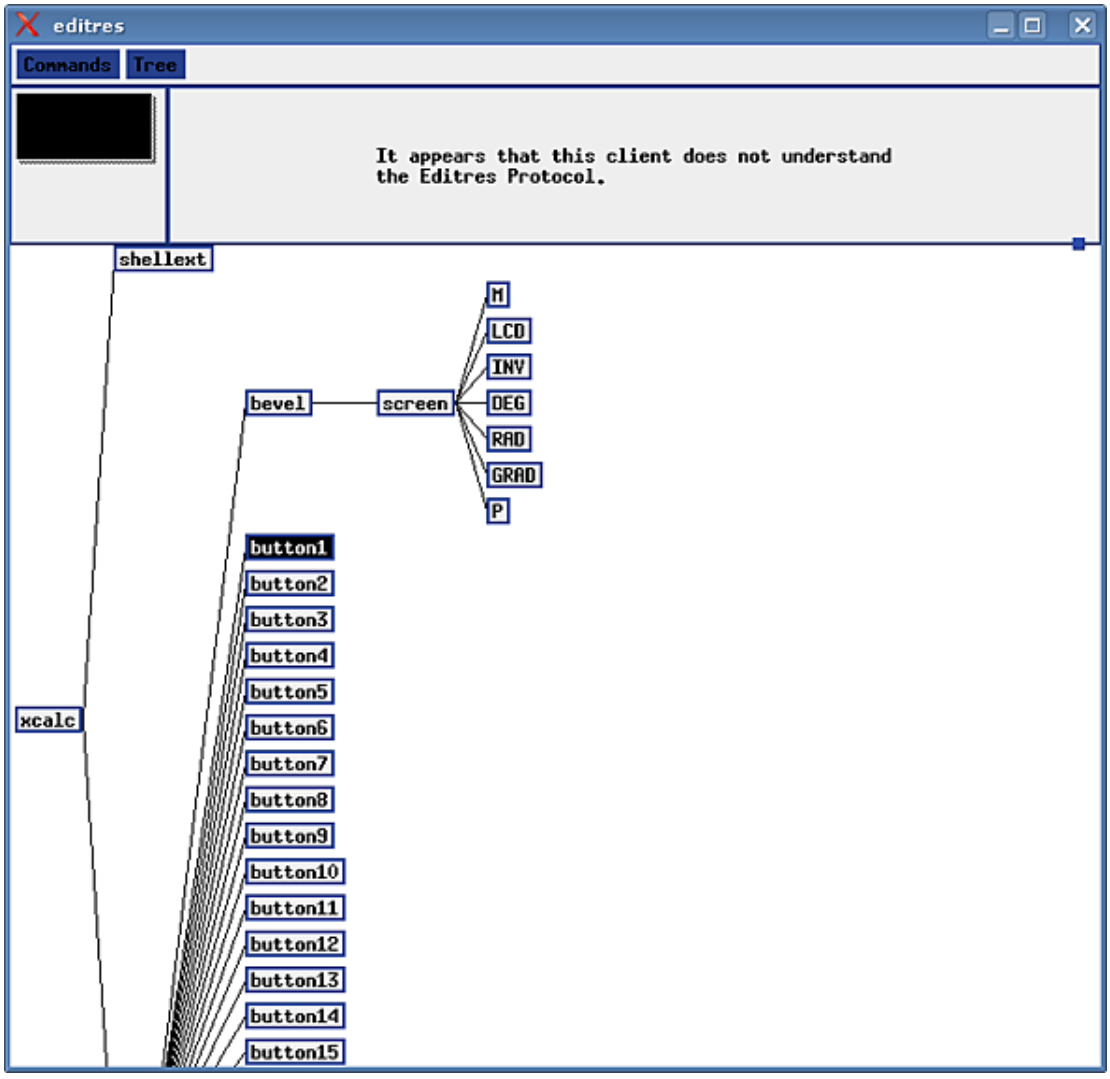
Mais si vous remplacez ti ou hp par une étoile « * » la définition de la ressource s'appliquera à l'ensemble. Les ressources elles-mêmes peuvent se décomposer en sous-ressources.

La commande **appres** prend en paramètre une instance ou une classe et retourne toutes les ressources associées. C'est idéal pour sauver les ressources d'une classe avant de les modifier. Il affiche cependant aussi toutes les ressources applicables à XCalc, dont les ressources par défaut commençant par « * » :

```
$ appres XCalc | grep -i xcalc
...
XCalc*ti.button13.fromVert:             button8
XCalc*ti.button30.translations:         #override
<Btn1Down>,<Btn1Up>:subtract()unset()
XCalc*ti.button30.background:           rgb:e/d/c
XCalc*ti.button30.borderColor:          rgb:9/8/7
XCalc*ti.button30.foreground:           gray5
XCalc*ti.button30.fromVert:             button25
XCalc*ti.button30.fromHoriz:            button29
XCalc*ti.button30.label:                -
XCalc*ti.button30.displayList:          foreground rgb:a/9/8;segments
8,-4,-9,-4,-4,-9,-4,8;draw-arc -14,-14,-4,-4,270,90
XCalc*ti.button14.label:                ln
XCalc*ti.button14.translations:         #override
<Btn1Down>,<Btn1Up>:naturalLog()unset()
XCalc*ti.button14.fromHoriz:            button13
XCalc*ti.button14.fromVert:             button9
XCalc*ti.Command.background:            rgb:c/d/e
XCalc*ti.Command.borderColor:           rgb:8/9/a
XCalc*ti.Command.shapeStyle:            roundedRectangle
XCalc*ti.Command.foreground:            gray5
XCalc*ti.Command.displayList:           foreground rgb:a/b/c;segments
8,-4,-9,-4,-4,-9,-4,8;draw-arc -14,-14,-4,-4,270,90
XCalc*ti.backgroundPixmap:
gray3?foreground=gray70&background=gray85
XCalc*bevel.screen.GRAD.fromHoriz:      RAD
XCalc*bevel.screen.GRAD.fromVert:       LCD
XCalc*bevel.screen.INV.fromVert:        LCD
XCalc*bevel.screen.INV.vertDistance:    4
XCalc*bevel.screen*INV.vertDistance:    2
XCalc*bevel.screen.P.label:             ()
```

```
XCalc*bevel.screen.P.fromHoriz:         GRAD
XCalc*bevel.screen.P.fromVert:          LCD
XCalc*bevel.screen.P.horizDistance:     2
XCalc*bevel.screen.LCD.label:           88888888888
XCalc*bevel.screen.LCD.fromHoriz:       M
XCalc*bevel.screen.LCD.horizDistance:   4
XCalc*bevel.screen.LCD.vertDistance:    2
XCalc*bevel.screen*LCD.foreground:      gray20
XCalc*bevel.screen.DEG.fromHoriz:       INV
XCalc*bevel.screen.DEG.fromVert:        LCD
XCalc*bevel.screen.DEG.horizDistance:   1
XCalc*bevel.screen.Label.horizDistance: 4
XCalc*bevel.screen.Label.vertDistance:  2
XCalc*bevel.screen.Label.internalHeight:    1
XCalc*bevel.screen.Label.internalWidth: 1...
```

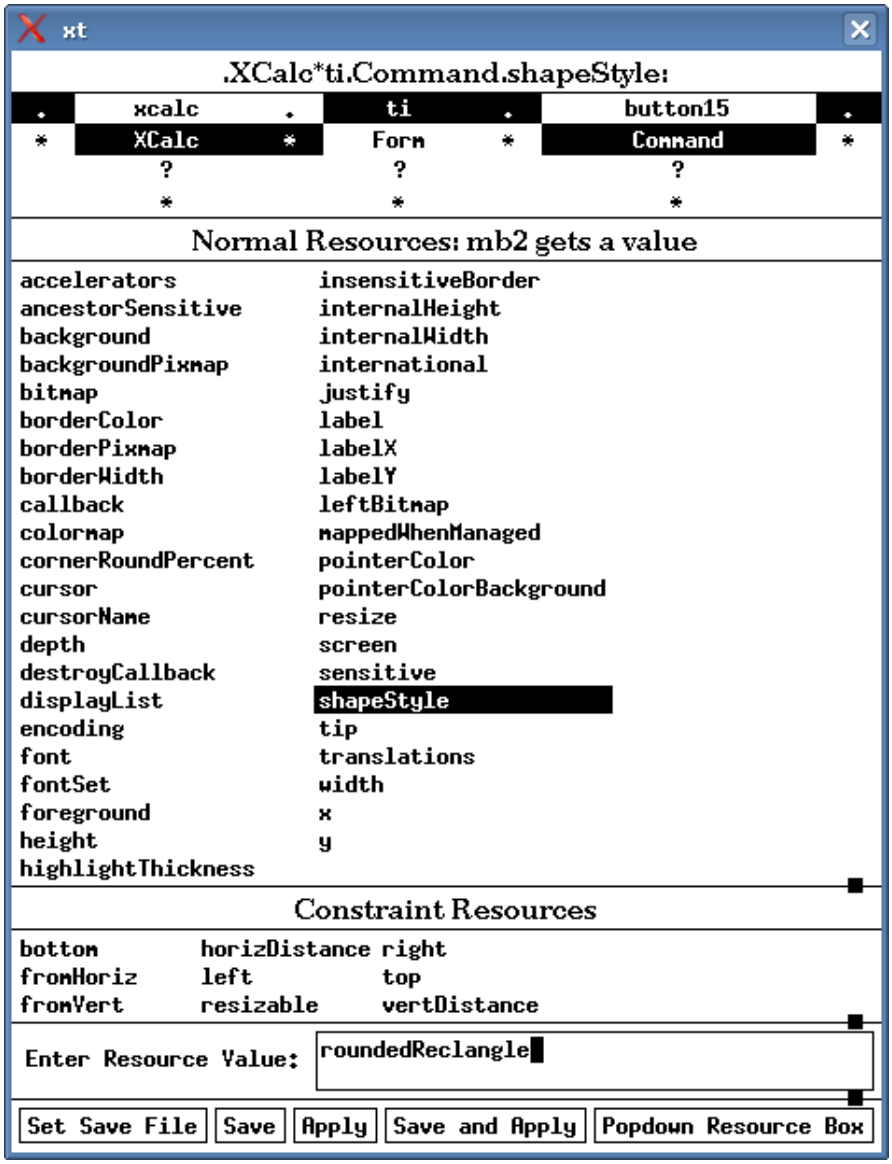
Il existe un éditeur simple pour éditer les ressources d'un programme et de sauver les paramètres. Il s'appelle **editres**. Lancez-le puis dans le menu **Commands** sélectionnez **Get Tree**. Cliquez ensuite sur la fenêtre X, celle de Xcalc.



editres : éditeur de ressources X

Pour modifier à la volée une valeur, cliquez sur une ressource, par exemple **button1**, puis dans **Commands** sur **Show Resource Box**. Dans cette boîte choisissez **background**, saisissez **red** puis cliquez sur **Apply**. Le fond du bouton **1/x** devient rouge.

Dans le menu **Tree**, sélectionnez **Show class name**. Cliquez sur l'un des **Command** puis retournez dans la boîte des ressources. Vous allez modifier toutes les ressources pour toutes les classes Command : cliquez en haut sur **XCalc**, *****, **ti**, **.**, **Command** puis dans le cadre central sur **shapeStyle** (en haut vous devez avoir **.XCalc*ti.Command.shapeStype**). Saisissez **roundedRectangle**, puis **Apply**. Les boutons sont tous arrondis !



Édition d'une ressource de xcalc avec editres

Vous pouvez sauver le réglage à l'aide des boutons **Set Save File** puis **Save**.

```
$ cat Xcalc.res
.XCalc*ti.Command.shapeStyle: roundedRectangle
```

c. xrdb

Vous pouvez sauver vos réglages dans une fichier quelconque mais X tente de lire le fichier $HOME/.Xdefaults à l'ouverture de session. C'est la commande **xrdb** qui est utilisée pour cela. Elle admet les paramètres suivants :

- **aucun** : le nouveau fichier écrase tous les anciens réglages ;
- -**merge** : le fichier est ajouté aux précédents réglages ;
- -**remove** : supprime une ressource. Sans rien : supprime toutes les ressources.

Soit le contenu suivant :

```
$ cat .Xdefaults
XCalc*title: Calculette
XCalc*ti.Command.shapeStyle: roundedRectangle
```

```
XCalc*ti.Command.background: cyan
```

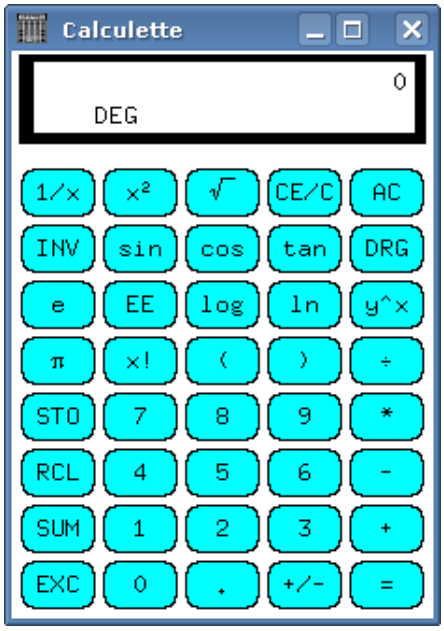
- Le titre de la fenêtre sera Calculette.
- Les boutons auront les coins arrondis.
- Le fond des commandes (les boutons) seront bleu ciel.

Chargez les réglages ainsi :

```
$ xrdb -merge .Xdefaults
```

Et regardez le résultat :

```
$ xcalc
```



xcalc modifié via le fichier des ressources.

Accessibilité

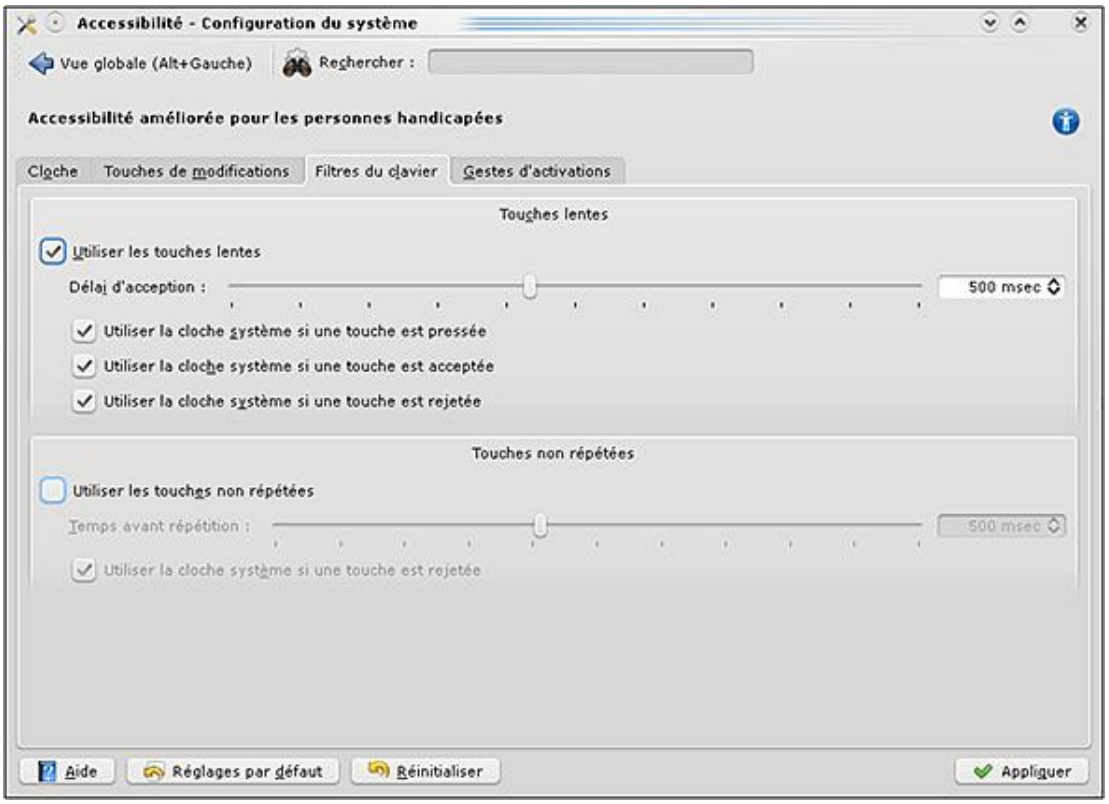
Si la plupart des utilisateurs n'ont pas de problème pour lire un écran, écrire sur un clavier et utiliser une souris, quelques uns souffrent parfois de troubles les empêchant de travailler de manière idéale. Selon leur handicap, Linux et ses outils proposent divers moyens pour aider ces personnes.

Il s'agit ici de survoler quelques moyens disponibles pour aider les personnes handicapées, notamment moteurs ou ayant des problèmes de vue.

1. Assistance au clavier et à la souris

Un simple réglage du clavier peut aider une personne ayant des problèmes moteurs. Par exemple, une personne lente va chercher longtemps une touche, quitte à se tromper, puis appuiera trop longtemps sur la touche voulue, ce qui aura pour effet de répéter le caractère saisi plusieurs fois.

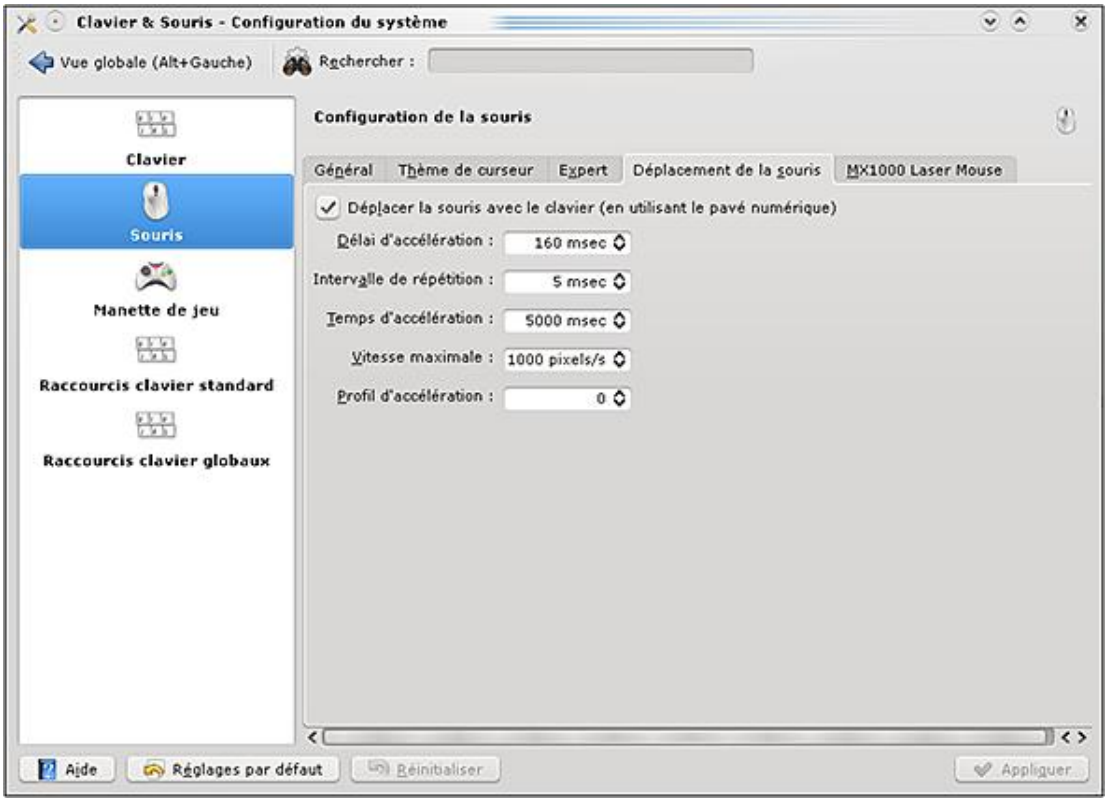
Le module AccessX de l'environnement graphique permet de nombreux réglages. Ceux-ci sont accessibles depuis les centres de configuration des environnements graphiques. Sous KDE4 par exemple, le module **Accessibilité** de l'onglet **Général** de la configuration du système propose de modifier de nombreux réglages par défaut, dont le fonctionnement du clavier. Par exemple, dans l'onglet **Filtres du clavier**, cochez **Utiliser les touches lentes**. Si une personne veut effectuer une action, elle ne sera prise en compte qu'après le délai choisi (0,5 secondes par défaut, réglable). En dessous, **Utiliser les touches non répétées** attend le délai indiqué avant de répéter une touche en cas d'appui trop long.



Les paramètres d'accessibilité pour le clavier de KDE4

Dans le même état d'esprit, vous pouvez activer un support visuel ou auditif selon les actions effectuées. L'appui sur une touche peut provoquer un bip, le clignotement de l'écran. Les mouvements de la souris peuvent contribuer à améliorer l'accessibilité.

Comme tout le monde ne peut pas utiliser la souris, il est possible de déplacer le curseur sur l'écran à l'aide du clavier. Reportez-vous à la documentation de votre environnement pour l'activation de ce mode. Toujours sous KDE4, dans le module **Clavier et Souris**, onglet **Déplacement de la souris**, si vous cochez **Déplacer la souris avec le clavier**, vous pouvez alors déplacer le curseur de la souris avec les touches directionnelles du clavier numérique. Les touches de 1 à 9 sauf 5 déplacent le curseur dans la direction souhaitée. La touche 5 simule un simple clic (5 deux fois rapidement simule un double clic). Il est possible de changer de bouton, de le maintenir, etc.



Utilisation des touches du clavier numérique pour déplacer la souris

Du côté de la souris, si vous disposez d'un modèle correct vous pouvez activer le profil pour gaucher qui va inverser les boutons et même inverser le sens de la roulette.

Le clavier virtuel peut enfin fournir une aide appréciable en cas de difficulté d'utilisation du clavier physique. Vous pouvez dès lors utiliser un clavier logiciel, avec la souris ou un écran tactile. Il suffit alors de cliquer sur les différentes touches pour l'utiliser comme un clavier ordinaire. Celui de KDE se nomme **kvkbd**, mais X Window en propose un par défaut appelé **xvkbd** et Gnome propose GOK.

Le clavier virtuel kvkbd permet de se passer du clavier physique

2. Assistance visuelle et auditive

Les personnes malvoyantes apprécieront de pouvoir utiliser des notifications visuelles comme le clignotement de l'écran. De même, tous les environnements de bureau dignes de ce nom, comme KDE3, KDE4 ou Gnome (par exemple) proposent de modifier l'affichage :

- **Couleurs** : en augmentant par exemple les contrastes ou avec des thèmes selon la représentation des

couleurs (daltonisme).

- **Styles** : en modifiant la taille des divers éléments visuels, comme les boutons, les cases, les champs, etc.
- **Polices de caractères** : en choisissant une police adaptée, de plus grande taille, d'un style donné.
- **Thèmes** : des thèmes spécifiques pour les malvoyants existent, modifiant l'ensemble des réglages ci-dessus en une fois.

Des outils simples proposent d'agrandir des zones de l'écran, avec une loupe virtuelle. KDE propose Kmagnifier. Il est aussi possible d'associer des actions (touches du clavier ou de la souris) à la loupe pour un accès plus rapide.

Les aveugles peuvent aussi travailler sous Linux, car il existe des solutions pour lire un écran en braille. Plusieurs solutions existent pour supporter le braille dont brltty, pour le transfert de la console sur un lecteur braille, et orca, produit phare sous Gnome, qui permet le support du braille dans l'environnement Gnome mais aussi la synthèse vocale et l'accessibilité à l'écran pour les non-voyants.

Enfin, il existe plusieurs moyens pour faire parler l'ordinateur. Les français sont un peu moins bien lotis dans ce domaine car la synthèse vocale libre y est moins courante. Mais des produits comme espeak (synthèse très « ordinateur ») et surtout mbrola avec freetts donnent des résultats très proches d'une voix humaine.

Partitionnement avancé RAID

1. Définitions

Le **RAID** (*Redundant Array of Inexpansive Disks*) a été défini par l'université de Berkeley en 1987 dans le double but de réduire les coûts et d'augmenter la fiabilité du stockage des données. Le but est de combiner plusieurs petits disques physiques indépendants en une matrice (array : tableau, ensemble, rangée, matrice) de disques dont la capacité dépasse celle du SLED (*Single Large Expensive Drive*). Une matrice apparaît comme une unité logique de stockage unique.

Le **MTBF** (*Mean Time Between Failure* - temps moyen entre pannes) de l'ensemble est égal au MTBF d'un disque individuel divisé par le nombre de disques dans l'ensemble et donc théoriquement, une solution RAID peut être inadaptée pour des tâches critiques. Heureusement, le RAID peut être tolérant aux fautes en stockant de manière redondante ses informations selon plusieurs méthodes :

- **RAID-0** : appelé **stripe mode** : deux disques au moins forment un seul volume. Les deux disques ont en principe la même taille. Chaque opération de lecture/écriture sera fractionnée et effectuée sur chacun des disques. Par exemple, 4 ko seront écrits sur le disque 0, 4 ko sur le disque 1, 4 ko sur le disque 2, puis 4 ko sur le disque 0, etc. Ainsi, les performances sont accrues puisque les opérations de lecture et d'écriture sont effectuées en parallèle sur les disques. Si N est le nombre de disques et P la vitesse de transfert, la vitesse de transfert du volume RAID est en principe proche de N*P mbps. Le RAID-0 n'a aucune redondance. En cas de panne d'un des disques, il est probable que l'ensemble des données soit perdu.

- **RAID-1** : appelé **mirroring** : premier mode redondant. Il peut être utilisé à partir de deux disques ou plus avec d'éventuels disques de secours (*Spare Disk*). Chaque information écrite sur un disque est dupliquée sur les autres. Si N-1 disques du RAID viennent à tomber, les données restent intactes. Si un disque de secours est présent, en cas de panne, il est automatiquement reconstruit et prend la place du disque défaillant. Les performances en écriture peuvent être mauvaises : écriture sur N disques en même temps, risquant de saturer le contrôleur disque et le bus. Les performances en lecture sont bonnes, car RAID emploie un algorithme qui peut lire les données sur chaque disque (puisqu'ils sont identiques).

- **RAID-5** : RAID avec bande de parité redistribuée. C'est le mode le plus utilisé car c'est celui qui offre le meilleur compromis entre le nombre de disques, l'espace disponible et la redondance. Il faut au moins trois disques avec d'éventuels disques de secours. La parité est présente sur chacun des disques. La taille finale est celle de N-1 disques. Le RAID-5 survit à une panne de disque. Dans ce cas, si un disque de secours est présent, il sera automatiquement reconstruit. Les performances en lecture sont équivalentes à celles du RAID-0 tandis qu'en écriture, elles dépendent de l'algorithme employé et de la mémoire de la machine.

2. Précautions et considérations d'usage

a. Disque de secours

Un disque de secours (*Spare Disk*>) ne fait pas partie intégrante d'une matrice RAID tant qu'un disque ne tombe pas en panne. Si cela arrive, le disque est marqué défectueux et le premier disque Spare prend le relais. Quoi qu'il arrive, il faut tout de même, le plus vite possible, changer le disque défaillant et reconstruire le RAID.

b. Disque défectueux

Un disque défectueux (*Faulty Disk*) est un disque qui a été reconnu défaillant ou en panne par le RAID. Dans ce cas, RAID utilise le premier disque Spare pour reconstruire sa matrice. Les disques Faulty appartiennent toujours à la matrice mais sont désactivés.

c. Boot

La partition de boot (celle qui contient le noyau, la configuration du bootloader, les fichiers images de disques) ne doit pas être placée dans une matrice RAID : le chargeur de démarrage est incapable de monter des partitions RAID (la prochaine version de GRUB en sera capable).

d. Swap

Vous pouvez installer un swap sur du RAID mais ce n'est en principe pas utile dans les cas courants. En effet, Linux est capable d'équilibrer l'utilisation du swap sur plusieurs disques/partitions seuls. Dans ce cas, déclarez n swaps dans **/etc/fstab** avec la même priorité.

```
/dev/sda2       swap            swap    defaults,pri=1   0 0
/dev/sdb2       swap            swap    defaults,pri=1   0 0
/dev/sdc2       swap            swap    defaults,pri=1   0 0
```

Cependant, en cas de besoin de haute disponibilité, le swap sur le RAID est possible.

e. Périphériques

Une matrice RAID est reconnue par le système comme un périphérique de type bloc, comme n'importe quel disque physique. Ainsi, un RAID peut être constitué avec des disques, des partitions (généralement, on crée une unique partition sur chaque disque). Le bus n'a aucune importance : vous pouvez construire une matrice RAID avec des disques SCSI et IDE mélangés. De même, on peut construire du RAID sur d'autres matrices RAID, par exemple du RAID-0+1 (2x2 disques en RAID-1, les deux matrices résultantes en formant une nouvelle en RAID-0). Les périphériques RAID sont sous la forme :

/dev/md0
/dev/md1

f. IDE

Si les disques IDE ont longtemps été le SCSI du pauvre (matériel de moins bonne qualité, lenteur, manque de fiabilité) ce n'est plus vraiment le cas. Les derniers modèles sont totalement identiques aux disques SCSI, contrôleur excepté. Vous pouvez donc monter pour un coût raisonnable des configurations RAID en IDE. Cependant une règle est à retenir :

UN SEUL DISQUE IDE PAR BUS IDE

En pratique, cela correspond à un disque par câble, sans rien d'autre. En effet, un bus IDE survit en principe à la déficience d'un disque mais il arrive régulièrement que le bus IDE devienne lui-même défectueux, entraînant la perte du second disque présent sur le bus et donc la perte de la matrice RAID. L'achat de cartes IDE supplémentaires (bas prix) permet de compenser le problème de fiabilité (deux disques par carte).

g. Hot Swap

- **IDE** : NE JAMAIS DEBRANCHER À CHAUD UN DISQUE IDE ! C'est le meilleur moyen de détruire le disque, si ce n'était pas encore le cas et de détruire le contrôleur IDE (et donc éventuellement la carte mère ou additionnelle). L'IDE n'est pas prévu pour.
- **SCSI** : les contrôleurs SCSI ne sont pas prévus pour le Hot Swap mais devraient en théorie tout de même fonctionner, si le disque est identique physiquement et logiquement.
- **SATA** : le SATA est reconnu comme du SCSI. La spécification SATA en version 2 supporte théoriquement le Hot Swap. Seulement, la plupart des contrôleurs actuels implémentent mal ou pas du tout cette possibilité, d'où les risques de plantages ou de griller son contrôleur. Référez-vous à la documentation du constructeur de votre carte mère (chipset).
- **SCA** : ce sont des disques SCSI spécifiques. Consultez le document « Software RAID Howto ».

3. RAID avec mdadm

a. Préparation

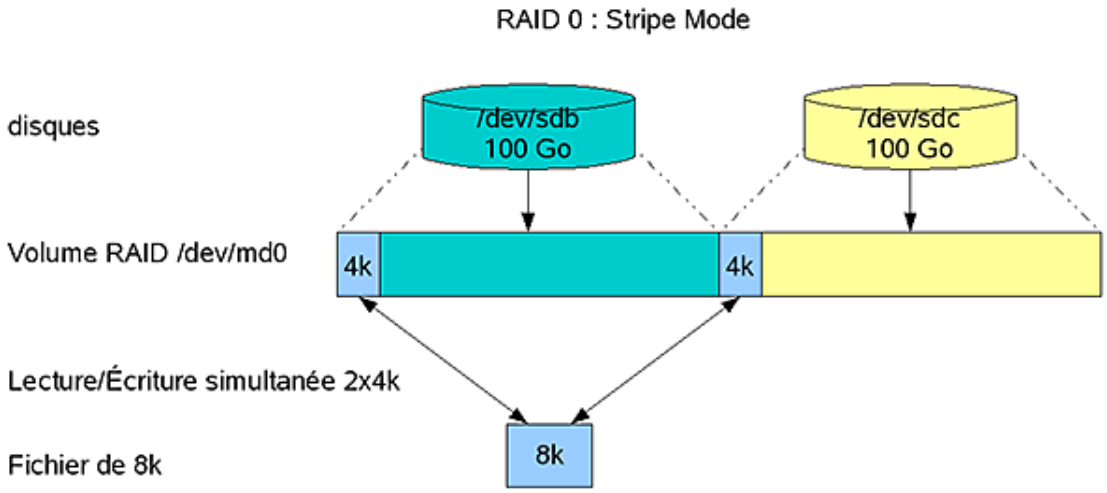
L'outil **mdadm** remplace les outils raidtools des anciennes distributions Linux. Cet outil unique est plus simple et permet d'effectuer l'ensemble des opérations. Son fichier de configuration est **/etc/mdadm.conf**.

Afin de créer des matrices RAID, il faut que les partitions qui vont servir à créer la matrice soient de type **0xFD** (Linux RAID autodetect). Les partitions doivent être logiquement sur des disques différents, mais pour des tests, le support RAID autorise des partitions sur le même disque. Dans ce cas, vous veillerez à ce que les partitions

disposent de la même taille.

b. Création

RAID-0



Principe du RAID 0 : Striping

Soient deux partitions **/dev/sdb1** et **/dev/sdc1**. Vous allez créer une partition RAID-0, assemblage de ces deux partitions.

```
# mdadm --create /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

`--create`

Créer un RAID.

`/dev/md0`

Nom du fichier périphérique de type bloc représentant la matrice RAID.

`--level`

Type de RAID à créer : `0`, `raid0` et `stripe` pour du RAID0.

`linear` n'est pas du RAID0 (remplissage au fur et à mesure).

`--raid-devices`

Nombre de partitions utilisées pour créer la matrice.

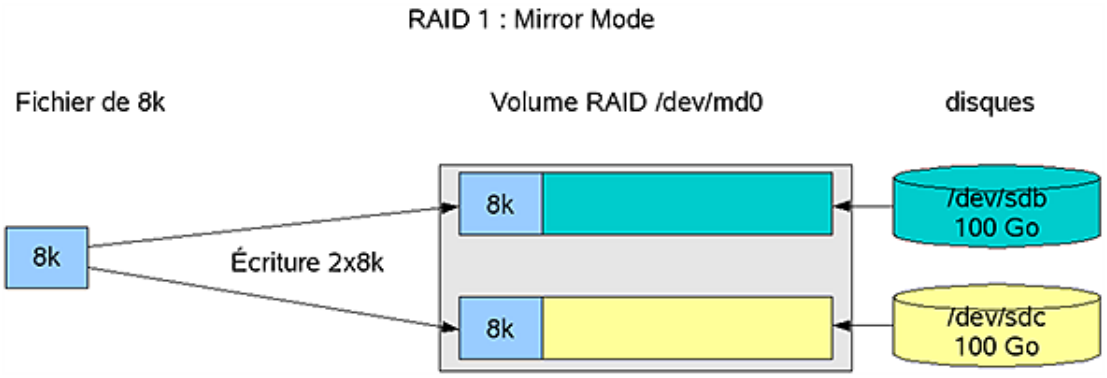
`/dev/sdb1, /dev/sdc1`

Partitions constituant la matrice, suivant le nombre indiqué dans `--raid-devices`.

Il ne reste plus qu'à installer le système de fichiers sur le disque RAID :

```
# mkfs -t ext3 /dev/md0
```

RAID-1



C'est le même principe. Vous allez cette fois rajouter une partition de secours **/dev/sdd1**.

```
# mdadm --create /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb1
/dev/sdc1 --spare-devices=1 /dev/sdd1
```

`--level 1`, `mirror` ou `raid1` sont de bonnes valeurs pour un RAID-1.

`--spare-devices` nombre de disques de secours à utiliser.

`/dev/sdd1` partitions constituant les disques de secours, suivant le nombre indiqué dans `-spare-devices`.

Puis :

```
# mkfs -t ext3 /dev/md0
```

RAID-0+1

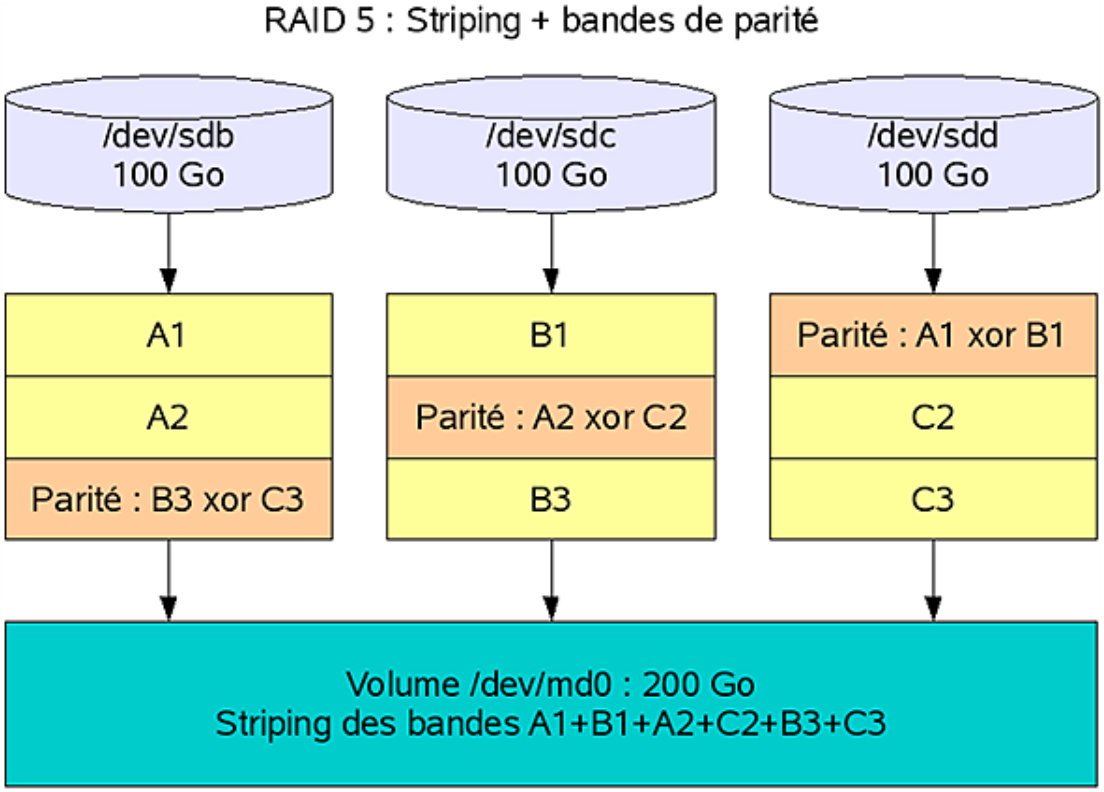
Il faut au moins quatre partitions. Vous devez créer deux matrices RAID-1 que vous allez regrouper en une matrice RAID-0.

```
# mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb1
/dev/sdc1

# mdadm --create /dev/md1 --level=raid1 --raid-devices=2 /dev/sdd1
/dev/sde1

# mdadm --create /dev/md2 --level=raid0 --raid-devices=2 /dev/md0
/dev/md1
```

Puis :

```
# mkfs -t ext3 /dev/md2
```

RAID 5



Vous allez utiliser trois disques de données /dev/sdb1, /dev/sdc1, /dev/sdd1 et un disque de secours /dev/sde1.

```
# mdadm --create /dev/md0 --level=raid5 --raid-devices=3 /dev/sdb1
/dev/sdc1 /dev/sdd1

--spare-devices=1 /dev/sde1
```

Puis on formate :

```
# mkfs -t ext3 /dev/md2
```

c. Sauver la configuration

Pour faciliter la tâche de l'outil **mdadm**, vous pouvez créer (ce n'est pas obligatoire) le fichier de configuration **`/etc/mdadm.conf`**. Ce fichier peut être créé manuellement mais l'outil **mdadm** sait le générer. Il est préférable de le faire APRÈS la création des matrices RAID.

```
# echo "DEVICE partitions" > /etc/mdadm.conf

# mdadm --detail --scan>> /etc/mdadm.conf
```

4. État du RAID

Le fichier virtuel **`/proc/mdstat`** contient des informations sur le RAID. C'est ici que vous pouvez voir le détail d'un RAID, notamment si un des volumes de la matrice est défectueux (Faulty).

```
Personalities : [raid1]
md0 : active raid1 hda10[2] hda9[1] hda8[0]
      104320 blocks [2/2] [UU]
```

La commande **watch** permet de vérifier un état en continu :

```
# watch cat /proc/mdstat
```

Vous pouvez aussi utiliser **mdadm** avec le paramètre `--detail` :

```
# mdadm --detail /dev/md0
/dev/md0:
        Version : 00.90.01
  Creation Time : Mon Jan 23 22:10:20 2006
     Raid Level : raid1
     Array Size : 104320 (101.88 MiB 106.82 MB)
    Device Size : 104320 (101.88 MiB 106.82 MB)
   Raid Devices : 2
  Total Devices : 3
Preferred Minor : 1
    Persistence : Superblock is persistent

    Update Time : Mon Jan 23 22:13:06 2006
          State : clean
 Active Devices : 2
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1

    Number   Major   Minor   RaidDevice State
       0       3        8        0      active sync   /dev/hda8
       1       3        9        1      active sync   /dev/hda9
       2       3       10       -1      spare   /dev/hda10
           UUID : 90e838b5:936f18c7:39f665d3:d9dad1a9
         Events : 0.4
```

Remarquez qu'avec cette dernière commande vous obtenez bien plus de détails, notamment quels sont les disques "spare" et "faulty".

5. Simuler une panne

Vous allez simuler une panne sur **/dev/hda8** :

```
# mdadm /dev/md0 -f /dev/hda8

mdadm: set /dev/hda8 faulty in /dev/md0
```

Regardez l'état du RAID dans **/proc/mdstat** durant l'exécution :

```
md0 : active raid1 hda10[2] hda9[1] hda8[0](F)
     104320 blocks [2/1] [U_]
     [=>..................]  recovery =  8.8% (9216/104320) finish=0.1min
speed=9216K/sec
```

Remarquez qu'un « (F) » est apparu près de hda8, indiquant un disque Faulty. On voit aussi que sur les deux disques, un est en panne et que le RAID reconstruit sa matrice avec le spare disk. Après l'exécution, vous obtenez :

```
md0 : active raid1 hda10[1] hda9[0] hda8[2](F)
      104320 blocks [2/2] [UU]
```

Le RAID est reconstruit et fonctionne à merveille.

```
# mdadm --detail /dev/md0
...
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 0

    Number   Major   Minor   RaidDevice State
       0       3        9        0      active sync   /dev/hda9
       1       3       10        1      active sync   /dev/hda10
       2       3        8       -1      faulty        /dev/hda8
...
```

Le disque Faulty est bien **/dev/hda8** ; **/dev/hda10** a pris sa place en tant que disque de secours. Ainsi, le disque de secours devient un disque RAID de la matrice.

6. Remplacer un disque

Puisque **/dev/hda8** est en panne, vous allez le remplacer. Retirez-le avec `-r` (ou `--remove`) :

```
# mdadm /dev/md0 -r /dev/hda8
mdadm: hot removed /dev/hda8

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 hda10[1] hda9[0]
      104320 blocks [2/2] [UU]
```

Constatez que `hda8` a disparu. Vous pouvez éteindre la machine puis remplacer le disque défaillant. Rallumez la machine, puis repartitionnez le disque correctement. Il n'y a plus qu'à rajouter le disque réparé dans la matrice RAID avec `-a` (`--add`) :

```
# mdadm /dev/md0 -a /dev/hda8
mdadm: hot added /dev/hda8

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 hda8[2] hda10[1] hda9[0]
      104320 blocks [2/2] [UU]
```

Le disque `hda8` apparaît à nouveau. Voyez le détail :

```
# mdadm --detail /dev/md0
...
         State : clean
 Active Devices : 2
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1


    Number   Major   Minor   RaidDevice State
       0       3        9        0      active sync   /dev/hda9
       1       3       10        1      active sync   /dev/hda10
       2       3        8       -1      spare         /dev/hda8
...
```

Le disque **/dev/hda8** a été remis et est devenu le nouveau disque de secours !

7. Arrêt et relance manuels

Vous pouvez arrêter ponctuellement une matrice RAID avec `-S` (`--stop`) APRÈS avoir démonté le périphérique :

```
# mdadm --stop /dev/md0
```

Vous redémarrez une matrice RAID avec `-As` (`--assemble -scan`). Cela implique que le fichier **/etc/mdadm.conf** est correctement renseigné (`--scan` recherche les informations dedans).

```
# mdadm --assemble --scan /dev/md0
```

Si le RAID ne redémarre pas, vous pouvez tenter avec `-R` (`--run`) : il est probable qu'il manque un disque ou qu'une reconstruction en cours n'est pas terminée :

```
# mdadm --run /dev/md0
```

Initiation au LVM

1. Principe

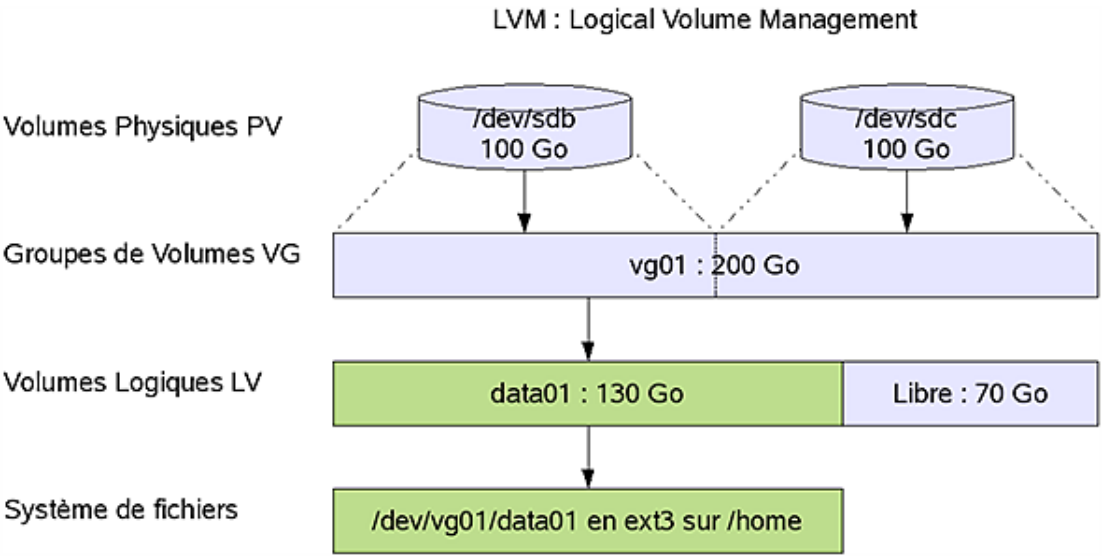
Le **Logical Volume Manager** est un système de gestion très perfectionné des supports de stockage. Le but est de dépasser, voire transcender la gestion physique des disques, et leur organisation logique basique (les partitions) pour étendre la capacité globale des supports, à l'aide d'une gestion entièrement logique de celle-ci.

Un LVM permet, tout comme le RAID 0 par exemple, de créer des espaces de données logiques sur plusieurs disques. Il permet aussi de faire du mirroring, comme le RAID 1. Mais la comparaison s'arrête là. Le RAID logiciel se contente de créer une « partition » dans un espace de stockage défini par le RAID lui-même (par exemple une partition de 100 Go dans un RAID 0 de deux disques de 50 Go).

Le LVM regroupe les disques physiques, ou tout autre support de stockage dit physique (disque, RAID matériel, RAID logiciel, support de stockage en provenance d'un SAN), qu'il appelle des volumes physiques PV (Physical Volume) en un groupe de volumes VG (Volume Group). Ce groupe VG est vu par le LVM comme une sorte de métadisque, dans lequel vous allez créer des volumes logiques LV (Logical Volume) à volonté.

- Volume physique PV : un support de stockage de données dit physique : disque dur par exemple ;
- Groupe de volumes VG : un regroupement logique de 1 à n VG ;
- Volume logique LV : un découpage logique au sein d'un VG.

Un volume logique est vu comme une partition, et est utilisable comme telle. Il peut contenir des données, il suffit de créer un système de fichiers ordinaire (ext3 par exemple) et de le monter de manière tout à fait classique.



Contrairement au RAID 0 Logiciel où la partition de données doit occuper tout l'espace, il est possible de créer autant de volumes logiques de toute taille que souhaité. Mais cela va bien plus loin.

Le LVM est dynamique. Il est possible d'ajouter et de supprimer des volumes physiques d'un groupe de volumes. En ajoutant des volumes physiques, la capacité, et donc l'espace disponible du groupe augmente. Le nouvel espace disponible peut permettre de créer des nouveaux volumes logiques, mais aussi d'agrandir un volume logique existant.

Un volume logique est dynamique : il peut être agrandi ou réduit à volonté, ce qui implique qu'il faut aussi pouvoir agrandir un système de fichiers, ou le réduire.

Notez enfin qu'une matrice RAID peut être utilisée comme volume physique.

La configuration du LVM est située dans les fichiers et répertoires présents dans **`/etc/lvm`**. Le fichier **`/etc/lvm/lvm.conf`** contient la configuration globale. La configuration des différents volumes (physiques, groupes et logiques) ne se trouve pas dans un fichier, mais dans une structure présente au sein des périphériques eux-mêmes, dans les premiers blocs : ce sont les métadatas des volumes physiques.

2. Les volumes physiques

a. Créer un volume physique

Un volume physique peut être un disque complet ou une partition classique au sein d'un disque. Dans ce cas, la partition doit être de type **0x8e**.

Voici le retour de la commande **fdisk** sur **/dev/sdb**. Distinguez les partitions primaires 2 et 3 de type **8e** qui vont servir pour les exemples suivants.

```
# fdisk -l /dev/sdb

Disque /dev/sdb: 160.0 Go, 160041885696 octets
255 heads, 63 sectors/track, 19457 cylinders
Units = cylindres of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000eab03

Périphérique Amorce    Début         Fin      Blocs    Id  Système
/dev/sdb1               2       16846   135307462+  83  Linux
/dev/sdb2           16847       18152    10490445   8e  Linux LVM
/dev/sdb3           18153       19457    10482412+  8e  Linux LVM
```

Une fois les partitions créées, utilisez la commande **pvcreate** sur une première partition (plusieurs partitions peuvent être précisées) :

```
# pvcreate /dev/sdb2
  Physical volume "/dev/sdb2" successfully created
```

b. Voir les volumes physiques

La commande **pvdisplay** permet de visualiser l'ensemble des volumes physiques accessibles sur votre système. Elle peut prendre aussi un nom de volume spécifique.

```
# pvdisplay /dev/sdb2
  "/dev/sdb2" is a new physical volume of "10,00 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               10,00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

Pour le moment, les informations sont réduites. Le PV n'appartient encore à aucun groupe de volumes (ligne `VG Name`). Sa taille est de 10 Go.Les lignes les plus intéressantes sont les lignes (pour l'instant vides car le PV n'appartient pas à un VG) où est indiqué `PE`. PE signifie Physical Extend, extension physique. Chaque VG, et donc PV le constituant, est découpé en tranches appelées PE. Le PE est l'unité de base de travail du LVM. Si un PE fait 4 Mo, cela signifie que l'espace pourra être découpé au sein du groupe de volumes par tranches de 4 Mo. L'allocation se fait par PE : la création d'un volume logique de 500 PE de 4 Mo fait donc 2000 Mo.

Les valeurs à zéro seront remplies dès que le PE sera dans un VG.

3. Les groupes de volumes

a. Créer un groupe de volumes

Pour créer un groupe de volumes, vous devez disposer d'au moins un volume physique. Vous pouvez créer un groupe de volumes avec la commande **vgcreate**. Un groupe de volumes porte un nom, celui que vous voulez. C'est le premier argument de la commande. Vous passez ensuite comme argument la liste des volumes physiques composant le groupe de volumes, ici un seul, **/dev/sdb2**.

```
# vgcreate vg01 /dev/sdb2
  Volume group "vg01" successfully created
```

b. Propriétés d'un VG

Le groupe de volumes a de nombreuses propriétés. Il peut être étudié avec la commande **vgdisplay**.

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               10,00 GB
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       0 / 0
  Free  PE / Size       2561 / 10,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

Notez les lignes `MAX LV` et `MAX PV`. La première indique le nombre maximum de volumes logiques qui pourront être créés dans ce groupe de volumes. La valeur zéro indique un nombre théoriquement infini. La seconde indique le nombre maximum de volumes physiques pouvant être ajoutés au groupe de volumes. Là encore, le zéro indique un nombre infini.

Votre attention doit être attirée sur le fait que Linux représente un cas particulier dans ce domaine. Le nombre de LV et de PV est ici virtuellement infini (dans le cas du lvm2 ce qui est le cas ici), ce qui n'est absolument pas le cas des autres UNIX où les valeurs MAX LV, MAX PV et PE Size sont automatiquement déterminées à la création du groupe de volumes, en fonction notamment des propriétés des volumes physiques qui composent le groupe. Les valeurs peuvent être positionnées à la création du groupe de volumes à l'aide des paramètres suivants de la commande **pvcreate** :

-l Nombre maximum de volumes logiques

-p Nombre maximum de volumes physiques

-s Taille des extensions physiques (avec un suffixe k, m, g ou t pour préciser l'unité).

Les dernières lignes concernent les PE (extensions physiques). Le groupe dispose actuellement de 2561 extensions de 4 Mo, soit 10 Go, toutes libres. Les volumes logiques occuperont ensuite un certain nombre de ces PE, selon leur taille.

La commande **vgdisplay** accepte le paramètre -v qui donne plus de détails, et notamment la liste des volumes physiques qui le composent.

```
# vgdisplay -v vg01
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
```

```
  Act PV                1
  VG Size               10,00 GB
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       0 / 0
  Free  PE / Size       2561 / 10,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG

  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 2561
```

Maintenant que le PV **/dev/sdb2** fait partie d'un VG, plus d'informations sont disponibles :

```
# pvdisplay /dev/sdb2
  --- Physical volume ---
  PV Name               /dev/sdb2
  VG Name               vg01
  PV Size               10,00 GB / not usable 589,00 KB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2561
  Free PE               2561
  Allocated PE          0
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

4. Les volumes logiques

a. Créer un volume logique

Un volume logique est un découpage d'un VG (groupe de volumes) qui est l'équivalent d'une partition dans laquelle vous pourrez créer un système de fichiers. Un volume logique LV occupe un certain nombre de PE (extensions physiques) d'un VG, contigus ou non. Ceci a son importance pour la suite car :

- il est possible d'agrandir un LV tant qu'il reste des PE de libres dans le VG.
- il est possible de réduire un LV, ce qui libérera des PE dans le VG, utilisables pour créer de nouveaux LV ou pour les agrandir.

Ceci signifie que le LVM gère une sorte d'index des PE, et leur ordre, pour savoir à quel LV appartient un PE.

Vous créez un volume logique avec la commande **lvcreate**. Un volume logique porte un nom, dispose d'une taille exprimée soit en extensions logiques LE (Logical Extension) qui sont la représentation des PE au sein d'un LV, soit en Ko, Mo, Go… La commande suivante crée un volume logique appelé **data01** au sein du VG **vg01**, d'une taille de 6 Go. Le `-L` précise que l'unité est en Mo (`m`), Go (`g`), To (Teraoctet, `t`), Po (Petaoctet), ou Eo (Exaoctet). Pour préciser un nombre de PE, utilisez « -l ».

```
# lvcreate -n data01 -L 6g vg01
  Logical volume "data01" created
```

Un LV est vu comme une partition, et dispose après sa création d'un fichier périphérique associé. Le fichier est dans le dossier **/dev/<nom_du_vg>/<nom_du_lv>**. Notez qu'il s'agit d'un lien symbolique vers un fichier de **/dev/mapper** pour garder une compatibilité avec les autres Unix.

```
# ls -l /dev/vg01/data01
lrwxrwxrwx 1 root root 23 sept. 13 09:27 /dev/vg01/data01 -> /dev/
mapper/vg01-data01
```

b. Propriétés d'un volume logique

Les propriétés d'un volume logique sont accessibles par la commande **lvdisplay** :

```
# lvdisplay /dev/vg01/data01
File descriptor 3 left open
File descriptor 4 left open
  --- Logical volume ---
  LV Name                /dev/vg01/data01
  VG Name                vg01
  LV UUID                6ucPwc-sxMJ-K9P3-MkWR-t28I-NyRM-ZKkTmm
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                6,00 GB
  Current LE             1536
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

Vous pouvez passer les paramètres `-v` et `-m`. Dans ce dernier cas, **lvdisplay** affiche aussi les segments qu'occupe le volume logique au sein des divers volumes physiques, donc la répartition des extensions physiques occupées par le volume logique au sein de chaque volume physique. Comme il n'y a pour l'instant qu'un seul PV au sein du VG, vous obtenez ceci :

```
# lvdisplay -m /dev/vg01/data01
...
  --- Segments ---
  Logical extent 0 to 1535:
    Type               linear
    Physical volume    /dev/sdb2
    Physical extents   0 to 1535
```

c. Accès au volume logique

Vous pouvez créer un système de fichiers et monter le LV comme pour n'importe quelle partition :

```
# mkfs -t ext3 /dev/vg01/data01
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
393216 i-noeuds, 1572864 blocs
78643 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=1610612736
48 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
8192 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Écriture des tables d'i-noeuds : complété
Création du journal (32768 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

Le système de fichiers sera automatiquement vérifié tous les 37 montages ou
après 180 jours, selon la première éventualité. Utiliser tune2fs -c ou -i
pour écraser la valeur.
```

Il ne reste plus qu'à monter le nouveau système de fichiers.

```
# mount -t ext3 /dev/vg01/data01 /mnt/data01
# df /mnt/data01
Sys. de fich.        1K-blocs       Occupé Disponible Capacité Monté sur
/dev/mapper/vg01-data01
```

```
                      6192704    143488    5734644    3% /mnt/data01
```

5. Agrandissements et réductions

a. Les groupes de volumes

Pour le moment, tout reste assez classique. La force du LVM est son dynamisme. L'étape suivante consiste à l'exploiter. Vous voulez maintenant créer un nouveau LV de 6 Go appelé **data02** au sein du VG **vg01**. Voici l'état actuel de **vg01** :

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               10,00 GB
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       1536 / 6,00 GB
  Free  PE / Size       1025 / 4,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

Il n'y a plus assez de place. Seuls 4 Go (1025 PE) sont disponibles. Il faut rajouter au sein de ce VG un nouveau volume physique. Ceci se fait avec la commande **vgextend** qui fonctionne de la même manière que **vgcreate** : indiquez le nom du VG suivi du ou des PV à rajouter.

```
# pvcreate /dev/sdb3
  Physical volume "/dev/sdb3" successfully created
# vgextend vg01 /dev/sdb3
  Volume group "vg01" successfully extended
```

Voici le nouvel état de **vg01**. Remarquez que le VG contient maintenant deux PV, et que 14 Go (3584 PE) sont disponibles.

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  <$[gras>Cur PV                2<$]gras>
  Act PV                2
  VG Size               20,00 GB
  PE Size               4,00 MB
  Total PE              5120
  Alloc PE / Size       1536 / 6,00 GB
  Free  PE / Size       3584 / 14,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

Il ne reste plus qu'à créer le LV **data02** de 6 Go :

```
# lvcreate -n data02 -L 6g vg01
Logical volume "data02" created
```

Quand vous créez un LV, le LVM cherche à optimiser l'utilisation des PE de manière à ce qu'ils soient les plus contigus possibles et si possible sur un même PV. Ceci se voit avec la commande **lvdisplay** et le paramètre `-m`, sur les lignes `Segments` et la liste des segments proposés.

```
# lvdisplay -m /dev/vg01/data02
File descriptor 3 left open
File descriptor 4 left open
  --- Logical volume ---
  LV Name                /dev/vg01/data02
  VG Name                vg01
  LV UUID                QozsE1-tA70-cj2c-RZeD-HfX0-dTm8-0wlYpj
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                6,00 GB
  Current LE             1536
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

  --- Segments ---
  Logical extent 0 to 1535:
    Type                linear
    Physical volume     /dev/sdb3
    Physical extents    0 to 1535
```

Les commandes suivantes créent le système de fichiers et montent celui-ci :

```
# mkfs -t ext3 /dev/vg01/data02
# mount -t ext3 /dev/vg01/data02 /mnt/data02
```

b. Agrandir un volume logique

Il se trouve que le LV **data01** de 6 Go est trop petit. Il doit en faire le double. Il faut lui ajouter 6 Go, ce qui est possible car il reste 8 Go (2048 PE) dans le groupe de volumes **vg01** :

```
# vgdisplay vg01|grep Free
  Free  PE / Size       2048 / 8,00 GB
```

L'agrandissement d'un volume logique se fait dans cet ordre :

- Agrandissement du LV avec la commande **lvextend**
- Agrandissement du système de fichier avec **resize2fs** (ext3)

Agrandissement du LV

La commande **lvextend** autorise les paramètres `-l` (nombre d'extensions logiques LE) ou `-L` comme pour **lvcreate**. Vous précisez ensuite la nouvelle taille du LV ou, si vous rajoutez un + en préfixe, la taille additionnelle souhaitée. Vous pouvez aussi préciser, en dernier argument, le nom du PV sur lequel forcer l'extension du LV (c'est aussi possible avec lvcreate). Ca ne marchera que si le ou les PV précisés disposent d'assez de PE.

La commande suivante rajoute 1536 LE (4x1536=6144 Mo soit 6 Go) dans **data01** :

```
# lvextend -l +1536 /dev/vg01/data01
  Extending logical volume data01 to 12,00 GB
  Logical volume data01 successfully resized
```

Regardez maintenant sur quels PV les données sont situées :

```
# lvdisplay -m /dev/vg01/data01
  --- Logical volume ---
  LV Name                /dev/vg01/data01
  VG Name                vg01
  LV UUID                6ucPwc-sxMJ-K9P3-MkWR-t28I-NyRM-ZKkTmm
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                12,00 GB
  Current LE             3072
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Segments ---
  Logical extent 0 to 2560:
    Type                linear
    Physical volume     /dev/sdb2
    Physical extents    0 to 2560

  Logical extent 2561 to 3071:
    Type                linear
    Physical volume     /dev/sdb3
    Physical extents    1536 to 2046
```

Le volume logique `data01` occupe bien 12 Go, sur deux segments de PE, ces segments étant sur les PV **/dev/sdb2** et **/dev/sdb3**. Le LVM a donc attribué un espace sur l'ensemble des PV du VG.

Extension du système de fichiers

Seul le volume logique a été agrandi. Pour le moment, la taille du système de fichiers contenu dans `data01` n'a pas changé :

```
# df -h /mnt/data01
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/mapper/vg01-data01
                      6,0G  141M  5,5G   3% /mnt/data01
```

La commande **resize2fs** permet de réduire et d'agrandir un système de fichiers. Le premier argument est le système de fichiers, le second la taille, avec un éventuel suffixe `K` (Ko), `M` (Mo), ou `G` (Go). Sans suffixe, c'est le nombre de blocs du système de fichiers qui est indiqué. Si la taille est absente, le système de fichiers sera adapté à la taille de la partition ou du LV.

La commande **resize2fs** peut être utilisée à chaud, c'est-à-dire système de fichiers monté, pour les agrandissements. Il faudra par contre démonter le système de fichiers pour le réduire.

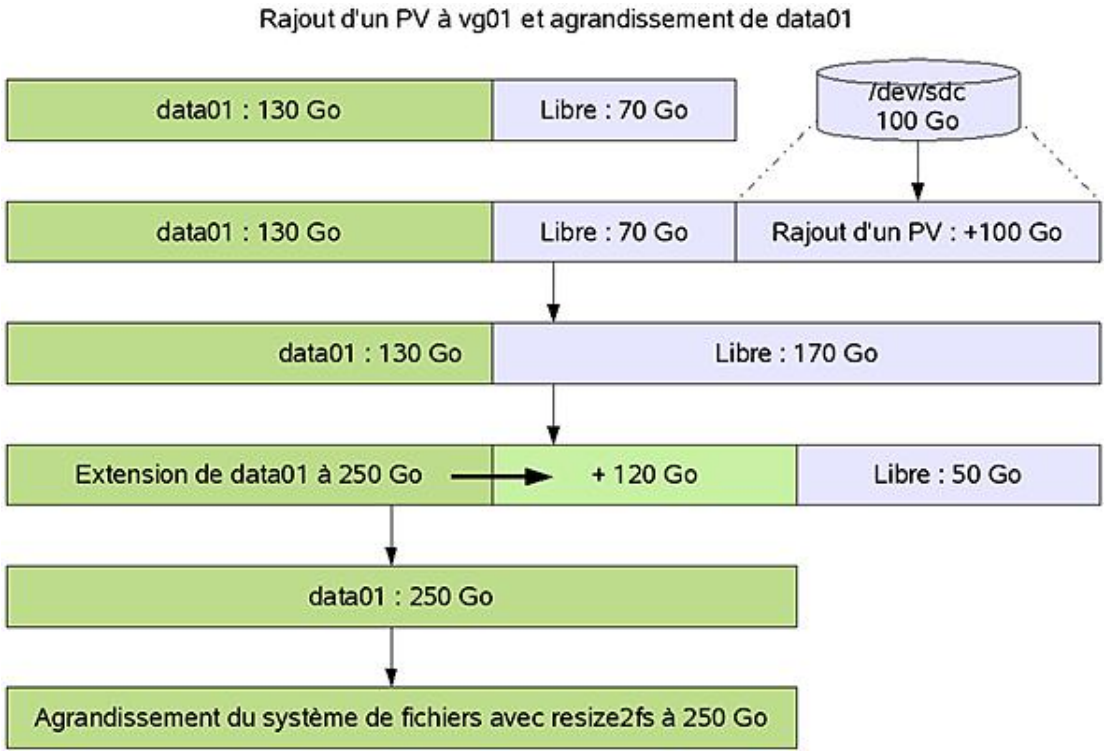
```
# resize2fs /dev/vg01/data01
resize2fs 1.40.8 (13-Mar-2008)
Filesystem at /dev/vg01/data01 is mounted on /mnt/data01; on-line
resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/vg01/data01 to 3145728 (4k) blocks.
Le système de fichiers /dev/vg01/data01 a maintenant une taille
de 3145728 blocs.
```

Regardez l'état du système de fichiers, il occupe maintenant 12 Go :

```
# df -h /mnt/data01
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/mapper/vg01-data01
                       12G  144M   12G   2% /mnt/data01
```

Vous voyez maintenant la puissance du LVM : rajout de volumes physiques et agrandissement de volumes logiques à la volée, de manière dynamique. Il n'y a plus de place ? Ce n'est pas grave : il suffit de rajouter un nouveau disque, le transformer en PV, l'ajouter dans le VG, et redimensionner le LV qui manque de place, sans avoir à

repartitionner, recréer de système de fichiers, faire des backups, etc.



c. Réduire un volume logique

Pour réduire la taille d'un volume logique, vous devez procéder dans cet ordre :

- Vérification du système de fichiers à réduire avec **fsck**.
- Réduction du système de fichiers contenu dans le volume logique avec **resize2fs**.
- Réduction du volume logique avec la commande **lvreduce**.

Vous allez réduire le LV **data01** à 4 Go. C'est uniquement possible si ses données occupent moins de 4 Go. Dans un premier temps, vérifiez la taille actuelle du système de fichiers. Ici, il est quasiment vide :

```
# df -h /mnt/data01
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/mapper/vg01-data01
                       12G  144M   12G   2% /mnt/data01
```

Le système de fichiers ne peut être réduit que s'il n'est pas monté ; démontez-le :

```
# umount /mnt/data01
```

Vérifiez le système de fichiers :

```
# fsck -f /dev/vg01/data01
fsck 1.40.8 (13-Mar-2008)
e2fsck 1.40.8 (13-Mar-2008)Passe 1 : vérification des i-noeuds,
des blocs et des tailles
Passe 2 : vérification de la structure des répertoires
Passe 3 : vérification de la connectivité des répertoires
Passe 4 : vérification des compteurs de référence
Passe 5 : vérification de l'information du sommaire de groupe

/dev/vg01/data01: ***** LE SYSTÈME DE FICHIERS A ÉTÉ MODIFIÉ *****
/dev/vg01/data01: 11/786432 files (9.1% non-contiguous),
86002/3145728 blocks
```

Redimensionnez le système de fichiers à 4 Go :

```
# resize2fs /dev/vg01/data01 4G
resize2fs 1.40.8 (13-Mar-2008)
Resizing the filesystem on /dev/vg01/data01 to 1048576 (4k) blocks.
Le système de fichiers /dev/vg01/data01 a maintenant une taille
de 1048576 blocs.
```

Vérifiez la nouvelle taille du système de fichiers. 4096*1048576 font bien 4 Go.

```
# dumpe2fs -h /dev/vg01/data01 | grep ^Block
dumpe2fs 1.40.8 (13-Mar-2008)
Block count:              1048576
Block size:               4096
Blocks per group:         32768
```

Enfin, redimensionnez le LV à 4 Go. La syntaxe de **lvreduce** est la même que lvextend, sauf qu'il n'est pas possible de préciser de PV. Veillez ici à ne pas vous tromper : si vous avez mal réduit le système de fichiers, vous risquez de le détruire. Répondez **y** à la question si vous êtes certain.

```
# lvreduce -L 4G /dev/vg01/data01
  WARNING: Reducing active logical volume to 4,00 GB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce data01? [y/n]: y
  Reducing logical volume data01 to 4,00 GB
  Logical volume data01 successfully resized
```

Remontez le système de fichiers :

```
# mount -t ext3 /dev/vg01/data01 /mnt/data01
# df -h /mnt/data01
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/mapper/vg01-data01
                      4,0G  141M  3,7G   4% /mnt/data01
```

d. Déplacer le contenu d'un volume physique

Il est courant en entreprise de déplacer un PV vers un autre. Ce peut être dans le but de remplacer un disque contenant le PV par un autre (pour agrandir par exemple). Dans ce cas, vous pouvez déplacer le contenu d'un PV vers un autre, voire des PE d'un LV vers un autre PV, ou encore certains PE précis. Sans rien préciser comme destination, le LVM va déplacer tous les PE du PV dans les autres PV du groupe de volumes. Attention : les volumes physiques doivent être dans le même groupe de volumes.

La commande **pvmove** permet de déplacer les PE d'un PV vers un autre. Il s'agit ici pour vous de déplacer le contenu du PV **/dev/sdb3** vers **/dev/sdb2**. **/dev/sdb3** contient 1536 PE d'utilisés. Il contient tous les LE du LV **data02**.

```
# pvdisplay -m /dev/sdb3
  --- Physical volume ---
  PV Name               /dev/sdb3
  VG Name               vg01
  PV Size               10,00 GB / not usable 748,50 KB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2559
  Free PE               1023
  Allocated PE          1536
  PV UUID               GwkOvR-DOD0-vpA1-zkVk-1Yb2-gcj3-8HbTl6

  --- Physical Segments ---
  Physical extent 0 to 1535:
    Logical volume      /dev/vg01/data02
    Logical extents     0 to 1535
  Physical extent 1536 to 2558:
    FREE
```

Vérifiez si le volume physique **/dev/sdb2** dispose d'assez de place pour accueillir le contenu de **/dev/sdb3**. Il reste 1537 PE dans ce dernier, c'est donc possible.

```
# pvdisplay /dev/sdb2
  --- Physical volume ---
  PV Name               /dev/sdb2
  VG Name               vg01
  PV Size               10,00 GB / not usable 589,00 KB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2561
  Free PE               1537
  Allocated PE          1024
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

Déplacez le PV **/dev/sdb3** vers le PV **/dev/sdb2**. Vous pouvez utiliser le paramètre -v pour suivre l'avancement. Notez que l'opération s'effectue alors qu'aucun système de fichiers n'est démonté :

```
# pvmove -v /dev/sdb3 /dev/sdb2
    Wiping cache of LVM-capable devices
    Finding volume group "vg01"
    Found volume group "vg01"
    Found volume group "vg01"
    Checking progress every 15 seconds
  /dev/sdb3: Moved: 4,6%
...
/dev/sdb3: Moved: 97,1%
  /dev/sdb3: Moved: 100,0%
    Found volume group "vg01"
    Found volume group "vg01"
    Loading vg01-data02 table
    Suspending vg01-data02 (253:1) without device flush
    Suspending vg01-pvmove0 (253:2) without device flush
    Found volume group "vg01"
    Found volume group "vg01"
    Found volume group "vg01"
    Resuming vg01-pvmove0 (253:2)
    Found volume group "vg01"
    Resuming vg01-data02 (253:1)
    Found volume group "vg01"
    Found volume group "vg01"
    Removing temporary pvmove LV
    Writing out final volume group after pvmove
    Creating volume group backup "/etc/lvm/backup/vg01" (seqno 9).
```

Vérifiez maintenant l'état du groupe de volumes :

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 1

  PV Name               /dev/sdb3
  PV UUID               GwkOvR-DOD0-vpA1-zkVk-1Yb2-gcj3-8HbTl6
  PV Status             allocatable
  Total PE / Free PE    2559 / 2559
```

Le second PV de **vg01** est entièrement libre. Il est alors maintenant possible de le supprimer du VG.

e. Réduire un groupe de volumes

La commande **vgreduce** permet de retirer un ou plusieurs PV d'un groupe de volumes. Pour cela, il faut tout d'abord que les PV en question soient vides : leurs PE doivent être entièrement libres. C'est le cas de **/dev/sdb3** que vous allez retirer du VG **vg01** :

```
# vgreduce vg01 /dev/sdb3
  Removed "/dev/sdb3" from volume group "vg01"@
```

Contrôlez que le VG ne contient plus ce PV :

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 1
```

6. Supprimer un groupe de volumes

a. Étapes

Pour supprimer un groupe de volumes, vous devez suivre les étapes suivantes :

- Démonter tous les systèmes de fichiers des LV associés.
- Supprimer tous les volumes logiques avec **lvremove**.
- Retirer tous les volumes physiques du VG avec **lvreduce**.
- Détruire le VG avec **vgremove**.

Vous allez détruire le groupe de volumes **`vg01`**.

b. Supprimer un volume logique

Démontez **`data01`** et **`data02`** :

```
# umount /data01
# umount /data02
```

Supprimez les volumes logiques avec **lvremove** :

```
# lvremove /dev/vg01/data01 /dev/vg01/data02
Do you really want to remove active logical volume "data01"? [y/n]: y
  Logical volume "data01" successfully removed
Do you really want to remove active logical volume "data02"? [y/n]: y
  Logical volume "data02" successfully removed
```

c. Retirer tous les volumes physiques

Utilisez la commande **vgreduce** avec le paramètre `-a` pour `ALL` :

```
# vgreduce -a vg01
  Can’t remove final physical volume "/dev/sdb2" from volume group
"vg01"
```

Remarquez que la commande **vgreduce** laisse toujours au minimum un PV dans le VG, car il faut au moins un PV pour constituer un VG.

d. Détruire un groupe de volumes

Utilisez la commande **vgremove** pour détruire un groupe de volumes :

```
# vgremove vg01
Volume group "vg01" successfully removed
```

Vérifiez que les fichiers et répertoires associés ont disparu :

```
# ls /dev/vg01
ls: ne peut accéder /dev/vg01: Aucun fichier ou dossier de ce type
```

Enfin, la commande **vgdisplay** ne retourne plus rien :

```
# vgdisplay
```

e. Supprimer un volume physique

Les deux volumes physiques peuvent maintenant être détruits puisqu'ils ne sont plus utilisés. Vous pouvez détruire les informations contenues dans le volume avec la commande **pvremove**. Cependant, si vous détruisez la partition via `fdisk` ou que vous créez un système de fichiers dessus, l'effet est le même.

7. Commandes supplémentaires

Vous n'avez eu qu'un bref aperçu des possibilités du LVM. De nombreuses autres commandes existent dont :

- **pvchange** : modifie l'état d'un volume physique, par exemple pour interdire l'allocation d'extensions physiques sur ce volume.
- **pvresize** : redimensionne un volume physique si sa partition ou disque d'origine a été agrandie ou réduite.
- **pvscan** : recherche tous les volumes physiques présents sur tous les supports de stockage du système.
- **vgchange** : modifie les attributs d'un groupe de volumes, pour l'activer ou le désactiver par exemple, mais aussi pour modifier les valeurs maximales de PV et de PE, ou pour interdire son agrandissement ou sa réduction.
- **vgscan** : recherche tous les groupes de volumes sur tous les supports.
- **vgrename** : renomme un groupe de volumes.
- **vgmerge** : regroupe deux groupes de volumes en un seul.
- **lvresize** : redimensionne un volume logique, équivaut tant à **lvextend** qu'à **lvreduce**.
- **lvchange** : modifie les attributs d'un volume logique.
- **lvrename** : renomme un volume logique.