

Marc SCHAEFER
Epervier 15
2053 Cernier
Ing. inf. dipl. EPFL

Conseil et réalisations en informatique libre http://www.cril.ch/
--

Email : schaefer@alphanet.ch
Tél. : +41 32 841 40 14
Fax : +41 32 841 40 81
Mobile : +41 79 502 56 92

Support de cours de préparation de l'examen **LPI-101**

Marc SCHAEFER
schaefer@alphanet.ch

C+R
Informatique libre

25 avril 2008

Avant-propos

Ce cours est basé sur mon expérience d'administration système UNIX depuis 1987, ainsi que de mes expériences faites dès 1995 sur GNU/Linux et en particulier avec **Debian GNU/Linux**, un système de qualité, à la fois à but personnel et en entreprise, comme conseiller indépendant, ingénieur système et développeur.

Il est également basé sur mes expériences de formateur en logiciels libres et notamment sur GNU/Linux, notamment dans le cadre des formations postgrades ES en cours du soir à l'**ESNIG** à Neuchâtel depuis 2000, ainsi que de nombreuses formations que j'ai proposées en entreprise ou dans le cadre d'associations comme **/ch/open** ou le **GULL**.

Il est également basé sur la lecture de documentations systèmes (page man, info), ainsi que de ressources Internet.

Réalisé dans le but d'une certification, le cours tente de suivre exactement le programme obligatoire, de donner les points essentiels, tout en proposant un contexte solide et des exemples à jour. J'essaie absolument d'éviter le bachottage uniquement en vue de la certification : le but est que la personne qui suit ce cours puisse en bénéficier largement pour son travail de tous les jours, notamment en apprenant comment trouver l'information et analyser les problèmes.

Ce cours écrit s'accompagne normalement d'extraits d'autres cours, dont les droits sont en général également en licence FDL, parfois co-écrits par d'autres auteurs. Il insiste surtout sur les compléments nécessaires pour la certification.

Bon apprentissage !

Licence et droits d'auteurs

Ce cours est ©2006-2007 CRIL - Marc SCHAEFER. Vous avez cependant le droit de le copier, transmettre, modifier et redistribuer, dans la mesure où vous respectez les termes de la licence GFDL et considérez l'invariant (les 2 premières pages).

Si vous ne désirez pas accepter les termes de la licence, je vous donne malgré tout le droit de consulter ce cours sans restrictions (ce qui devrait être évident !)

Dans tous les cas, vous devez accepter le fait que je décline toute responsabilité quant à l'utilisation que vous pourriez faire de ce cours et ne m'engage en rien à ce propos.

Marc SCHAEFER
Ing. inf. dipl. EPFL
Conseil et réalisations en informatique libre (CRIL)
<http://www.cril.ch/>

Table des matières

1	Architecture et matériel	4
1.1	BIOS	5
1.2	Cartes modems et son	6
1.3	Périphériques SCSI	10
1.4	Configuration de cartes d'expansion	15
1.5	Périphériques de communication	17
1.6	Périphériques USB	19
2	X Window System	20
2.1	The X Window System	21
2.2	Configuration de XFree86	23
2.3	Configuration d'un gestionnaire de connexion	25
2.4	Configuration d'un gestionnaire de fenêtre et des ressources	27
2.5	Travail distant et sécurité	28
3	Installation et gestion des packages	29
3.1	Définition du partitionnement	30
3.2	Configuration du démarrage	32
3.3	Compilation et installation locale depuis la source	35
3.4	Gestion des bibliothèques dynamiques	36
3.5	Système de packaging Debian	37
3.6	Système de packaging Red Hat	40
4	Périphériques et standard de hiérarchie FHS	41
4.1	Création de partitions et de systèmes de fichiers	42
4.2	Maintenir l'intégrité des systèmes de fichiers	46
4.3	Montage et démontage	49
4.4	Gestion des quotas	51
4.5	Gestion des permissions d'accès aux données	53
4.6	Propriétaires et groupes	54
4.7	Liens symboliques et durs	55
4.8	Trouver et placer les fichiers	56
5	Commandes UNIX et GNU	58
5.1	Travailler en ligne de commande	59
5.2	Filtres de traitement de texte	61
5.3	Gestion des fichiers	63
5.4	Fichiers, pipes et redirections	65
5.5	Créer, surveiller et tuer des processus	66
5.6	Modification des priorités d'exécution	67
5.7	Recherche textuelle avec expressions régulières	69
5.8	Edition de texte avec vi	70
6	Corrigé des exercices	71

1. Architecture et matériel

Contenu du chapitre

- introduction générale au support du matériel sous GNU/Linux
- configuration du BIOS, de cartes modems et son, de périphériques SCSI, de cartes d'expansion
- configuration de périphériques de communication (y compris accès à Internet)
- configuration de périphériques USB

Buts du chapitre

- savoir quel sous-système gère quel type de périphériques
- savoir comment déterminer l'identification d'un périphérique
- savoir comment configurer manuellement certains périphériques
- savoir où trouver l'information (système, Internet)
- connaître les détails spécifiques pour certains bus et types de périphériques

support de cours additionnel : **cours Matériel**

Ce premier chapitre traite principalement du matériel et de la place de celui-ci dans GNU/Linux, ainsi que des outils et moyens à disposition pour déterminer le type de périphérique et son identification, activer et configurer celui-ci. Des cas pratiques sont étudiés. Le démarrage bas niveau et la géométrie des disques sont également traités.

BIOS – 1.1

Résumé des concepts importants

2

- différence entre CHS/LBA/LARGE (mapping incompatible entre nombre de blocs et (cylindre, tête, secteurs)) et raison de l'emploi du mode LBA (> 1024 cylindres) pour des disques jusqu'à 8 GB
- activation et désactivation de périphériques intégrés dans le BIOS et raison (interruptions libérées)
- configuration du mode d'erreur du BIOS (p.ex. serveur sans clavier)
- configuration du démarrage (disque, CD, floppy, etc)
- /proc, dmesg, hdparm, ...

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/Large-Disk-HOWTO.html>

Exercices

1. déterminez le mode de mapping actuel via le log du kernel (indication : dmesg)
2. activez et désactivez certains périphériques (p.ex. port parallèle) et vérifiez leur présence/absence dans le système
3. vérifiez que vous pouvez démarrer sans clavier
4. vérifiez que le système ne démarre *que* sur le disque-dur
5. donnez deux méthodes pour déterminer la mémoire totale disponible dans le système (indication pour la 2^e méthode : grep et quelque chose dans /proc)
6. consultez la page man de hdparm et :
 - (a) consultez la géométrie du disque
 - (b) déterminez l'état de consommation / alimentation / économie d'énergie du disque
 - (c) vérifiez si le DMA est activé, changez l'état et testez la performance
 - (d) comment rendre le disque en lecture seulement (ne pas le faire)
7. en consultant les lectures supplémentaires ci-dessus, répondez aux questions suivantes :
 - (a) avec votre version du kernel (uname -r), avez-vous des problèmes avec des grands disques ?
 - (b) si le kernel n'a pas de problème, quels peuvent être les autres problèmes ?
 - (c) à quoi peut servir l'option lba32 de LILO ? que faut-il faire avec un grand disque sans l'option lba32 ?

Cartes modems et son – 1.2

Modems sous GNU/Linux : principes généraux

- soit modem série connecté à un UART (port série RS-232C/V24-V28)
- soit modem interne (intégré à la carte-mère ou carte PCI/ISA)
 - soit émulant un UART supplémentaire
 - soit pilote spécifique, voire Winmodem
- soit modem externe USB
 - compatible CDC/ACM (Communication Device Class, Abstract Control Model), via pilote `acm.o`
 - incompatible
- les *modems* ADSL ou câble ne sont pas traités ici

Compatibilité

bonne pour les UARTs, bonne en USB compatible CDC, mauvaise à bonne pour les Winmodems

3

Winmodem

Un modem simplifié sans intelligence, qui est géré par le système hôte (DSP). Utilise plus de ressources, pilote propriétaire (s'il existe). Moins cher en général.

Configuration

Le port série standard est en général configuré automatiquement. Parfois il faut faire appel à `setserial` puis s'arranger pour que le système sauvegarde cette configuration.

S'il s'agit d'une carte ISA PNP, l'outil `isapnp` est nécessaire. Le BIOS PnP peut aussi faire la configuration automatiquement.

Périphérique

`/dev/ttyS0`, ou `/dev/ttyI0` (émulation ISDN), ou encore
`/dev/usb/ttyACM0`,...

Dial-in

Réception d'appel par GNU/Linux via contrôle de `tty` (un `getty`), p.ex. `getty`, `agetty`, `mgetty`.

Le tout est géré par `init`, configuré dans `/etc/inittab`. Après changement lancer `telinit q` pour recharger la configuration.

Dial-out

Fonctionnement typique d'un accès à Internet : ouverture du périphérique (p.ex. `/dev/ttyS0`), verrouillage (p.ex. fichier dans `/var/lock`), composition et interactivité (`chat`), lancement du programme nécessaire (p.ex. `pppd`).

Généralement :

- programme de terminal : `minicom`, simplifié : `cu`
- accès Internet : `pppd`, `wvdial`, `wvdialconf`
- automatique : `dip`, `diald`, `pppd`
- historique : UUCP, SLIP/CSLIP, FTN

Carte son

L'ancienne méthode de configuration est via l'utilitaire `ncurses` (mode texte) `sndconfig`. Il détectera la carte son et proposera une configuration par défaut, ou un mode manuel où l'on choisit la carte son.

6

Les modifications aux fichiers `/etc/isapnp.conf` et `/etc/conf.modules` (ou `/etc/modules.conf`) sont automatiques.

Un test automatique est effectué.

largement obsolète aujourd'hui avec les autodétecteurs

Résumé des concepts importants

- compatibilité : différence modem-UART ou Winmodem (et où trouver les pilotes éventuellement)
- IRQ, E/S et DMA uniques et correctes
- cartes PNP : configuration ISA PNP
- configuration carte son de base
- configuration modem pour dial-up (y compris PPP/CSLIP/SLIP) et dial-in (getty)
- `setserial`, `getty`, `agetty`, `mgetty`, `/etc/inittab`, `telinit q`, `minicom`, `sndconfig`, `isapnp`, `pnpdump`, `/etc/isapnp.conf`, `pppd`, `slmodem`.

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/Modem-HOWTO.html>
- <http://linmodems.org/>
- <http://www.tldp.org/HOWTO/Winmodems-and-Linux-HOWTO.html>
- <http://www.linux-usb.org/USB-guide/x332.html>

Exercices

1. quelle est la différence entre un modem interne, un modem externe sur port série (RS-232/V24-V28) et un modem USB ?
2. où trouver les pilotes pour un Winmodem propriétaire ?
3. déterminez le nom de périphérique du modem et configurez-le pour 115 kBit/s (`setserial`)
4. activez le dial-in sur ce modem (`/etc/inittab`, `telinit q`) et vérifiez que le processus `getty` concerné tourne (utilisez plutôt `agetty` de préférence), comprenez les différents arguments !
5. comment créer les périphériques (fichiers spéciaux de type périphériques) UNIX pour 4 modems USB compatibles CDC ?
6. comment créer une configuration `minicom` ?
7. quel utilitaire peut-être utilisé pour configurer simplement une carte son ?

Périphériques SCSI – 1.3

Principes généraux

8

- protocole très répandu !
- SCSI Parallel Interface (SPI)
 - 50 (*narrow*) ou 68 broches (*wide*)
 - 8 respectivement 16 périphériques (SCSI ID 0-7 ou 0-15)
 - plusieurs bus incompatibles électriquement
 - vitesses de 1 à 320 MByte/s, négociable pour chaque équipement
 - terminaison de bus nécessaire à chaque extrémité, y compris demi-bus
 - longueurs de bus variables
 - Host Adapter (HA), carte SCSI
 - ID unique (SCSI ID) pour chaque périphérique. Le HA également !
 - concept de LUN
 - nombreux types de périphériques

SCSI est un protocole très général qui est beaucoup plus répandu que l'on pourrait penser : il est présent par exemple dans l'interface 50 ou 68 broches SCSI parallèle bien connue (SCSI Parallel Interface, SPI), mais aussi dans le bus USB notamment pour les périphériques de stockage, FireWire (IEEE-1394), FibreChannel, iSCSI (SCSI sur TCP/IP), etc.

L'interface SCSI parallèle (SPI) propose deux *largeurs* de bus : 50 (*narrow*, 8 bits) ou 68 broches (*wide*, 16 bits) pour 8 respectivement 16 périphériques (SCSI ID 0-7 ou 0-15). Le bus large est, à même fréquence de bus, 2 fois plus rapides pour les transferts de données.

Electriquement, ces bus sont soit *Single-Ended*, *Differential* ou aujourd'hui *Low-Voltage Differential (LVD)*. Le type de bus définit la vitesse maximum (fréquence max) du bus, qui va de 1 à 320 MByte/s aujourd'hui en tenant compte de la largeur de bus. La vitesse est négociée pour chaque équipement. Une terminaison de bus nécessaire à chaque extrémité, y compris demi-bus (p.ex. un bus interne narrow et un bus externe wide : la carte doit être terminée uniquement sur les 8 bits supérieurs : *High On*, *Low Off* puis les deux bus doivent être terminés à leur extrémité (p.ex. sur un périphérique). Les longueurs des bus (mesurés depuis les extrémités les plus longues) sont variables suivant la vitesse configurée et le type de bus : p.ex. 12m pour LVD, 3m pour Ultra-SCSI (40 MByte/s en wide, 20 MByte/s en narrow).

Si des équipements plus anciens sont connectés au bus, le bus entier passe au fonctionnement électrique compatible (p.ex. SE) si supporté, puis négocie une vitesse plus faible pour cet équipement.

Chaque périphérique a un ID unique (SCSI ID), y compris le HA, configuré par des cavaliers ou par logiciel ; un protocole d'assignation automatique (SCAM, *SCSI Configured Auto Magically*) n'a jamais été déployé.

Les types de périphériques supportés sont : disque, cassette, scanner, CD-ROM, graveur, communi-

cation, processeur, etc. Des périphériques complexes peuvent être décomposés en sous-périphériques adressés par LUN (Logical Unit) : p.ex. changeur de cassette

BIOS SCSI

- liste des périphériques avec leur type, description courte et ID (évt. aussi LUN)
- possibilité de modifier le SCSI ID du HA (usuellement 7 sur les cartes SCSI des PC)
- possibilité de configurer le SCSI ID/LUN de démarrage
- possibilité de modifier les paramètres des périphériques un à un (vitesse, largeur de bus, ignorer, etc)
- possibilité de modifier les paramètres de terminaison du bus
- fonctions de vérification et de formatage bas-niveau pour les disques

BIOS système

Configuration ordre IDE/SCSI

Le PC accède au bus SCSI via un Host Adapter (HA), soit une carte SCSI, qui comporte en règle générale un BIOS spécifique qui permet de définir les paramètres de négociation (vitesse, largeur de bus, etc) pour chaque périphérique, configurer le HA lui-même (numéro ID SCSI, terminaison, démarrage ID/LUN) et offre des outils de test et de formatage pour les disques.

Le BIOS système permet parfois de configurer l'ordre de démarrage entre SCSI et IDE.

SCSI et (GNU/Linux)

10

- logiciellement Linux voit en général les disques SATA et graveurs ainsi que les périphériques de stockage USB comme des périphériques SCSI (/dev/sda1, /dev/scd0, /dev/st0, /dev/sg0, ...) (via p.ex. `ide-scsi.o` ou `usb-storage.o`).
- IDE/ATA : pour ce faire ajouter `hdc=ide-scsi`, pour ici un graveur sur /dev/hdc, à la ligne de commande du kernel (sera accédé via /dev/scd0, et pour le gravage transparent via /dev/sgX)
- option de kernel `max_scsi_luns=1` pour problème au démarrage
- on peut ajouter/supprimer un périphérique dynamiquement (hotplug) via `echo "scsi add-single-device 0 1 2 3" >/proc/scsi/scsi` ("Host Channel Id Lun"; voir aussi `remove-single-device`)
- les outils `scsi_info`^a permettent de visualiser des paramètres des disques, formater ou faire d'autres opérations spécifiques.

^anom réel de la commande : `scsiinfo`

Exemple de `/proc/scsi/scsi` :

```
machine:~# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: TSSTcorp Model: CD/DVDW SH-W162C Rev: TS10
  Type:    CD-ROM                               ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 00 Lun: 00
  Vendor: MAXTOR   Model: ATLAS10K4_36SCA Rev: DFV0
  Type:    Direct-Access                       ANSI SCSI revision: 03
Host: scsi1 Channel: 00 Id: 03 Lun: 00
  Vendor: MAXTOR   Model: ATLAS10K4_36SCA Rev: DFV0
  Type:    Direct-Access                       ANSI SCSI revision: 03
```

On voit ici deux bus virtuels (un formé probablement de `ide-scsi`, et un d'un HA SCSI avec 2 disques-durs).

Résumé des concepts importants

- BIOS SCSI
- BIOS système : ordre de démarrage SCSI/IDE/SATA
- kernel : options de démarrage (LUNs, ide-scsi)
- outils SCSI spécifiques
- `scsi_info`, `/proc/scsi`, `/proc/scsi/scsi`, `ide-scsi.o`, `/etc/fstab`,
`/dev/scd0`, `/dev/cdrom`, `cdrecord`.

11

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html>
- <http://www.tldp.org/HOWTO/SCSI-2.4-HOWTO/>
- <http://www.paralan.com/sediff.html>
- <http://www.scsita.org/>

Exercices

1. qu'y a-t-il dans `/proc/scsi` ?
2. combien de périphériques peuvent tenir place sur un bus SCSI *narrow* ?
3. comment rendre possible le gravage sur un graveur IDE/ATA ? Que devez-vous faire ensuite pour pouvoir tout de même monter des CD-R comme avant ?
4. vous démarrez votre système, et lors de la détection SCSI, un périphérique est vu à tous les LUNs et/ou bloque, que proposez-vous ?
5. si vous enlevez le disque 2 SCSI ID, et qu'il reste les 0 et 3, que se passe-t-il pour le nom usuel du disque 3 ?
6. comment déterminez-vous le nom de périphérique pour `cdrecord` ?
7. comment obtenez-vous des informations sur un disque SCSI ? (proposez 2 commandes)

Configuration de cartes d'expansion – 1.4

Résumé des concepts importants

12

- détection par le BIOS
- configuration automatique (PCI, PCMCIA, ISA PNP), voire manuelle (ISA pur)
- éviter le partage de ressources (IRQ p.ex.) dans le cas général
- `lspci`, `/proc/pci`, `/proc/dma`, `/proc/interrupts`, `/proc/ioports`, `pnpdump`, `isapnp`, `/etc/isapnp.conf`, `setpci`.

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/PCI-HOWTO.html>
- <http://www.faqs.org/docs/Linux-HOWTO/SMP-HOWTO.html>
- <http://www.roestock.demon.co.uk/isapnptools/isapnpfaq.html>

Exemple isapnp :

```
mls-firewall:~# cat /proc/isapnp
Card 1 'EDI0216:PLUG & PLAY ETHERNET CARD' PnP version 1.0
  Logical device 0 'EDI0216:Unknown'
    Supported registers 0x2
    Compatible device PNP80d6
    Device is active
    Active port 0x260
    Active IRQ 11 [0x2]
    Active memory 0xd8000ff,0xff
    Resources 0
  Priority preferred
  Port 0x260-0x3e0, align 0x1f, size 0x20, 10-bit address decoding
  IRQ 3,4,5,2/9,10,11,12,15 High-Edge
```

Ensuite, avec `pnpdump`, créer un fichier de configuration squelette pour `/etc/isapnp.conf`, y modifier des paramètres éventuels (comme p.ex. activer ou désactiver des jeux de configuration, choisir une interruption, etc) qui seront lus et appliqués.

Des cartes non PNP ou fonctionnant mal peuvent nécessiter un chargement explicite du pilote ou la spécifications d'options dans `/etc/conf.modules` (ou `/etc/modules.conf`), voire le char-

gement manuel dans `/etc/modules`.

Exercices

1. listez les **interruptions**, les **ports d'E/S**, les canaux **DMA** et les périphériques **PCI** détectés (deux méthodes). Y-a-t-il des interruptions partagées ?
2. comptez le nombre de périphériques PCI (`wc -l`)
3. y-a-t-il des bridges PCI ou ISA ?
4. consultez la page man de `lspci` et :
 - (a) affichez un arbre du bus PCI (`lspci`)
 - (b) listez uniquement les périphériques du fabricant Intel
5. consultez la page man de `setpci` et :
 - (a) déterminez le rôle de cet utilitaire
 - (b) montrez un exemple d'accès à un registre
6. utilisez-vous un XT-PIC ou un APIC ?
7. y-a-t-il des cartes ISA-PNP sur votre machine ?

Périphériques de communication – 1.5

Traité dans la section modem, sauf :

- différence routeur/switch DSL
- `isdn4linux`, émulation `/dev/ttyI`, MSN

13

Différence routeur-switch/modem DSL Un routeur/switch est adressé directement en IP sur Ethernet. On configure simplement la carte réseau comme d'habitude¹, en indiquant l'adresse IP du routeur – configurable par GUI WWW voir la documentation constructeur – comme passerelle par défaut, et en activant éventuellement le DHCP. Cette méthode est un peu plus chère mais la plus simple.

Un modem DSL est accédé soit comme un modem USB (il faut donc des pilotes USB, éventuellement pour ce modem spécifique, peut-être propriétaires voire inexistantes !), soit via Ethernet dans un mode spécial (PPPoE). Dans ce cas, aucune configuration IP n'est nécessaire sur cette interface. Les données IP/PPP sont échangées simplement via Ethernet plutôt qu'USB. Cette variante s'appelle PPP-over-Ethernet (PPPoE) et est plus simple à configurer sous GNU/Linux que la variante USB. Dans les deux cas, c'est `pppd` qui est utilisé pour l'échange de trames IP via PPP.

Emulation de tty en ISDN De manière à utiliser des cartes ISDN PCI ou USB avec des logiciels s'attendant à une interface série et à des commandes AT Hayes, le sous-système `isdn4linux` implémente une couche de compatibilité tty (`isdn-tty`). Les périphériques sont usuellement nommés `/dev/ttyI0` et suivants, supportent le jeu de commande AT.

Certains pilotes supportent le fax ou la voix via ce biais également, encore qu'en général cela sera utilisé pour des connexions numériques X.75 (p.ex. échange de fichiers).

Pour un fonctionnement correct, le MSN (*Multiple Subscriber Number*, le numéro de téléphone) utilisé doit être configuré, soit via des commandes AT, soit via l'outil `isdnctrl`. Enfin, le mode

¹commandes `ifconfig`, fichier `/etc/network/interfaces`, cf LPI-102

synchrone (ippd) n'utilise pas ces émulations de tty.

Périphériques USB – 1.6

Résumé des concepts importants

14

- hiérarchie des pilotes
- types de Host Controller (courant : `usb-uhci.o`, Compaq : `usb-ohci.o`, rapide : `usb-ehci.o`)
- listage : `lsusb`, `usbview`, `/proc/bus/usb`
- recherche de pilote manuelle : `usbmodules`
- hotplug, le kernel lance `/etc/hotplug/pci.agent`,
`/etc/hotplug/usb.agent`; `/etc/init.d/hotplug` au démarrage du système.
- applications USB
 - `gphoto2 --summary`

Classes de pilotes Si les périphériques USB sont en général supportés par des pilotes kernel génériques (p.ex. disque-dur ou USB stick via `usb-storage`), parfois des pilotes kernel spécifiques sont nécessaires, malheureusement certains propriétaires.

Dans deux cas, des pilotes spécifiques ne sont pas nécessaires dans le kernel :

scanners on utilise un pilote en user-space, dans **sane**

caméras les caméras (pas en mode stockage) sont accédés via le pilote user-space **libgphoto**, p.ex. via GIMP ou KDE.

Exercices

1. de quel type(s) de Host Controller dispose votre machine ? (indication : aussi `lsmod`)
2. listez le bus USB

2. X Window System

Contenu du chapitre

- configuration de XFree86 (y.c. serveur de fontes)
- configuration d'un gestionnaire de connexion (Display Manager, DM)
- configuration d'un gestionnaire de fenêtre et des ressources X11
- travail distant et sécurité X11

Buts du chapitre

- connaître la structuration de X
- savoir configurer un serveur X
- savoir configurer un serveur de fontes
- savoir configurer un environnement graphique simple X
- être capable d'ouvrir des sessions distantes (TCP ou tunnel SSH) tout en gérant la sécurité X11

support de cours additionnel : **Guide UNIX**

Ce chapitre traite principalement de X11, le système de fenêtrage multi-plateforme multi-écran multi-application standard ainsi que de la mise en place simple d'un gestionnaire de connexion (y compris pour un serveur de terminaux), de la configuration d'un gestionnaire de fenêtres et des ressources X11, du travail dans l'environnement ainsi que de la sécurité X11.

The X Window System – 2.1

Résumé des concepts importants

- X11
 - réseau
 - client-serveur (inversé !)
 - display-manager, window-manager, session-manager
 - serveur de fontes (polices) optionnel
 - aujourd’hui extensions pour performance
 - pas dans le kernel (mais ...)
- XFree86 (et X.org aujourd’hui) implémente un *serveur X*
- LPI traite de XFree86 3.3.x

Lectures supplémentaires

- http://en.wikipedia.org/wiki/X_Window_System
- <http://www.x.org/>

The X Window System, X, X11, X11R6 est le système de fenêtrage standard sous UNIX. Le nom X vient du fait que le système de fenêtrage précédent s’appelait *w*, comme Window System. Développé originalement au MIT dès 1984, la version utilisée aujourd’hui, X11R6¹, dispose d’une implémentation libre originellement pour plateforme IA32 (i386), mais qui aujourd’hui est multi-plateforme : **X.Org**.

Des implémentations propriétaires existent également.

Le principe général de X repose sur un modèle client-serveur réseau, où le serveur est celui qui offre les services d’accès à l’écran : cela signifie que bien souvent le client tourne sur un serveur UNIX, et le serveur désigne le poste de travail qui affiche les données du client.

Cela signifie aussi que le support de *Terminal Services* est un simple effet de bord du protocole et non pas un ajout après-coup. Sauf peut-être pour quelques applications modernes qui utilisent des extensions locales, comme par exemple l’accès direct au *frame buffer* de l’écran pour des raisons de performance et ne supporteraient pas un mode plus lent.

XDMCP (X Display Manager Control Protocol) offre des services de localisation d’un display-manager, ce qui permet de se connecter graphiquement depuis un terminal X.

Les composants de X sont :

client une application qui désire afficher quelque chose

serveur X ce qui contrôle l’affichage (peut dans certains cas être un client également)

terminal X équivalent de serveur X, en général plutôt dédié et sans applications locales

¹X11R7 annoncé en décembre 2005.

window-manager client particulier qui gère le placement, la décoration des fenêtres et les actions sur celles-ci

display-manager client X spécial, disposant souvent d'un serveur XDMCP.

session-manager gestion des applications lancées, support de sauvegarde/restauration de leur contexte.

desktop-environment combinaison d'un session-manager, d'un window-manager et d'outils spécialisés (comme des menus, une barre des tâches, un gestionnaire de fichiers, etc).

Les fontes peuvent être bitmap ou générées (True Type p.ex.). Elles sont accédées soit du système de fichiers par le serveur X directement, soit via un serveur de polices qui peut être centralisé.

Sous GNU/Linux et la plupart des systèmes d'exploitation, le serveur X n'est pas implémenté dans le kernel mais est une application privilégiée. Une partie de l'implémentation matérielle peut, cependant, se situer dans le kernel (frame-buffer, DGA (Direct Graphics Access), etc).

Configuration de XFree86 – 2.2

Résumé des concepts importants

17

- configuration du fichier `/etc/X11/XF86Config`, via divers outils, comme :
`XF86Setup`, `xf86config`
- adaptation fine avec `xvidtune`
- démarrage/arrêt de diverses manières, p.ex. via `telinit 5` et `telinit 3` si configuré dans `/etc/inittab`
- serveur de fontes `xf86`

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/XFree86-HOWTO/>
- <http://www.linuxfibel.de/xf86setup.htm>

En général, la configuration automatique fonctionne bien. Parfois il faut l'adapter ou la modifier pour différentes raisons. Il y a plusieurs niveaux de configuration :

1. GUI de configuration : en général, elle se borne à choisir parmi des préconfigurations dans le fichier de configuration de X11, `/etc/X11/XF86Config`¹
2. auto-détecteur / auto-installateur
3. interface de configuration Debian
`dpkg-reconfigure xserver-xfree86`²
(peut cohabiter avec des modifications manuelles)
4. configuration avec `XF86Setup` (XFree86 3.x seulement), graphique en Tcl/Tk
5. configuration en mode texte avec `xf86config`
6. configuration **standard** via `XFree86 -configure`³

Parfois il faut modifier le fichier de configuration `/etc/X11/XF86Config` à la main. On peut aussi employer l'utilitaire `xvidtune` pour modifier des paramètres très finement et graphiquement, ou encore `Xfine2`.

Sous GNU/Linux, le serveur X peut être démarré de plusieurs manières suivant la distribution et les besoins :

¹Version 4 : `/etc/X11/XF86Config-4`; actuellement : `xserver-xorg`
`/etc/X11/xorg.conf`

²dès Debian etch et Ubuntu : `dpkg-reconfigure xserver-xorg`

³`xorg -configure`

1. via une entrée dans `/etc/inittab`, par exemple dans le niveau d'exécution 5 :
`telinit 5` pour démarrer
`telinit 3` pour stopper
2. via le lancement d'un service dans le niveau 2 par exemple, voir un des display-managers :
`/etc/init.d/*dm` et `/etc/rc2.d/*dm`, ou manuellement (ici display-manager kdm) :
`/etc/init.d/kdm start` pour démarrer
`/etc/init.d/kdm stop` pour stopper
3. manuellement, via la commande `xinit` ou `startx` :
`startx - :1` pour démarrer l'écran virtuel 1 si le 0 est déjà pris
`xinit - :1` idem, moins spécifique à GNU/Linux
 (sous votre utilisateur, pas root). On peut spécifier dans votre fichier `.xinitrc` quel window-manager et quels clients doivent être spécifiés (aussi `.xsession`).
4. on peut aussi démarrer le display-manager manuellement, simplement en tapant p.ex. `xdm` (display-manager xdm).

Pour arrêter de manière brusque, on peut faire `CTRL-ALT-BACKSPACE`, ce qui tue le serveur X (mais s'il y a un display-manager actif, il le relancera !)

Le font server `xf86` (ou `xfstt` pour True Type) est installé simplement, puis on doit indiquer au serveur X qu'il doit le contacter, puis le redémarrer.

Exemple de configuration :

```
Section "Files"
    FontPath          "unix/:7100" # local font server
    ...
```

Exercices

1. comment est démarré le serveur X dans votre distribution ? (est-ce via `/etc/inittab` ou via `/etc/init.d/*dm` ?)
2. que se passe-t-il si vous tapez `cat /dev/psaux` et que vous déplacez la souris ? (sortez avec `CTRL-C`).
3. démarrez un deuxième serveur X sous *votre* utilisateur :
 - (a) revenez à la console 1 : `CTRL-ALT-F1`
 - (b) connectez-vous sous votre utilisateur
 - (c) lancez un serveur X manuellement sur le 2^e écran virtuel avec : `startx -- :1`
 - (d) passez d'un serveur X à l'autre avec `CTRL-ALT-F7` et `CTRL-ALT-F8`
 - (e) tuez le 2^e avec `CTRL-ALT-BACKSPACE`
4. quel fichier permet d'indiquer à `startx` quel window manager / quelle session démarrer ?
5. un font-server est-il configuré ? sinon installez-le, configurez-le et vérifiez qu'il est utilisé (p.ex. via `XFree86 2>/tmp/start-log`)
6. configurez votre serveur X avec : `xf86config`
7. configurez votre serveur X avec : `XFree86 -configure`
8. adaptez les paramètres avec `xvidtune`

Configuration d'un gestionnaire de connexion – 2.3

Résumé des concepts importants

- gestionnaire de connexion (Display Manager, DM)
- protocole XDMCP
- `/etc/inittab`, `/etc/init.d/*dm`, `/etc/rc?.d/*dm`,
`/etc/X11/{x,k,g}dm/*`

18

Lectures supplémentaires

- <http://www.ltsp.org/>

Le principe général est que si l'on désire un login graphique, on utilise un Display Manager. Les choix sont `xdm`, `kdm`, `gdm` pour respectivement une version simple, KDE et GNOME. Le Display Manager permet de lancer une session sous un utilisateur particulier (quel que soit l'environnement (`window-manager`, `bureau`, etc) désiré et aussi par exemple une session sans échec (*failsafe*) qui comporte en général un simple terminal (`xterm`).

Le Display Manager gère un ou plusieurs écrans de façon fixe (et lance les serveurs X considérés ou y accède (terminaux X)). Il supporte également, optionnellement, un protocole nommé XDMCP, X Display Manager Control Protocol, qui a pour but de permettre à des terminaux et serveurs X d'interroger un DM pour obtenir la liste des DM disponibles (`chooser`) ou d'accéder directement un DM donné sur un réseau.

Le démarrage automatique du DM est géré comme déjà vu dans la section précédente.

On retrouve les configurations dans `/etc/X11/xdm/`, `/etc/X11/gdm/` et `/etc/kde3/kdm/` par exemple.

Exercices

1. activez et désactivez le Display Manager
2. changez le message de bienvenue du Display Manager
3. changez le nombre de couleurs (bitplanes) du Display Manager
4. activez le protocole XDMCP pour votre voisin
5. lancez de façon permanente un deuxième serveur X (fichier Xservers)
6. testez
 - (a) `X -query localhost :1`
 - (b) `X -indirect localhost :1`
 - (c) sous votre utilisateur et sous X11 : `Xnest -query localhost :1`
 - (d) de même pour le XDMCP de votre voisin !
7. désactivez le XDMCP et vérifiez p.ex. avec `netstat -anp`

Configuration d'un gestionnaire de fenêtre et des ressources – 2.4

Résumé des concepts importants

- gestionnaire de fenêtres (Window Manager, WM)
- gestionnaire de bureau / d'environnement (p.ex. GNOME et KDE)
- configuration par utilisateur dans `.Xresources`
- `.xinitrc`, `.Xdefaults`

Lectures supplémentaires

- <http://www.linuxdocs.org/HOWTOs/XWindow-User-HOWTO-6.html>

On peut configurer les paramètres d'affichage dans votre fichier `.Xresources` (ou `.Xdefaults`). Il existe de nombreux Window Managers, comme `fvwm2`, `WindowMaker`, `olwm`, `vtwm`, etc.

Exercices

1. lancez un `xterm` avec la couleur de fond rouge (indication : rouge c'est red)
2. modifiez votre `.Xresources` de manière à ce qu'`xterm` se lance *toujours* avec un fond rouge. Activez durant la session via `xrdb -merge .Xresources`. Faut-il faire un lien symbolique de `.Xdefaults` pour que cela soit actif à chaque session ?
3. quelles autres propriétés sont modifiables ? consultez notamment `man xterm` et aussi `/etc/X11/app-default/XTerm`.
4. installez le package `menu`, le `window-manager blackbox` et malgré la mise en garde modifiez le menu dans `/etc/X11/blackbox/blackbox-menu` et mettez à jour avec `update-menus`.
5. connectez-vous en *failsafe* et lancez `gnome-session`. Quittez. Observez.

Travail distant et sécurité – 2.5

Résumé des concepts importants

- variables `DISPLAY`, `XAUTHORITY`
- commandes `xhost`, `xauth`
- fichier `Xservers`
- `ssh -X user@host`

Lectures supplémentaires

- Guide UNIX, section X Window

Exercices

1. est-ce que votre serveur X local autorise les connexions TCP distantes ? sinon, activez-le et vérifiez que `telnet localhost 6000` accepte une connexion. (indication : cherchez un fichier `Xservers`, p.ex. `/etc/X11/xdm/Xservers`, ou `/etc/kde3/kdm/Xservers`, ou encore pour `gdm` la variable `DisallowTCP` dans le fichier `/etc/gdm/gdm.conf`)
2. autorisez toute la machine de votre voisin à vous connecter sur votre serveur. Expérimentez le lancement de `xclock`, `xterm`.
3. interdisez l'accès par adresse IP, vérifiez.
4. désactivez les connexions distantes et vérifiez avec `telnet localhost 6000`.
5. connectez-vous à la machine de votre voisin via `ssh` et lancez des application X11 sur votre machine via le tunnel SSH. Que valent la variable `DISPLAY` et le cookie d'authentification X11 ? (indication : vérifiez que le serveur SSH autorise `X11Forwarding`, cf `/etc/ssh/sshd_config`).
6. depuis votre session graphique :
 - (a) faites `su -` pour devenir root dans un `xterm`
 - (b) faites `su` pour devenir root dans un `xterm`
 - (c) lancez un `xterm` dans la première session root.
 - (d) lancez un `xterm` dans deuxième session root. Que se passe-t-il ? Consultez le man de `su` pour comprendre la différence.
 - (e) que faire pour que root ait les droits d'accès à votre écran localement ? (trois méthodes : une avec une variable d'environnement, une avec `xauth`, une dernière moins sûre avec `xhost`)

3. Installation et gestion des packages

Contenu du chapitre

- définition d'un schéma de partitionnement
- configuration d'un programme de démarrage
- compiler et installer un programme source
- gérer les bibliothèques dynamiques
- système de packaging Debian
- système de packaging Red Hat

Buts du chapitre

- savoir planifier la structure des systèmes de fichiers et partitions de manière correcte
- savoir configurer LILO et grub
- savoir déployer correctement des logiciels non packagés
- savoir gérer les packages installés

supports de cours additionnels : **cours Installation et packaging**

Ce chapitre traite principalement de la planification d'installation d'un système (notamment partitionnement), de la gestion d'un système installé (y compris adaptation de la configuration des programmes de démarrage courants), notamment l'installation manuelle de logiciels non packagés (depuis leur source : compilation, installation, déinstallation ; y compris bibliothèques partagées), ainsi que bien sûr les outils les plus courants de gestion de packages.

Définition du partitionnement – 3.1

Résumé des concepts importants

- mémoire d'échange (*swap*)
- motivations des choix
 - emplacement de `/boot` (< cylindre 1024)
 - performance suivant type d'application,
 - indépendance
 - limite
 - plusieurs types de fs
- points de montage, partitions, système de fichiers : `/` (racine, root), `/var`, `/home`
- `/bin/mount`, `/sbin/mkswap`, `/sbin/fdisk`, `/sbin/mkfs.*`,
`/sbin/swapon`

Lectures supplémentaires

- <http://www.debian.org/releases/stable/i386/ch06s03.html.en#di-partition>

Une installation de base ne nécessite que deux¹ partitions, la partition racine `/` et une partition de swap (mémoire virtuelle). Le kernel Linux 2.4 est notamment plus performant lorsqu'un swap est disponible : il peut ainsi y déplacer des données non fréquemment utilisées et donc gagner en performance disque (puisqu'les caches disques sont allouées de la mémoire libre).

On peut également subdiviser le système en plusieurs partitions si désiré. Les raisons d'un tel partitionnement plus fin sont :

- maintenir le démarrage sous la limite de 1024 cylindres (disques IDE et anciens BIOS) : petit / séparé de `/usr`, ou carrément partition de démarrage `/boot`.
- performance : p.ex. plusieurs périphériques swap distribués sur plusieurs partitions de disques-durs différents ; séparation des logs et des queues (`/var/log` et `/var/spool`) sur un serveur de mail très chargé (noter que cela augmente le risque de panne !)
- indépendance à la réinstallation ou au changement de distribution : mettre `/home` sur un répertoire séparé
- limiter le risque de casse en cas de problème, ou assurer une limite sans quotas par utilisateurs/groupes
- spécifier des options de montages spéciales pour la performance (`noatime`, `async`) ou pour la sécurité (`noexec`, `nosuid`, etc)
- utiliser plusieurs systèmes de fichiers différents (p.ex. supportés ou non par le boot-loader grub ; ou pour des utilisations variées).

La performance est souvent un problème négligé, mais il est essentiel. Un bon partitionnement peut

¹on peut utiliser un fichier de swap à la place d'une partition (moins efficace) ; on peut aussi se passer totalement de swap dans des cas très particuliers, mais en règle générale il faut l'éviter.

contribuer à améliorer la performance globale du système. Il faut commencer par définir les tâches ou les applications que le système doit exécuter.

`/` peu modifié, données cachées : peut être monté depuis un disque lent.

`/var` varie beaucoup (en particulier `/var/log`) : on a intérêt à le placer sur un disque rapide

`/usr` disque rapide conseillé p.ex. dans le cas d'un serveur d'application (Terminal Server)

La rapidité d'un disque est à la fois facteur de sa vitesse brute (liée à la vitesse de rotation et à sa densité) et de la latence (durée de déplacement des têtes).

Une fois qu'un système est installé, la fragmentation peut diminuer, au fur et à mesure de l'utilisation, la performance d'un système de fichier. Sous GNU/Linux, ce problème est presque inexistant tant que les systèmes de fichiers ne sont pas remplis complètement. C'est une des raisons pour laquelle on peut réserver une partie de la place disque pour le super-utilisateur.

Le partitionnement lui-même se fait avec la commande `fdisk`, ou d'autres implémentations comme `cdisk`.

On initialise une partition de swap avec `mkswap`, et un système de fichiers avec les diverses instances de `mkfs` (p.ex. `mkfs.ext2` ou `mke2fs`, `mkfs.ext3` ou `mke2fs -j`, etc)

Une fois initialisé, une partition de swap est activée avec `swapon`. Un système de fichiers est rattaché à un répertoire existant de l'arborescence via `mount`. Il remplace le contenu éventuel du répertoire temporairement.

De manière à automatiser le montage des systèmes de fichiers et l'activation du swap, on modifie `/etc/fstab`.

Exercices

1. quel est le rôle des options `-a` et `-s` de `swapon` ?
2. quel est le rôle de l'option `-a` de la commande `mount` ?
3. laissez tourner la commande `vmstat` quelques temps et déterminez si le système swappe de manière active.
4. combien de swap est utilisé ? désactivez tous les espaces de swap et comparez.
5. proposez un schéma de partitionnement pour une application particulière et justifiez.

Configuration du démarrage – 3.2

Résumé des concepts importants

- étapes du démarrage
 - Master Boot Record (MBR) : *first stage boot loader*
 - second stage boot loader : LILO ou grub
- `/etc/lilo.conf`, `/boot/grub/grub.conf`^a
- `lilo` (à chaque modification), `grub-install` (une fois)

Lectures supplémentaires

- <http://tldp.org/HOWTO/LILO.html>
- <http://www.cri74.org/linux/howto/grub-howto.html>
- <http://www.tldp.org/HOWTO/Multiboot-with-GRUB.html>
- <http://www.tldp.org/HOWTO/Bootdisk-HOWTO/index.html>
- <http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>

^aDebian et dérivés : `/boot/grub/menu.lst`

Le chargeur de premier niveau (*first stage boot loader*) peut être soit le MBR standard MS-DOS ou celui livré par GNU/Linux (package `mbr` sous Debian), ou encore LILO. Le MBR consiste en les 2 premiers bloc du disque : un bloc de démarrage et un bloc contenant la table des partitions. La taille de bloc est usuellement de 512 bytes.

Le chargeur de deuxième niveau n'est, dans le cas de LILO, nécessaire que si LILO est installé dans une partition démarrable, et non pas dans le MBR. Une partition démarrable est dénotée par le drapeau `A` dans `vdisk` et ne peut être qu'une partition Linux, et pas de swap. Si LILO est installé dans le MBR, le démarrage est possible de n'importe quelle partition Linux ou non Linux (sauf swap), y compris partitions logiques ou d'autres disques.

Si un autre programme de démarrage est préféré (p.ex. `NTLOADER.EXE`, on peut aussi démarrer GNU/Linux avec ce chargeur).

En cas de configurations très particulières (p.ex. démarrage sur un disque que le BIOS ne considère pas pour le démarrage), il faut installer LILO dans le MBR du premier disque. On peut alors, en général, faire en sorte que LILO charge son kernel et l'`initrd` éventuel d'un autre disque.

Exemple de `/etc/lilo.conf` :

```
# BIOS moderne, disque large
lba32

# Eventuellement: correspondance noms Linux <-> numéro pour le BIOS
#disk=/dev/hde
#bios=0x81
#disk=/dev/sda
#bios=0x80

boot=/dev/hda # ici installé dans le MBR
#boot=/dev/hda1 # ici serait installé dans une partition

root=/dev/hda3 # partition de / (racine)

# Choix de boot sector (mise en page différente)
install=/boot/boot-menu.b

map=/boot/map

# 2 secondes
delay=20

vga=normal

# Arguments du kernel (par défaut)
append="ide-scsi=hdc"

default=linux

# Image Linux (pas d'initrd)
image=/vmlinuz
    label=linux
    append="hdc=ide-scsi ide2=autotune"

# Image non Linux
other=/dev/hda4
label=OTHER
```

Comme LILO mémorise la suite des adresses de blocs à charger là où il est installé, il faut le relancer (`/sbin/lilo`) si l'on modifie l'image à charger (kernel, `initrd` ou paramètres de config). Par contre, `grub` utilise le système de fichiers et ne doit pas être relancé lorsqu'on change quelque chose. Comme inconvénient, `grub` ne supporte pas le chargement depuis LVM ou RAID ou des systèmes de fichiers très spécifiques.

En cas de problème de démarrage, on peut démarrer sur un CD, monter les systèmes de fichiers p.ex. sous `/mnt`, puis lancer `lilo -r /mnt` (ou encore `chroot /mnt /sbin/lilo`) de manière à lancer LILO dans le contexte du système installé.

Exercices

1. créez une disquette de démarrage contenant uniquement un kernel, et modifiez les options ad-hoc via `rdev`
2. créez une disquette de démarrage via `SYSLINUX`
3. démarrez un autre système avec `grub` (ou `memtest86`)
4. passez de `grub` à `LILO` pour le démarrage de votre machine
5. mettez un message spécifique et un mot de passe sur votre `LILO`, que devez-vous faire ? que faudrait-il encore faire pour sécuriser la machine ?
6. à quoi sert le package `mbr` ?
7. pouvez-vous démarrer sur le RAID logiciel avec Linux ?

Compilation et installation locale depuis la source – 3.3

Résumé des concepts importants

- télécharger, vérifier, extraire, configurer, compiler, installer
- gzip, gunzip, bzip2, tar
- configure, make, make install

Lectures supplémentaires

- cours installation depuis source

Gestion des bibliothèques dynamiques – 3.4

Résumé des concepts importants

- ldd
- ldconfig, /etc/ld.so.conf,
- LD_LIBRARY_PATH, LD_PRELOAD
- /lib, /usr/lib

25

Exercices

1. quel est le chemin de recherche actuel du système en ce qui concerne les bibliothèques dynamiques ?
2. de quelles bibliothèques partagées (dynamiques) dépend le logiciel /bin/ls ? et /usr/bin/konqueror ? et /sbin/ldconfig ? comparez aussi avec file.
3. lancez /usr/lib/mozilla-firefox/firefox-bin, quel est le problème ? corrigez-le manuellement temporairement. proposez ensuite 3 solutions (variable d'environnement permanente, wrapper, configuration système activée) et testez-les.
4. quel est le rôle de la commande fakeroot ? quel est son principe technique (indication : préchargement d'une bibliothèque d'interception) ?
5. que faire si la configuration système des bibliothèques est sabotée ?

Systeme de packaging Debian – 3.5

Résumé des concepts importants

- /etc/dpkg/dpkg.cfg, /etc/apt/apt.conf, /etc/apt/sources.list
- /var/lib/dpkg/*
- dpkg, dpkg-reconfigure
- apt-get
- alien
- dselect

Lectures supplémentaires

- <http://www.debian.org/doc/manuals/apt-howto/>
- <http://people.debian.org/~debacle/refcard/refcard-fr-a4.pdf>

Les commandes principales :

Commande	description
dpkg -get-selections	obtenir l'état des packages
dpkg -set-selections	configurer l'état des packages
dpkg -l	liste des packages (optionnel : argument de recherche, pouvant contenir des jokers du shell, exemple : dpkg -l '*truc*')
dpkg -s PACKAGE	obtenir des informations sur un package <i>installé</i>
dpkg -i package.deb	installer manuellement un package
apt-get install PACKAGE	installer un package et ses dépendances
dpkg -r PACKAGE	supprimer un package manuellement (optionnellement : --purge, pour supprimer également les configurations)
apt-get remove PACKAGE	supprimer un ou plusieurs packages et ses/leurs dépendances (optionnellement : --purge, pour supprimer également les configurations)
dpkg-reconfigure PACKAGE	reconfigurer avec debconf
apt-get install -f	correction de problèmes de dépendances
debsums -ca	vérifier les fichiers installés

Pour faire des recherches dans des packages non installés, on peut utiliser :

Commande	description
<code>apt-cache search RECHERCHE</code>	recherche dans les sources installables
<code>apt-cache show package</code>	détails sur un package
<code>apt-get update</code> <code>apt-get upgrade</code>	il faut bien sûr que la liste des packages soit à jour ! mise à jour des packages automatisée
<code>apt-get -u dist-upgrade</code>	mise à jour à une distribution
<code>http://packages.debian.org/</code>	recherche d'un fichier dans un package non installé, dans d'autres versions, etc.
<code>http://www.apt-get.org/</code>	packages non officiels
<code>http://volatile.debian.net/</code>	packages changeant souvent (anti-virus, etc)
<code>http://www.backports.org/</code>	ports de versions plus récentes

Les fichiers de configuration :

Commande	description
<code>/etc/apt/sources.list</code>	liste des sources d'installation
<code>/etc/apt/apt.conf</code>	configuration par défaut d'APT

On peut utiliser `apt-setup`¹ (ou `apt-cdrom` directement pour les CD/DVD) pour ajouter des sources d'installation.

Pour reconstruire un package depuis sa source, on télécharge le package source manuellement ou via `apt-get source`, puis on utilise `dpkg-buildpackage` :

```

schaefer@shakotay:~$ mkdir /tmp/tt
schaefer@shakotay:~$ cd /tmp/tt
schaefer@shakotay:/tmp/tt$ apt-get source hello
Reading Package Lists... Done
Building Dependency Tree... Done
Need to get 405kB of source archives.
Get:1 http://archive.alphanet.ch sarge/main hello 2.1.1-4 (dsc) [562B]
Get:2 http://archive.alphanet.ch sarge/main hello 2.1.1-4 (tar) [389kB]
Get:3 http://archive.alphanet.ch sarge/main hello 2.1.1-4 (diff) [15.4kB]
Fetched 405kB in 0s (3090kB/s)
dpkg-source: extracting hello in hello-2.1.1
schaefer@shakotay:/tmp/tt/hello-2.1.1$ cd hello-2.1.1
schaefer@shakotay:/tmp/tt/hello-2.1.1$ dpkg-buildpackage -us -uc \
-rfakeroot
[ ... ]
dpkg-deb: building package 'hello' in `../hello_2.1.1-4_i386.deb'.
[ ... ]

```

¹plus disponible dès Debian etch.

On peut aussi consulter un fichier de package directement sans l'installer :

Commande	description
<code>dpkg-deb -c package.deb</code>	listage du contenu du fichier de package
<code>dpkg-deb -x package.deb /tmp/tt</code>	extraire manuellement
<code>dpkg-deb -I package.deb</code>	informations

`aptitude` est un frontal qui permet d'effectuer les mêmes opérations qu'`apt-get` mais aussi de déterminer, par exemple, les packages installés localement et/ou non maintenus par Debian, en particulier après une mise à jour.

Exercices

- consultez les manpages de `dpkg`, `dpkg-reconfigure`, `alien`, `apt-cache` et `apt-get`
- essayez les diverses commandes proposées ci-dessus.
- consultez quelques fichiers sous `/var/lib/dpkg`
- comment déterminer dans quel package se trouve `ls` (package déjà installé). Et si le package n'était pas déjà installé ?
- déterminez la liste des packages installés.
- quels packages ont `-dev` dans leur nom ?
- examinez l'entête de la sortie de `dpkg -l | more` et expliquez les 2 premiers caractères.
- quelle commande proposeriez-vous pour télécharger chaque nuit les packages candidats à une mise à jour, sans les installer proprement dit ? (indication : `man apt-get`)
- changez le frontend de `debconf` et expérimentez
- téléchargez manuellement un package Debian qui n'a pas ses dépendances satisfaites et installez-le via `dpkg`. Corrigez le problème avec `apt-get install -f`. Est-ce une bonne idée d'installer des packages qui ne sont pas dans Debian standard ? comment en obtenir la liste (aussi p.ex. après une mise à jour) ?
- où trouvez-vous des packages plus récents maintenus pour la version stable ?
- comment déterminez-vous quels packages ne sont pas des packages disponibles dans les listes de sources disponibles ?
- ajoutez une source d'installation (p.ex. un backport) et installez un logiciel.
- que fait l'option `APT::Get::Assume-Yes` ?
- téléchargez la source unstable du package `hello`, et recréez le package binaire. Interrogez le package sans l'installer. Installez le. Vérifiez que la version installée est plus récente.
- à quoi peut-il servir de mettre un package en état **hold** ?
- documentez-vous sur le *pinning*, l'option `-t` d'`apt-get` ainsi que sur les syntaxes `apt-get install package` et `apt-get install package+VERSION`.
- doit-on être root pour générer des packages ? et pour les installer ?
- trouvez, via <http://www.rpmfind.net/>, le package `hello`, convertissez-le en package Debian avec `alien` et installez-le.
- lancez `dselect` et essayez de comprendre plus ou moins comment cette *chose* fonctionne.

Systeme de packaging Red Hat – 3.6

Résumé des concepts importants

- /etc/rpmrc, /usr/lib/rpm/*
- rpm, grep

Lectures supplémentaires

- <http://www.rpm.org/RPM-HOWTO/>
- http://fr.wikipedia.org/wiki/Red_Hat_Package_Manager
- <http://www.rpmfind.net/> – de quoi installer n'importe quoi

27

Commande	description
rpm -qf /bin/ls	déterminer (q, query) dans quel package (f) se trouve /bin/ls
rpm -qpi package.rpm	obtenir des informations (iq) du fichier-package (p) package.rpm sans l'installer
rpm -qpl package.rpm	liste du contenu du package non installé
rpm -i package.rpm	installer un package
rpm -U package.rpm	mettre à jour un package
rpm -e package.rpm	supprimer un package
rpm --rebuild PACKAGE-SOURCE	reconstruire un package
rpm -qa grep package	chercher le package package dans la liste des packages installés
rpm -qi package	informations sur un package installé.
rpm -Va	vérifier le système et déterminer les fichiers manquants.

4. Périphériques et standard de hiérarchie FHS

Contenu du chapitre

- partitionnement, création, montage, démontage et maintenance de systèmes de fichiers
- gestion des quotas
- gestion des permissions d'accès aux données
- gestion des propriétaires et groupes des données
- liens symboliques et durs
- trouver et placer les fichiers au bon endroit

Buts du chapitre

- savoir gérer les données d'un système UNIX, y compris le placement standardisé de celles-ci

supports de cours additionnels : **cours UNIX**

Création de partitions et de systèmes de fichiers – 4.1

Résumé des concepts importants

- système de fichiers
- périphérique de type bloc (*block device*)
- MAJOR, MINOR
- pilote kernel
- partitionnement
- `fdisk`, `mkfs`

29

Un périphérique de type bloc est un périphérique adressé, du point de vue UNIX, par un numéro de bloc. Les blocs sont de taille fixe, usuellement 512 bytes par bloc. On y accède par un fichier spécial de type périphérique bloc :

```
schaefer@voyager:/data/home/schaefer$ ls -l /dev/hda
brw-rw---- 1 root disk 3, 0 Mar 31 15:31 /dev/hda
```

Les numéros 3, 0 correspondent aux numéros de MAJOR et de MINOR. Le MAJOR est en général le type général de périphérique associé à un pilote kernel donné :

```
schaefer@voyager:/data/home/schaefer$ grep ide /proc/devices
3 ide0
22 ide1
```

Quant au MINOR, c'est un point d'entrée dans le périphérique lui-même (p.ex. numéro de disque sur un bus, numéro de partition, etc).

On crée ces entrées (fichiers spéciaux) via la commande `mknod`, ou via le script `MAKEDEV` :

```
root@voyager:/dev # mknod hda b 3 0
root@voyager:/dev # mknod hda1 b 3 1

root@voyager:/dev # ./MAKEDEV isdnmodem
```

Ces entrées figurent réellement sur le disque (et sont créés automatiquement à l'installation de base, voire plus tard à l'installation de packages particuliers) ou dans un système de fichiers dynamique virtuel.

L'outil `fdisk` d'obtenir la table des partitions et la géométrie d'un disque-dur, ainsi qu'en mode interactif de partitionner celui-ci :

```
root@voyager:~ # fdisk -l /dev/hda
```

```
Disk /dev/hda: 60.0 GB, 60060155904 bytes
255 heads, 63 sectors/track, 7301 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	4	32098+	83	Linux
/dev/hda2		5	7299	58597087+	5	Extended
/dev/hda5		5	1098	8787523+	83	Linux
/dev/hda6		1221	3652	19535008+	83	Linux
/dev/hda7		3653	7299	29294496	83	Linux
/dev/hda8		1099	1220	979933+	82	Linux swap

Partition table entries are not in disk order

```
root@voyager:~ # fdisk /dev/hda
```

```
Command (m for help): p
```

```
Disk /dev/hda: 60.0 GB, 60060155904 bytes
255 heads, 63 sectors/track, 7301 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	4	32098+	83	Linux
/dev/hda2		5	7299	58597087+	5	Extended
/dev/hda5		5	1098	8787523+	83	Linux
/dev/hda6		1221	3652	19535008+	83	Linux
/dev/hda7		3653	7299	29294496	83	Linux
/dev/hda8		1099	1220	979933+	82	Linux swap

Les opérations principales de `fdisk` sont :

p créer une partition

1. partition primaire (1-4), étendue (1 max) ou logique (dans la partition étendue)
2. numéro de partition (si primaire ou étendue)
3. cylindre de début de la partition (RETURN pour valeur proposée)
4. dimension de la partition (valeur absolue en cylindres ou tailles comme 16M, 5000M ou 5G, etc)

d supprimer une partition (numéro)

t changer le type de la partition (numéro, type ; l pour liste)

w pour inscrire les changements

q pour quitter

La création d'un système de fichiers se fait via la commande `mkfs` (et ses dérivés : `mkfs.ext3` ou `mke2fs -j`, etc).

Un système de fichiers peut aussi être monté à partir d'un périphérique bloc émulé sur un fichier (pour pour essais, maintenance d'`initrd`, ou chiffrement de données) :

```
% # Création d'un fichier de 16 MB contenant des NULs
% dd if=/dev/zero of=un_fichier bs=1024k count=16
16+0 records in
16+0 records out
16777216 bytes transferred in 0.093541 seconds (179356606 bytes/sec)

% # Création d'un système de fichiers ext3
% mke2fs -j -i 65536 -b 4096 -c -L test -m 10 un_fichier
mke2fs 1.35 (28-Feb-2004)
un_fichier is not a block special device.
Proceed anyway? (y,n) y
Filesystem label=test
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
256 inodes, 4096 blocks
409 blocks (9.99%) reserved for the super user
First data block=0
1 block group
32768 blocks per group, 32768 fragments per group
256 inodes per group

Checking for bad blocks (read-only test): done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.

% # Montage de ce système de fichiers
# mkdir /mnt/test
# mount -o loop un_fichier /mnt/test

# df /mnt/test
/tmp/un_fichier          16336          4124          10576    29% /mnt
```

Exercices

1. dans l'exemple ci-dessus, quels droits doit-on avoir pour créer le système de fichiers ? monter le système de fichiers ? et dans le cas général (pourquoi) ?
2. créez plusieurs systèmes de fichiers différents (p.ex. `ext2`, `ext3`, `reiserfs`, `vfat`, `xfs`) et comparez

p.ex. les méta-données POSIX. Utilisez pour ce faire l'option `-t` de `mkfs`, ou les utilitaires spécifiques `mkfs.*`

3. comment faire une vérification simple du média avant de créer le système de fichiers ?

Maintenir l'intégrité des systèmes de fichiers – 4.2

Résumé des concepts importants

- démontage non propre, vérification, journalisation
- espace disque et inodes
- du, df
- fsck, e2fsck
- mke2fs
- tune2fs, debugfs, dumpe2fs

Au démarrage du système, le fichier `/etc/fstab` est consulté et les systèmes de fichiers non marqués `noauto` sont montés automatiquement. S'ils n'ont pas été démontés proprement, une passe de vérification et d'auto-correction simple est lancée (`fsck`, *file system check*) et ils ne sont montés qu'après.

Si une défectuosité qui ne peut être réparée automatiquement est détectée, une réparation manuelle est à faire (le système reste en mode mono-utilisateur), p.ex. avec `e2fsck /dev/hda3` pour réparer un système de fichiers `ext2` ou `ext3` situé sur la partition 3 du maître du primaire IDE/ATA.

Dans tous les cas, par défaut, une vérification automatique est faite tous les 190 jours ou tous les 30 montages (30 redémarrages en général).

Les seuls cas où le lancement de `fsck` est possible est si un système de fichiers est monté `read-only` (`mount / -o remount,ro`) ou s'il n'est pas monté. Le cas de `/` est spécial : il est monté en lecture seulement.

En cas de panne franche du système (p.ex. panne de courant), une partie des données dans le cache n'aura pu être écrite. Le rôle de `fsck` est d'éviter l'apparition de données incorrectes (p.ex. en tronquant les fichiers au dernier endroit encore correct) en contrôlant l'état des métadonnées. Sur de grands systèmes de fichiers, cette opération peut prendre un temps considérable.

En conséquence, les systèmes de fichiers journalisés ont fait leur apparition (p.ex. `ext3`) : ils permettent d'assurer une certaine atomicité des opérations sur les métadonnées (optionnellement aussi sur les données) via l'utilisation d'un journal, qui est rejoué en cas de panne, très rapidement : cela évite l'utilisation de `fsck` et une perte de temps.

Il est donc recommandé, sur un système de fichiers journalisé d'une certaine taille de supprimer

la vérification automatisée régulière, via la commande `tune2fs`, de manière à éviter des temps d'attente en cas de redémarrage, et de *planifier* un `fsck` manuel régulier (p.ex. tous les 3 mois), même sur un système de fichiers journalisé ! Ce n'est pas obligatoire, mais cela vous donnera un sentiment de sécurité supplémentaire. Ce sera aussi l'occasion de vérifier que votre système redémarre – on oublie souvent ce qu'est le démarrage d'un système GNU/Linux ... :)

La commande `fsck` prend les arguments :

- A** vérifier tous les systèmes de fichiers listés dans `/etc/fstab`
- t** restreindre à un type pour `-A`
- C** *progress bar*
- a** réparer automatiquement
- r** réparer interactivement

La commande `tune2fs` permet d'obtenir ou de modifier les paramètres d'un système de fichiers.

- L nom** configurer le label (nom) du système de fichiers
- l** lister les informations du superblock du système de fichiers
- i duree** configuration de l'intervalle entre les vérifications automatiques (0 : jamais).
- c nombre** configuration du nombre de montage entre les vérifications automatiques (0 : jamais).

La commande `dumpe2fs` permet d'obtenir des informations internes d'un système de fichiers.

- b** afficher la liste des blocks mauvais (*bad blocks*)
- h** afficher uniquement les informations du superblock

La commande `debugfs` permet de consulter les métadonnées directement, p.ex. pour restaurer un fichier effacé – ce qui est très difficile par défaut sur un système UNIX. Ses commandes principales sont :

- help** aide sur les commandes
- stat** informations sur un objet du fs
- logdump** ext3 : sauver le journal dans un fichiers

L'espace disque disponible est résumé via la commande `df` (on peut y donner un argument de répertoire, la sortie sera alors limitée au système de fichiers sur lequel se répertoire se trouve). Les systèmes de fichiers POSIX ont une deuxième limite en plus de la place disponible : le nombre d'inodes (de fichiers et de répertoires) maximum. On peut visualiser ces valeurs avec `df -i`. L'option `-a` montrera également les systèmes de fichiers virtuels (p.ex. `/proc`).

La commande `du` permet d'obtenir la taille utilisée par des fichiers et répertoires, arrondie aux blocs utilisés, aussi récursivement. Ses options sont :

- a** tous les objets sont affichés (pas seulement les répertoires)
- b** afficher la taille non arrondie au blocs, et en unité bytes
- c** effectuer un total
- h** format plus humain

Ses arguments sont les objets que l'on désire compter (répertoires, fichiers, etc).

Exercices

1. vérifiez un système de fichiers non monté avec `fsck` Vous pouvez forcer la vérification si `fsck` pense que le fs est propre.
2. comment réparer automatiquement un fs ? comment répondre oui à toutes les questions ?
3. comment réparer un fs ext3 alors que le superblock a été détruit ?
4. obtenez les informations du superblock d'un système de fichiers à l'aide de la commande `tune2fs` (et/ou `dumpe2fs`)
5. changez les paramètres de vérification de manière à ce que la vérification automatique ne soit plus faite sur la racine.
6. peut-on ajouter une journalisation sur un fs ext2 ? comment ?
7. combien d'inodes reste-t-il sur / ? combien de blocs ? combien de bytes ? et en mégabytes ou gigabytes ? (humain)
8. listez *tous* les systèmes de fichiers montés, y compris les systèmes de fichiers virtuels cachés.
9. ne listez que les systèmes de fichiers d'un type donné.
10. combien de place prend le répertoire `/boot` ?
11. trie les sous-répertoires de `/usr` par taille décroissante

Montage et démontage – 4.3

Résumé des concepts importants

- montage et point de montage
- `/etc/fstab`
- `mount`, `umount`

31

Un système de fichiers est une organisation de données et de métadonnées (permissions, nom de fichier, propriétaire, ACL, attributs étendus, *fork*) organisée en générale hiérarchiquement (arbre en général). Sous UNIX, on y accède en rattachant le *block device* qui contient le filesystem à l'arborescence système à un point de montage système via le concept de montage (*mounting*).

Un système de fichiers ne peut être démonté que s'il n'est plus utilisé. La commande `lsof` ou `fuser -m` permet de déterminer les processus utilisant un fs.

Les options de montage sont :

noauto pas de montage au démarrage

exec on peut exécuter des programmes depuis ce fs

suid le SUID est actif

user un utilisateur peut monter ce fs

unhide systèmes de fichiers CD-ROM : montrer les noms et leurs alias

async E/S asynchrone

sync E/S synchrone

ro en lecture seulement

defaults `rw, suid, dev, exec, auto, nouser, async`

La commande `mount` elle-même supporte l'option additionnelle `remount` (qui permet de changer les options de montage d'un fs déjà monté).

Exercices

1. comment automatiser le montage d'un système de fichiers au démarrage ?
2. comment autoriser le montage d'un système de fichiers par les utilisateurs et ne pas le monter automatiquement au démarrage (p.ex. floppy, CD-ROM, etc)
3. comment vérifier un système de fichiers alors qu'il est monté et ne peut être démonté complètement (indication : consultez les options `remount, ro` et `remount, rw` de la commande `mount`).
4. comment mouter sans modifier l'état de montage `/etc/mtab` ?
5. comment assurer au montage des requêtes d'E/S synchrones ?

Gestion des quotas – 4.4

Résumé des concepts importants

- quota, repquota
- quotaon, edquota

Lectures supplémentaires

- <http://www.tldp.org/HOWTO/Quota.html>

Linux supporte deux types de quotas : les quotas par utilisateurs, et les quotas par groupe. De plus, un quota spécifie 2 genres de limites : les limites sur les fichiers et objets et les limites sur le nombre de blocs utilisés (place disque). Enfin, un quota peut être dur ou faible (hard and soft quota). Un quota dur interdit l'allocation de plus de ressources. Un quota faible est un avertissement et se change en quota dur après un certain délai.

On active les quotas en ajoutant l'option `usrquota` et/ou `grpquota` dans `/etc/fstab`, en créant les fichiers de gestion des quotas, les renseignant sur l'état actuel avec `quotacheck` puis en activant les quotas avec `quotaon -av` (pour tous les systèmes de fichiers avec quota).

```
apt-get install quota

cd /tmp
dd if=/dev/zero of=a_file bs=1024k count=100
mke2fs -j a_file
mount a_file -o loop /mnt

mount /mnt -o usrquota,remount

# Quotas version 2
touch /mnt/aquota.user
chmod 600 /mnt/aquota.user
quotacheck -v /mnt
```

```
# Quotas version 1
rm -f /mnt/aquota.user
quotacheck -mcF vfsold /mnt
chmod 600 /mnt/quota.user

quotaon /mnt
```

Exercices

1. ajoutez des quotas par groupe et par utilisateur sur /home.
2. éditez les quotas de votre utilisateur, puis de votre groupe
3. éditez les configurations générales de quota (`edquota -t`)
4. affichez vos quotas avec `quota`, et ceux d'un autre utilisateur ou groupe.
5. affichez un rapport de quota avec `repquota`
6. copiez une configuration de quota d'un utilisateur à l'autre

Gestion des permissions d'accès aux données – 4.5

Résumé des concepts importants

- permissions UNIX : `chmod`, `umask`
- attributs ext[23] : `chattr`

Lectures supplémentaires

- cours **sécurité**

Exercices

1. comment changer récursivement des permissions ?
2. à quoi sert la permission `t` sur un répertoire ? et sur un fichier exécutable ?
3. que se passe-t-il si le bit `sgid` est mis sur un répertoire ?
4. comment rendre un fichier immutable ou en ajout seulement avec la commande `chattr`. Est-ce que cela fonctionne aussi sur un `fs ni ext2`, ni `ext3` ?

Propriétaires et groupes – 4.6

Résumé des concepts importants

– chown, chgrp

Lectures supplémentaires

– cours **sécurité**

34

Exercices

1. pouvez-vous changer le propriétaire d'un fichier, en tant qu'utilisateur normal ?
2. quels groupes pouvez-vous choisir pour un fichier ?
3. comment changer l'utilisateur et le groupe simultanément ?

Liens symboliques et durs – 4.7

Résumé des concepts importants

35

- `ln -s` (soft/symlink) et `ln` (hardlink)
- option `-f` (force)
- soft-link (-s) : concept d'original et de lien ; risque de *dangling link*
- hard-link : limité au même système de fichiers ; lien dans la structure POSIX du système de fichiers (partage d'inode : métadonnées ; et de données) ; pas de concept d'original.

Exercices

1. comment trouver, avec `find`, des *dangling links* (liens symboliques ne pointant pas/plus sur un fichier existant)
2. expérimentez avec les liens symboliques sur des répertoires : y-a-t-il une différence entre `ls lien` et `ls lien/` ?
3. vérifiez que le concept de hardlink n'a pas d'original.
4. est-ce que `find` suit les liens symboliques ?

Trouver et placer les fichiers – 4.8

Résumé des concepts importants

- chercher
 - en temps réel : `find`
 - dans le path : `which`
 - dans le système : `whereis`
 - dans un index (BD) : `locate`, `slocate`, `updatedb`, `/etc/updatedb.conf`

Lectures supplémentaires

- `man 7 hier`
- <http://www.pathname.com/fhs/>

Exercices

1. quelle est la différence fondamentale entre `find` et `locate` ?
2. comment régénérer la base de données utilisée par `locate` ?
3. quelle est la différence entre `locate` et `slocate` ?
4. où se trouve en général la configuration système d'un programme ?
5. où se trouvent les configurations par utilisateur ?
6. où installez-vous des logiciels maintenus localement ?
7. où sont installés en général des logiciels optionnels ?
8. où sont installés les binaires prévus pour le superutilisateur ? et lorsqu'ils ne sont pas essentiels au démarrage ?
9. où se trouve le répertoire utilisateur de `root` sous GNU/Linux ?
10. où se trouvent les périphériques ?
11. où se trouvent les fichiers utilisés par le démarrage de la machine ?
12. où se trouvent les bibliothèques partagées par défaut (2 réponses) ?
13. où sont les points de montage temporaires ?
14. où se trouvent les fichiers temporaires ?
15. quel genre de données se trouvent dans `/usr` ?
16. où sont en général les fichiers de X11 ?
17. où sont les documentations des packages installés ?

18. où sont les jeux ?
19. où sont les fichiers de définition de développement pour le C ? (fichiers *includes*)
20. où sont les données non essentielles au démarrage et partageables sur plusieurs machines (en lecture seulement)
21. où peut-on placer les sources ?
22. où place-t-on les données systèmes qui changent ? les logs ? les queues ? les données de cache ?

5. Commandes UNIX et GNU

Contenu du chapitre

- travailler en ligne de commande
- filtres de traitement de texte
- gestion de fichiers
- fichiers, pipes et redirections
- créer, surveiller, modifier les priorités et tuer des processus
- recherche textuelle avec expressions régulières
- édition de texte avec **vi**

Buts du chapitre

- savoir travailler efficacement et trouver l'information sur son système UNIX
- savoir gérer les processus du système
- savoir utiliser l'éditeur **vi**

supports de cours additionnels : **cours UNIX et shell** et exercices, **Guide UNIX**

Travailler en ligne de commande – 5.1

Résumé des concepts importants

- .
- bash, man
- echo, exec, pwd,
- env, export, unset
- set
 - paramètres positionnels manuels
 - options du shell
- ~/.bash_history, ~/.profile

Lectures supplémentaires

- Guide UNIX et cours shell
- info bash (package bash-doc)

Le shell `bash` a déjà été décrit lors du cours shell. Nous donnons ici quelques compléments utiles et quelques exercices supplémentaires.

Le shell exécute les commandes en les cherchant dans un ordre défini :

1. dans les alias
2. dans les mots-clés comme `if`, `for`, `while`, `case`, `do`, ...
3. dans les fonctions définies
4. dans les commandes internes (`cd`, `help`, `type`, ...)
5. finalement dans le `$PATH` (programmes exécutables, scripts exécutables, notamment avec `#!` she-bang, etc)

L'ordre de *sourcing* des scripts initiaux *au login* est le suivant : (documenté dans la section `INVOCATION` du `man` de `bash`)

1. `/etc/profile`
2. `~/.bash_profile`, ou `~/.bash_login` ou `~/.profile` (ce dernier compatible `sh/ksh`)

Dans le cas d'un sous-shell (pas un shell de login), les fichiers `/etc/bash.bashrc` puis `~/.bashrc` sont sourcés.

La commande `set` permet de définir manuellement les paramètres positionnels (`$1 ... ${N}`), ou d'activer des options du shell :

- o no clobber (pas d'écrasement)
- v verbose : debugging
- e exit on error

Exercices

1. quelles sont les variables définies dans votre shell ? comparez le résultat des commandes `env` et `set` (indication : `env | sort -u > /tmp/env.out` et `diff`)
2. visualisez et modifiez la variable `PS1`
3. à quoi servent les fichiers `.bashrc`, `.bash_profile` et `.bash_logout` ? ajoutez du debugging (`echo`) et déterminez à quelle occasion ces scripts sont exécutés (comparez en particulier un login, p.ex. sur une console texte, et le lancement d'un nouveau sous-shell ou d'un `xterm`)
4. quelle est la différence entre `programme` et `exec programme` ?
5. quelle sont les différences entre
 - `VAR=valeur; programme`
 - `export VAR=valeur; programme`
 - `env VAR=valeur programme`
6. documentez-vous sur la commande `passwd` et répondez aux questions suivantes :
 - (a) comment l'administrateur peut bloquer et débloquer un compte ?
 - (b) comment changer les délais minimum et minimum de validité d'un mot de passe ?
 - (c) quels fichiers `passwd` peut-il accéder/modifier ?
7. qu'y-a-t-il dans le fichier `.bash_history` et comment peut-on contrôler ses paramètres ?
8. `.profile` et `.bash_profile` sont-ils équivalents ?
9. quelle est la différence entre la commande interne `kill` et l'exécutable `/bin/kill` ? même question pour `echo`.
10. créez un script dans votre `bin`, exécutez-le avec son chemin entier, puis changez le `PATH` de manière à pouvoir le lancer sans spécifier son emplacement puis assurez-vous qu'à chaque nouvelle connexion et nouvel `xterm` la modification de `$PATH` soit activée !
11. et si vous vouliez faire de même, mais pour tous les utilisateurs du système ?
12. à quoi sert la commande `apropos` ? et l'option `-k` de `man` ? trouvez les commandes qui permettent de compresser des données.
13. utilisez la commande `date` pour afficher la date en format ISO (2006-03-24)

Filtres de traitement de texte – 5.2

Résumé des concepts importants

- `cat`, `cut`, `expand`, `fmt`, `head`, `join`, `nl`, `od`, `paste`, `pr`, `sed`, `sort`, `split`, `tac`, `tail`, `tr`, `unexpand`, `uniq`, `wc`

Lectures supplémentaires

- Guide UNIX et manpages

Exercices

1. documentez-vous sur chacun des programmes listés ci-dessus
2. coupez un gros fichier en petits morceaux
 - (a) comment le recréer ?
 - (b) vérifiez la taille en bytes du fichier, sans utiliser `ls`
 - (c) comment vérifier que le fichier n'a pas été corrompu (indication : `md5sum`, `cmp`, `diff`)
3. affichez uniquement les champs utilisateur, répertoire par défaut et shell par défaut du fichier `/etc/passwd`, séparés par des espaces, le tout trié sur le nom d'utilisateur .
4. déterminez la liste des shells effectivement utilisés par les utilisateurs du système, sans doublons
5. comparez la longueur en bytes d'un fichier après traitement par `expand` et `unexpand`
6. générez un long texte avec `fortune -l` et repaginez-le à 50 caractères par ligne.
7. affichez les 7 premières lignes de `/etc/passwd`, la dernière ligne de `/etc/group`, les changements en continu d'un fichier de log, toutes les lignes de `/etc/group` sauf la première.
8. numérotez les lignes de `/etc/motd`
9. vérifiez que les fins de lignes des fichiers sous UNIX sont composées du caractère **LF** (*line-feed*), code hexadécimal `0x0a` uniquement.
10. comment préparer un fichier pour impression en double interligne ?
11. remplacez toutes les occurrences de `/bin/bash` par `/bin/sh` dans `/etc/passwd` en stockant le résultat dans `/tmp/passwd`

12. afficher les lignes d'un fichier, à l'envers
13. remplacer les minuscules par les majuscules correspondantes dans un fichier
14. à l'aide des fichiers `/etc/passwd` et `/etc/group`, créez un fichier dont chaque ligne contiennent un utilisateur et le *nom* de son groupe (indication : commencez par trier sur des fichiers temporaires !)
15. (difficile) numérotez les lignes de `/etc/motd` sans utiliser `nl` ni `cat -n` (indication : utilisez la commande `seq` et les backticks ou le `$()` ainsi qu'uniquement des commandes vues ici)
16. quel est le problème avec la commande `cut` lorsqu'il y a plus d'un espace séparateur ?

Gestion des fichiers – 5.3

Résumé des concepts importants

- `cp`, `mkdir`, `mv`, `rm`, `rmdir`, `touch`
- `ls`
- `find`
- le *globbing* (expansion des *jokers* `*?{}[]~`)

40

C'est le shell qui s'occupe de l'expansion des arguments, que cela soit les variables d'environnement, les opérateurs `` `` ou `$ ()`, les calculs, et les wildcards (jokers, métacaractères du shell).

Exercices

1. documentez-vous sur chacun des programmes listés ci-dessus
2. comment créer toute une hiérarchie de répertoire d'un seul coup ? comment la supprimer (deux méthodes) ? comment spécifier la permission utilisée ?
3. peut-on supprimer un répertoire qui n'est pas vide avec `rmdir` ?
4. comment créer un fichier avec une date de modification dans le passé ?
5. comment copier plusieurs fichiers à la fois dans un répertoire ? en conservant leurs permissions ? copier aussi les sous-répertoires ?
6. comment éviter l'écrasement avec `cp` et `mv` ? quelle est la configuration par défaut ? comment la changer ? est-ce une bonne idée ?
7. quels sont les deux rôles de `mv` ?
8. trouvez au moins une autre façon de créer un fichier que la commande `touch`
9. testez et documentez les options principales de `ls`
 - `-l`
 - `-F`
 - `-a`
 - `-R`
 - `-d`
 - `-1`

peut-on les combiner ? comment ?

10. faites une recherche complexe avec `find`, par exemple les fichiers dont le nom comprend un `s`, de taille inférieure à 64 kilobytes, appartenant à `root` et de type fichier (par opposition à block device, etc), dont le dernier accès est plus récent que hier
11. comment exécuter une commande pour chaque fichier trouvé avec `find`? et avec confirmation ?
12. comparez les commandes suivantes, prévoyez et testez :
 - (a) `echo ceci est mon repertoire: $HOME: *`
 - (b) `echo ceci est mon repertoire: \ $HOME: *`
 - (c) `echo "ceci est mon repertoire: $HOME: *"`
 - (d) `echo "ceci est mon repertoire: \ $HOME: *"`
 - (e) `echo 'ceci est mon repertoire: $HOME: *'`
 - (f) `echo 'ceci est mon repertoire: \ $HOME: *'`
13. comparez et expliquez :
 - (a) `ls /bin/*cho`
 - (b) `ls '/bin/*cho'`
 - (c) `ls "/bin/*cho"`
 - (d) `ls /bin/*cho`
14. montrez deux façons d'afficher un texte avec une apostrophe
15. comment afficher la taille d'un fichier de manière plus intuitive (kilobytes, mégabytes, etc) avec `ls` ?
16. à quoi peut servir le caractère tilda `~` (deux rôles principaux, consultez `man bash`, paragraphe `Tilde Expansion`)
17. que fait la commande `rename` et en quel langage est-elle écrite ?

Fichiers, pipes et redirections – 5.4

Résumé des concepts importants

- tee, xargs
- <, <<, >, >>, |, ``

41

Exercices

1. quels sont les 3 flots standard ouverts par défaut pour chaque processus sous UNIX ?
2. comment peut-on rediriger un flot dans un autre ? dans un fichier ?
3. comment fonctionne un HERE document (<<EOF)
4. comment afficher les fichiers dont le nom contient la date du jour ?
5. à quoi servent les commandes xargs et tee ?

Créer, surveiller et tuer des processus – 5.5

Résumé des concepts importants

42

- `&`, `bg`, `fg`, `jobs`
- `kill`, `nohup`
- `ps`, `top`
- numéro de processus usuels, de jobs (`%N`) et spéciaux (p.ex `-1`)
- `kill -9`

La commande `nohup` permet de lancer un processus protégé contre le signal de raccrochage `SIGHUP`, avec sa sortie dans un fichier `nohup.out`.

Exercices

1. quelle est la différence entre `jobs` et ses processus ? comment lister l'un et l'autre ? comment lister tous les processus du système ?
2. qu'est-ce qu'apporte le lancement avec `&` ?
3. comment lister les signaux que `kill` peut envoyer ? quel signal est envoyé par défaut ?
4. que fait la commande `killall` sous GNU/Linux ? et sous Solaris ?
5. comment tuer un processus sans lui permettre d'intercepter l'opération ? (ni p.ex. sauver ses données)

Modification des priorités d'exécution – 5.6

Résumé des concepts importants

– nice, renice

43

UNIX est un système multitâche préemptif en temps partagé. Cela signifie que l'on peut associer des priorités aux processus (de -20 à 19, la priorité -20 étant la plus grande et 19 la plus faible). Comme UNIX est temps partagé (*time-sharing*) les priorités ne sont pas absolues mais pondérées par le bout de temps CPU déjà consommé par le processus (*quantum*). Cela signifie qu'un processus de priorité supérieure ne va pas bloquer complètement les processus de priorités inférieures, seulement les ralentir de manière très importante.

Linux propose aussi, pour des besoins spécifiques, des processus de classe temps réel qui sont de priorité plus grande que les processus *time-sharing* et bloquants.

La priorité par défaut est 0. La commande `ps` (et `top`) indiquent les processus de priorité augmentée (négative) par un < et les processus de priorités diminuées (positives) par N, comme *nice*.

Exemples

```
% sleep 3600 &
[2] 10265
schaefer@voyager:/data/home/schaefer$ renice 0 10265
10265: old priority 0, new priority 0
schaefer@voyager:/data/home/schaefer$ renice 1 10265
10265: old priority 0, new priority 1
schaefer@voyager:/data/home/schaefer$ renice 1 10265
10265: old priority 1, new priority 1
schaefer@voyager:/data/home/schaefer$ renice 0 10265
renice: 10265: setpriority: Permission denied
```

```
schaefer@voyager:/data/home/schaefer$ renice -1 10265
renice: 10265: setpriority: Permission denied
schaefer@voyager:/data/home/schaefer$ renice -19 10265
renice: 10265: setpriority: Permission denied
schaefer@voyager:/data/home/schaefer$ renice 19 10265
10265: old priority 1, new priority 19
```

Exercices

1. y-a-t-il déjà des processus prioritaires ou non prioritaires sur votre système ?
2. lancez un `sleep 3600` de manière non prioritaire, puis augmentez sa priorité, et diminuez la : quelles opérations se font sous root ?
3. comment diminuer la priorité de tous les processus d'un utilisateur ? d'un terminal (process group) ?

Recherche textuelle avec expressions régulières – 5.7

Résumé des concepts importants

- grep, sed
- **regex** (expressions régulières)

44

Il est important de ne pas confondre les *wildcards* ou *jokers* du shell et les regex – expressions régulières.

Beaucoup d'utilitaires proposent le support de regex en interne, notamment pour des recherches de sous-chaînes ou des remplacements. Il n'existe malheureusement pas qu'un seul type de regex, mais les différences sont surtout dans l'étendue des possibilités, la base reste compatible.

La commande `sed` permet de remplacer des textes dans un fichier. Elle applique une ou plusieurs commandes sur un flot (*stream*) de données. Sa syntaxe générale est :

```
sed [address1][,address2][!]command[options] [files...]
```

Les adresses sont des expressions régulières ou des numéros de lignes.

Exercices

1. étudiez les options `-i`, `-l`, `-c`, `-n` et `-v` de `grep`.
2. quelle est la différence entre `grep` et `egrep` ?
3. que font les commandes
 - (a) `sed '1,3s/aa/bb/g' < file`
 - (b) `sed '/here/, $d' < file`
 - (c) `sed '/ll/, /ff/!s/maison/house/g'`
4. que désigne la regex `a{1,3}` ?

Edition de texte avec vi – 5.8

Résumé des concepts importants

45

- recherche : /, ?
- déplacement d'un caractère : h, j, k, l
- déplacement : G, H, L
- modification : i, c, d, dd, p, o, a
- on retourne au mode déplacement avec ESC
- sauver/quitter/charger/forcer ZZ, :w, :q, :e, :w!
- commandes : :!COMMANDE (en mode déplacement uniquement)

Lectures supplémentaires

- résumé des commandes vi
- vintutor (package vim)

6. Corrigé des exercices

Architecture et matériel

BIOS

1. `dmesg | grep 'CHS='`
2. (config du BIOS, enable/disable on-chip peripherals)
3. (config du BIOS, halt on errors/not keyboard)
4. (config du BIOS, ordre de démarrage)
5. `free` et `grep MemTotal /proc/meminfo`
6.
 - (a) `hdparm -g /dev/hda`
 - (b) `hdparm -C /dev/hda`
 - (c) `hdparm -v /dev/hda;hdparm -d1 /dev/hda;hdparm -d0 /dev/hda;hdparm -t -T /dev/hda`
 - (d) `hdparm -r 1 /dev/hda`
7.
 - (a) non, voir `uname -r` et le HOWTO Large Disk
 - (b) – d'autre OS qui scratchent les données de GNU/Linux
– démarrage non fonctionnel : petite partition
 - (c) support gros disque ; sans lba32, créez une petite partition /boot ou / entièrement sous cylindre 1024

Cartes modems et son

1. pilotes spécifiques ou génériques selon les cas propriétaires (Winmodem)
2. voir les HOWTOs et linmodem.org
3. `/dev/ttyS0, setserial /dev/ttyS0 spd_vhi`
4. voir exemple commenté dans `/etc/inittab`, activer avec `telinit q`
5. `mknod`, ou script MAKEDEV
6. `minicom -s`
7. `sndconfig`

Périphériques SCSI

1. la liste des périphériques SCSI (avec des détails) et le numéro SCSI ID et LUN, classé par carte SCSI (réelle ou virtuelle come `ide-scsi` ou `usb-storage`)
2. 7 (8 si l'on considère que le HA est un périphérique)
3. pour un graveur sur le maître du secondaire IDE, configurer p.ex. LILO avec `append="hdc=ide-scsi"`, redémarrer. Ensuite éditer `/etc/fstab` et/ou créer un lien symbolique :
`ln -s /dev/scd0 /dev/cdrom`. Noter qu'avec des distributions récentes avec `udev`, on modifierait plutôt `/etc/udev/rules.d/z25_persistent-cd.rules`

4. `append="max_scsi_luns=1"` dans LILO p.ex.
5. il change de `/dev/sdb` à `/dev/sda`
6. `cdrecord -scanbus`
7. `scsiinfo -a /dev/sda` ou `hdparm -gv /dev/sda`

Cartes d'expansion

1.

```
for i in interrupts ioports dma pci
do cat /proc/$i; done
grep , /proc/interrupts
```
2. `lspci | wc -l`
3. `lspci | grep bridge`
4. (a) `lspci -t`
(b) `lspci -d 8086:`
5. (a) interroger et configurer des cartes PCI
(b) `setpci -d *:* latency_timer=40`
6. `cat /proc/interrupts`
7. `apt-get install isapnptools`, puis `pnpdump`

Périphériques USB

1. `lsmod | grep usb`
2. `lsusb`

X Window System

Configuration de XFree86

1. dans la Debian, le démarrage est fait via un script d'init du niveau 2 à 5 (`/etc/rc[2-5].d/*dm`), lancé par `init` lorsque le niveau 2-5 est atteint. Dans la Red Hat et dérivés, la configuration est directement faite dans `/etc/inittab`, et le niveau 3 est sans X (avec réseau) et le niveau 5 avec X et réseau.
2. on voit de l'activité dans le protocole de la souris (utile pour debugging, p.ex. déterminer quel port est utilisé pour une souris série).
3. manipulations simples
4. le fichier `.xinitrc` (sous Debian GNU/Linux et les autres distributions modernes, ce fichier est remplacé par `.xsession`: le plus simple est de faire un lien symbolique de l'un à l'autre).
5. Debian installe un font-server par défaut si l'on installe le paquet `x-window-system`. On peut vérifier qu'il est bien lancé avec `ps auxw | grep xfs` et voir qu'il est bien utilisé en regardant les logs du serveur X, p.ex. via `XFree86 2>/tmp/start-log`, ou en vérifiant que la section **Files** du fichier `/etc/X11/XF86Config` (ajouter -4 pour Debian) contient bien une entrée pour le serveur de fonte (socket UNIX port 7100).
6. manipulation simple. Lier ensuite `/dev/mouse` à la souris réellement utilisée (p.ex. `/dev/psaux` si l'on ne donne pas le périphérique manuellement).
7. `XFree86 -configure`. La configuration est générée dans `/root/` et peut être testée.
8. `xvidtune` permet de générer des `ModeLines`.

Configuration d'un gestionnaire de connexion

1. Sous Debian, `/etc/init.d/xdm stop` et `start`; sous Red Hat, `telinit 3` et `telinit 5`.
2. `/etc/X11/xdm/Xresources`, champ `xlogin*greeting`
3. `/etc/X11/xdm/Xservers`, option `-depth` (ou `-bpp`)
4. `/etc/X11/xdm/xdm-config`, commenter avec `! DisplayManager.requestPort: 0`, puis autoriser dans `/etc/X11/xdm/Xaccess`, décommenter `*` et aussi l'option `CHOOSE` si désiré (lire les commentaires).
5. ajouter `:1 local /usr/X11R6/bin/X vt8 -dpi 100 -nolisten tcp` dans le fichier `/etc/X11/xdm/Xservers`
6.
 - (a) login sur le serveur localhost
 - (b) choix du login (chooser)
 - (c) login sur le serveur localhost, avec client spécial X dans X
 - (d) manipulation simple
7. manipulation simple

Configuration d'un gestionnaire de fenêtres et des ressources

1. `xterm -background red` (ou `-bg red`)
2. `xterm*background: red`
- 3.
- 4.
5. on voit que les décorations des fenêtres et menus sont gérés par le window-manager

Travail distant et sécurité

1. non, car `-nolisten tcp` dans `/etc/X11/xdm/Xservers`
2. `xhost +1.2.3.4`, où 1.2.3.4 est l'adresse IP ou le nom de la machine de votre voisin. Votre voisin fait ensuite `xclock -display 1.2.3.5:0`, où 1.2.3.5 est votre adresse IP ou le nom de votre machine (ou encore `export DISPLAY=1.2.3.5:0; xclock`)
3. `xhost -1.2.3.4, xhost`
4. `-nolisten tcp`
5. `ssh -X 1.2.3.4` (suppose que `/etc/ssh/sshd_config` autorise le `X11Forwarding`). La variable `DISPLAY` montre un serveur X virtuel géré par SSH (tunnel). Utiliser `xauth list` pour la liste des cookies.
6.
 - (a) `su` conserve les variables et le répertoire courant du shell ayant appelé le `su`; `su -` fait l'équivalent d'un nouveau login, la variable `DISPLAY` n'est pas passée !
 - (b) `export XAUTHORITY=~user/.Xauthority`, avec `user` le nom de l'utilisateur ayant lancé la session X; `xhost +localhost` (dangereux car trop général) `xauth add VA` où `VA` est quelque chose comme la sortie de `xauth list | grep unix` sous l'utilisateur qui a lancé la session (mot de passe généré).

Installation et gestion des packages

Partitionnement

1. activer tous les swaps définis dans `/etc/fstab`; donner les statistiques d'utilisation des swaps actifs
2. monter tous les filesystems définis dans `/etc/fstab`;
3. colonnes `si` et `so`, restent zéro si pas d'activité swap
4. `free; swapoff -a; free`
5. p.ex. serveur de mail écrivant beaucoup de logs et de mails, partitionner `/var/spool` et `/var/log` séparément

Démarrage

1. sur un kernel optimisé sans `initrd` avec tout le support nécessaire : `cp /vmlinuz /dev/fd0`, puis utiliser `rdev /dev/fd0 /dev/hda3` pour définir le `/` à utiliser sur `/dev/hda3`.
2. `mformat A:`
`syslinux /dev/fd0`
`mount /floppy \`
`&& cp /boot/vmlinuz-2.4.20 /floppy/linux \`
`&& cp syslinux.cfg /floppy \`
`umount /floppy`

avec, par exemple :

```
DEFAULT linux
TIMEOUT 0
DISPLAY boot.msg
PROMPT 1
LABEL linux
        KERNEL linux
        APPEND root=/dev/md2 vga=normal
```

3. par exemple

```
title          Ubuntu, kernel memtest86+
root           (hd0,0)
kernel         /memtest86+.bin
savedefault
boot
```

(sous Debian : il suffit de décommenter la ligne qui parle de `memtest86` et l'entrée sera générée automatiquement. Rappelons que tout ce qui se trouve entre les marques automatiques Debian dans `/boot/grub/menu.lst` est écrasé et régénéré !)

4. installer `mbr` et `lilo`, installez LILO dans la partition, `mbr` dans le MBR, configurez `/etc/lilo.conf`, lancez `lilo` pour prendre en compte les modifications. (sous Debian `liloconfig` se charge de tout !)
5. mot de passe : `password=toto`, ensuite choisir, soit `restricted` (empêcher la modification des paramètres au démarrage sans mot de passe) ou `mandatory` : refuser le boot sans

démarrage. On doit encore relancer `lilo` pour prendre en compte les modifications. Pour sécuriser véritablement la machine, mettre un mot de passe sur le BIOS, interdire le démarrage sur autre chose que le disque-dur, sécuriser le petit MBR ou y mettre LILO. Et espérer que le BIOS n'a pas de *back-door* !

6. il peut servir à démarrer tous les OS dont la partition est active sans les déclarer dans LILO, voire démarrer sur d'autres périphériques. Inutilisé si LILO ou GRUB sont installés dans le MBR directement (et pas une partition).
7. oui, notamment avec LILO. Sinon, le plus simple est de ne pas avoir `/boot` en RAID ou LVM.

Bibliothèques dynamiques

1. `man ld.so` : variables `LD_LIBRARY_PATH` et `LD_PRELOAD`, le fichier généré via `/etc/ld.so.conf` et `ldconfig`, puis dans les chemins standards `/usr/lib` puis `/lib`.
2. manipulation avec `ldd PROGRAMME`. On remarque que `ldconfig` est statiquement lié (cf file aussi)
3. il manque une bibliothèque partagée au moins.

```
env LD_LIBRARY_PATH=/usr/lib/mozilla-firefox /usr/lib/mozilla-firefox/firefox-bin
```

 On peut configurer un *wrapper* script qui configure la variable au moment du lancement, voire ajouter le répertoire sous `/etc/ld.so.conf` et lancer `ldconfig`.
4. précharger via `LD_PRELOAD` une bibliothèque qui intercepte les fonctions de la bibliothèque C standard (ouvrir un fichier, changer des permissions, propriétaires, etc) de manière à pouvoir créer des archives d'installation (paquets Debian) sans être root, tout en ayant les permissions correctes.
5. démarrer avec `init=/bin/sh` et faire :


```
mount -n / -o remount,rw
mount -a
ldconfig -r /
umount -a
mount -n / -o remount,ro
sync
reboot -f
```

ou : redémarrer avec le CD d'installation, monter les systèmes de fichiers puis lancer `ldconfig`, soit avec l'argument `-r`, soit via `chroot`.

Packaging Debian

1. manipulation simple
2. manipulation simple
3. manipulation simple
4. `dpkg -S /bin/ls`; <http://packages.debian.org/>, dernier champ de recherche.
5. `dpkg --get-selections | grep 'install|hold'`
6. `dpkg -l '.*-dev'`
7. premier caractère : état désiré ; deuxième : état
8. `apt-get update && apt-get -y --download-only -u dist-upgrade`

9. `dpkg-reconfigure debconf`
10. `aptitude` (ou `Synaptic`), *Locally installed or obsolete Packages*
11. <http://www.backports.org/>
12. `aptitude` (ou `Synaptic`), *Locally installed or obsolete Packages*
13. manipulation simple
14. suppose que toutes les questions simples sont répondues par oui
15. manipulation simple
16. empêcher sa mise à jour (p.ex. en cas de modifications manuelles) (attention : différencier le hold de `dpkg` de celui d'`aptitude`)
17. permet d'installer des packages d'une autre version sans mettre à jour tout le système, grâce à des priorités
18. non, `fakeroot` suffit. Oui, pour les installer.
19. manipulation simple, p.ex. avec
`ftp://fr2.rpmfind.net/linux/SuSE-Linux/i386/9.2/suse/i586/hello-2.1.1-311.i586.rpm`
20. complexe, non ? :)

Commandes UNIX et GNU

Travail en ligne de commande

1. manipulation simple
2. p.ex. `PS1='\h:\t> '`
3. `.bashrc` est sourcé en cas de shell NON login ; `.bash_profile` en cas de shell login et `.bash_logout` en cas de logout ?
4. `exec programme` remplace le shell : lorsque programme se termine, le shell n'est déjà plus là
5. – programme ne verra pas la variable (locale au shell)
– tous les programmes lancés après l'export verront la variable
– seulement programme verra la variable
6. (a) `-l` et `-u`
(b) `-x` et `-n`
(c) `/etc/passwd` et `/etc/shadow`, car `passwd` est SUID root
7. l'historique des commandes tapées, sauvée à la fin de la session bash, voir la section **HISTORY** de `man bash`.
8. `.profile` est utilisé lorsque `.bash_profile` n'est pas disponible ou que le shell est lancé sous le nom `sh` (pour compatibilité) ; dans tous les cas uniquement pour un shell de login
9. il n'y a pas de différence fondamentale, sauf que l'on lance les commandes externes uniquement si l'on spécifie leur chemin
10. p.ex. `export PATH=${PATH}:${HOME}/bin`, à déposer dans `.bashrc` (sous Debian, le `.bash_profile` source aussi le `.bashrc` ! sinon mettre dans les deux)
11. modifier alors `/etc/profile` et/ou `/etc/bash.bashrc`
12. à chercher dans l'index des pages man (description courte). `man -k compress`
13. `date --iso`

Filtres

- manipulation simple
- split < /vmlinuz && ls -lh x??
 - cat x?? > f1
 - wc -c f1 /vmlinuz
 - cmp f1 /vmlinuz && diff f1 /vmlinuz && md5sum f1 /vmlinuz
- cut --output-delimiter ' ' -d: -f1,6,7 < /etc/passwd \
| sort -k 1,1
- cut --output-delimiter ' ' -d: -f7 < /etc/passwd | sort | uniq
ou cut --output-delimiter ' ' -d: -f7 < /etc/passwd | sort -u
- manipulation simple
- fortune -l | fmt -w 50
- head -7 /etc/passwd
tail -1 /etc/group
tail -f /var/log/syslog
tail +2 /etc/group | diff /etc/group -
- nl /etc/motd
- utiliser od -h (ou hexdump)
- pr -d < fichier > out
- sed s%/bin/bash%/bin/sh% < /etc/passwd > /tmp/passwd
- tac fichier
- tr a-z A-Z < fichier > out
- sort -t : -k 4,4 < /etc/passwd > /tmp/p.SORTED
sort -t : -k 3,3 < /etc/group > /tmp/g.SORTED
join -t : -1 4 -2 3 /tmp/p.SORTED /tmp/g.SORTED
| cut --output-delimiter ' ' -d: -f2,8
- seq 1 \$(wc -l < /etc/motd) | paste - /etc/motd
- cut -f est sensible à plusieurs espaces (plusieurs colonnes) ce qui pose un problème p.ex. avec ls -l. On peut passer par un sed avant, ou utiliser awk à la place de cut.

Gestion des fichiers

- manipulation simple
- mkdir -p hier/arch/ie
rmdir -p hier/arch/ie
rm -rf hier/arch/ie
mkdir -m=755 dir
- non
- touch --date=yesterday /tmp/fichier (date de modif)
- cp f1 f2 f3 repertoire
cp -p f1 f2 f3 repertoire
cp -r dir1 dir2

6. `-i`, par défaut écrasement. Faire un alias. Non, car cela crée de très mauvaises habitudes de ne pas lire les messages ou si on travaille sur un système sans cette config.
7. renommer et déplacer des fichiers
8. `cat > fichier` puis `CTRL-D`
`> fichier`
9. `-l` : long (détails : taille, permissions, etc)
`-F` : ajouter un caractère de type en fin du nom
`-a` : aussi les fichiers *cachés* `.*?*`
`-R` : aussi les sous-répertoires
`-d` : pas le contenu des répertoires
`-l` : un fichier/répertoire par ligne
on les combine p.ex. `-l -a -F` ou `-laF` (ou encore `-alF`)
10. `find / -name '*s*' -user root -type f -atime -1 -size -64k`
11. `find . -type f -print | xargs echo`
`find . -type f -print0 | xargs -0 echo` (plus sûr)
`find . -type f -exec echo {} \;`
`find . -type f -ok echo {} \;`
12. le backslash permet d'empêcher l'interprétation du caractère qui le suit. Les apostrophes empêchent l'interprétation de tous les caractères spéciaux, y compris le backslash. Les guillemets empêchent l'interprétation (expansion) des wildcards (jokers).
13. on voit que la commande `ls` n'a pas le support pour les wildcards, c'est le shell qui les étend.
14. `echo \'`
`echo "' "`
15. `-h` (human)
16. votre répertoire : `~`. Le répertoire de l'utilisateur `user` : `~user`. Quant à `~-` et `~+`, ils permettent d'obtenir le répertoire courant précédent et actuel, respectivement. On peut aussi indiquer le numéro dans une pile de répertoires sauvegardés (voir la commande `popd`)
17. renommage de groupes de fichiers ou de répertoires (spécifiés comme wildcard du shell) selon des règles de substitution spécifiées par une regexp Perl. Il est écrit en Perl (et n'est pas forcément toujours disponible).

Pipes et redirections

1. `stdin, stdout, stderr` (0, 1, 2)
2. `2>&1, 2>fichier`
3. `command <<EOF`
(ici les données)
`EOF`
4. `ls *`date --iso`*`
5. exécuter des commandes groupées, dériver (dupliquer) un pipe dans un fichier

Processus

1. `jobs` : processus de *ce shell*. `jobs -l` et `ps x` (ou `ps auxw` pour tous)
2. lancement en tâche de fond sans bloquer le clavier (`bg`)

3. `kill -l, SIGTERM (15)`
4. envoie un signal indiqué au processus nommé (nom de commande); sous Solaris (Sun) cela envoie un signal donné à tous les processus !

Priorités d'exécution

1. indiqués par < et N respectivement après l'état du processus (S, R, ...)
2. baisser la priorité (augmenter la valeur ..) est toujours possible sans root.
3. `renice -u user +2`
`renice -g pgrp +2`

Recherche textuelle et regexps

1. `-i` (ignorer la casse), `-l` (noms de fichiers correspondants), `-c` (comptage des lignes correspondantes), `-n` (numéro de lignes) et `-v` (inversion de la condition)
2. `egrep` supporte les regexps étendues
3. (a) `sed '1,3s/aa/bb/g' < file` : remplacer aa par bb dans les lignes 1 à 3, autant de fois que nécessaire
(b) `sed '/here/, $d'` < file : supprimer de la ligne contenant here jusqu'à la fin
(c) `sed '/ll/, /ff/!s/maison/house/g'` : sauf la ligne content ll à la ligne contenant ff, remplacer maison par house autant de fois que nécessaire
4. la chaîne doit contenir a de 1 à 3 fois.

Périphériques et standard de hiérarchie FHS

Systemes de fichiers

1. pas de droits particuliers pour le créer dans un fichier, droits root pour le monter (sauf dans `/etc/fstab` avec option `user`). Dans le cas général, seul root a accès aux périphériques physiques de `/dev` (cf permissions)
2. manipulation simple
3. `-c`

Intégrité

1. `e2fsck -f /dev/hda3`
2. `-ay`
3. `-b`
4. manipulation simple
5. `tunefs -i 0 -c 0 /dev/hda3`
6. `tune2fs -j`
7. `df -i, df, df -B 1, df -h`
8. `df -a, mount`
9. `df -t ext3`
10. `du -s /boot`
11. `du /usr/* | sort -nr`

Montage et démontage

1. `/etc/fstab`, sans option `noauto`
2. `/etc/fstab`, options `noauto` et `user`
3. `mount -o remount,ro /`
4. `-n`
5. `-o sync`

Quotas

1. manipulation simple
2. `edquota user,edquota -g group`
3. manipulation simple
4. `quota -v,quota -vg`
`quota -v user quota -vg group`
5. manipulation simple
6. `edquota -p demo schaefer`

Permissions

1. `chmod -R a+r rep/`
2. sticky bit ; interdire effacement de fichiers d'autres propriétaires même si `w` sur répertoire pour tous. Originellement, sur les fichiers : maintenir en mémoire l'exécutable après terminaison (performance au prochain lancement), pas utilisé sous GNU/Linux (ou alors éventuellement *mandatory locking*, si activé.)
3. héritage du droit `g+s` sur les répertoires créés, objets (fichiers et répertoires) créés dans le groupe du répertoire et pas le groupe actif de l'utilisateur, pratique dans les répertoires partagés
4. `chattr +i fichier,chattr +a fichier`; uniquement `ext2/ext3`

Propriétaires et groupes

1. non, pas sous GNU/Linux (sinon serait facile de tricher avec les quotas¹)
2. votre groupe primaire ou un de vos groupes secondaires
3. `chown user:group fichier ... fichier`, encore que la syntaxe `user.group` est toujours tolérée (mais dangereuse, car seul `:` est un caractère interdit dans `/etc/group`)

Liens

1. `find -P / -xtype l -print` (voir info `find`)
2. `ls lien/` montre le contenu du répertoire pointé.
3. manipulation : créer plusieurs liens, effacer "l'original", le recréer, comparer `ls -li`.
4. la plupart des commandes qui traversent les répertoires savent faire la différence : l'option `-L` de `find` suit les liens symboliques.

¹mais liens durs sur des fichiers ... pensez-y

Trouver et placer les fichiers

1. `find` : état actuel ; `locate` : état au lancement de la commande de recherche, vu d'un utilisateur spécial, p.ex. `nobody`
2. `updatedb` & (ou `/etc/cron.daily/find`)
3. `slocate` est la version plus sécurisée
4. `/etc`, p.ex. `/etc/PROGRAMME/`
5. `~/.*rc`
6. `/usr/local`
7. `/opt`
8. `/usr/sbin` (ou `/sbin` si essentiels)
9. `/root`
10. `/dev`
11. `/boot`
12. `/usr/lib`, `/lib`
13. `/mnt`
14. `/tmp` (et `/var/tmp`)
15. des programmes et données non essentiels au démarrage
16. `/etc/X11` pour les configurations et `/usr/X11R6` pour le reste
17. `/usr/share/doc/PACKAGE/`
18. `/usr/games` (`/usr/share/games`)
19. `/usr/include`
20. `/usr/share`
21. `/usr/src`
22. `/var`, `/var/log`, `/var/spool`, `/var/cache`