

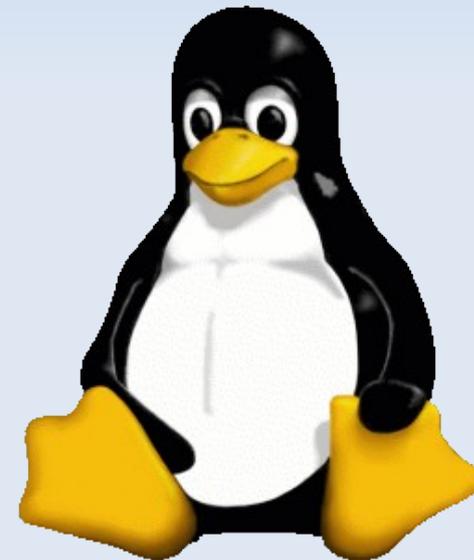
**Administration**

**Systeme**

**GNU/Linux**

# Administration Système GNU/Linux

- Historique
- Composants
- Organisation
- Shell
- Commandes de base
- Utilisateurs & Droits
- Disques & Filesystems

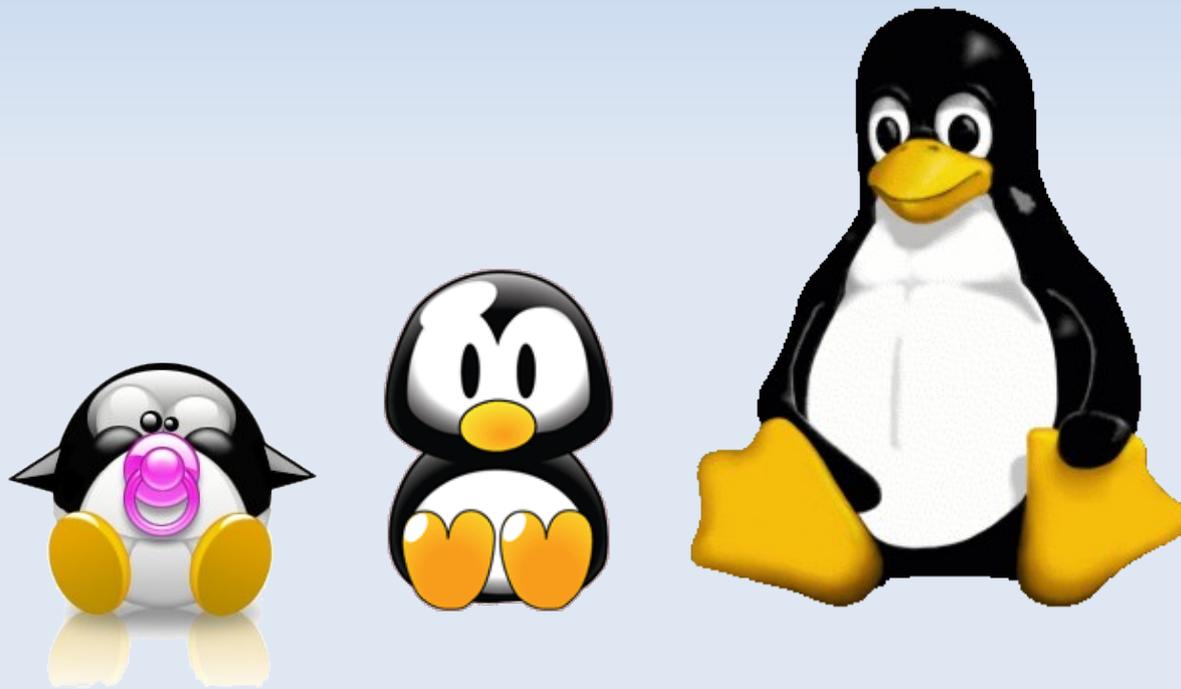


# Administration Système GNU/Linux

- Editeurs
- Shell & commandes avancés
- Packages
- Boot & Runlevels
- Réseau
- Interfaces graphiques
- Fichiers spéciaux
- Etudes de cas



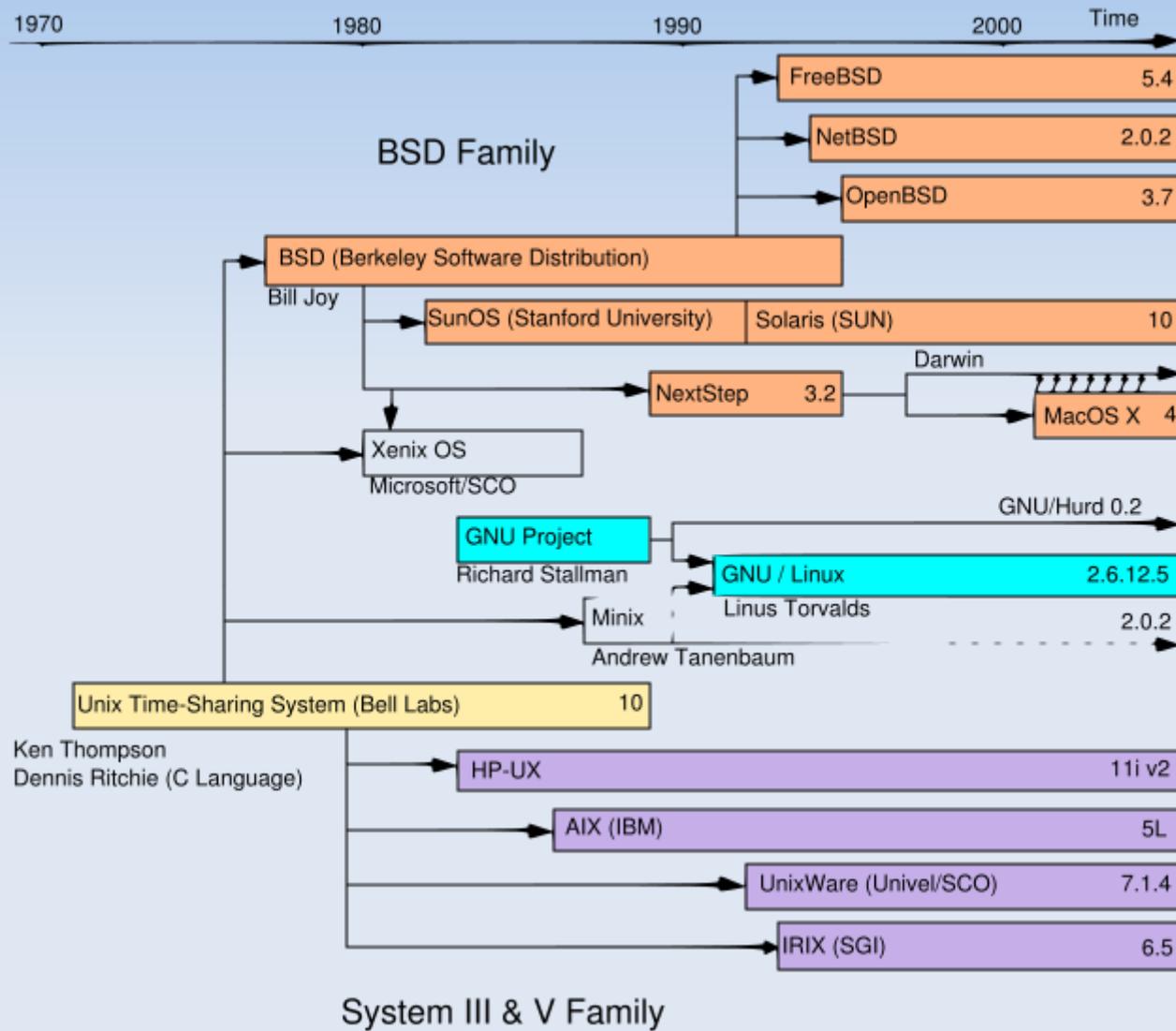
# Historique



*« I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. »*

Linus Torvalds

# Historique Unix



# Unix

## Philosophie

- 1) Ce qui est petit est beau
- 2) Chaque programme fait une chose et la fait bien
- 3) Construire un prototype dès que possible
- 4) Choisir la portabilité plutôt que l'efficacité
- 5) Enregistrer les données dans des fichiers plats
- 6) Utiliser le logiciel comme une force
- 7) Utiliser les scripts shells pour accroître cette force
- 8) Éviter les interfaces utilisateur captives
- 9) Faire de chaque programme un filtre

# Unix

## Philosophie

- 1) Small is beautiful.
- 2) Make each program do one thing well.
- 3) Build a prototype as soon as possible.
- 4) Choose portability over efficiency.
- 5) Store data in flat text files.
- 6) Use software leverage to your advantage.
- 7) Use shell scripts to increase leverage and portability.
- 8) Avoid captive user interfaces.
- 9) Make every program a filter.

# Unix

## Philosophie

« `timtowski` »

There is more than one way to do it

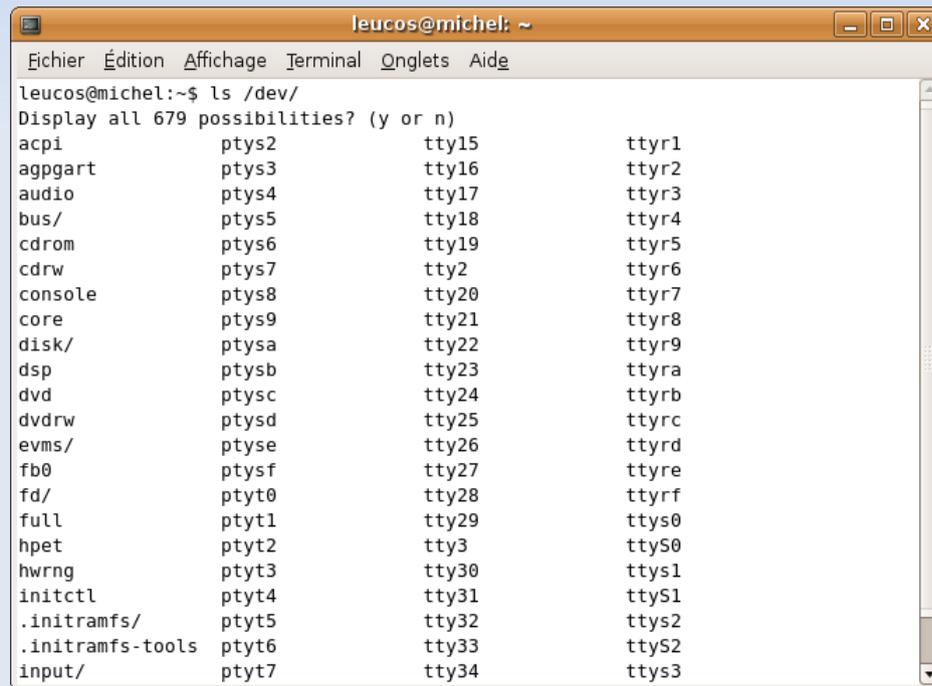
*leitmotiv perl, qui s'applique très bien à Unix*

# Unix

## Philosophie

### Sous unix, tout est fichier

- fichiers (!)
- répertoires
- devices
- liens
- pipes
- sockets



```
leucos@michel: ~  
Fichier Édition Affichage Terminal Onglets Aide  
leucos@michel:~$ ls /dev/  
Display all 679 possibilities? (y or n)  
acpi          ptys2        tty15        ttyr1  
agpgart       ptys3        tty16        ttyr2  
audio         ptys4        tty17        ttyr3  
bus/          ptys5        tty18        ttyr4  
cdrom         ptys6        tty19        ttyr5  
cdrw          ptys7        tty2         ttyr6  
console       ptys8        tty20        ttyr7  
core          ptys9        tty21        ttyr8  
disk/         ptysa        tty22        ttyr9  
dsp           ptysb        tty23        ttyra  
dvd           ptysc        tty24        ttyrb  
dvdrw        ptysd        tty25        ttyrc  
evms/         ptyse        tty26        ttyrd  
fb0           ptysf        tty27        ttyre  
fd/           ptyt0        tty28        ttyrf  
full          ptyt1        tty29        ttys0  
hpet          ptyt2        tty3         ttys0  
hwrng         ptyt3        tty30        ttys1  
initctl       ptyt4        tty31        ttys1  
.initramfs/   ptyt5        tty32        ttys2  
.initramfs-tools ptyt6      tty33        ttys2  
input/        ptyt7        tty34        ttys3
```

# Unix

## Philosophie

Sous unix, les fichiers doivent être

- lisibles par l'homme
- exploitables simplement par la machine

...donc sous Unix :

- tous les fichiers de configuration sont texte...  
...ou générés à partir de fichiers texte
- pas de binaire : les fichiers de configuration doivent être éditables simplement ( $\neq$ registry)
- pas de formats exotiques ( $\neq$ xml)
- pas de formats fermés ( $\neq$ doc,byo,...)

# Unix

## Philosophie

- Multitâche
  - plusieurs processus s'exécutent en même temps
  - l'OS à le contrôle des processus
- Multiutilisateur
  - plusieurs utilisateurs peuvent se connecter simultanément
  - plusieurs utilisateurs peuvent exécuter des processus simultanément

# Historique GNU

- GNU : « Gnu's Not Unix »
- Promu par la Free Software Foundation
- Objet : développement d'outils et de licences libres
  - 1984 : emacs
  - 1987 : gcc
  - 1989 : GPLv1
  - 1991 : GPLv2
  - 1997 : lesstif



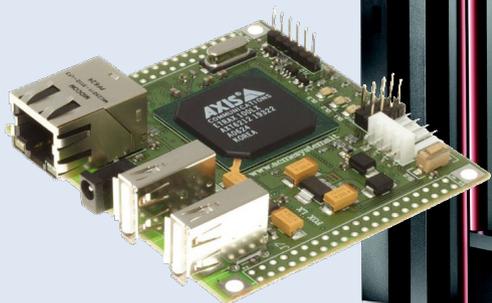
# Historique Linux

- Août 1991
  - 1<sup>ère</sup> version
  - code source : 63 Ko
- 1994 :
  - 2 branches
  - code source : 1 Mo
- Juin 1996 : v2.0
  - code source : 5 Mb
  - port Alpha
  - SMP
- Mars 2006 : v2.6.11, >200 Mb
- 20+ architectures supportées (x86, ppc, sparc, arm, α, avr32, ...)

```
> ----- Begin post from Linus -----
> From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
> Newsgroups: comp.os.minix
> Subject: What would you like to see most in minix?
> Summary: small poll for my new operating system
> Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
> Date: 25 Aug 91 20:57:08 GMT
> Organization: University of Helsinki
>
> Hello everybody out there using minix -
>
> I'm doing a (free) operating system (just a hobby, won't be big and
> professional like gnu) for 386(486) AT clones. This has been brewing
> since april, and is starting to get ready. I'd like any feedback on
> things people like/dislike in minix, as my OS resembles it somewhat
> (same physical layout of the file-system (due to practical reasons)
> among other things).
>
> I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
> This implies that I'll get something practical within a few months, and
> I'd like to know what features most people would want. Any suggestions
> are welcome, but I won't promise I'll implement them :-)
```

# Historique Linux

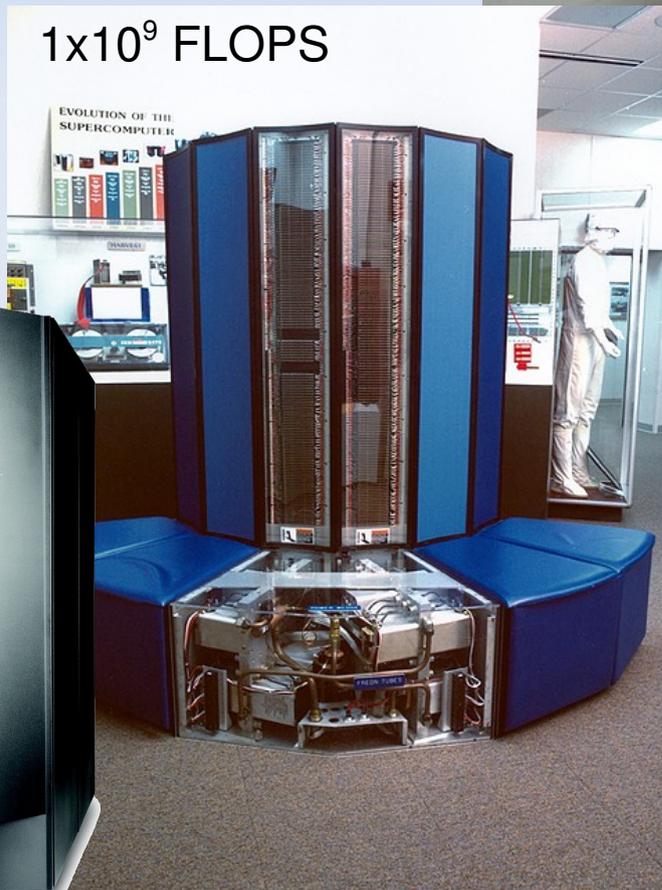
1 MIPS



$20 \times 10^3$  MIPS



$1 \times 10^9$  FLOPS



$360 \times 10^{12}$  FLOPS



# GNU/Linux

## Le kernel et l'OS

- Kernel : Linux
  - mise à disposition des ressources machines
    - entrées /dev, /proc, /sys
    - allocation mémoire
  - gestion de l'accès aux ressources machines
    - droits
    - ordonnancement
    - gestion des accès
  - modulaire
    - chargement déchargement de modules (drivers)
    - automatique ou manuel

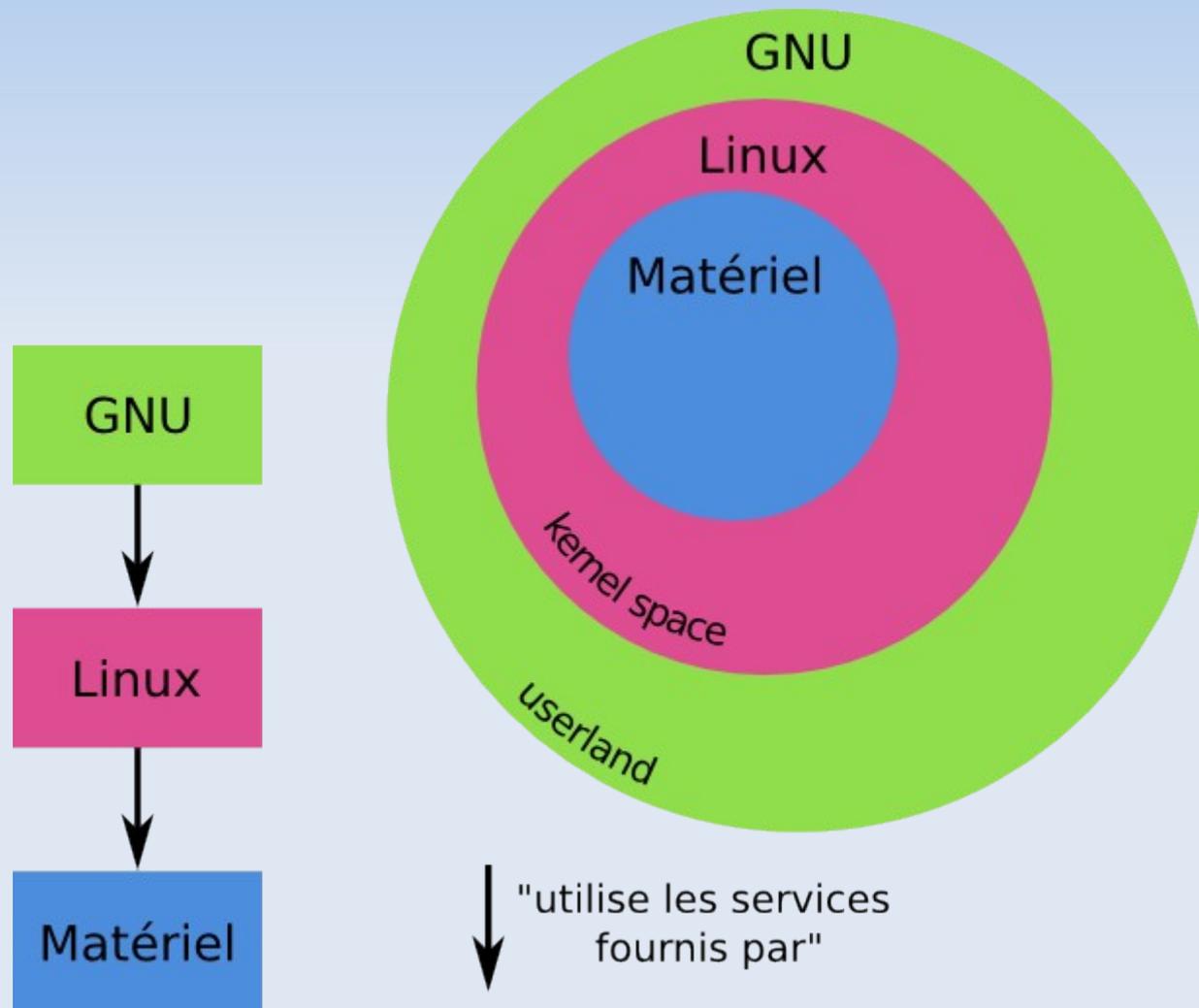
# GNU/Linux

## Le kernel et l'OS

- OS : GNU
  - gestion du système via l'interface proposée par le kernel
    - systèmes de fichiers
    - réseau
    - droits
    - périphériques
    - ...
  - sous forme d'utilitaires, ou de bibliothèques
    - shell, ls, rm, ...
    - libusb, libpam, ...

# GNU/Linux

## Le kernel et l'OS



# GNU/Linux

## Userland et kernel space

- « Kernel Space »
  - espace sensible
  - espace protégé
  - contient le kernel et ses modules
- « Userland »
  - espace utilisateur
  - espace libre
  - espace cloisonné

# GNU/Linux

## Les distributions

### « Versions » différentes de GNU/Linux

- Payantes (RHEL), semi-payantes (Mandriva) ou gratuites (presque toutes)
- Pour l'expert (Debian), le débutant (Kubuntu), le maniaque (LFS), le patient (Gentoo), le nostalgique (Yggdrasil, Slackware)
- Orientée bureautique (Ubuntu), appliance (Damn Small), serveur (Trustix) ou généraliste (SuSE, Fedora)
- Religieuses (Crux, Ubuntu CE, Ubuntu SE), païennes (les autres)
- Plus de 350 distributions sur le «marché»

# GNU/Linux

## Les distributions

- Points communs ?

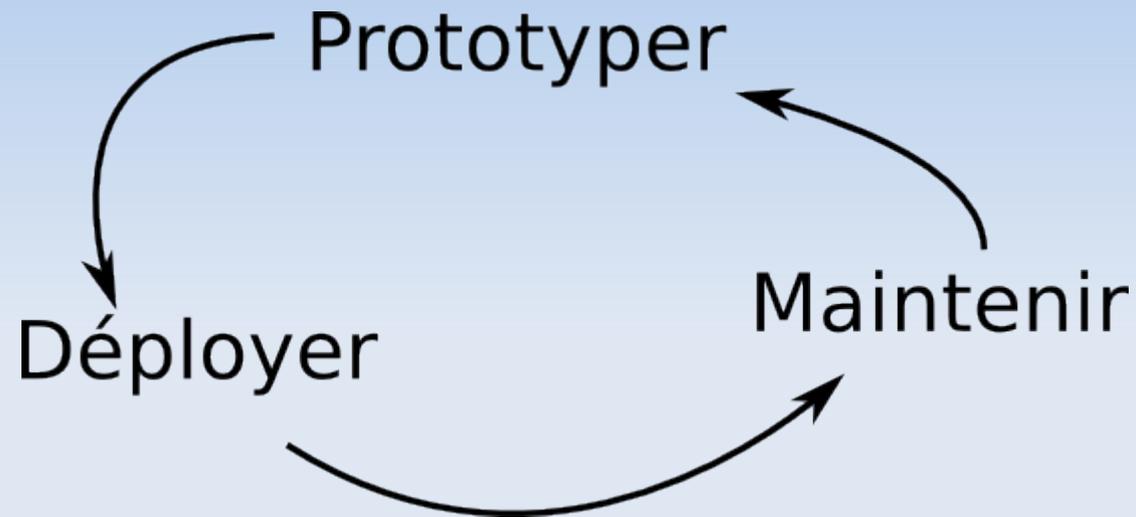
- kernel
- outils GNU

- Différences ?

- kernel
- outils GNU
- système de packages
- fichiers de configuration
- fichiers de démarrage
- organisation et type du filesystem
- philosophie
- canaux de distribution
- méthode d'installation, de configuration
- outils
- ....

# Administration Système

## Définition



« Faire un maximum de scripts pour en faire le moins possible »

*E. Arzur*

# Le Shell



*« Less sucks less more than more.  
That's why I use more less, and less more.. »*

Inconnu

# Shell

## Intérêt ?

- Interface utilisateur non graphique
  - terminaux texte
  - accès distant
- Interpréteur de scripts
  - traitement "par lots"
  - automatisation
- Lancement de tâches multiples
  - tâche combinées (pipes)
  - job control

# Shell

## Oui mais...

- C'est un programme "normal"
- Le choix est vaste
  - sh, sash, ksh, csh, tcsh, ash, bash, psh, fish,  
...
- On peut exécuter plusieurs shells en parallèle (unix est multitâche)
- Plusieurs utilisateurs peuvent exécuter un shell en même temps (unix est multi-utilisateurs)



# Shell



## Autopsie d'un « login »

/bin/login

```
QEMU
Using /etc/random-seed to initialize /dev/urandom.
INIT: Entering runlevel: 3
Going multiuser...
Starting syslogd daemons: /usr/sbin/syslogd /usr/sbin/klogd -c 3 -x
Starting PCMCIA services:
  <Probing for PCIC: edit /etc/rc.d/rc.pcmcia>
Activating IPv4 packet forwarding.
Starting Internet super-server daemon: /usr/sbin/inetd
Starting OpenSSH SSH daemon: /usr/sbin/sshd
Updating shared library links: /sbin/ldconfig
Starting ACPI daemon: /usr/sbin/acpid
Loading /usr/share/kbd/keymaps/i386/az
assuming iso-8859-1 cedilla
assuming iso-8859-1 acute
assuming iso-8859-1 diaeresis
assuming iso-8859-1 brokenbar
assuming iso-8859-1 threequarters
assuming iso-8859-1 currency
assuming iso-8859-1 onehalf
assuming iso-8859-1 onequarter

Welcome to Linux 2.6.18 (tty1)

darkstar login: _
```

/bin/bash

```
QEMU
  <Probing for PCIC: edit /etc/rc.d/rc.pcmcia>
Activating IPv4 packet forwarding.
Starting Internet super-server daemon: /usr/sbin/inetd
Starting OpenSSH SSH daemon: /usr/sbin/sshd
Updating shared library links: /sbin/ldconfig
Starting ACPI daemon: /usr/sbin/acpid
Loading /usr/share/kbd/keymaps/i386/azerty/fr-latin9.map.gz
assuming iso-8859-1 cedilla
assuming iso-8859-1 acute
assuming iso-8859-1 diaeresis
assuming iso-8859-1 brokenbar
assuming iso-8859-1 threequarters
assuming iso-8859-1 currency
assuming iso-8859-1 onehalf
assuming iso-8859-1 onequarter

Welcome to Linux 2.6.18 (tty1)

darkstar login: root
Password:
Linux 2.6.18.
Last login: Sat Nov  4 00:38:08 +0100 2006 on tty1.
You have mail.
root@darkstar:~# _
```

# Shell

## Environnement

- Un shell donné s'exécute dans un environnement propre et clos contenant des variables
- Cet environnement est :
  - est modifiable
  - est transitoire
- L'environnement initial
  - est défini au niveau système
  - peut être modifié par l'utilisateur

# Shell Environnement



```
leucos@michel: ~  
Fichier Édition Affichage Terminal Onglets Aide  
leucos@michel:~$ TOTO="shell numero 1"  
leucos@michel:~$ echo $TOTO  
shell numero 1  
leucos@michel:~$ █
```

```
leucos@michel: ~  
Fichier Édition Affichage Terminal Onglets Aide  
leucos@michel:~$ echo $TOTO  
leucos@michel:~$ █
```

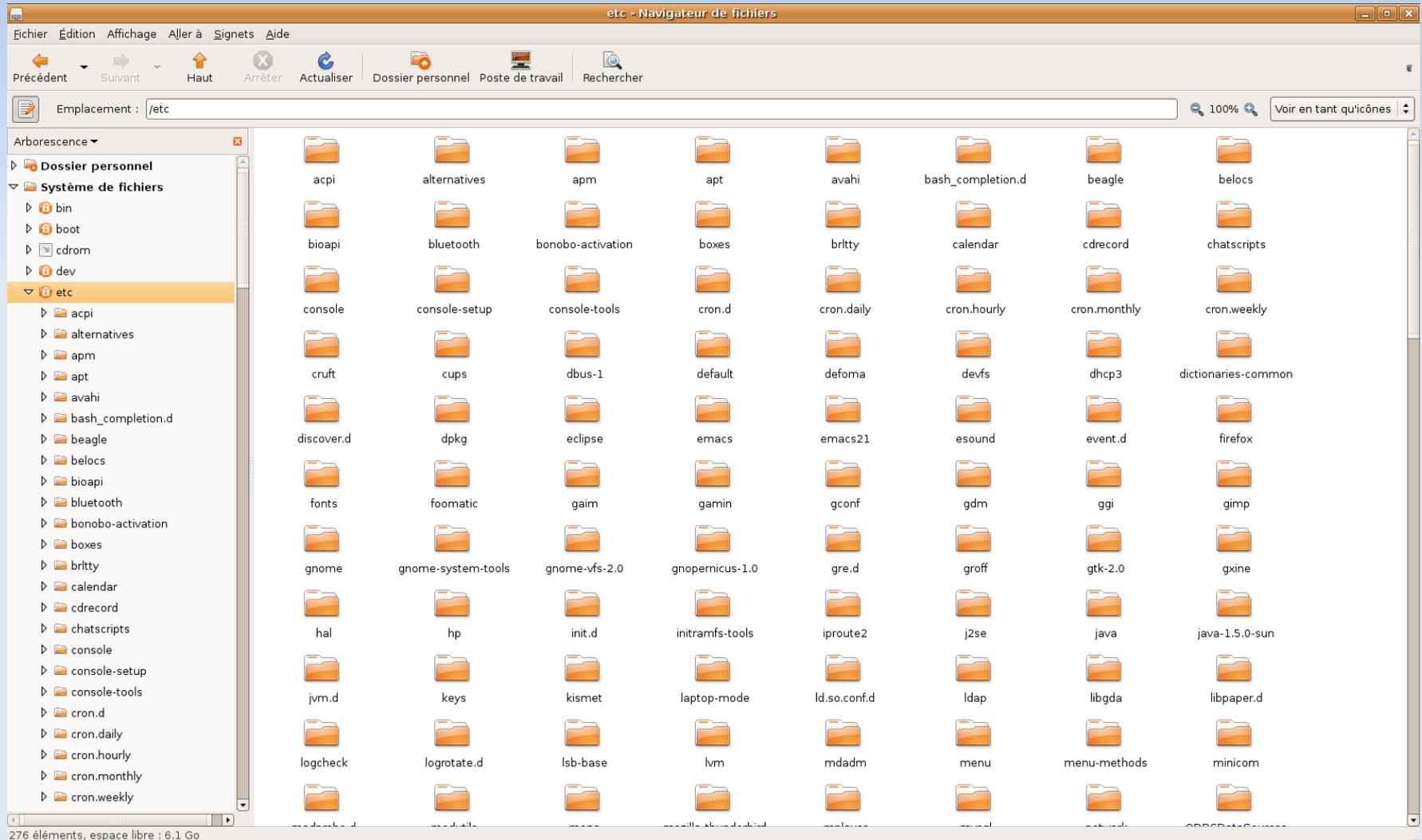
# Filesystem

## Le système de fichiers (filesystem)

- Organisé hiérarchiquement dans sa totalité depuis la racine («/»)
- Sensible à la casse (des caractères)
- Utilise '/' pour séparer les répertoires dans un chemin
- Peut contenir n'importe quel caractère
- Taille pratique d'un nom de fichier illimitée
- Pas de notion «d'extensions»
- Distro-dépendant malgré le «standard»

# Filesystem

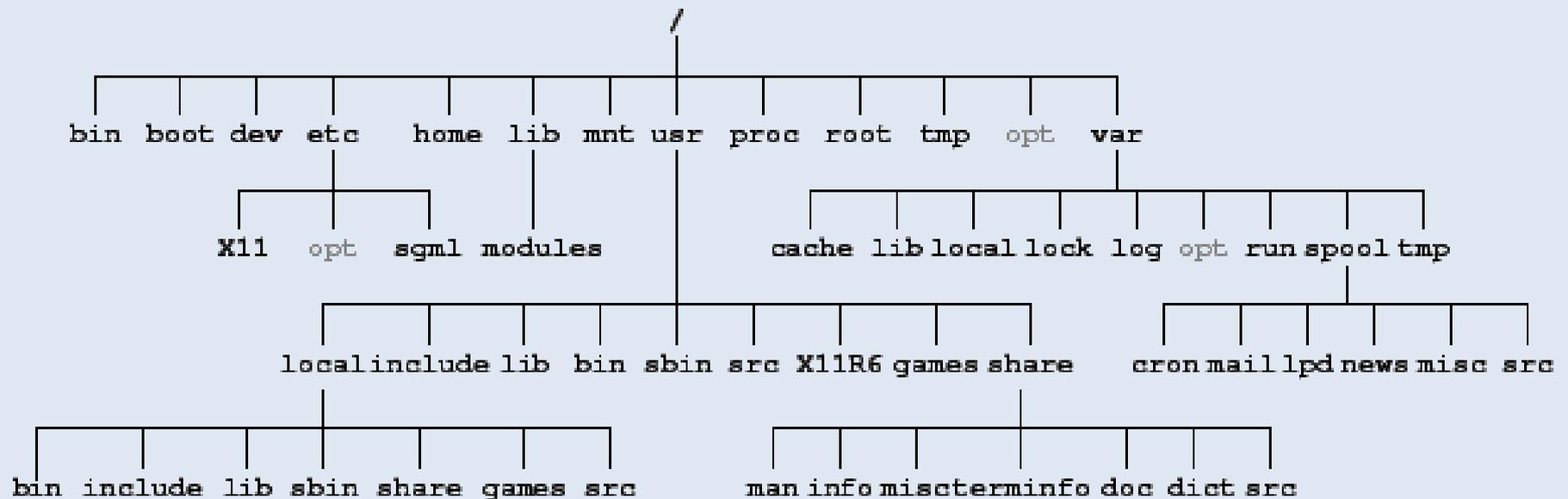
## Le système de fichiers (filesystem)



# Filesystem

## L'organisation du filesystem (fs)

### *Une seule arborescence*



# Filesystem



## Balade dans le «fs»

- Se déplacer dans le filesystem
  - `cd chemin` (chemin relatif)
  - `cd /chemin` (chemin absolu)
- Voir le contenu du répertoire
  - `ls` (contenu du répertoire courant)
  - `ls chemin` (contenu de « chemin »)
- Connaître le répertoire courant
  - `pwd` (print working directory)



### Répertoires à part :

- / : racine du filesystem
- . : répertoire courant
- .. : répertoire parent
- ~ : répertoire maison («home dir», correspond à \$HOME)

# Filesystem

## Lieux touristiques du «fs»

### Répertoire essentiels au fonctionnement

/ : racine

/bin, /sbin : binaires systèmes

/etc : configuration système

/boot : kernel et 2<sup>eme</sup> étage du bootloader

# Filesystem

## Lieux touristiques du «fs»

/usr : binaires d'usage courant

/home : répertoires utilisateurs

/var : données variables (sgbd, logs)

/proc : information système temps réel

/mnt, /media : point montage temporaire

/sys : bus systèmes

/tmp : répertoire temporaire

lost+found : objets trouvés

# Shell

## \$PATH



- Variable d'environnement
- Contient une suite de chemins séparés par ':'
- Les exécutable seront recherchés dans ces chemins

A terminal window titled 'leucos@michel: ~' with standard window controls. The terminal shows the command 'echo \$PATH' being executed, resulting in the output: '/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11:/usr/games'. The prompt 'leucos@michel:~\$' is visible at the end of the line.

```
leucos@michel: ~  
leucos@michel:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin  
:/bin:/usr/bin/X11:/usr/games  
leucos@michel:~$
```



## Répertoire '.' danger !

- risque d'exécuter un trojan
  - n'importe qui peut écrire dans /tmp (entre autres) !
- le shell perd du temps à chaque 'cd'

# Shell

## Entrée/Sorties

- Par défaut tout processus possède :
  - 1 entrée
  - 2 sorties
- Ces I/O sont distinctes pour chaque processus
- Pour les programmes interactifs (comme les shells) :
  - les entrées proviennent du clavier
  - les sorties s'affichent sur l'écran

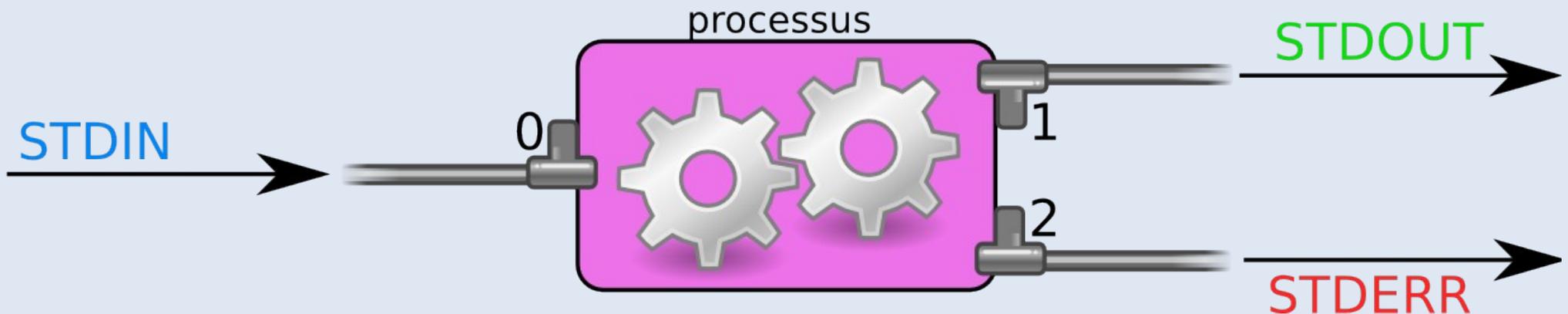
# Shell

## Entrée/Sorties

**STDIN** (entrée standard) : ce qui est envoyé vers le processus

**STDOUT** (sortie standard) : ce qui est envoyé par le processus

**STDERR** (sortie erreur standard) : les **erreurs** renvoyés par le processus



# Shell

## Entrée/Sorties

Ces entrées/sorties standard sont en fait des noms symboliques, correspondant à des « descripteurs de fichiers » :

- **STDIN** (INput) : descripteur de fichier 0
- **STDOUT** (OUTput) : descripteur de fichier 1
- **STDERR** (ERRor) : descripteur de fichier 2

# Shell



## Entrée/Sorties : cat

cat : copie **STDIN** (ou un fichier) sur **STDOUT**  
(écrit les erreurs, s'il y a, sur **STDERR**)

```
user@host:~$ cat
azerty
azerty
qsdffg
qsdffg

user@host:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      michel

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0     ip6-localnet

user@host:~$ cat /etc/bidon
cat: /etc/bidon: Aucun fichier ou répertoire de ce type
user@host:~$
```

# Shell

## Redirections d'E/S

Les E/S peuvent être redirigées de ou vers un fichier

**processus < fichier**

STDIN provient du fichier (et non plus du clavier)

**processus > fichier**

STDOUT est écrit dans fichier (et non plus sur le terminal)

**processus 2> fichier**

STDERR est écrit dans fichier (et non plus sur le terminal)

**processus > fichier1 2> fichier2**

STDOUT est écrit dans fichier1 et STDERR dans fichier2

# Shell

## Redirections d'E/S



```
user@pluton:~$ cat < /etc/hostname
pluton
user@pluton:~$ cat /etc/hostname
pluton
user@pluton:~$ cat /etc/hostname > /tmp/test
user@pluton:~$ cat /tmp/test
pluton
user@pluton:~$ cat /etc/hostname > /dev/null
user@pluton:~$ cat < /etc/hostname > /dev/null
user@pluton:~$ cat /etc/hostname
pluton
user@pluton:~$ cat /etc/portnaouak
cat: /etc/portnaouak: Aucun fichier ou répertoire de ce type
user@pluton:~$ cat /etc/portnaouak 2> /dev/null
user@pluton:~$ cat /etc/hostname /etc/portnaouak > out.txt 2> err.txt
user@pluton:~$ cat out.txt
pluton
user@pluton:~$ cat err.txt
cat: /etc/portnaouak: Aucun fichier ou répertoire de ce type
```

# Shell

## Redirections d'E/S



```
user@pluton:~$ cat /etc/hostname
pluton
user@pluton:~$ cat /etc/hostname >> /tmp/test
user@pluton:~$ cat /tmp/test
pluton
pluton
user@pluton:~$ cat /etc/hostname > /tmp/test
user@pluton:~$ cat /tmp/test
pluton
user@pluton:~$ cat /etc/hostname > /tmp/test
user@pluton:~$ cat /tmp/test
pluton
user@pluton:~$ cat /etc/hostname >> /tmp/test
user@pluton:~$ cat /tmp/test
pluton
pluton
user@pluton:~$
```

# Shell



## Entrées/sorties : devices spéciaux

Plusieurs "devices" (fichiers dans /dev) ont une vocation particulière :

### **/dev/null**

trou noir annihilant tout ce qui lui est envoyé

### **/dev/zero**

envoie des zéros ad-vitam

### **/dev/random /dev/urandom**

fournisseurs officiels de hasard

### **/dev/full**

dispositif hypochondriaque : se plaint toujours (d'être plein)



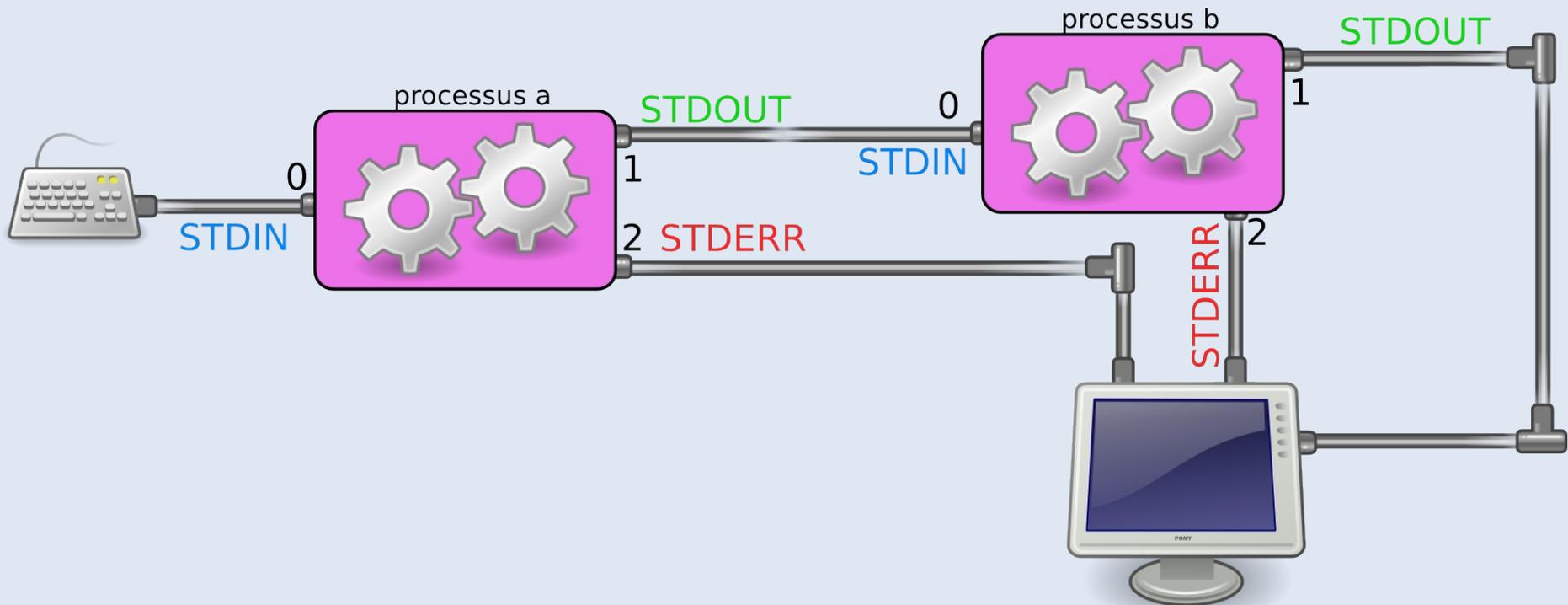
## Redirections d'E/S : travaux

- 1) Copier le contenu de `/etc/passwd` dans le fichier `/tmp/users.txt`
- 2) Ecrire «`linus`» à la fin de `users.txt`
- 3) Vider le fichier `users.txt`
- 4) Remplir `users.txt` de "zéros" (utiliser le dispositif fournisseur de zéros : `/dev/zero`)
- 5) Rediriger l'erreur standard de `'ls -lR /'` dans `/tmp/users.txt`
- 6) Vider le fichier `users.txt` (d'une autre manière qu'en 3)

# Shell Pipes



- Les «pipes» (pipelines) permettent d'envoyer la sortie d'une commande (**STDOUT**) à l'entrée d'une autre (**STDIN**) :



# Shell

## Pipes



- On trouve très souvent la commande **grep** au milieu de pipelines
- **grep** permet de n'afficher une ligne que si elle contient une chaîne de caractères donnée
- Sa syntaxe est :

```
grep chaîne fichier
```

affiche les lignes de fichier contenant "chaîne"

```
grep chaîne
```

affiche les lignes lues sur l'entrée standard (STDIN) contenant "chaîne"

# Shell



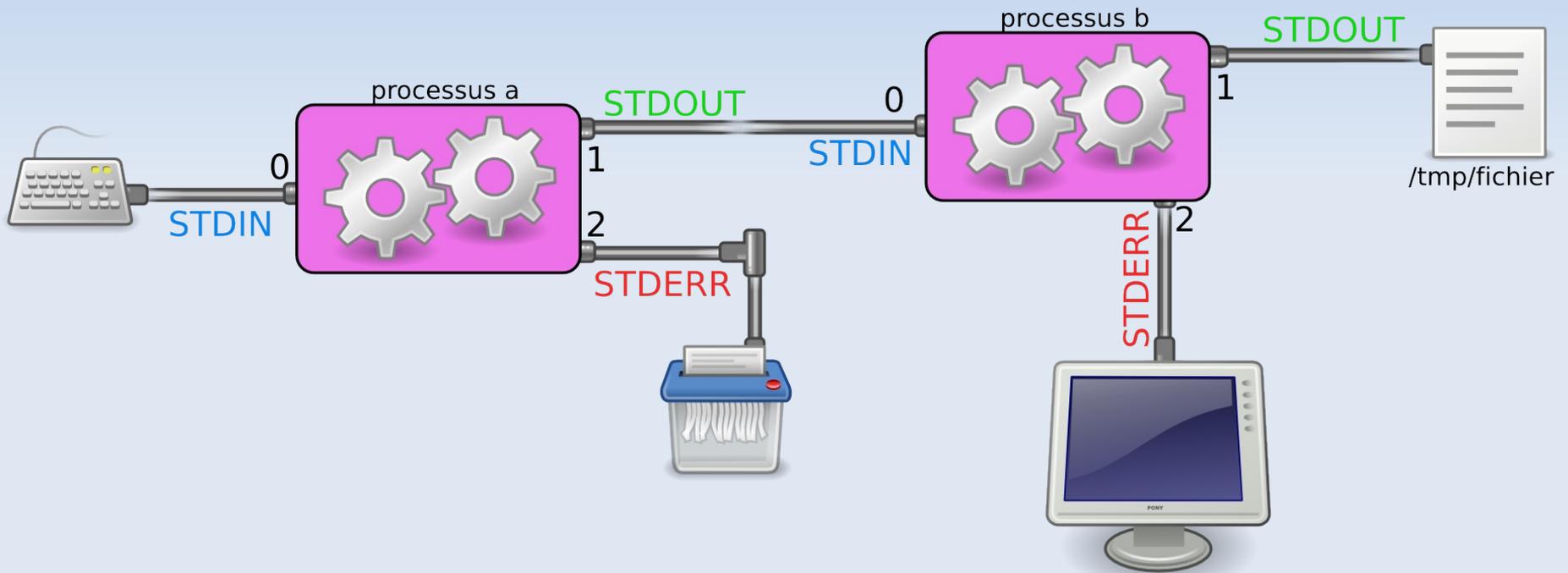
## Pipes : exemples (compliqués)

```
user@pluton:~$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
user@pluton:~$ ls | grep test
test.txt
user@pluton:~$ ip link | grep UP
1: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
3: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,10000> mtu 1500 qdisc pfifo_fast qlen 100
user@pluton:~$ ip link | grep UP > uplinks.txt
user@pluton:~$ cat uplinks.txt | grep eth
3: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
user@pluton:~$ ip link | grep UP | grep eth
3: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
user@pluton:~$ history | awk '{ print $2 }' | sort | uniq -c | sort -nr -k1 | head -10
 164 ls
   74 cd
   62 ssh
   55 ping
   55 man
   53 make
   51 ip
   47 more
user@pluton:~$
```

Il n'y a pas de limitation pratique au nombre de "pipes" que l'on peut enchaîner...

# Shell Pipes

- Les pipes et les redirections peuvent être combinées



```
a 2> /dev/null | b > /tmp/fichier
```

# Shell



## Pipes & Redirections

Attention à l'ordre des redirections et des pipelines

**a | b 2> c**

- le résultat de la commande a est passé à b
- les erreurs de b sont redirigées dans le fichier c

**a 2> c | b**

- le résultat de la commande a est passé à b
- les erreurs de a sont redirigées dans le fichier c

```
user@pluton:~$ cat /etc/toto /etc/passwd | grep root 2> /dev/null
cat: /etc/toto: Aucun fichier ou répertoire de ce type
root:x:0:0:root:/root:/bin/bash
user@pluton:~$ cat /etc/toto /etc/passwd root 2> /dev/null | grep root
root:x:0:0:root:/root:/bin/bash
user@pluton:~$
```

# Shell

## Complétion



- Permet de compléter une saisie utilisateur dans le shell
- Affecté à la touche « tab »
- La complétion affiche la plus grande correspondance unique
- S'applique aux commandes et à leurs arguments (selon configuration)
- « tab-tab » affiche toutes les correspondances possibles

# Shell

## Editeur de ligne



- l'éditeur de ligne permet d'éditer....  
... la ligne !
- raccourcis clavier identiques à Emacs

**CTRL** + **a** : début de ligne

**CTRL** + **e** : fin de ligne

**CTRL** + **k** : coupe de la position courante jusqu'à la fin de la ligne

**CTRL** + **y** : colle ce qui a été précédemment coupé

# Shell

## Historique

- bash mémorise toutes les commandes dans un d'historique
- cet historique est enregistré dans un fichier lorsque le shell se termine
- cet historique est lu depuis un fichier au démarrage d'un shell
- le nom du fichier est défini dans `$HISTFILE` et la taille de l'historique dans `$HISTFILESIZE`
- le rappel de commandes :
  - par numéro (!numéro)
  - par nom (!debutnom)
  - après recherche (ctrl-r)

# Shell

## Historique



```
user@host:~$ echo $HISTFILE
/home/user/.bash_history
user@host:~$ echo $HISTFILESIZE
1500
user@host:~$ history | tail -n 5
 1429  history
 1430  kill -9 18487
 1431  echo $HISTFILE
 1432  echo $HISTFILESIZE
 1433  history | tail -n 5
user@host:~$ tail -n 5 $HISTFILE
ls
history
ls /ttest
history
kill -9 18487
user@host:~$ !1431
user@host:~$ echo $HISTFILE
/home/user/.bash_history
user@host:~$ !echo
echo $HISTFILE
/home/user/.bash_history
user@host:~$
```

# Shell

## Historique



```
user@host:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
...
postfix:x:111:118:./var/spool/postfix:/bin/false
user@host:~$ !cat | grep root
root:x:0:0:root:/root:/bin/bash
user@host:~$ bash
user@host:~$ history | tail -n 5
1476 ls
1427 history
1428 ls /ttest
1429 history
1430 kill -9 18487
user@host:~$
```

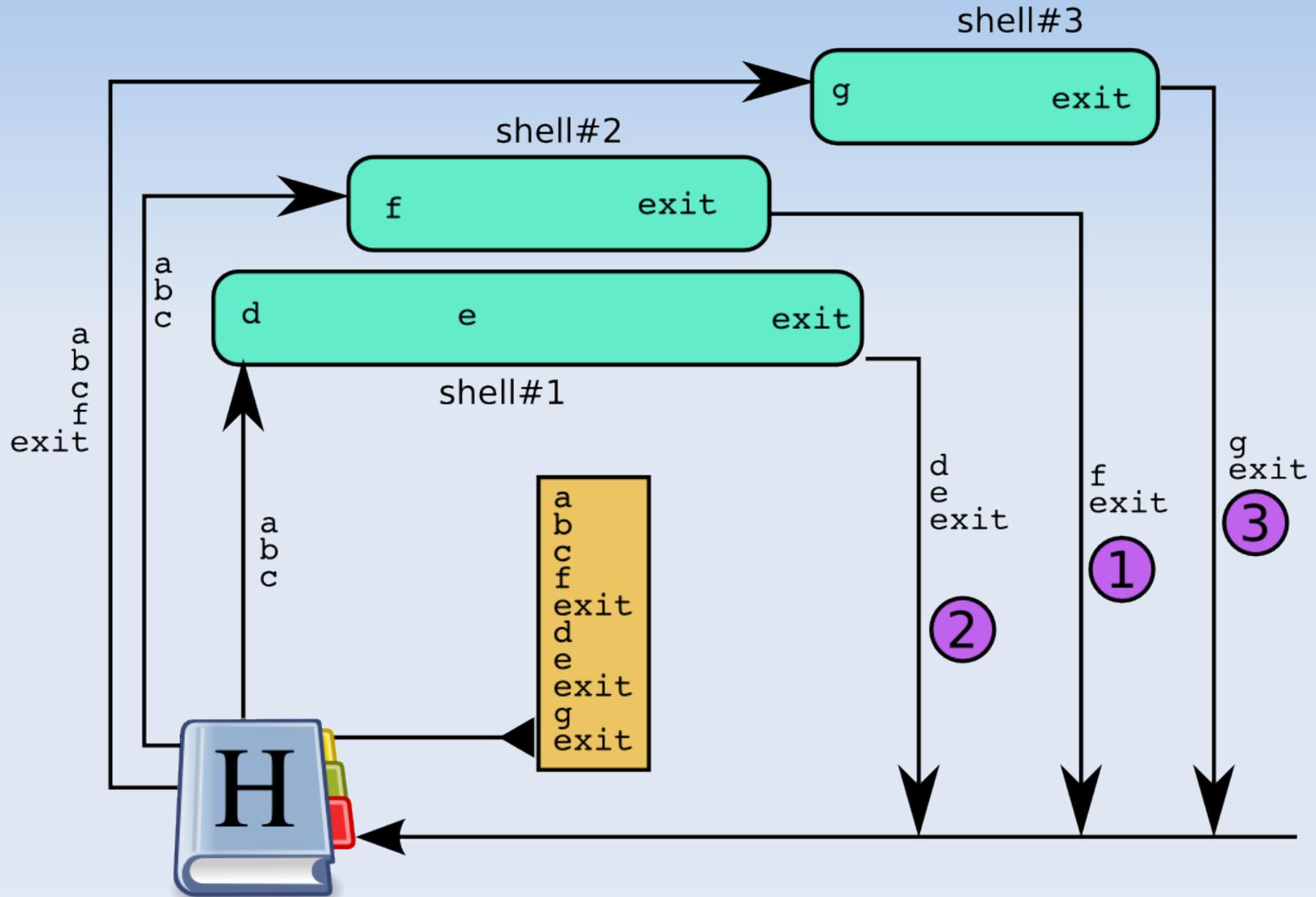
# Shell

## Historique

### Remarques

- si le shell B est ouvert alors que le shell A est en cours, B ne verra pas les commandes de A dans l'historique
- si le shell B est fermé avant le shell A, les commandes de B seront plus haut dans l'historique
- si le shell est tué, il ne pourra pas mettre l'historique à jour

# Shell Historique



# Shell

## Globbing

Permet de nommer plusieurs fichiers d'un seul coup grâce à des « jokers » (wildcards)

? : accepte un seul caractère

\* : accepte n'importe quel caractère 0 ou n fois

[**chars**] : denote une liste de caractères acceptée (liste, suite alphabétique ou suite numérique)

[**^chars**] ou [**!chars**] : dénote une classe de caractères refusée (liste, suite alphabétique ou suite numérique)

{**e1, e2, e3**} : remplace les arguments par les éléments de la liste

{**e1, e2**}{**e3, e4**} : remplace les arguments par le produit cartésien des deux listes

# Shell

## Globbing

- abc?** : accepte un fichier de 4 lettres commençant par abc
- abc\*** : accepte un fichier de 3 lettres ou plus commençant par abc
- \*abc\*** : accepte un fichier de 3 lettres ou plus contenant abc
- [ab]c** : accepte un fichier de 2 lettres commençant par 'a' ou 'b'
- [a-m]\*** : accepte un fichier commençant par une lettre comprise alphabétiquement entre 'a' et 'm' inclus
- \*[0-1][0-9]** : accepte un fichier se terminant par un nombre compris entre '00' et '19'
- [13579]\*[a-z]** : accepte un fichier commençant par un nombre impair et se terminant par une minuscule

# Shell

## Globbing



`[^13579]*[^a-z]` : accepte un fichier commençant par un nombre pair et se terminant par une majuscule

`{a,b,c}` : accepte les fichiers nommés 'a', 'b' ou 'c'

`{a*,*b*,*c}` : accepte les fichiers dont le nom commence par 'a', contient 'b' ou se termine par 'c'

`*{.doc,.odt,.rtf}` : accepte les fichiers nommés '\*.doc', '\*.odt' ou '\*.rtf'

`{photo,image}{.jpg,.png}` : accepte les fichiers nommés 'photo.jpg', 'photo.png', 'image.jpg' ou 'image.png'

`{jean,pierre}{,-jean}` : accepte les fichiers dont le nom est 'jean', 'jean-jean', 'pierre' ou 'pierre-jean'



## Globbing : classes

**Les classes sont des listes prédéfinies de caractères, utilisables entre [: et :] :**

`alnum` : [a-zA-Z0-9]  
`alpha` : [a-zA-Z]  
`ascii` : caractère ascii (...)  
`blank` : espace ou tabulation  
`cntrl` : caractère de contrôle  
`digit` : [0-9]  
`graph` : caractères  
    imprimables et visibles  
`lower` : [a-z]  
`print` : caractère imprimable

`punct` : ponctuation  
`space` : espace, tabulation, ...  
`upper` : [A-Z]  
`word` : [a-zA-Z0-9\_]  
`xdigit` : [a-fA-F0-9]

Exemple :

[[:alnum:]] : accepte un  
    caractère alphanumérique

Voir : man 7 glob



## Globbing : les limites

- On ne peut pas matcher '/'
- On doit matcher explicitement le caractère '.'
- Ce ne sont pas des expressions régulières
- On ne peut donc pas appliquer de numération à des classes

Il est par exemple impossible de matcher tous les noms de fichiers se terminant par des nombres

# Commandes de base

## De l'aide : man



### man *commande*

affiche le manuel de la commande commande

- options
  - t : produit une sortie postscript pour impression
- exemples

```
man -t ls | lpr
man man
```

```
leucos@michel: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
MAN(1)                                     Manual pager utils                                     MAN(1)

NAME
  man - an interface to the on-line reference manuals

SYNOPSIS
  man [-c|-w|-tZ] [-H[browser]] [-T[device]] [-adhu7V] [-i|-I] [-m system[,...]] [-L locale]
  [-p string] [-C file] [-M path] [-P pager] [-r prompt] [-S list] [-e extension] [[section]
  page ...] ...
  man -l [-7] [-tZ] [-H[browser]] [-T[device]] [-p string] [-P pager] [-r prompt] file ...
  man -k [apropos options] regexp ...
  man -f [whatis options] page ...

DESCRIPTION
  man is the system's manual pager. Each page argument given to man is normally the name of
  a program, utility or function. The manual page associated with each of these arguments
  is then found and displayed. A section, if provided, will direct man to look only in that
  section of the manual. The default action is to search in all of the available sections,
  following a pre-defined order and to show only the first page found, even if page exists
  in several sections.

  The table below shows the section numbers of the manual followed by the types of pages
  they contain.

  1 Executable programs or shell commands
Manual page man(1) line 1
```

# Commandes de base

## De l'aide : man



man divise la documentation en 9 sections :

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in /dev)
5. File formats and conventions (e.g. /etc/passwd)
6. Games
7. Miscellaneous (including macro packages and conventions)
8. System administration commands (usually only for root)
9. Kernel routines [Non standard]

cela permet d'éviter les ambiguïtés :

```
man glob ≠ man 7 glob
```

# Commandes de base



## Aide : help, apropos, whatis

### help *commande*

affiche le manuel d'une commande interne (builtin)

### apropos *sujet*

affiche les pages de man correspondant au sujet

### whatis *commande*

affiche une information succincte sur la commande

A terminal window titled 'leucos@michel: ~/tmp' with a menu bar containing 'Fichier', 'Édition', 'Affichage', 'Terminal', 'Onglets', and 'Aide'. The terminal shows the following commands and their outputs:

```
leucos@michel:~/tmp$ apropos bash
airmon is a bash script designed to turn wireless cards into monitor mode. (1) [airmon] - (sujet inconnu)
airmon-ng is a bash script designed to turn wireless cards into monitor mode. (1) [airmon-ng] - (sujet inconnu)
bash (1) - GNU Bourne-Again SHell
bash-builtins (7) - bash built-in commands, see bash(1)
bashbug (1) - report a bug in bash
builtins (7) - bash built-in commands, see bash(1)
checkbashisms (1) - check for bashisms in /bin/sh scripts
rbash (1) - restricted bash, see bash(1)
leucos@michel:~/tmp$ whatis bash
bash (1) - GNU Bourne-Again SHell
leucos@michel:~/tmp$ help bash
bash: help: aucun sujet d'aide ne correspond à `bash'. Essayez `help help' ou `man -k bash' ou `info bash'.
leucos@michel:~/tmp$
```

# Commandes de base



## cd

### `cd` *argument*

« rentre » dans le répertoire *argument*

Après un changement de répertoire courant, l'ancien répertoire courant est stocké dans '-', permettant d'y revenir avec 'cd -'

- exemples

```
cd
  (equivaut à 'cd ~' et 'cd $HOME' )
cd ..
cd -
cd /tmp
cd tmp
```

# Commandes de base



## \$CDPATH

- \$CDPATH permet d'offrir des chemins directement accessibles via « cd »
- Comme pour \$PATH, les chemins sont concaténés avec ':' :

```
$ echo $CDPATH
```

```
/home/user:/var:/home/user/doc
```

```
$
```

# Commandes de base



## ls

`ls -a`

Liste tous les fichiers (incluant .\*)

`ls -l`

Affichage long (droits, propriétaire, groupe, taille, date)

`ls -R`

Liste les fichiers récursivement

`ls -t`

Affiche les plus récents en premier

`ls -S`

Affiche les plus gros en premier

`ls -1`

Affiche le listing sur une colonne

Les options se combinent directement entre-elles

ex : `ls -laRt`

# Commandes de base



## ls

### *ls option argument*

liste les fichiers/répertoires correspondant à argument ou dans le répertoire argument

- options
  - a : affiche tous les fichiers (y compris ceux commençant par '.')
  - l : listing étendu
  - R : récursif
  - S : tri par taille
  - t : tri par date de modification
  - 1 : affichage sur une colonne

- exemples

```
ls *.txt
ls /etc
ls /etc/host*
ls /etc/rc[1-3].d
ls /media/win/Program\ Files/
ls -laR ~
```

# Commandes de base



## {mk,rm}dir

`mkdir [-p] répertoire1 répertoire2 ...`

crée les répertoires *répertoire1*, *répertoire2*, ...

l'option -p per et *destination* (fichier ou répertoire)

- options
  - p : crée les répertoires supérieurs si nécessaire
- exemples

```
mkdir $HOME/documents
```

```
mkdir test $HOME/images_iso
```

```
mkdir -p $HOME/documents/personnel/{photos, factures}
```

`rmdir répertoire1 répertoire2`

supprime les répertoires *répertoire1*, *répertoire2*, ...

ces répertoires doivent être vides pour pouvoir être supprimés (utiliser `rm -rf` sinon)

- exemples

```
rmdir $HOME/documents
```

```
rmdir test $HOME/images_iso
```

```
rmdir $HOME/documents/personnel/{photos, factures, }
```

# Commandes de base



## cp & mv

### cp *source destination*

copie la *source* (fichier ou répertoire) vers la *destination* (fichier ou répertoire)

- options
  - p : préserve les droits
  - R : récursif
  - f : force l'écrasement de la destination

#### - exemples

```
cp *.txt /tmp
cp test.txt toast.txt
cp -Rf /home /var/backup
```

### mv *source destination*

déplace la *source* (fichier ou répertoire) vers la *destination* (fichier ou répertoire)  
permet aussi de renommer un fichier

#### - exemples

```
mv *.txt /tmp
mv test.txt toast.txt
```

# Commandes de base



## rm & touch

### rm *argument*

Supprime le fichier ou répertoire *argument*

- options
  - R : récursif
  - f : force la suppression

- exemples

```
rm -rf /  
rm toast.txt
```

### touch *fichier*

Crée le fichier s'il n'existe pas, ou met la date de modification du fichier à l'heure courant s'il existe

La commande ':>' permet aussi de créer un fichier

- exemples

```
touch toast.txt
```

# Commandes de base

## egrep

`egrep options patron fichier`

liste les fichiers/répertoires contenant une chaîne correspondant à *patron*.  
si *fichier* n'est pas spécifié, `grep` travaille sur STDIN

- options
  - v : inverse le comportement de `egrep` (n'affiche que les lignes qui ne correspondent pas)
  - i : insensible à la casse
  - R : récursif
  
- patrons d'expression régulières
  - si l'on utilise `grep` au lieu de `egrep`, il faut mettre un `\` devant les caractères `?`, `+`, `{`, `}`, `|`, `(`, et `)`
  
  - `.` n'importe quel caractère
  - `*` le caractère précédent 0 ou plusieurs fois
  - `+` le caractère précédent 1 fois au moins
  - `?` le caractère précédent 0 ou 1 fois
  - `{n}` le caractère précédent exactement n fois
  - `{m,n}` le caractère précédent de m à n fois
  - `{n,}` le caractère précédent n fois ou plus
  - `[a-z]` un caractère en minuscule
  - `[a-zA-Z]` une lettre
  - `[0-9]` un chiffre
  - `^/$` le début/la fin de ligne
  - `|` séparateur pour spécifier de multiples expressions (ou logique)

# Commandes de base



## egrep

- Matcher une ligne commençant par « foo » :  
egrep "^foo.\*"
- Matcher une ligne commençant par « foo » ou contenant « bar » :  
egrep "^foo|bar"
- Matcher une ligne commençant par « foo » ou commençant par « bar » :  
egrep "^(foo|bar)"
- Matcher une ligne commençant par « foo » et contenant « bar » :  
egrep "^foo.\*bar"
- Matcher une ligne commençant par « foo » et se terminant par « bar » :  
egrep "^foo.\*bar\$"
- Matcher une ligne se terminant par un espace suivi d'un nombre de 1 à 3 chiffres :  
egrep "[[:space:]]+[0-9]{1,3}\$"
- Matcher une ligne se terminant par un espace suivi d'un nombre de 1 à 3 chiffres :  
egrep "[[:space:]]+[0-9]{1,3}\$"
- Matcher une ligne contenant le caractère '\*' :  
egrep "\*"
- Matcher une ligne contenant le caractère '\*' ou le caractère '?' :  
egrep "\*|?"

# Commandes de base



## cat, less, tee et wc

`cat fichier1 fichier2 ...`

affiche le contenu de *fichier1 fichier2 ...* sur la sortie standard  
si `cat` est appelé sans arguments, la source est l'entrée standard

- exemple  
`cat /dev/urandom`

`less fichier1 fichier2 ...`

comme `cat`, affiche le contenu de *fichier1 fichier2 ...* sur la sortie standard mais effectue un arrêt à chaque page  
si `less` est appelé sans arguments, la source est l'entrée standard

- exemple  
`less /etc/password`

`tee fichier`

duplique l'entrée standard vers la sortie standard et dans un fichier

- exemple  
`vmstat 1 | tee toto`

`wc option fichier`

compte le nombre de lignes (-l), bytes (-c), mots (-w) ... dans *fichier* (ou sur STDIN si aucun fichier n'est spécifié)

- exemple  
`wc -l /etc/passwd`

# Commandes de base



## tail et head

`head [-nX] fichier1 fichier2 ...`

affiche les X dernières lignes de fichier1 fichier2 ... sur la sortie standard  
si `tail` est appelé sans arguments, la source est l'entrée standard

- exemple

```
head -n1 /etc/passwd
```

`tail [-nX] [-f] fichier1 fichier2 ...`

affiche les X dernières lignes de *fichier1 fichier2 ...* sur la sortie standard  
si `tail` est appelé sans arguments, la source est l'entrée standard et le nombre de lignes est 10

l'option `-f` permet de faire un 'tail' continu sur un fichier qui croit

- exemple

```
tail -n5 /var/log/syslog
```

```
tail -f /var/log/syslog
```

Une combinaison des deux permet d'afficher la nième ligne d'un fichier :

```
head -n10 /etc/passwd | tail -n1 : affiche la 9ème (10-1) ligne de /etc/passwd
```

```
head -n20 /etc/group | tail -n3 : affiche les lignes 17 à 20 (20-3 -> 20) lignes de /etc/group
```

# Commandes de base



## Versions 'z' & 'bz'

`zcat fichier1 fichier2 ...`

`bzcat fichier1 fichier2 ...`

comme `cat`, mais sur des fichiers gzippés/bzippés

si `zcat/bzcat` est appelé sans arguments, la source est l'entrée standard

- exemples

```
zcat myfile.gz
```

```
bzcat myfile.bz2
```

`zless fichier1 fichier2 ...`

`bzless fichier1 fichier2 ...`

comme `less`, mais sur des fichiers gzippés/bzippés

si `less` est appelé sans arguments, la source est l'entrée standard

- exemples

```
bzless myfile.bz2
```

`zgrep options patron fichier`

duplique l'entrée standard vers la sortie standard et dans un fichier

- exemples

```
zgrep user myfile.gz
```



## Commandes multiples

- Le shell permet l'exécution de commandes successives en les séparant par ';' :

```
ls > ~/liste.txt; cd -; ls >> ~/liste.txt
```

- Le shell permet d'exécuter une commande uniquement si la précédente a marché :

```
cat fichier > /dev/null && echo "le fichier existe"  
grep -i "effacez moi" fichier && rm -f fichier
```

- Le shell permet d'exécuter une commande uniquement si la précédent a échoué :

```
cat fichier > /dev/null|| echo "le fichier n'existe pas"  
grep -i "gardez moi" ~/fichier.txt || rm -f fichier.txt
```

# Shell

## Job control

- Fonctionnalité interne au shell
- Permet de gérer des tâches (processus) multiples dans un même shell :
  - suspension temporaire : le processus est arrêté jusqu'à nouvel ordre
  - arrêt définitif : le processus est terminé
  - mise en arrière/avant plan : le processus reçoit l'entrée clavier
- Un processus peut ignorer l'arrêt définitif, mais pas la suspension

# Shell

## Job control



- **Suspendre** (stopper) un processus : `<Ctrl>z`
- **Arrêter** (terminer) un processus : `<Ctrl>c`  
(si le processus est à l'avant plan)
- **Arrêter** (terminer) un processus : `kill %n`  
(si le processus est à l'arrière plan)
- **Voir** la liste des processus du terminal (jobs) : `jobs`  
    '+' est le job « courant » (`%%` ou `%+`)  
    '-' est le job « précédent » (`%-`)
- **Mettre** un job en **arrière plan** : `bg %n` OU `%n&`
- **Mettre** un job en **avant plan** : `fg %n` OU `%n`
- **Lancer** un job en **arrière plan** : `commande &`

# Shell

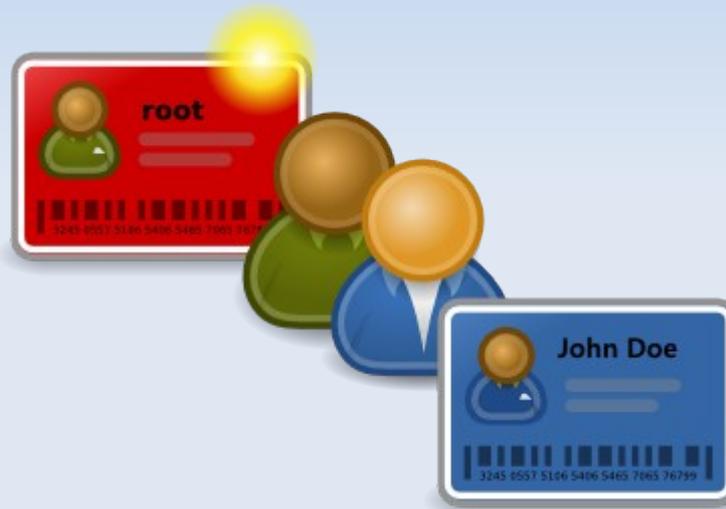
## Job control



```
$ sleep 100
<ctrl-z>
[1]+  Stopped                  sleep 100
$ sleep 200 &
[2] 19858
$ jobs
[1]+  Stopped                  sleep 100
[2]-  Running                  sleep 200 &
$ bg %1
[1]+ sleep 100
$ jobs
[1]-  Running                  sleep 100
[2]+  Running                  sleep 200 &
$ kill %1
$ <return>
[1]-  Complété                  sleep 100
$ fg
sleep 200 <ctrl-c>

$ jobs
$
```

# Utilisateurs



« *The cause of the problem is:  
Bad user karma.* »

BOFH Excuse Server  
<http://www.cs.wisc.edu/~ballard/bofh/bofhserver.pl>

# Utilisateurs

Utilisateurs : /etc/passwd

- Définis dans le fichier /etc/passwd
  - root (UID 0) :
    - propriétaire de presque tous les fichiers système
    - possède tous les droits
  - les autres :
    - utilisateurs « système » (UID < 1000) : *daemon, postfix, sshd, ...*
    - vrais utilisateurs (UID  $\geq$  1000) : *toto, marcel*
- Mots de passes définis dans le fichier /etc/shadow

# Utilisateurs

Utilisateurs : /etc/passwd

Pour chaque utilisateur, le fichier /etc/passwd contient sept champs :

- login
- mot de passe ('x' pour les shadow passwords)
- UID (User ID)
- GID (Group ID, groupe principal)
- champ GECOS (nom complet, adresse, téléphones)
- répertoire personnel (« home dir »)
- le shell exécuté au login

```
root:x:0:0:Linux Torvalds,0,123,456:/root:/bin/bash
```

# Utilisateurs

## Utilisateurs : /etc/shadow

- Seul root peut lire/modifier ce fichier
- Pour chaque utilisateur, le fichier /etc/shadow contient le mot de passe de connexion et ses paramètres de validité :
  - login
  - mot de passe chiffré
  - 6 champs décrivant la validité du compte
  - 1 champ réservé
- Il n'est pas possible de se logger directement si le mot de passe est '\*' ou '!'

```
mb lanc : $1$QJ//btH...jL:13428:0:99999:7:::
```

# Utilisateurs



## /etc/passwd & /etc/shadow

```
$ cat /etc/passwd
```

```
root:x:0:0:Linus Torvads:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
gdm:x:106:111:Gnome Display Manager:/var/lib/gdm:/bin/false
acox:x:1000:1000:Alan Cox,Kernel St,0625081221,0474701221:/home/acox:/bin/bash
```

```
$
```

```
$ cat /etc/shadow
```

```
root:*:13428:0:99999:7:::
daemon:*:13428:0:99999:7:::
bin:*:13428:0:99999:7:::
sys:*:13428:0:99999:7:::
sync:*:13428:0:99999:7:::
games:*:13428:0:99999:7:::
man:*:13428:0:99999:7:::
lp:*:13428:0:99999:7:::
mail:*:13428:0:99999:7:::
gdm!:13428:0:99999:7:::
acox:$1$QN//abU4$nHskZjoAb3nx23J2z.WVJeqz.:13428:0:99999:7:::
```

# Utilisateurs

Groupes : `/etc/group`

- Chaque ligne contient 4 champs :
  - le nom du groupe
  - le mot de passe du groupe
  - l'ID du groupe (GID)
  - la liste des membres du groupe, séparés par des ','
- Chaque utilisateur possède en général un groupe à son nom (Unique Private Group), c'est son groupe primaire
- Chaque utilisateur peut aussi appartenir à  $n$  groupes secondaires
- Les groupes systèmes permettent souvent de permettre aux utilisateurs de manipuler des devices (dialout, fax, audio, ...)

# Utilisateurs



## Groupes : /etc/group

```
$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:acox,ttso,ltorvalds
dialout:x:20:cupsys,acox,ttso,ltorvalds
fax:x:21:hugo,corentin
cdrom:x:24:haldaemon,acox,ttso,ltorvalds
floppy:x:25:haldaemon,acox,ttso,ltorvalds
tape:x:26:acox,ttso,ltorvalds
sudo:x:27:
audio:x:29:ttso,ltorvalds
www-data:x:33:
backup:x:34:
shadow:x:42:
utmp:x:43:
video:x:44:acox,ttso,ltorvalds
sasl:x:45:
plugdev:x:46:haldaemon,acox,ttso,ltorvalds
acox:x:1000:
ttso:x:1001:
ltorvalds:x:1002:
$
```

# Utilisateurs

## Création & gestion



- Outils « posix » :  
`{user,group} {add,mod,del}`
- Outils « distro-dependant » :  
`{add,del} {user,group}`
- Outils divers :  
`chsh, chfn`
- Edition directe des fichiers
  - déconseillée pour les créations/suppressions
  - acceptable pour les modifications

# Utilisateurs

## Voir



- Qui suis-je ?  
`whoami, id, groups, who needs sleep`
- Qui est là ?  
`who, users`
- Qui était là ?  
`last, lastlog`
- Qui fait quoi ?  
`w`
- Qui existe ?  
`lastlog, cat /etc/passwd`

# Utilisateurs

## Authentification : PAM

- L'authentification est gérée par PAM (Pluggable Authentication Modules)
- PAM permet de changer la façon dont on va identifier un utilisateur (globalement, ou pour un service donné)
- PAM gère 4 aspects de l'authentification :
  - **account** : validité du compte, expiration du mot de passe, ...
  - **authentication** : vérification de l'identité de l'utilisateur
  - **password** : modification du mot de passe
  - **session** : liste des tâches à effectuer avant la mise à disposition du service (ou après sa terminaison)
- PAM se configure dans
  - `/etc/pam.conf` (configuration globale)
  - `/etc/pam.d/*` (configurations spécifiques, par modules, ...)

# Utilisateurs

## Authentification : PAM

Grâce à PAM, on peut authentifier un utilisateur de multiples façons :

- par son identifiant/mot de passe situés dans les fichiers `passwd` et `group` (méthode standard)
- par son identifiant/mot de passe stockés dans une base de données (MySQL par exemple)
- par un dispositif biométrique (empreintes digitales...)
- par un dispositif RFID/Bluetooth
- par une clef USB

...en résumé : l'authentification peut se faire :

- en fonction de n'importe quel(s) élément(s)
- service par service

(par exemple Bluetooth pour login graphique, login/password pour la console)

# Utilisateurs

## Changer d'identité : su et sudo

- **su** («switch user») :
  - permet d'exécuter une commande ou d'ouvrir un shell «en tant que»
  - nécessite de taper le mot de passe de l'utilisateur désiré (sauf si l'on est root)
- **sudo** :
  - permet d'exécuter une commande «en tant que»
  - nécessite de taper *SON* mot de passe
  - les possibilités sont définies dans `/etc/sudoers`

# Droits



# Droits

## Généralités

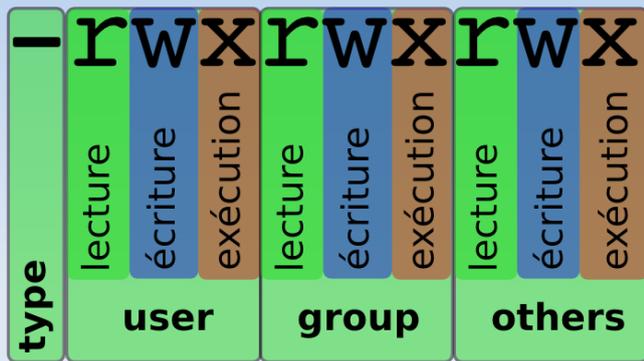
- Chaque fichier ou répertoire possède :
  - un propriétaire
  - un groupe propriétaire
- Chaque fichier ou répertoire possède 3 listes de droits :
  - les droits pour le propriétaire
  - les droits pour le groupe
  - les droits pour les autres
- Chaque liste donne les opérations possibles pour cet utilisateur :
  - **lecture (r) :**
    - fichier* : donne la possibilité de lire le **contenu** du fichier
    - répertoire* : donne la possibilité de lire le contenu d'un répertoire (donc la liste des fichiers qu'il contient)
  - **écriture (w) :**
    - fichier* : permet d'écrire dans ce fichier
    - répertoire* : permet d'y créer/renommer/supprimer des fichiers
  - **exécution (x) :**
    - fichier* : permet d'exécuter ce fichier
    - répertoire* : permet d'y 'rentrer' (cd) et de voir son contenu

# Droits



## Fonctionnement

```
user@host:~$ ls -la /fichier
```



...

Type	Droit	Destinataire
b	r	u
c	w	g
d	x	o
l	x	
s	s	
p		
-		

b Block device  
c Character device file  
d Répertoire (Directory)  
l Lien symbolique  
s Socket  
p FIFO  
- Fichier normal

r Read  
w Write  
x Execute (fichier)  
x chdir (répertoire)  
s SUID bit

u User (propriétaire)  
g Groupe  
o Others (les autres, ni 'u' ni 'g')

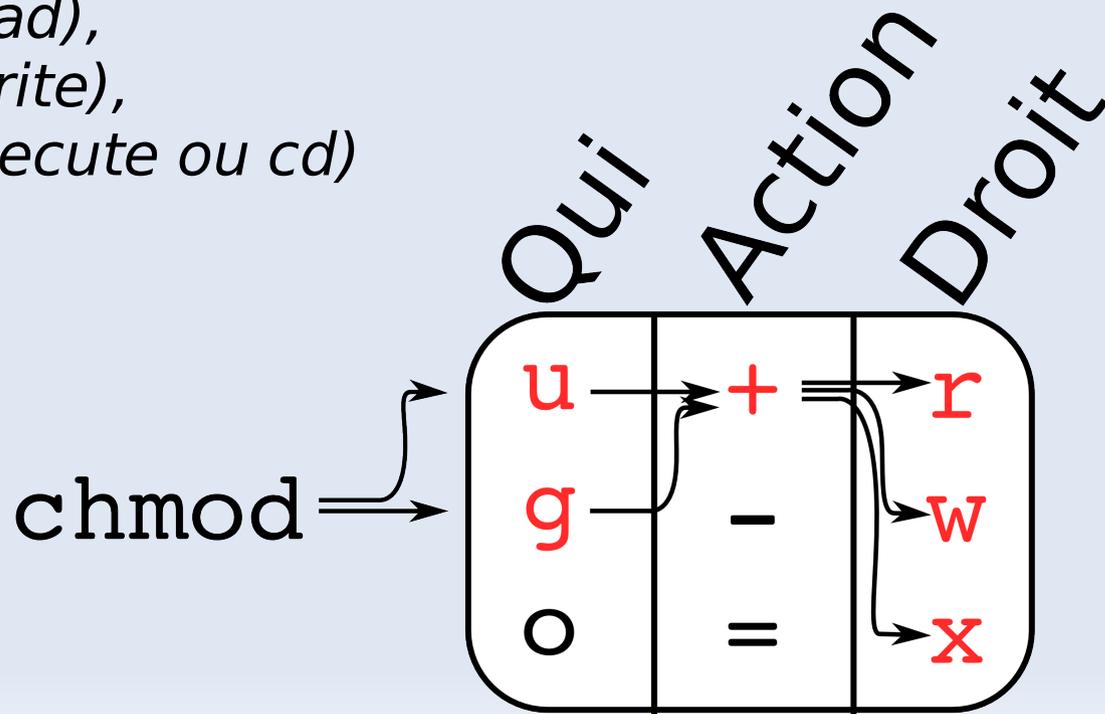
# Droits

## chmod

**chmod** : permet de gérer les droits et peut fonctionner selon un mode littéral :

`chmod destinataire(s) opération droits, ...`

- *destinataire* : **u** (**u**ser), **g** (**g**roup), **o** (**o**ther) ou **a** (**a**ll)
- *opération* : **+** (ajouter), **-** (supprimer), **=** (mettre à la valeur)
- *droits* : **r** (**r**ead),  
**w** (**w**rite),  
**x** (**e**xecute ou **c**d)



# Droits

## chmod

- chmod : permet de gérer les droits et peut fonctionner selon un mode littéral :

`chmod destinataire(s) opération droits, ...`

- *destinataire* : u (**u**ser), g (**g**roup), o (**o**ther) ou a(**a**ll)
- *opération* : + (ajouter), - (supprimer), = (mettre à la valeur)
- *droits* : r (**r**ead), w (**w**rite), x (**e**xecute ou cd)

Exemples :

```
chmod ugo+rwx fichier1 fichier2
```

```
chmod u+rw, g+r, o-rwx fichier3
```

```
chmod ug=rwx,o=rx fichier4
```

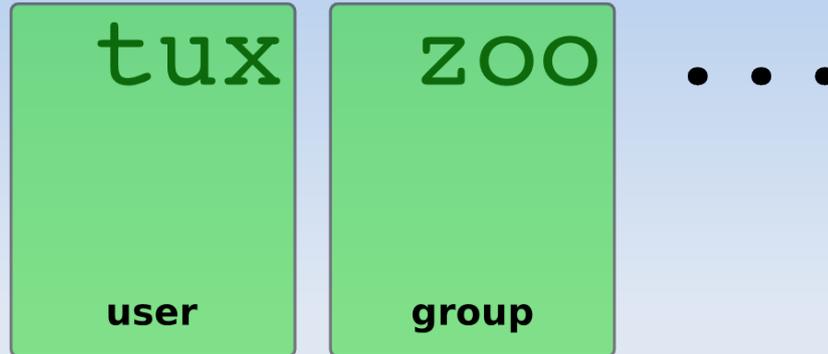
```
chmod a+rx,u+w repertoire
```

# Droits

chmod

```
user@host:~$ ls -la /fichier
```

	<b>-</b>	<b>r</b>	<b>w</b>	<b>x</b>	<b>r</b>	<b>w</b>	<b>x</b>	<b>r</b>	<b>w</b>	<b>x</b>
<b>type</b>		lecture	écriture	exécution	lecture	écriture	exécution	lecture	écriture	exécution
		<b>user</b>			<b>group</b>			<b>others</b>		
<b>bit</b>		<b>210</b>			<b>210</b>			<b>210</b>		
<b>valeur</b>		<b>421</b>			<b>421</b>			<b>421</b>		



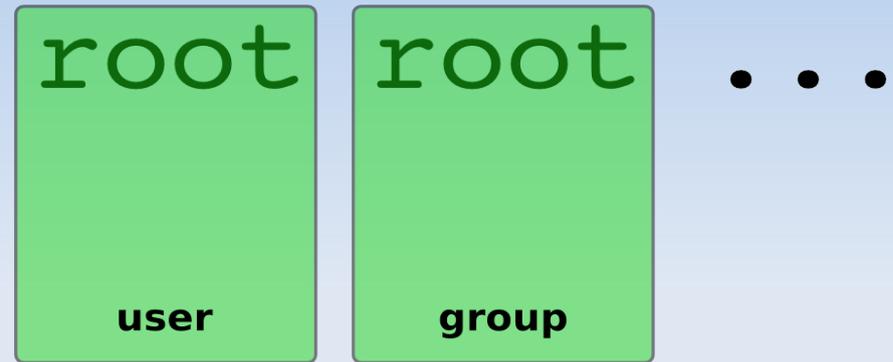
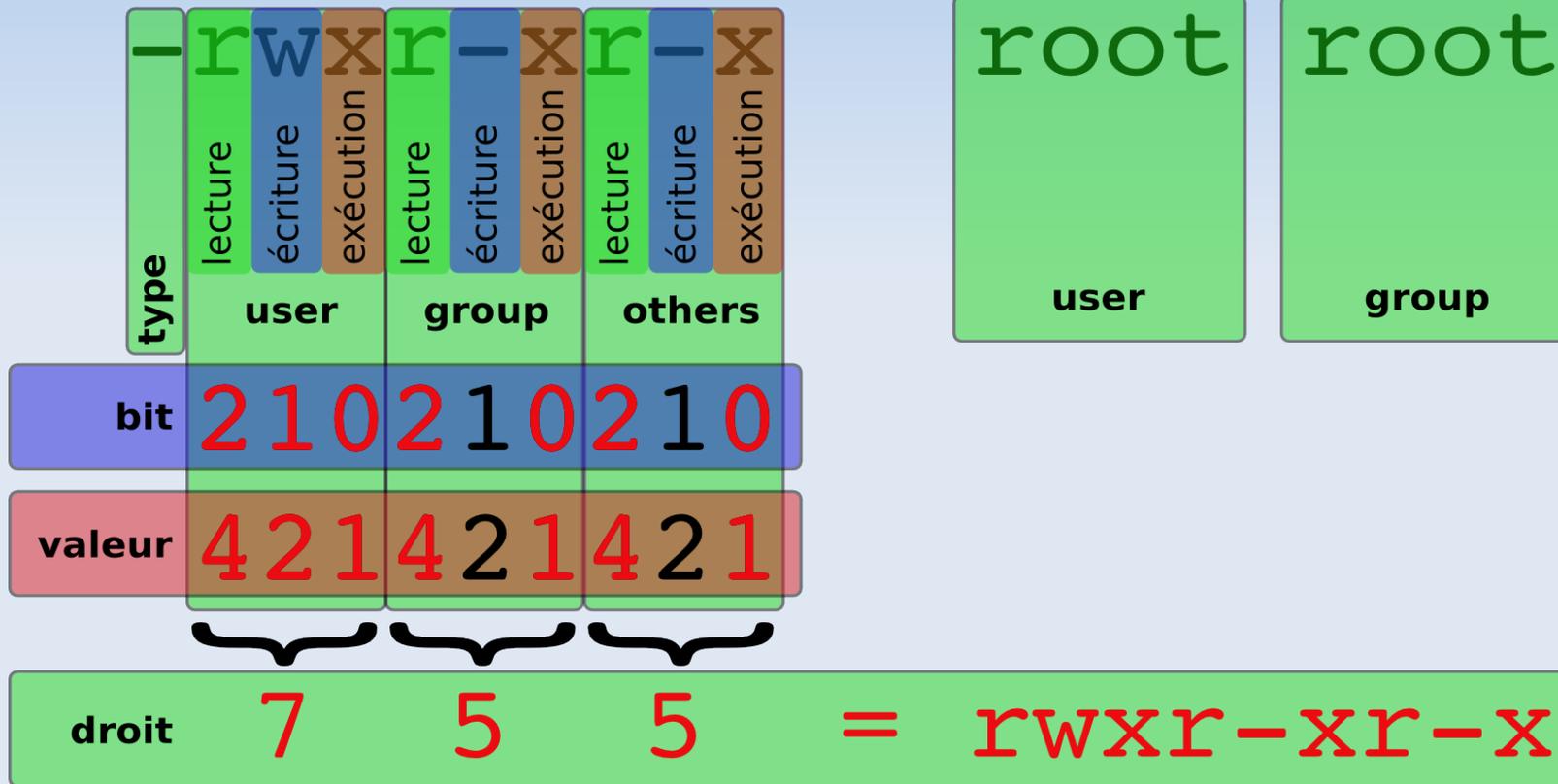
- chmod : peut aussi fonctionner en mode octal :
  - trois groupes d'utilisateurs (u, g, o)
  - trois bits par groupe correspondant à r, w et x
  - le mode s'exprime alors en octal

# Droits



## chmod

```
user@host:~$ ls -la /bin/ls
```



```
chmod 755 <=> chmod u=rwx,g=rx,o=rx
```

# Droits

## chown

**chown** : permet de changer le propriétaire ou le groupe propriétaire d'un fichier

`chown propriétaire:groupe fichier ...`

- *propriétaire* : nouveau propriétaire du fichier ou répertoire
  - *groupe* : nouveau groupe propriétaire du fichier ou répertoire
  - *fichier ...* : fichiers ou répertoires dont il faut changer la propriété
- 
- Si le groupe est omis, chown ne change que le propriétaire  
`chown root /etc/passwd`
  - Si propriétaire est omis, chown ne change que le groupe  
`chown :cdrom /dev/cdrom`

# Droits

## umask



- Les droits d'un fichier ou répertoire à sa création sont défini par umask
- A sa création, un fichier aura les permissions :  
**666 - (umask)**
- A sa création, un répertoire aura les permissions :  
**777 - (umask)**

```
$ umask 0022
$ umask
0022
$ umask -S
u=rwx,g=rx,o=rx
$ touch fichier; mkdir repertoire; ls -ld fichier repertoire
-rw-r--r-- 1 tux zoo ... fichier
drwxr-xr-x 2 tux zoo ... repertoire
$
```

# Droits

## suid/sgid bits



- suid/sgid bits (s) : celui qui exécute prend temporairement l'identité du propriétaire ou groupe propriétaire du fichier

```
$ ls -l zzz
-rwxr-xr-x 1 ltorvalds ltorvalds 0 2007-01-16 23:02 zzz
$ chmod u+s zzz
$ ls -l zzz
-rwsr-xr-x 1 ltorvalds ltorvalds 0 2007-01-16 23:02 zzz
$ chmod g+s zzz
$ ls -l zzz
-rwsr-sr-x 1 ltorvalds ltorvalds 0 2007-01-16 23:02 zzz
$
```

très utile



mais très dangereux

# Droits

## sticky bit



- sticky bit (t) : seul le propriétaire du fichier ou répertoire peut renommer ou supprimer un fichier dans un répertoire affligé du « sticky bit »

```
$ touch fichier
$ chmod a+rxw fichier
$ chmod +t .
$ sudo chown root . fichier
$ ls -la
total 12
drwxr-xr-t 2 root          ltorvalds 4096 2007-01-17 00:17 .
drwxr-xr-x 3 ltorvalds   ltorvalds 4096 2007-01-16 23:02 ..
-rwxrwxrwx 1 root          ltorvalds   0 2007-01-17 00:17 fichier
$ rm -f fichier
rm: ne peut enlever `fichier': Permission non accordée
$
```

# Partitions & Filesystems



# Filesystem



## Types de filesystems

- minix
  - fs original
  - simple, encore utilisé sur les disquettes
  - limité (64 Mb)
- ext2
  - développé en 1993
  - limité (2 Tb)
- ext3
  - ext2 journalisé

# Filesystem



## Types de filesystems

- ext4
  - supporte jusqu'au Pb ( $10^{15}$ )
  - en cours d'intégration au kernel
- jfs/xfs
  - journalisés
  - respectivement IBM/SGI
- ReiserFS/Reiser4
  - journalisé
  - 10 fois plus performant sur les petits fichiers
  - problème de pérennité ?

# Filesystem



## Types de filesystems

- iso9660/udf
  - respectivement CDRoms/DVDRoms
- fat/vfat/ntfs
  - fat/vfat bien gérés
  - support ntfs plus délicat (ro)
- unionfs
  - agrégat de filesystems différents
  - mélange possible entre ro/rw
- nfs/smbfs/coda
  - filesystems réseau

# Filesystem



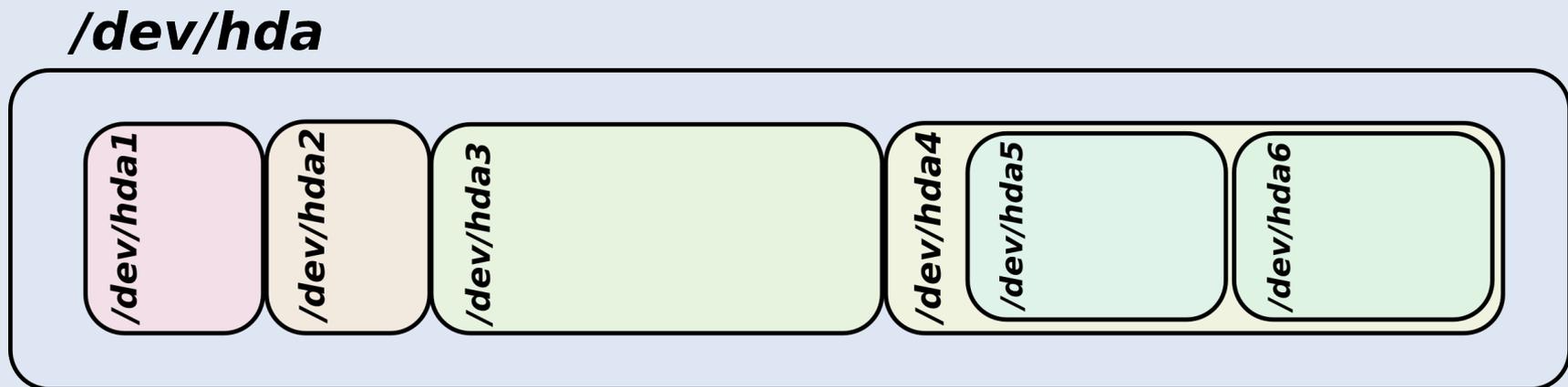
## Filesystems journalisés

- Un filesystem journalisé note toutes les transactions à venir avant de les exécuter
- En cas de crash, le système peut savoir ce qui a été fait et ce qui ne l'a pas été
- La journalisation procure donc quelques avantages :
  - la durée d'un fsck au boot ne dépend plus de la taille du filesystem (on sait où chercher)
  - intégrité des données accrue (et paramétrables)
- ext3/ext4/jfs/xfs/ReiserFS/Reiser4 sont journalisés

# Filesystem

## Partitions

- les disques sont découpés en partitions primaires (max : 4) et étendues
- les partitions sont représentées par des numéros à la fin du device représentant le disque

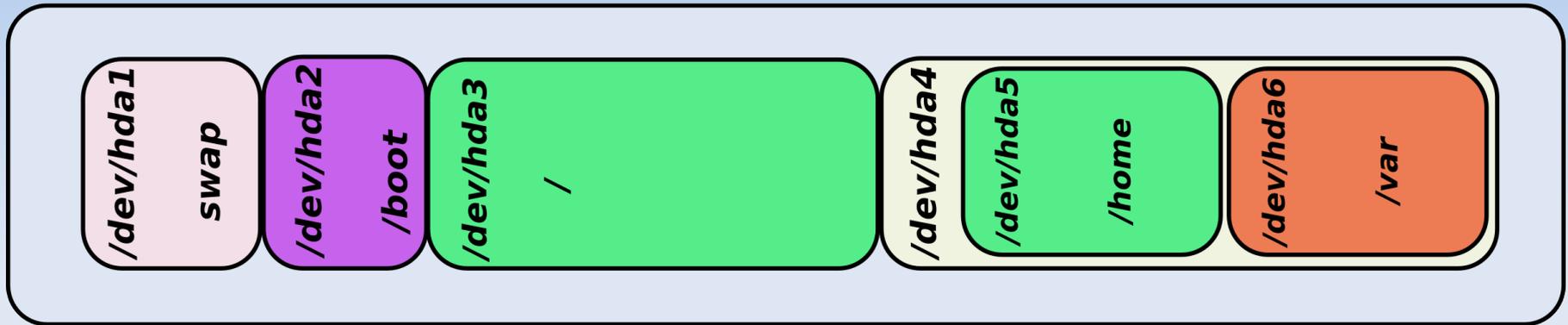


- les devices dépendent de l'interface du disque et de sa position (e.g. `/dev/hda` pour le 1<sup>er</sup> disque IDE, `/dev/hdb` pour le 2<sup>ème</sup>, etc...)

# Filesystem

## Filesystems

***/dev/hda***



linux swap



ext2



ext3



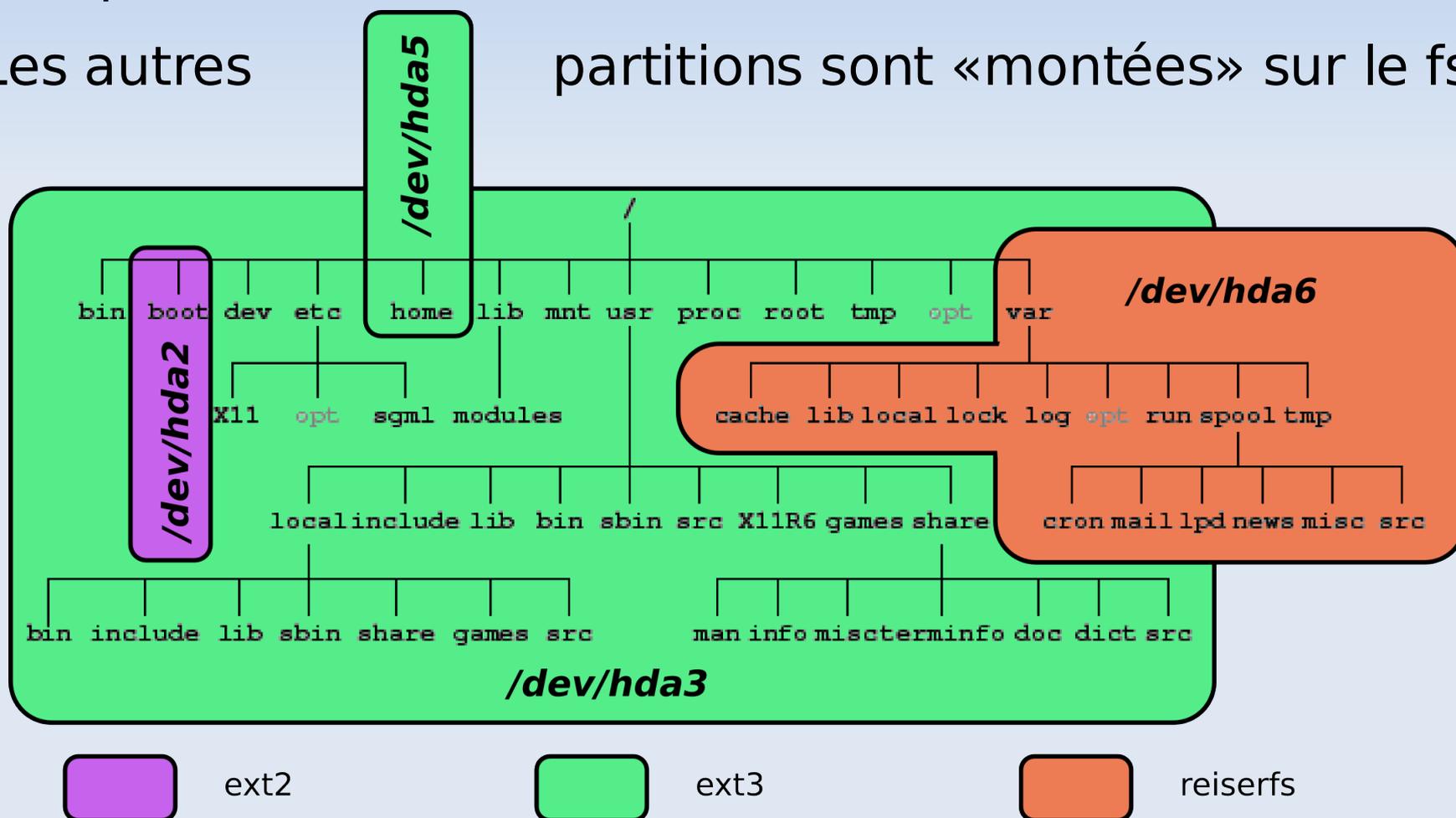
reiserfs

chacune des partitions  
contient un filesystem  
(«fs») qui sera monté dans  
l'arborescence du système

# Filesystem

## Devices et points de montage

- Les disques et les partitions sont des devices
- Une partition contient la «racine» du fs
- Les autres partitions sont «montées» sur le fs



# Filesystem

## Création de partitions



`sfdisk` : ligne de commande (CLI)

`fdisk`, `cdisk` : menus texte (curses)

`gparted` : interface graphique (gnome)

The screenshot shows the GParted application window titled "/dev/sda - GParted". The interface includes a menu bar (GParted, Edition, Vue, Périphérique, Partition, Aide), a toolbar with icons for "Nouveau", "Supprimer", "Redimensionner/Déplacer", "Copier", "Coller", and "Défaire", and a dropdown menu showing the selected device "/dev/sda (74.53 Gio)".

The main display area shows a visual representation of the disk layout with three partitions highlighted: /dev/sda1 (30.86 Gio), /dev/sda6 (12.72 Gio), and /dev/sda7 (29.98 Gio).

Partition	Système de fichiers	Point de montage	Taille	Utilisé	Inutilisé	Drapeaux
/dev/sda1	ntfs	/media/win	30.86 Gio	24.41 Gio	6.44 Gio	boot
▼ /dev/sda2	extended		43.67 Gio	---	---	
/dev/sda5	linux-swap		996.19 Mio	---	---	
/dev/sda6	ext3	/	12.72 Gio	7.18 Gio	5.54 Gio	
/dev/sda7	ext3	/home	29.98 Gio	26.66 Gio	3.31 Gio	

0 opérations en attente

# Filesystem

## Création de filesystems



```
mkfs.<put_you_favorite_filesystem_here> <device>
```

```
user@host:~$ ls /sbin/mkfs.*
/sbin/mkfs.cramfs  /sbin/mkfs.ext3
/sbin/mkfs.minix   /sbin/mkfs.reiser4
/sbin/mkfs.vfat    /sbin/mkfs.ext2
/sbin/mkfs.jfs     /sbin/mkfs.msdos
/sbin/mkfs.reiserfs /sbin/mkfs.xfs
user@host:~$
```

- minix : fs linux historique, encore utilisé pour les disquettes
- cramfs : fs pour embarqué et initrd
- ext{2,3} : filesystem «standard» linux (ext3 journalisé)
- reiser4/jfs/xfs : reiserFS, IBM, SGI (journalisés)
- msdos/vfat : dos (fat) & windows (fat-32)



dd est un cat sous stéroïdes :

- il peut contrôler le nombre d'octets écrits
- il peut convertir ces octets à la volée
- il peut lire/écrire des fichiers, des devices, des pipes, stdin/stdout...
- il peut sauter une partie de l'entrée ou de la sortie

```
dd if=/dev/zero of=mess_zeros count=1 bs=1024
```

bs : taille des blocs en octets (ici 1ko)

count : nombre de blocs

*la taille totale sera donc (bs\*count)*

# Filesystem

## Création de filesystems



```
user@host:~/temp$ dd if=/dev/zero of=monfs.raw bs=1024 count=10240
10240+0 enregistrements lus
10240+0 enregistrements écrits
10485760 octets (10 MB) copiés, 0,08864 seconde, 118 MB/s
user@host:~/temp$ sudo losetup /dev/loop0 monfs.raw
user@host:~/temp$ sudo mkfs.ext2 /dev/loop0
mke2fs 1.39 (29-May-2006)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=1024 (log=0)
Taille de fragment=1024 (log=0)
2560 inodes, 10240 blocs
512 blocs (5.00%) réservé pour le super utilisateur
Premier bloc de données=1
Nombre maximum de blocs du système de fichiers=10485760
2 groupes de blocs
8192 blocs par groupe, 8192 fragments par groupe
1280 inodes par groupe
Superblocs de secours stockés sur les blocs :
    8193

Écriture des tables d'inodes : complété
Écriture des superblocs et de l'information de comptabilité du système de fichiers : complété

Ce système de fichiers sera automatiquement vérifié tous les 23 montages ou
tous les 180 jours, selon la première éventualité. Utiliser tune2fs -c ou -i pour écraser la valeur.
user@host:~/temp$ sudo losetup -d /dev/loop0
```

# Filesystem

## Montage de partitions



- {u,}mount : {dé,}montage des partitions sur le filesystem

```
mount -t <fstype> <device> <mountpoint>
```

```
umount <device|mountpoint>
```

```
user@host:~/temp$ mkdir -p /mnt/mountpoint/

user@host:~/temp$ sudo mount -t ext2 -o loop monfs.raw /mnt/mountpoint/
user@host:~/temp$ df /mnt/mountpoint/
Sys. de fich.          1K-blocs          Occupé Disponible Capacité Monté sur
/home/leucos/temp/monfs.raw
                        9911              92           9307    1% /mnt/mountpoint
user@host:~/temp$ sudo umount /mnt/mountpoint/
```

# Filesystem



## Partitions : /etc/fstab

- Les partitions à monter au boot sont décrites dans /etc/fstab

```
LABEL=ROOTFS          /          ext3          defaults      1 1
UUID=ed2d8d56-5c08-4bd4-977e-673f7a1966b2 /usr      ext3          defaults      0 0
/dev/hda1             /var      reiserfs     defaults      0 0
none                  /dev/pts  devpts       gid=5,mode=620 0 0
none                  /dev/shm  tmpfs        defaults      0 0
none                  /proc     proc         defaults      0 0
none                  /sys      sysfs        defaults      0 0
/dev/cciss/c0d0p2     swap      swap         defaults      0 0
/dev/cdrom             /mnt/cdrom udf,iso9660 noauto,owner,ro 0 0
/dev/fd0              /mnt/floppy auto         noauto,owner  0 0
/dev/ida/c0d0p1       /home     ext3         defaults      1 1
```

- Pour chaque filesystem à monter, /etc/fstab contient :
  - la partition à monter (/dev/hda1)
  - le point de montage (/var)
  - le type de filesystem (reiserfs, ext3, ...)
  - les options de montage (lecteur seule, propriétaire, etc...)
  - un booléen à 1 si le fs doit être sauvegardé par dump
  - un numéro d'ordre pour la vérification de fs au boot

# Filesystem



## Partitions : /etc/fstab

```
LABEL=ROOTFS          /                ext3      defaults    1 1
UUID=ed2d8d56-5c08-4bd4-977e-673f7a1966b2 /usr            ext3      defaults    0 0
/dev/hda1             /var            reiserfs  defaults    0 0
```

- Les partitions à monter sont désignées par :
  - un device (/dev/hda1)
  - un label (ROOTFS), créé par `e2label` ou `tune2fs -L`
  - un UUID, identifiant unique (ed2d8d56-5c08-4bd4-977e-673f7a1966b2), créé par `tune2fs -U`
- Les UUID et labels permettent une indépendance par rapport au device
- `findfs` permet de retrouver un fs par label ou UUID

# Filesystem Gestion



- `df` : donne l'occupation des filesystems montés  
`df -Th | grep "^/dev/"`
- `hdparm` : tuning du disque (DMA, energie, ...)  
`sudo hdparm /dev/sda`
- `tune2fs` : tuning du filesystem (max-mount, UUID)  
`sudo tune2fs -l /dev/sda2`
- `fsck` : vérification du filesystem (boot single-user)  
`fsck -a /dev/hda6`

# Filesystem

## Filesystems spéciaux



- **procfs**
  - monté dans /proc
  - contient des informations sur les processus
  - zone fourre-tout pour les variables exportées du kernel
- **sysfs**
  - monté dans /sys
  - contient des informations sur les devices présents



- L'espace de swap permet de décharger temporairement la mémoire physique (RAM)
- La taille empirique recommandée est  $2 * RAM$
- Lorsqu'un process n'utilise pas une zone de sa mémoire («page»), le kernel peut décider de la mettre dans le swap («swap out»)
- Lorsque la mémoire physique se fait rare, les zones nécessaires aux process :
  - doivent être stockées sur le swap («swap out»)
  - puis relues depuis le swap lorsque le process s'exécute («swap in»)

# Filesystem

## inodes & fichiers

- Dans un fs unix, chaque fichier est représenté par un inode
- Le fs contient une table d'association  
"nom de fichier"  $\Leftrightarrow$  "inode"
- L'inode contient toutes informations nécessaires concernant le fichier :
  - numéro d'inode
  - permissions, propriétaire
  - dates (création, accès, modification)
  - références vers les blocs de données

# Filesystem

## inodes & fichiers



inode

inum	datablocks	permissions	create_time	change_time	etc...
2156	34,35,44,...	rwXrwxrwx uid : 0 gid : 0	12 jan 1006 15h27	12 jan 1006 18h27	

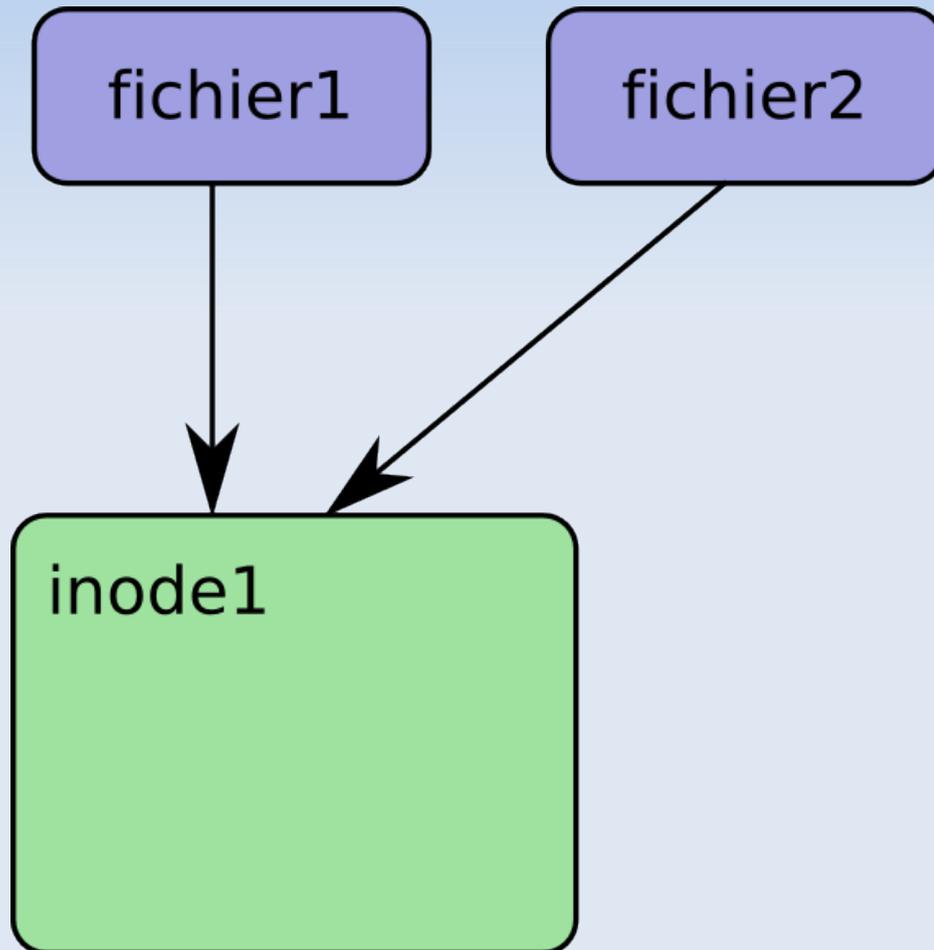
2156	passwd
------	--------

directory table



# Filesystem

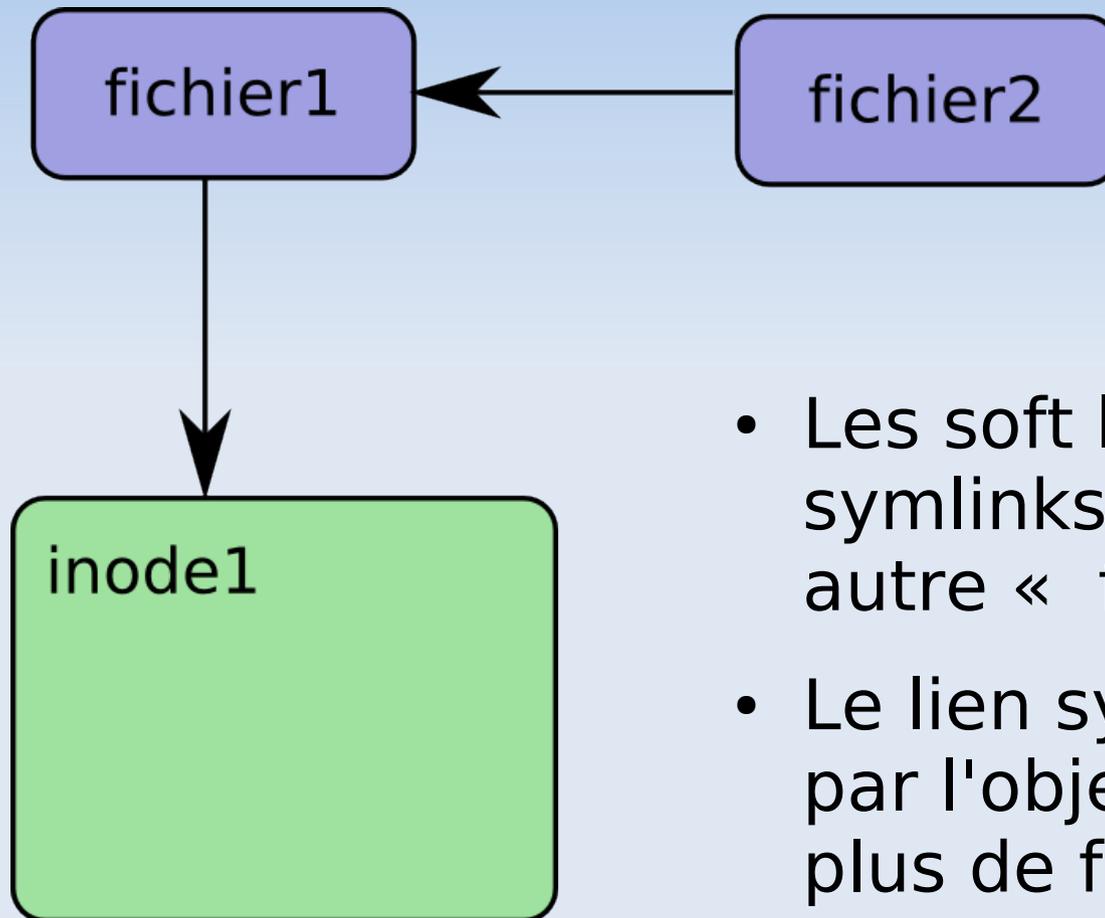
## Liens symboliques : hard links



- Un hard link est simplement une entrée de la DT qui pointe vers un inode
- Il peut y avoir plusieurs hard-links sur un inode (et donc plusieurs « noms » pour un même contenu)
- Les inodes existent tant qu'il y a au moins un hard-link pointant dessus

# Filesystem

## Liens symboliques : soft links



- Les soft links (symbolic links / symlinks) pointent vers un autre « fichier » de la DT
- Le lien symbolique n'existe que par l'objet qu'il pointe : sans lui, plus de fichier

# Filesystem

## Hard links et soft links : choisir

- Utilisation des liens :
  - hard links :
    - ne peuvent fonctionner que sur un même filesystem (la tables des inodes est spécifique au filesystem)
    - performances maximum (rien ne distingue un hard-link d'un autre)
  - soft links
    - peuvent fonctionner entre les filesystems
    - impact sur les performances (indirection)
    - un symlink n'est rien sans son hard-link !

# Filesystem



## Liens symboliques : ln et stat

- ln permet de créer des liens

`ln original destination`

crée un hard link (*destination*) pointant sur l'inode de *original*

`ln -s original destination`

crée un lien symbolique *destination* pointant sur *original*

- stat permet de connaître toutes les infos d'un inode

`stat fichier`

affiche le contenu de l'inode pointé par *fichier*

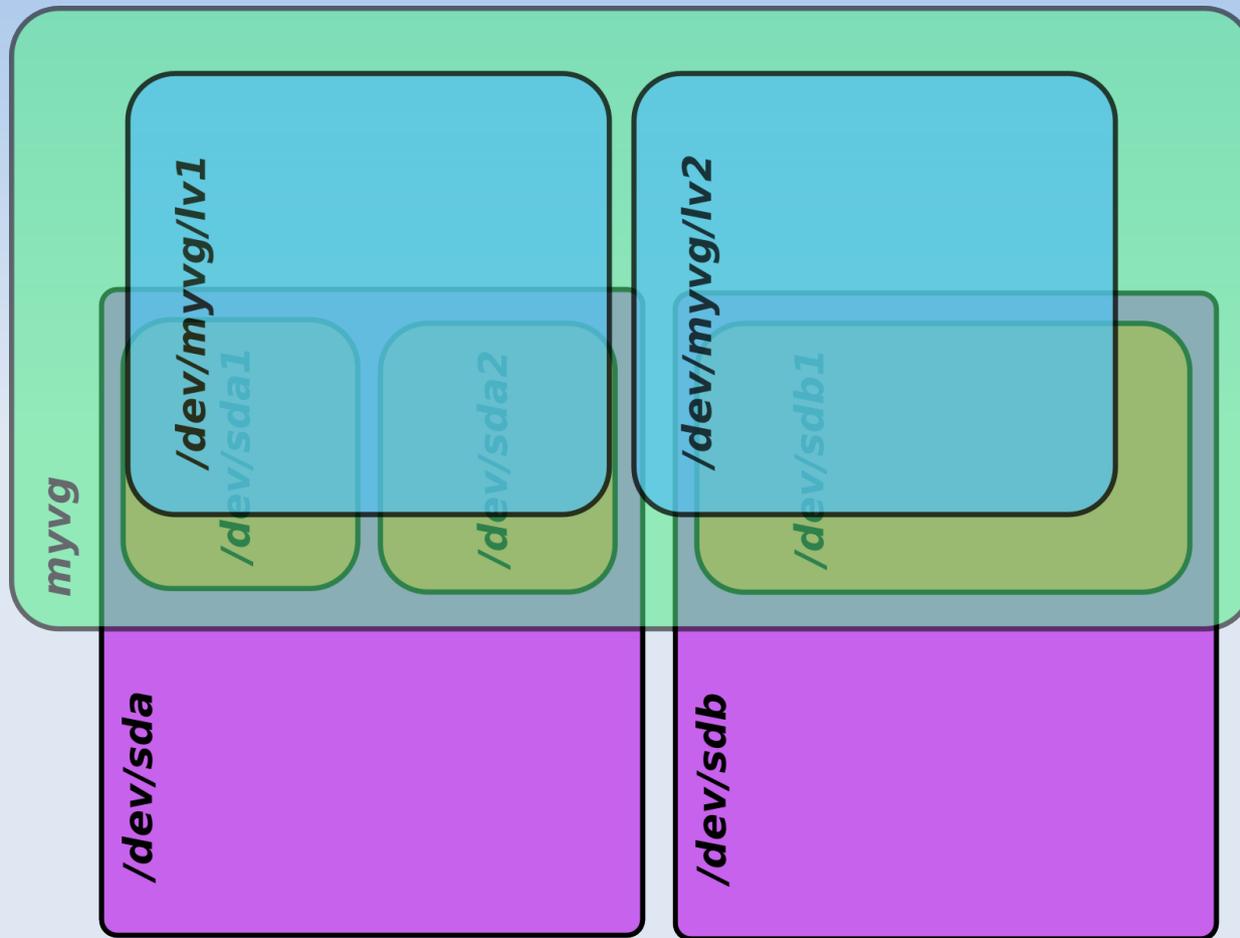
# Filesystem

## LVM : Logical Volume Management

- Fournit un abstraction du matériel de stockage :
  - on travaille sur des volumes logiques et non plus sur des partitions
  - on travaille sur des volume groups au lieu de disques
  - on peut ajouter des partitions dans des volumes groups
  - on peut agrandir les volumes logiques si nécessaire
- LVM permet au final de construire des systèmes de fichiers sur des devices ayant des tailles modulables
- Le système de fichier doit supporter le redimensionnement pour en profiter !

# Filesystem

## LVM



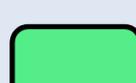
disque



logical volume



partition



volume group

# Filesystem

## RAID

- Niveaux de RAID (**linux-2.6.9+**)
  - **jbod (linear)** : les disques sont concaténées
  - **raid0 (stripping)** : un morceau de donnée sur chaque disque
  - **raid1 (mirroring)** : un morceau de donnée sur tous les disques
  - raid3 : la parité est stockée sur un disque
  - **raid5** : la parité est distribuée sur les disques de l'ensemble RAID
  - **raid6** : la parité est distribuée en double sur les disques de l'ensemble RAID
  - raid 0+1 : un ensemble raid0 en miroir
  - **raid10 (1+0)** : un ensemble raid1 strippé
- Gestion
  - /sbin/mdadm
  - /etc/mdadm/mdadm.conf

# Filesystem

## Filesystems chiffrés

- CryptFS, EncFS, Loop-AES, losetup -e, TrueCrypt, dm-crypt... *à l'aide !*
- Les choses se stabilisent, et deux méthodes émergent :
  - **TrueCrypt**
    - multi-plateforme (Linux/WindowsXP/200x)
    - volumes chiffrés indétectables et indécélabes
  - **Linux Unified Key Setup (luks)**
    - CryptoAPI kernel
    - multiclefs
    - volumes détectables

# Editeurs



*« An infinite number of monkeys typing into GNU emacs would never make a good program. »*

Linus Torvalds

# Editeurs

cat, echo >



- toujours disponible
- performances imbattables

```
user@host:~$ echo "chat {"noir," ,blanc} > texte.txt
user@host:~$ cat texte.txt
chat noir, chat blanc
user@host:~$ cat > texte.txt
chat blanc, chat noir
user@host:~$ cat texte.txt
chat blanc, chat noir
user@host:~$
```



- pas vraiment un éditeur...

# Editeurs

vi



- (presque) toujours disponible
  - très petit
  - incontournable
- 
- cryptique  
(pour les emacsiens)
  - apprendre avant d'en avoir besoin
  - incontournable



# Editeurs

## emacs



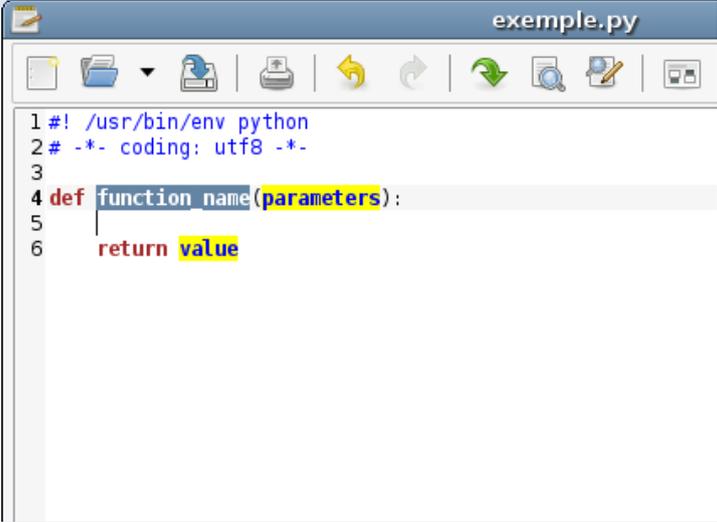
- sait tout faire...
- ...même le reste (extensible)
- cryptique (pour les vi-istes)
- très gros
- (presque) jamais disponible
- apprendre avant d'en avoir besoin



# Editeurs

## ...et les autres

- CLI : aee, ed, e3, joe, jed, nano, pico,....
- XWindow : xed, xwpe, axe,...
- Gnome : gedit, gnotepad+, scite, scribes...
- KDE : kate, kedit, ...



```
1 #! /usr/bin/env python
2 # -*- coding: utf8 -*-
3
4 def function name(parameters):
5
6     return value
```



- vi a deux modes :
  - mode commande : ce que l'on tape est considéré comme une commande et n'affiche rien
  - mode insertion : ce que l'on tape s'écrit dans le fichier
- Du mode commande au mode insertion : i, o, ...
- Du mode insertion au mode commande : <Esc>

# Editeurs

## Commandes vi



- `n` +  : avancer de  $n$  lignes
- `i` : passer en mode insertion
- `o` : insérer une ligne en dessous et passer en mode insertion
- `dd` : effacer la ligne courante
- `dw` : effacer le mot courant
- `D` : effacer la fin de la ligne courante
- `/` : rechercher
- `n` : occurrence de recherche suivante
- `:w` : écrit le fichier sur le disque
- `:q` : quitte vi
- `:q!` : quitte vi malgré les avertissements
- `:x` : écrit le fichier sur le disque et quitte vi (équivalent à `:wq`)
- `n` + `G` : aller à la ligne  $n$ . Si  $n$  est omis, va à la fin du fichier

# Shell & commandes avancées



sed cut reg exp awk expr ssh su sort uniq

# Personnaliser le shell

## Aliases



- Les alias permettent de créer une commande personnalisée

```
alias cd..="cd .."
```

```
alias la="ls -la"
```

```
alias cat="echo Impossible de lire "
```

- alias sans arguments donne la liste des alias actuellement définis
- unalias permet de supprimer un alias  

```
unalias cd..
```

# Personnaliser le shell



`~/.bash{rc,_profile}`

- L'environnement au démarrage du shell est paramétrable
  - `~/.bashrc` : exécuté à chaque shell
  - `~/.bash_profile` : exécuté à chaque login shell
- Il est préférable d'utiliser `~/.bashrc`, les login shell se font rares...
- La configuration par défaut d'un nouvel utilisateur est dans `/etc/skel`
- La configuration « system-wide » est définie dans `/etc/bash.bashrc` et `/etc/profile`
- Penser à personnaliser `http_proxy`

# Commandes avancées



## Boucles : for

- `for/in` : permet d'exécuter une série de commandes sur chacun des éléments d'une liste

```
for var in list; do things; done
```

Exemples :

```
for i in 1 2 3
do echo i vaut $i
done
```

```
for i in mnt boot var; do echo Contenu de /$i; ls /$i; done
```

```
for i in b{in,oot}; do echo -n "$i prend "; du -skh /$i; done
```

# Commandes avancées



## Boucles : backticks & seq

- **seq** : permet de générer une liste sur STDOUT  
seq 1 9 → affiche les chiffres de de 1 à 9  
seq 1 2 9 → affiche les nombres impairs entre 1 et 9
- Les `` (backticks) exécutent une commande et se remplacent par la sortie de cette commande  
rm `seq 1 3` : supprime les fichiers nommés 1 2 et 3

# Commandes avancées



## Boucles : for, `` et seq ensemble

### for, `` et seq

```
for i in `seq 3`; do echo i vaut $i; done
```

```
for i in `seq 1 2 9`; do echo i vaut $i; done
```

```
for i in `seq 9 -2 1`; do echo i vaut $i; done
```

```
for i in `ls ~`; do echo Il y a $i dans $HOME; done
```

```
for i in ~/tmp/*; do echo -n Le fichier $i est de type" "  
    file -b $i  
done
```

```
for i in ~/tmp/*; do echo -n Le fichier $i est de type" "  
    file -b "$i"  
done
```

```
for i in ~/tmp/*; do echo -n Le fichier \'$i\' est de type" "  
    file -b "$i"  
done
```

# Commandes avancées



## Traiter le texte : cut

**cut** : permet d'extraire une colonne dans un fichier à champs délimités

```
cut -f<i> -d<délim> <file>
```

- cut lit sur STDIN si *file* est omis
- le séparateur de champ est *délim* (blanc si omis)
- extrait le *i*<sup>ème</sup> champ sur chaque ligne et affiche sur STDOUT

```
cut -f1 -d':' /etc/passwd → affiche tous  
les utilisateurs déclarés dans /etc/passwd
```

# Commandes avancées



## Traiter le texte : `sort`

**sort** : permet de trier les lignes lues dans un fichier (éventuellement à champs délimités)

```
sort -k<k> [-n] [-r] -t<délim> <file>
```

- `sort` lit sur STDIN si *file* est omis
- le séparateur de champ est *délim* (ou du blanc si *délim* est omis)
- compare en fonction de la clef à la position *k*
- avec l'option *-r*, l'ordre est renversé (z->a, 9->0)
- avec l'option *-n*, la comparaison n'est plus alphabétique mais numérique

```
sort -k3 -nt ':' /etc/passwd → affiche  
/etc/passwd dans l'ordre numérique des UID
```

# Commandes avancées



## Traiter le texte : `uniq`

**uniq** : permet de supprimer les lignes consécutives doublon lues dans un fichier

```
uniq [-c] <file>
```

- sort lit sur STDIN si *file* est omis
- n'affiche qu'une ligne lorsque plusieurs lignes consécutives sont lues
- avec l'option `-c`, `uniq` affiche le nombre d'occurrences pour chaque ligne

```
echo -e "aaa\naaa\naaa\nbbb\nbbb" | uniq  
→ n'affiche que 'aaa' et 'bbb'
```

# Commandes avancées

## (Mal)traiter le texte : sed

**sed** : remplacement de texte ("stream editor")

```
sed [-r] 's/motif1/motif2/' [fichier]
```

- lit *fichier* ou STDIN si aucun fichier n'est spécifié
- -r permet d'utiliser des expressions régulières (presque) compatibles avec egrep
- remplace 'motif1' par 'motif2' (les motifs sont des expressions régulières compatibles avec grep)
- \1...\n dans motif2 sont des variables positionnelles représentant un match dans motif1
- écrit le résultat sur STDOUT

```
echo "chat noir" | sed -r 's/noir/blanc/'
```

→ affiche "chat blanc"

# Commandes avancées

## (Mal)traiter le texte : sed

**sed** : remplacement de texte («stream editor»)

- le modificateur 'g' permet d'attraper plusieurs fois le motif sur la même ligne :

```
echo "chat noir, très noir" | sed -r 's/noir/blanc/'  
→ affiche « chat blanc, très noir»
```

```
echo "chat noir, très noir" | sed -r 's/noir/blanc/g'  
→ affiche « chat blanc, très blanc»
```

- le modificateur 'i' permet d'ignorer la casse lors des évaluations :

```
echo "chat NOIR, très noir" | sed -r 's/noir/blanc/g'  
→ affiche « chat NOIR, très blanc»
```

```
echo "chat NOIR, très noir" | sed -r 's/noir/blanc/gi'  
→ affiche « chat blanc, très blanc»
```

# Commandes avancées

## (Mal)traiter le texte : sed

**sed** : remplacement de texte («stream editor»)

- \1...\n dans motif2 sont des variables positionnelles
- ces variables référenceront les expression entre () dans motif1
- elles ne sont initialisées que s'il y a une concordance

```
echo -e "chat noir" | sed -r 's/chat (.*)/\1 chat/'  
→ affiche "noir chat"
```

```
echo -e "abc" | sed -r 's/(.)(.)(.)/\3\2\1/'  
→ affiche "cba"
```

# Commandes avancées

## Traiter le texte : sed



### – Rappel sur les expressions régulières

.	n'importe quel caractère	
*	le caractère précédent 0 ou plusieurs fois	
+	le caractère précédent 1 fois au moins	
?	le caractère précédent 0 ou 1 fois	
{n}	le caractère précédent exactement n fois	
{m,n}	le caractère précédent de m à n fois	
{n,}	le caractère précédent n fois ou plus	
[a-z]	un caractère en minuscule	
[a-zA-Z]	une lettre	
[0-9]	un chiffre	[[[:alnum:]]]: [a-zA-Z0-9]
^/\$	le début/la fin de ligne	[[[:alpha:]]]: [a-zA-Z]
	séparateur pour spécifier de multiples expressions (ou logique)	[[[:ascii:]]]: caractère ascii (...)
		[[[:blank:]]]: espace ou tabulation
		[[[:cntrl:]]]: caractère de contrôle
		[[[:digit:]]]: [0-9]
		[[[:graph:]]]: caractères imprimables et visibles
		[[[:lower:]]]: [a-z]
		[[[:print:]]]: caractère imprimable
		[[[:punct:]]]: ponctuation
		[[[:space:]]]: espace, tabulation, ...
		[[[:upper:]]]: [A-Z]
		[[[:word:]]] : [a-zA-Z0-9_]
		[[[:xdigit:]]] : [a-fA-F0-9]

### – Classes de caractères

# Commandes avancées



## Rechercher des fichiers : `find`

### `find argument options`

liste les fichiers/répertoires situés dans *argument* selon des critères définis dans *options*.

- options

- name *glob* : liste les fichiers correspondant au shell glob *glob*
- type *f/d/l* : ne liste respectivement que les fichiers, répertoires, liens
- exec : exécute une commande récursif

- patrons de globs

.	n'importe quel caractère
*	le caractère précédent 0 ou n fois
+	le caractère précédent 1 ou n fois
?	le caractère précédent 0 ou 1 fois
[a-z]	un caractère en minuscule
[a-zA-Z]	une lettre
[0-9]	un chiffre

- exemples

```
find ~/mes_docs -type f -name "*.doc"
find ~/mes_docs -type f -name "*facture*" -exec rm -f '{}' \;
find / -name "*log*" -type d
```

# Commandes avancées



## xargs, companion de find

- xargs permet d'utiliser la sortie d'une commande et de la passer en argument à une autre

```
cut -f6 -d':' /etc/passwd | xargs du -skh
```

- La commande xargs est souvent utilisée en combinaison avec find :

```
find arguments | xargs commande
```

- C'est l'équivalent de 'find -exec', la performance en plus :
  - find -exec exécute un process pour chaque résultat
  - xargs cumule la sortie pour maximiser le passage d'arguments

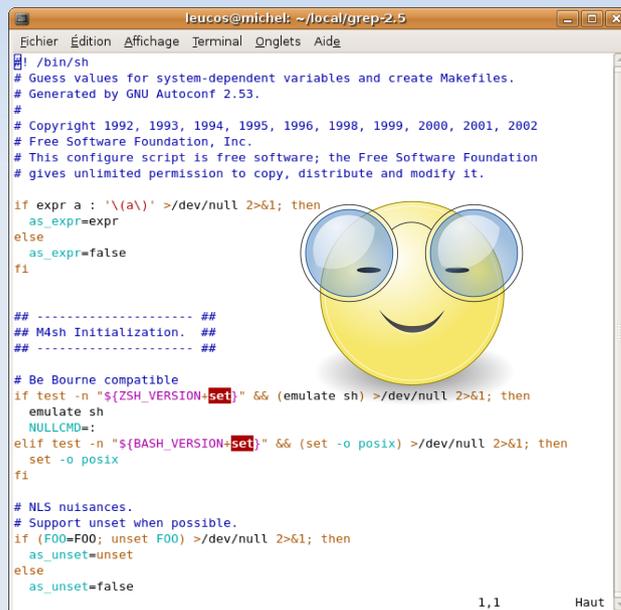
# Commandes avancées



## Rechercher des fichiers : locate

- `find` effectue une recherche sur le filesystem en temps réel : long
- Pour la recherche par nom de fichier, `locate` permet de faire la même chose
- `locate` utilise une base de données pré-générée par `updatedb`
- `locate` permet de rechercher des expressions régulières (`find` ne peut rechercher que des globs)

# Shell scripting



```
leucos@michel: ~/local/grep-2.5
# /bin/sh
# Guess values for system-dependent variables and create Makefiles.
# Generated by GNU Autoconf 2.53.
#
# Copyright 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002
# Free Software Foundation, Inc.
# This configure script is free software; the Free Software Foundation
# gives unlimited permission to copy, distribute and modify it.

if expr a : '\(a\)' >/dev/null 2>&1; then
  as_expr=expr
else
  as_expr=false
fi

## ----- ##
## M4sh Initialization. ##
## ----- ##

# Be Bourne compatible
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
  emulate sh
  NULLCMD=:
elif test -n "${BASH_VERSION+set}" && (set -o posix) >/dev/null 2>&1; then
  set -o posix
fi

# NLS nuisances.
# Support unset when possible.
if (F00=F00; unset F00) >/dev/null 2>&1; then
  as_unset=unset
else
  as_unset=false
fi
```

« *Talk is cheap.  
Show me the code* »

Linus Torvalds

# Shell scripting

## Bases



- Un shell script est un fichier exécutable contenant une série de commandes shell
- Un script commence généralement par :  
`#!/bin/sh`
- Il doit être exécutable, ou appelé par 'sh script'
- Par convention, il se termine en général par « .sh »
- Les lignes commençant par '#' sont des commentaires

# Shell scripting

## Variables



- Dans un script, on peut utiliser des variables locales ou issues de l'environnement

```
echo $HOME
```

- En dehors de l'affectation, la variable doit être précédée par '\$'

```
mavar=bonjour  
echo $mavar
```

- Les arguments passés aussi scripts sont automatiquement affectés aux variables \$1, \$2, ...

# Shell scripting

## Variables



- La variable \$? donne la valeur de sortie numérique de la dernière commande
- Un résultat nul indique (en général !) que la commande a réussi

```
user@host:~$ touch f
user@host:~$ cat f
user@host:~$ echo $?
0
```

```
user@host:~$ rm f
user@host:~$ cat f
cat: f: Aucun fichier ou répertoire de ce type
user@host:~$ echo $?
1
```

# Shell scripting

## Branchements



- La construction if/then/else/fi permet d'effectuer des branchements conditionnels
  - en fonction de valeur de sortie de commandes :

```
echo -n 192.168.0.254 est
if ping -c1 192.168.0.254
then
    echo joignable
else
    echo injoignable
fi
```

# Shell scripting

## Branchements



- La construction if/then/else/fi permet d'effectuer des branchements conditionnels
  - en fonction de valeur de variables :

```
if [ "$USER" = "root" ]
then
    echo Bonjour maitre
else
    echo Encore un utilisateur de base...
fi
```

- La construction [ ] est l'équivalent de la commande 'test'

# Shell scripting

## Comparaisons



- La commande `test` (ou `[]`) permet de comparer des valeurs entre-elles
  - Comparaison numériques
    - `-eq` : égal (equal)
    - `-ne` : différent (not equal)
    - `-gt` : plus grand que (greater than)
    - `-ge` : plus grand que ou égal à (greater or equal)
    - `-lt` : plus petit que (less than)
    - `-le` : plus petit que ou égal à (less of equal)
  - Comparaison de chaînes
    - `=` : égal (chaînes identiques)
    - `!=` : différent
    - `<` : précède (tri alphabétique)
    - `>` : suit (tri alphabétique)

# Shell scripting

## Arithmétique



- Le shell offre trois possibilités pour formuler des calculs :
  - la commande `expr expression` :  
`valeur=`expr 10 \* 20``  
`valeur=`expr $valeur + 2``
  - la construction `$( expression )`  
`valeur=$(10 * 20)`  
`valeur=$( $valeur + 2 )`
  - la construction `$(( expression ))`  
`valeur=$((10*20))`  
`valeur=$(( $valeur+2 ))`

# Shell scripting

## Boucles



- while condition/do/done permet de boucler tant que la condition est vraie

```
num=1
while [ $num -le 5 ]; do
    echo $num
    num=$(( $num + 1 ))
done
```

```
while ping -c1 $host > /dev/null 2>&1; do
    echo $host est joignable
done
echo $host est Injoignable
```

# Gérer et superviser les ressources



« *We all know Linux is great...it does infinite loops in 5 seconds* »  
Linus Torvalds

# Gérer les resssources

## Gérer le temps

- Mettre le système à l'heure
  - Depuis un serveur ntp :  
`ntpdate time.erasme.org`
  - A la main :  
`date --set="20061224 23:59:00"`
- Lire l'heure  
`date +FORMAT`  
ex : `date +%Y%m%d" "%H:%M:%S`
- Chronométrer  
`time commande`

# Gérer les ressources

## Gérer les processus

- Chaque exécutable, script, démon, apparaît comme un processus sur le système lorsqu'il est lancé
- Lorsqu'un processus est lancé, le kernel lui attribue un numéro (PID)
- La commande `jobs` ne permet de voir que les processus exécutés dans le terminal courant
- Pour visualiser les autres processus, il faut utiliser `'ps'`

# Gérer les ressources



## Gérer les processus : ps

- Voir tous les processus

```
ps -edf
```

- Voir tous les processus, un peu plus de détail

```
ps awux
```

- Voir les processus de l'utilisateur courant

```
ps wux
```

### Mnémoniques

PID	: ID du process
PPID	: ID du process parent
VSZ	: consommation totale du process (RAM+SWAP)
RSS,RES	: taille occupé sur la mémoire physique (RAM)
TTY	: terminal associé
STAT	: état (R=runnable, S=sleep, T=stopped, Z=zombie)
START	: heure de lancement
TIME	: cumul temps CPU consommé

# Gérer les ressources



## Gérer les processus : `kill{,all}`

- `kill` permet d'envoyer un signal à un processus
- ce processus peut intercepter ce signal et agir en conséquence
- `kill` n'est pas forcément malfaisant !
- `man kill` donne la liste de signaux possibles
- `killall` permet d'envoyer un signal à des processus par leur nom
- un utilisateur ne peut signaler que ses processus
- attention : `kill` est souvent un built-in (et non `/bin/kill`), donc `help kill` et non `man kill` pour de l'aide !

# Gérer les ressources



## Gérer les processus : `kill{,all}`

```
kill -signal pid
```

```
killall -signal nom
```

- pid : numéro du process
- signal : numéro ou nom du signal  
(-1 envoie le signal à tous les processus)

### Signaux courants

STOP/CONT : équivalents à <Ctrl>-z et fg/bg pour les process du terminal

INT (2) : arrêt demandé par l'utilisateur (généralement via <Ctrl-C>)

TERM (15) : kill demande gentiment au process de se terminer

KILL (9) : le processus est tué sans sommation

HUP (1) : signal de déconnexion du terminal, maintenant surtout utilisé pour demander une reconfiguration

USR1 : généralement utilisé pour demander une reconfiguration

# Gérer les ressources

## Quotas disque

- La suite d'outils 'quota' permet de limiter l'espace disque et le nombre d'inodes :
  - par usager
  - par groupe
- Ces limites sont soit :
  - rigides : dès qu'elles sont atteintes, impossible de créer d'autres fichiers
  - souples : lorsque les limites sont atteintes, l'utilisateur peut encore consommer de l'espace pendant une période de « grâce »

# Gérer les ressources

## Limites « PAM »

- PAM permet d'appliquer un certain nombre de limites à un utilisateur ou à un groupe
  - Ces limites sont définies dans `/etc/security/limits.conf`
  - Elles sont appliquées à la catégorie « session »
    - nombre max de process
    - mémoire maximum
    - priorité des processus
    - ...
- ➔ voir `ulimit -a`

# Superviser le système

## Problèmes classiques

- Filesystem plein
  - 90% des problèmes spontanés
  - résolu facilement en augmentant la capacité du fs
  - ➔ **installez LVM**
- Filesystem corrompu
  - pertes très rares avec les fs journalisés
  - résolu facilement avec une politique de backup efficace
  - ➔ **installez bontmia/dump/arkeia/...**
- Crash disque
  - Un setup RAID (soft ou hard) permet de se prémunir facilement d'un crash disque
  - Un reboot sera peut-être nécessaire en fonction du bus du disque
  - ➔ **utilisez le RAID (hard ou fourni par le kernel)**

# Superviser le système

## Problèmes classiques

- Surcharge système
  - traquer les gourmands
  - optimiser le système
  - ➔ **utilisez ps/top/vmstat/sar...**
- Facteur humain
  - de loin, la plus forte cause d'indisponibilité du système
  - résolu laborieusement par de la pratique et la destruction de nombreux systèmes innocents (de production de préférence)
  - ➔ **expérimentez, cassez, paniquez, recommencez**
  - ➔ **ne donnez pas de shell à n'importe qui**
- Piratage
  - deux catégories d'admins système : ceux qui savent qu'ils ont été piratés, et ceux qui ne le savent pas
  - luttez au quotidien pour être dans la 1<sup>o</sup> catégorie
  - ➔ **utilisez iptables, auditez et faites auditer vos systèmes**

# Gérer les ressources



## Fichiers ouverts : lsof

- Voir tous les fichiers ouverts  
`lsof`
- Voir tous les processus qui ouvrent *fichier*  
`lsof fichier`
- Voir tous les processus qui ouvrent un *fichier*  
sous *répertoire*  
`lsof répertoire`
- Voir tous les fichiers ouverts par le processus  
*PID*  
`lsof -pPID`
- Afficher en boucle toutes les *S* secondes  
`lsof -rS`

# Gérer les ressources



## Fichiers ouverts : lsof & fuser

- Voir tous les processus ayant une connexion ouverte avec *host*  
`lsof -i @host`
- Voir tous les process ayant une connexion tcp sur le port *port*  
`lsof -i TCP:port`
- Voir tous les processus ouvrant des fichiers sur un device ou point de montage  
`fuser -vm <device|mountpoint>`
- Idem, en effectuant le « ménage »  
`fuser -kvm <device|mountpoint>`

# Superviser le système

## Gérer l'espace disque



- Voir l'occupation de tous les filesystems  
`df -Th`
- Voir l'occupation du filesystem contenant le répertoire *rep*  
`df -Th rep`
- Voir l'occupation de tous les filesystems ext3  
`df -ht ext3`
- Voir l'occupation de chaque sous répertoires de *rep*  
`du -h rep`
- Voir l'occupation du répertoire *rep*  
`du -sh rep`

# Superviser le système

## Charge système



- Le noyau gère l'ordonancement de tous les processus du système
- Les processus peuvent être dans plusieurs états, les plus courants étant :
  - R (Runnable) : le processus demande le processeur ou d'E/S
  - S (Sleep) : le processus dort
  - T (sTopped) : le processus a été arrêté (job control)
- La charge système («load average») est le nombre moyen de processus dans la run-queue (état 'R') pendant une période donnée
- La charge système ne reflète pas forcément la charge processeur

# Superviser le système

## Charge système



- La charge système ne tient pas compte du nombre de processeurs (ou de «cores»)
- `uptime` renvoie les load averages des 1, 5 et 15 dernières minutes

```
user@host:~$ uptime
16:19:59 up 101 days, 6 users, load average: 5.10, 3.23, 2.31
```

(`uptime` donne aussi l'heure (!) et le temps de fonctionnement total du système)

- La charge est toujours disponible dans `/proc/loadavg`

```
user@host:~$ cat /proc/loadavg
5.10 3.23 2.31 4/40 20403
```

# Superviser le système

## Charge CPU



- `vmstat` permet de connaître la charge globale des processeurs

```
user@host:~$ vmstat
```

```
procs -----memory----- ---swap-- -----io----- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa
 0  0   17708  63868 119956 389016   0   0   60   51  211  605  7  1  91  1
```

- `mpstat` permet de connaître la charge des processeurs

```
user@host:~$ mpstat -P ALL
```

```
Linux 2.6.17-10-generic (host)
```

```
27.12.2006
```

16:56:12	CPU	%user	%nice	%sys	%iowait	%irq	%soft	%idle	intr/s
16:56:12	all	6,91	0,11	0,88	1,13	0,02	0,58	90,37	422,11
16:56:12	0	4,84	0,10	1,11	1,97	0,05	1,15	90,78	422,11
16:56:12	1	8,98	0,11	0,65	0,28	0,00	0,00	89,96	0,00

- lorsqu'un nombre  $n$  est ajouté en ligne de commande, l'affichage défile en continu toutes les  $n$  secondes

# Superviser le système

## Linux et la mémoire



- Le kernel utilise la mémoire pour :
  - son propre code
  - ses propres données
  - le code des applications exécutés
  - les données des applications exécutées
  - le cache disque
- La mémoire est divisée en «pages»
- Les pages les moins utilisées sont mises en mémoire virtuelle (dans le swap), ce processus est appelé «swapping out» ou «paging out»
- Pour de bonnes performances, il faut, si possible, minimiser les opérations de page out

# Superviser le système

## Mémoire



- `free` permet de connaître l'état de la mémoire

```
user@host:~$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	1009	985	24	0	89	333
-/+ buffers/cache:		562	447			
Swap:	996	17	978			

- Mem décrit la mémoire physique (RAM)
  - Swap décrit la mémoire swap
  - La quantité de mémoire (physique) réellement disponible est sur la ligne '-/+ buffers/cache' (ici 447 Mb)
- `vmstat` donne à peu près les mêmes informations

```
user@host:~$ vmstat
```

procs	-----memory-----	...				
r	b	swpd	free	buff	cache	...
0	0	17708	19284	92748	338840	...

*la mémoire réellement disponible vaut **free+buff+cache***

# Superviser le système

## Mémoire virtuelle



- `vmstat` et `free` permettent aussi de connaître l'état de la mémoire virtuelle

```
user@host:~$ vmstat
```

```
.... ---swap-- -----io----- -system-- ----cpu----
....  si  so   bi   bo   in   cs  us  sy  id  wa
....  20  590   59   47  211  578  9  2  88  1
```

```
user@host:~$ free
```

	total	used	free	shared	buffers	cached
Mem:	1034224	1016120	18104	0	43712	440104
-/+ buffers/cache:		532304	501920			
<b>Swap:</b>	<b>1020088</b>	<b>17720</b>	<b>1002368</b>			

- `swapon -s` permet de connaître la liste des partitions utilisées pour le swap

```
user@host:~$ swapon -s
```

Filename	Type	Size	Used	Priority
/dev/sda5	partition	1020088	17720	-1

# Superviser le système

## Couteaux suisses



top : montre la liste des processus trié selon les critères choisis

```
leucos@michel: ~
Eichier  Édition  Affichage  Terminal  Onglets  Aide
1:CPU - 22:02:41 up 11:49, 13 users, load average: 0.43, 0.29, 0.25
Tasks: 135 total, 3 running, 132 sleeping, 0 stopped, 0 zombie
Cpu0  : 25.0%us, 12.5%sy, 0.0%ni, 62.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  : 62.5%us, 12.5%sy, 0.0%ni, 25.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   1034224k total, 962468k used, 71756k free, 33940k buffers
Swap:  1030320k total, 17716k used, 1012604k free, 379228k cached

 1  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TIME+  COMMAND
 4931 root        16   0 144m 114m 9188 R   60 11.3 69:44.85 Xorg
 6202 leucos      15   0 72072 33m  10m S   24  3.3  1:02.99 gnome-terminal
11933 leucos      15   0 2384 1128 836 R   24  0.1  0:01.87 top
 5942 leucos     16   0 17048  9m 7348 S   12  1.0  0:59.04 metacity
   1 root        16   0 1632  504 420 S    0  0.0  0:02.34 init
   2 root        RT   0   0     0  0 S    0  0.0  0:00.00 migration/0
   3 root        34  19   0     0  0 S    0  0.0  0:00.02 ksoftirqd/0
   4 root        RT   0   0     0  0 S    0  0.0  0:00.00 watchdog/0
   5 root        RT   0   0     0  0 S    0  0.0  0:00.00 migration/1
   6 root        34  19   0     0  0 S    0  0.0  0:00.00 ksoftirqd/1
   7 root        RT   0   0     0  0 S    0  0.0  0:00.00 watchdog/1

 3  PID %MEM  VIRT  SWAP  RES  CODE  DATA  SHR  nFLT  nDRT  S  PR  NI  %CPU  COMMAND
11024 12.8  344m 215m 129m 540 128m 60m 1478  0  S  15  0  0  soffice.bin
 9503 12.0  238m 117m 121m 72 179m 25m 347  0  R  15  0  0  firefox-bin
 4931 11.3  144m 30m 114m 1508 101m 9188 31  0  R  16  0  60  Xorg
 6202 3.3  72072 36m 33m 276 31m 10m 32  0  S  15  0  24  gnome-terminal
 6118 3.2  76452 42m 32m 872 25m 14m 78  0  S  15  0  0  deskbar-applet
 5949 3.1  96620 62m 31m 1144 52m 19m 272  0  S  15  0  0  nautilus
 7442 2.8  65508 36m 27m 1664 13m 17m 201  0  S  15  0  0  tomboy
 5947 2.4  52140 26m 24m 472 6312 17m 187  0  S  15  0  0  gnome-panel
 6268 2.3  64724 39m 23m 1664 37m 9828 3  0  S  15  0  0  beagled-helper
 6046 1.3  68624 53m 13m 36 36m 9672 6  0  S  15  0  0  trashapplet
 6129 1.2  34396 21m 12m 40 2956 9328 1  0  S  15  0  0  mixer_applet2
```

## Tri

M : tri par mémoire

P : tri par CPU

N : tri par PID

T : tri par temps processeur

## Selection

1 : affiche tous les processeurs

i : affiche les tâches runnables

k : tue un process

r : renice

# Superviser le système

## Couteaux suisses



`saïdar` : affiche un résumé très complet des différents sous-systèmes

- `loadavg`
- CPU (global)
- processus
- mémoire
- réseau
- disques

```
leucos@michel: ~/doc/perso/formations/linux
Fichier Édition Affichage Terminal Onglets Aide
Hostname : michel      Uptime : 11:52:31      Date : 2007-01-27 22:05:28
Load 1   : 0.37      CPU Idle : 85.61% Running : 1   Zombie  : 0
Load 5   : 0.29      CPU System: 2.27% Sleeping : 134  Total   : 135
Load 15  : 0.25      CPU User  : 12.12% Stopped  : 0   No. Users : 13
Mem Total : 1009M Swap Total: 1006M Mem Used : 91.94% Paging in : 8
Mem Used  : 928M Swap Used : 17716K Swap Used : 1.72% Paging out: 16
Mem Free  : 83396K Swap Free : 988M Total Used: 46.91%
Disk Name      Read      Write      Network Interface      rx      tx
sda            8192B     16384B     lo                      0B      0B
               eth0                      0B      0B
Total          8192B     16384B     eth1                    15946B  4744B
               sit0                      0B      0B
               tun0                      0B      0B
Mount Point      Free      Used
/                2977M    75.54%
/home            597M     97.92%
/media/win       1698M    94.62%
```

# Superviser le système

## Couteaux suisses



statgrab : permet d'obtenir des statistiques «formatées»

```
user@host:~$ statgrab net.eth0.up net.eth1.up
net.eth0.up = false
net.eth1.up = true
user@host:~$ statgrab load.
load.min1 = 0.220000
load.min15 = 0.110000
load.min5 = 0.140000
user@host:~$ statgrab -u mem.cache mem.free
414134272
63537152
user@host:~$ statgrab -u mem.cache mem.free |
  tr '\n' ' ' | awk '{ print ($1+$2)/(1024*1024)"M" }'
455.188M
```

# Superviser le système



## Analyse « off-line » : sar

- sar permet de collecter des informations sur le système à intervalles réguliers
- ces informations sont stockées sur le système et peuvent être analysées à posteriori

```
sar option -s debut -e fin
```

- |                                     |                                |
|-------------------------------------|--------------------------------|
| -b : entrées sorties                | -q : load average              |
| -B : paging                         | -r : utilisation de la mémoire |
| -c : création de process            | -R : statistiques mémoire      |
| -d : activité sur les devices block | -u : utilisation du CPU        |
| -n : activité réseau                | -v : utilisation des inodes    |
| -P : activité processeur            | -y : activité des tty          |

# Superviser le système

## Analyse « off-line » : sar



```
user@lazyserver:~$ sar -u -s 10:00:00 -e 11:00:00
```

```
Linux 2.6.17-10-generic (lazyserver)          28.12.2006
```

10:05:01	CPU	%user	%nice	%system	%iowait	%steal	%idle
10:15:01	all	0,53	0,05	1,09	0,10	0,00	98,22
10:25:01	all	0,16	0,07	0,98	0,01	0,00	98,77
10:35:01	all	1,23	0,06	1,24	0,05	0,00	97,42
10:45:01	all	0,58	0,06	1,19	0,06	0,00	98,11
10:55:01	all	3,47	0,17	1,94	2,35	0,00	92,06
Moyenne:	all	1,20	0,08	1,29	0,51	0,00	96,92

- sar peut aussi monitorer en temps réel l'activité d'un processus :

```
sar -x PID 1 0
```

```
user@lazyserver:~$ sar -x 6702 1 0
```

```
Linux 2.6.17-10-generic (lazyserver)          28.01.2007
```

11:18:45	PID	minflt/s	majflt/s	%user	%system	nswap/s	CPU
11:18:46	6702	2702,02	0,00	97,98	2,02	0,00	0
11:18:47	6702	12310,89	0,00	91,09	7,92	0,00	0
11:18:48	6702	7212,00	0,00	94,00	5,00	0,00	1
11:18:49	6702	3215,00	0,00	97,00	2,00	0,00	0
11:18:50	6702	10741,41	0,00	93,94	5,05	0,00	1
11:18:51	6702	8178,43	0,00	93,14	4,90	0,00	1

# Superviser le système

## Rebooter ?

- Raisons de rebooter un linux :
  - changement de kernel
  - ajout de matériel
  - déménagement

`uptime`

donne la durée de fonctionnement du système

`shutdown -h time`

arrête le système (equivalent à 'halt')

`shutdown -r time`

reboote le système (equivalent à 'reboot')

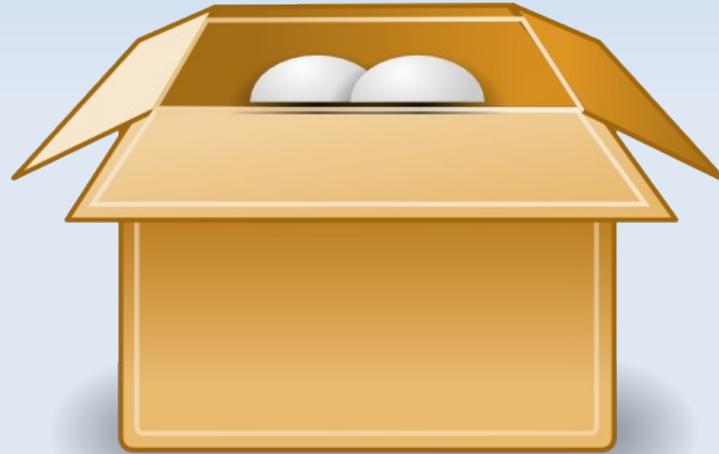
`shutdown -c`

annule un shutdown programmé

- *time* : hh:mm, +m ou now

# Packages

## Installation de logiciels



# Packages

## Installation de logiciels

### **Les packages (paquetages) ont pour objectif :**

- de permettre une installation simple
- ..... désinstallation .....
- ..... mise à jour .....
- de gérer des dépendances entre packages
- d'être recompilable facilement

# Packages

## Installation de logiciels

### **Encore des chapelles !**

- RPM sous Fedora, Redhat, Mandriva, SuSE...
- DEB sous Debian, Ubuntu
- tar.gz binaires sous Slackware
- scripts + tar.gz sources sous Gentoo
- OpenPKG, sorte d'espéranto du packaging

...sans compter les packages source

# Packages

## Redhat Package Manager

- Fonctionne sur les distributions basées RPM  
[http://en.wikipedia.org/wiki/List\\_of\\_Linux\\_distributions#RPM-based\\_free\\_distributions](http://en.wikipedia.org/wiki/List_of_Linux_distributions#RPM-based_free_distributions)
- Deux commandes suffisent pour gérer les paquetages sur les systèmes basés RPM
  - `rpm` : permet de manipuler les fichiers `.rpm` et de gérer les paquetages installés
  - `yum` : permet d'installer des paquetages depuis différentes sources
- Les packages RPM ont une extension de fichier `<arch>.rpm` (paquetages binaires) ou `.src.rpm` (paquetages source)
- La commande `rpm` accepte des URL (ftp/http)

# Packages

## Redhat Package Manager : rpm

- Les packages sont manipulés avec la seule commande rpm
- **Installer** un package  
`rpm -i package.rpm`
- **Supprimer** un package  
`rpm -e package`
- **Mettre à jour** un package installé  
`rpm -u package.rpm`
- Les options `-v` et `-h` rendent l'affichage plus 'humain'.

# Packages

## Redhat Package Manager : rpm

- La commande query (-q) permet d'interroger un package ou la base de données des packages installés

- **Voir** les informations sur un package

```
rpm -qip package.rpm (query information)
```

```
rpm -qi package (pour un package installé)
```

- **Lister** les fichiers d'un package

```
rpm -qlp package.rpm (query list)
```

```
rpm -ql package (pour un package installé)
```

- **Rechercher** dans quel package se trouve *fichier*

```
rpm -qf fichier (query find)
```

# Packages

## Redhat Package Manager : yum

- La commande `yum` permet de rechercher ou d'installer un paquetage directement depuis un dépôt (Internet, CD/DVD, disque)
- La liste des dépôts est définie dans `/etc/yum*`

- **Installer** le paquetage `nom`

```
yum install nom
```

- **Rechercher** les paquetages contenant la chaîne `nom`

```
yum search nom
```

- **Mettre à jour** les informations depuis les dépôts

```
yum update
```

# Packages

## Packages Debian

- Fonctionne sur les distributions basées Debian  
[http://en.wikipedia.org/wiki/List\\_of\\_Linux\\_distributions#Debian-based\\_free\\_distributions](http://en.wikipedia.org/wiki/List_of_Linux_distributions#Debian-based_free_distributions)
- Les systèmes debian possèdent l'équivalent (approximatif) du monde RPM pour gérer les paquetages :
  - `dpkg` : permet de manipuler les fichiers `.deb` et de gérer les paquetages installés
  - `apt-get` : permet d'installer des paquetages depuis différentes sources
- Les packages Debian ont une extension de fichier `.deb` (paquetages binaires)
- La commande `dpkg` n'accepte pas d'URL

# Packages



## Packages Debian : dpkg

Les fichiers packages sont manipulés avec la commande dpkg

- **Installer** ou **mettre à jour** un package

```
dpkg -i package.deb
```

- **Supprimer** un package

```
dpkg -r package
```

- **Supprimer** un package et sa configuration

```
dpkg -P package
```

- **Reconfigurer** un package

```
dpkg-reconfigure package
```

# Packages



## Packages Debian : dpkg

- **Rechercher** un fichier dans les paquetages installés

```
dpkg -S fichier
```

- **Lister** les fichiers *actuellement* installés pour un paquetage.

```
dpkg -L paquetage
```

- **Lister** les fichiers installés par un paquetage.

```
dpkg -l paquetage
```

- Donne **l'état** du *paquetage* installé

```
dpkg -l paquetage
```

# Packages



## Packages Debian : apt-get

- La commande `apt-get` permet de rechercher ou d'installer un paquetage directement depuis un dépôt (Internet, CD/DVD, disque)
- `apt-get` est *très* performant par rapport à `yum`
- La liste des dépôts est définie dans `/etc/apt/{sources.list,sources.list.d/}`
- **Installer** le paquetage `nom`  
`apt-get install nom`

# Packages



## Packages Debian : sources apt

- les outils de la famille apt utilisent des *sources* pour obtenir les packages et les métadonnées associées
- ces sources, définies dans `/etc/apt/{sources.list,sources.list.d/}` ont le format suivant

```
deb http://fr.archive.ubuntu.com/ubuntu/ edgy main restricted
      ↑                               ↑   ↑   ↑
      url du dépôt                   distribution sections...
```

- attention en ajoutant des sources :
  - il est impératif d'utiliser des sources de confiance
  - il faut vérifier la signature des paquets (`apt-key` permet de gérer les identités)

# Packages



## Packages Debian : apt-get

- **Upgrader** tous les paquetages installés  
`apt-get upgrade`
- **Mettre à jour** l'information depuis les dépôts  
`apt-get update`
- **Supprimer** le paquetage  
`apt-get remove package`
- **Passer** à une distribution plus récente  
`apt-get dist-upgrade`
- **Vérifier** l'état de la base de données  
`apt-get check`

# Packages



## Packages Debian : apt-\*

- **apt** est en fait une famille de commandes dédiées à la gestion de paquets.  
Quelques membres de la famille :
  - `apt-file` : rechercher un fichier dans les dépôts apt
  - `apt-cache` : rechercher un package dont la description ou le nom contient une expression régulière

```
apt-file search `which bash`
```

```
apt-cache search tcpdump
```

```
apt-cache -n search '*.dump.*'
```

# Packages



## tar, {g,b}zip

- tar permet de créer un fichier contenant d'autres fichiers
- Il est souvent utilisé en combinaison avec gzip ou bzip2
- tar est parfois utilisé comme système de gestion de paquetage dans certaines distributions (Slackware)
- **Créer** une archive *archive.tar* contenant *fichiers*  
`tar cvf archive.tar fichiers`
- **Créer** une archive tar compressée gzip *archive.tar.gz* contenant *fichiers*  
`tar cvzf archive.tar.gz fichiers`

# Packages

tar, {g,b}zip



- **Créer** une archive tar compressée bzip2  
*archive.tar.bz2* contenant *fichiers*

```
tar cvjf archive.tar.bz2 fichiers
```

- **Lister** le contenu de l'archive

```
tar tvf archive
```

```
tar tvzf archive.tar.gz
```

```
tar tvjf archive.tar.bz2
```

- **Extraire** une archive tar

```
tar xvf archive
```

```
tar xvzf archive.tar.gz
```

```
tar xvjf archive.tar.bz2
```

# Packages

## Installer à partir des sources

- Il est préférable d'installer des logiciels avec le gestionnaire de paquetage de la distribution
  - intégration plus fine dans la distribution
  - gestion des dépendances et des conflits
  - mises à jour automatiques
  - désinstallation aisée
- Tous les logiciels ne sont cependant pas disponible sous forme de paquetages

**Il est donc parfois nécessaire d'installer un logiciel à partir de ses sources**

# Packages

## Installer à partir des sources

- Les logiciels installés en dehors du gestionnaire de package doivent l'être dans `/usr/local/`
- La plupart des logiciels sont construits autour d'outils de développement communs :
  - *automake/autoconf* : définition des paramètres de compilation en fonction de la plateforme
  - *make* : automatisation de la construction du binaire à partir des sources et installation
- Des cas particuliers existent souvent; il n'y a pas de recette miracle...

# Packages



## Installer à partir des sources, v1

Exemple : installer le logiciel 'soft-1.2.tar.gz'

*Méthode 1 :*

- décompacter les sources 

```
$ tar xvzf soft-1.2.tar.gz
```

```
$ cd soft-1.2
```
- configurer la compilation 

```
$ ./configure
```

```
....
```
- compiler 

```
$ make
```

```
...
```
- installer 

```
$ make install
```

```
...
```
- A ce stade, si tout s'est bien passé, 'soft' doit être installé dans divers répertoires sous /usr/local/
- Cette méthode ne permet pas de désinstallation simple

# Packages



## Installer à partir des sources, v2

Exemple : installer le logiciel 'soft-1.2.tar.gz'

*Méthode 2 :*

- décompacter les sources 

```
$ tar xvzf soft-1.2.tar.gz
```

```
$ cd soft-1.2
```
- configurer la compilation 

```
$ ./configure
```

```
....
```
- compiler 

```
$ make
```

```
...
```
- installer 

```
$ sudo checkinstall
```

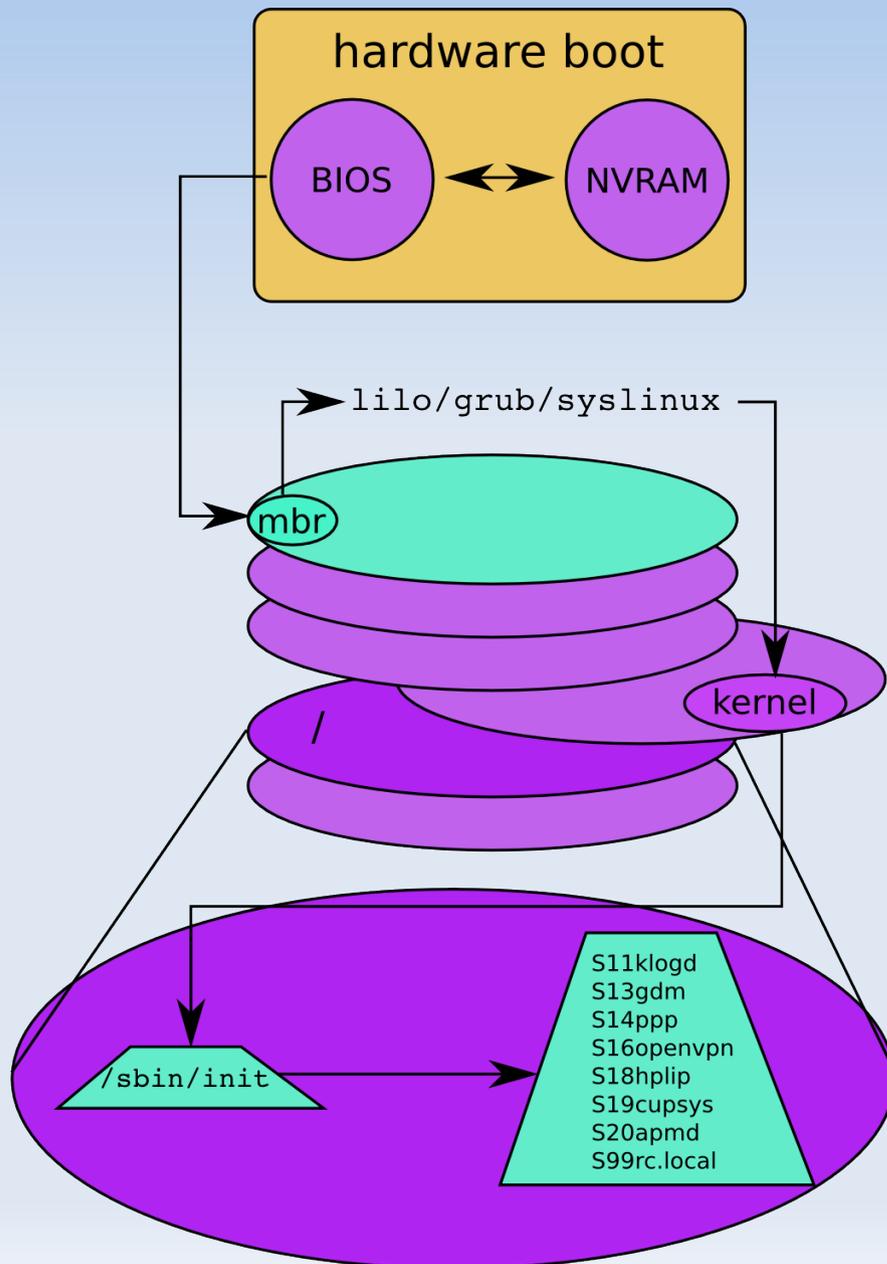
```
...
```
- Après quelques questions, checkinstall va faire un paquet binaire pour votre distribution et l'installer
- Cette méthode permet de transformer un logiciel source en paquet binaire automatiquement

# Boot process, runlevels, cron



# Boot

## Les 5 étapes



### (1) Boot hardware

exécution du bios  
paramètres dans la NVRAM

### (2) Chargeur d'OS

exécuté depuis le MBR par le BIOS  
limité à 512 bytes

### (3) Boot kernel

montage du root fs (/)  
chargement des drivers  
exécution d'*init*

### (4) init

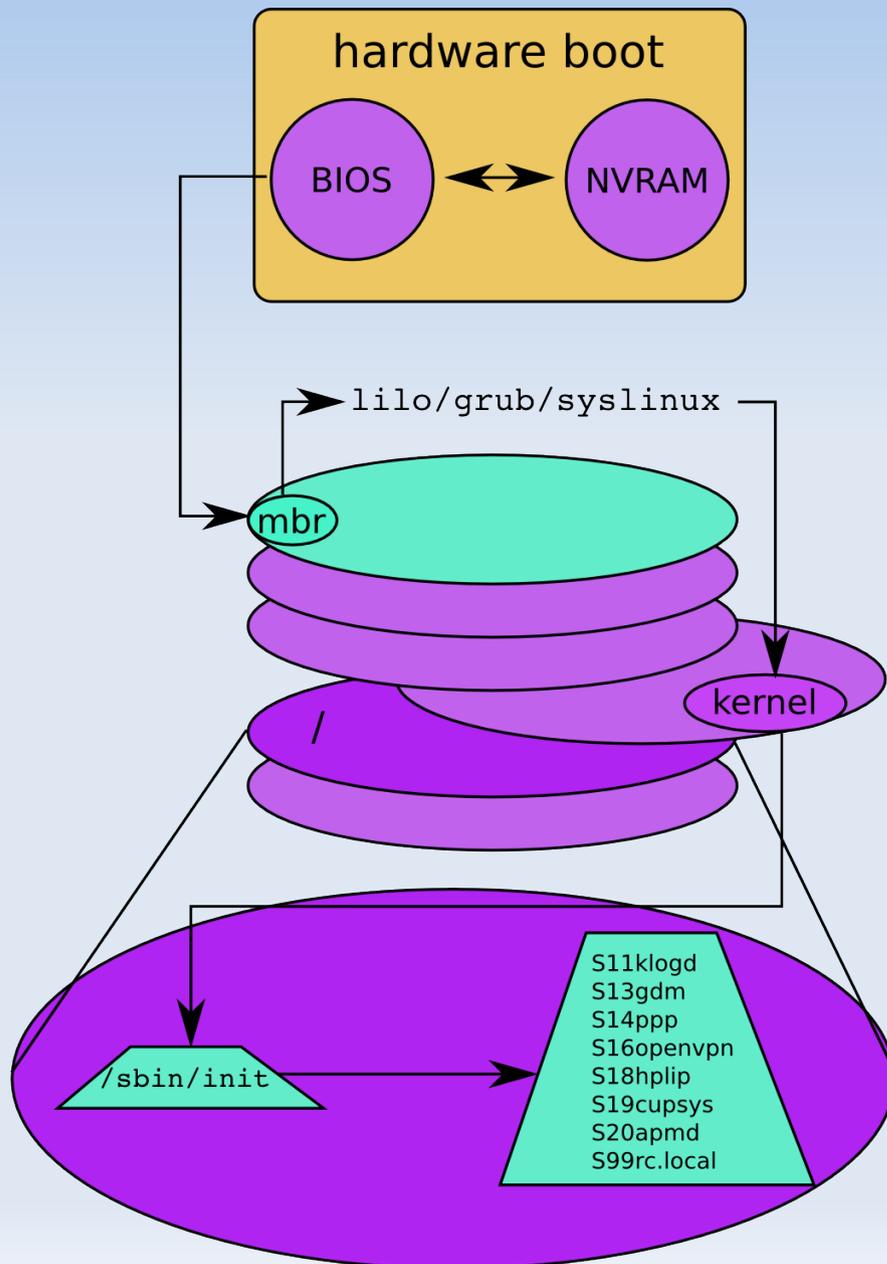
passage au runlevel demandé  
exécution séquencée des scripts  
console

### (5) Scripts SysV

démarrage des services

# Boot

## Les 5 étapes



(1) Boot hardware  
exécution du bios  
paramètres dans la NVRAM

(2) **Chargeur d'OS**  
exécuté depuis le MBR par le BIOS  
limité à 512 bytes

(3) Boot kernel  
montage du root fs (/)  
chargement des drivers  
exécution d'*init*

(4) *init*  
passage au runlevel demandé  
exécution séquencée des scripts  
console

(5) Scripts SysV  
démarrage des services

# Boot

## Bootloaders : lilo, grub & co.

- Le travail du bootloader : charger le kernel en mémoire
- Il doit donc :
  - savoir où le trouver
  - savoir le charger
- Après des années de domination de lilo, Grub à pris le dessus
- D'autres bootloaders existent (notamment syslinux, loadlin, silo, milo, emile, etc...)

# Boot

## Bootloaders : grub

- GRUB fonctionne en 3 temps :
  - 1<sup>er</sup> étage (512b d'assembleur) booté sur le MBR
  - 1.5<sup>ème</sup> étage (~10kb), booté au delà du MBR
  - 2<sup>ème</sup> étage (~100kb), chargé depuis /boot/grub
- GRUB sait donc (grâce à 1.5), lire un fichier sur un filesystem ext3/reiserfs/xfs/..
- Il lit aussi sa configuration directement sur le filesystem : pas besoin de «réinstaller» GRUB à chaque changement
- Le 2<sup>ème</sup> étage trouve le kernel et lui passe la main

# Boot

## grub : menu.lst

- /boot/menu.lst est le fichier de configuration de grub
- il contient des scripts qui sont interprétés par grub lors du boot
- ces scripts peuvent être édités au boot si besoin
  - démarrage en single user
  - changement du root filesystem
- grub-install permet d'écrire les étages 1 et 1.5 sur le disque

```
sudo grub-install /dev/sda
```
- la commande grub se présente comme un shell; par exemple les commandes équivalentes à grub-install sont :

```
[root@server ~]# grub
grub> device (hd0) /dev/sda
grub> root (hd0,0)
grub> setup (hd0)
```

# Boot

## grub : menu.lst

```
default 0
```

```
title          Ubuntu, kernel 2.6.17-10-generic
root           (hd0,0)
kernel         /boot/vmlinuz-2.6.17-10-generic root=/dev/sda6
initrd         /boot/initrd.img-2.6.17-10-386
quiet
boot
```

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

```
title          Windows 95/98/NT/2000
root           (hd0,0)
makeactive
chainloader    +1
```

# Boot

## grub : edition



```
Ubuntu, kernel 2.6.17-10-generic
Ubuntu, kernel 2.6.17-10-generic (recovery mode)
Ubuntu, memtest86+
```

Use the ↑ and ↓  
Press enter to  
commands before

```
root (hd0,0)
kernel /vmlinuz-2.6.17-10-generic root=/dev/mapper/Ubuntu-root ro
initrd /initrd.img-2.6.17-10-generic
quiet
savedefault
boot
```

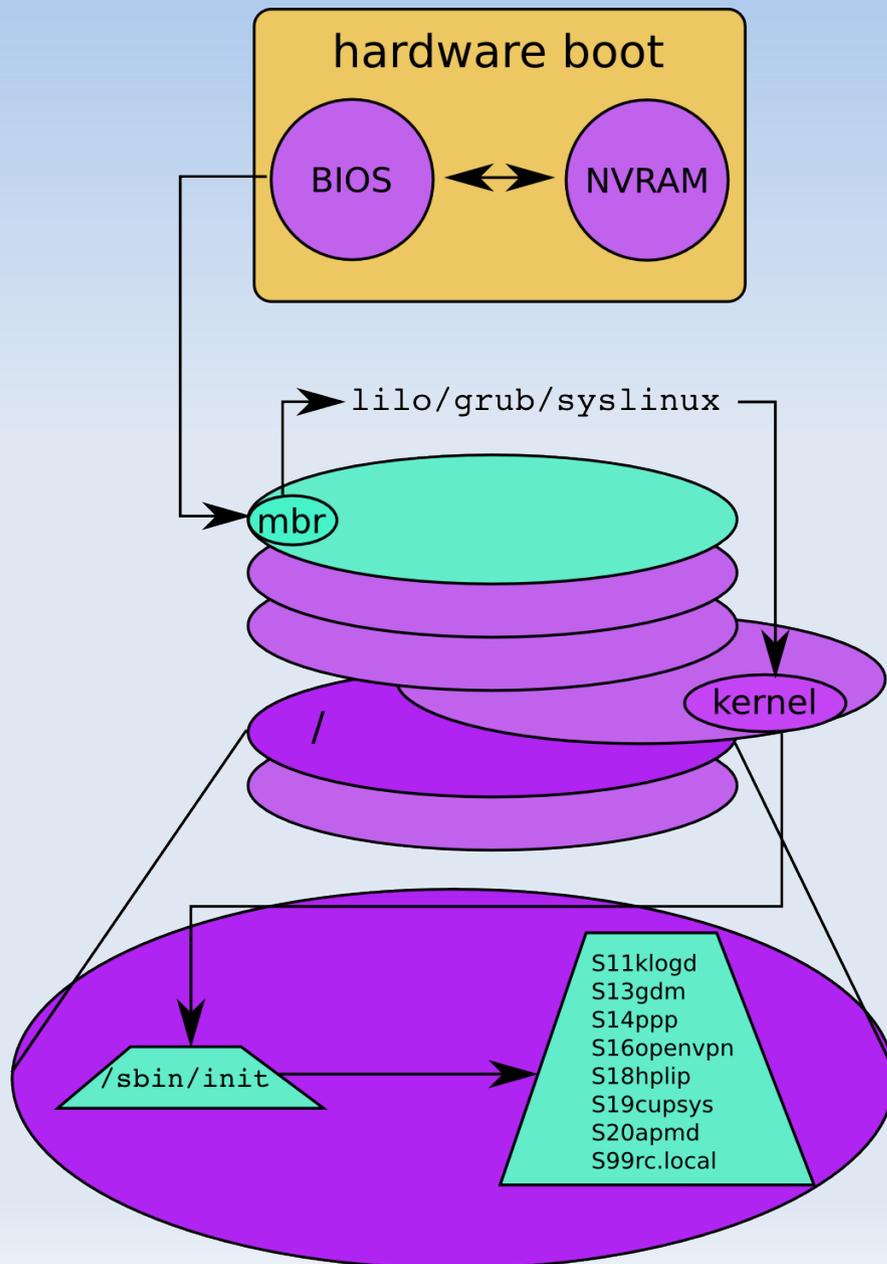
Use the ↑ and ↓ keys to select which entry is highlighted.  
Press 'b' to boot, 'e' to edit the selected command in the  
boot sequence, 'o' to boot the selected OS, 'r' to boot the  
selected line, or 'c' to boot the selected command.

[ Minimal BASH-like line editing is supported. For  
the first word, TAB lists possible command  
completions. Anywhere else TAB lists the possible  
completions of a device/filename. ESC at any time  
exits. ]

```
grub edit> kernel /vmlinuz-2.6.17-10-generic root=/dev/mapper/Ubuntu-root $ r>
```

# Boot

## Les 5 étapes



(1) Boot hardware  
exécution du bios  
paramètres dans la NVRAM

(2) Chargeur d'OS  
exécuté depuis le MBR par le BIOS  
limité à 512 bytes

(3) **Boot kernel**  
montage du root fs (/)  
chargement des drivers  
exécution d'*init*

(4) **init**  
passage au runlevel demandé  
exécution séquencée des scripts  
console

(5) **Scripts SysV**  
démarrage des services

# Boot

## Kernel

- Le kernel se décompresse lui même en RAM, et s'exécute, passant par différentes phases :
  - initialisation des sous-systèmes (mémoire, ACPI, CPU, ...)
  - initialisation des drivers (IDE, SCSI, ...)
  - chargement ramdisk
  - chargement modules du ramdisk
  - initialisation des (nouveaux) devices
  - montage des pseudo-filesystems (/proc, /sys)
  - montage de la (vraie) partition /
  - exécution d'init

# Kernel Versions

## Versionnement du noyau

### Versions a.b.c[.d]

- a.b : branche principale
  - b impair : branche de développement  
2.1, 2.3, 2.5., ...
  - b pair : branche stable  
2.0, 2.2, 2.4, 2.6, ...
- c : identifiant unique dans la série
- d : depuis 2.6.8/2.6.11, sous versions mineures intégrant des bugfixes (stabilisation de la version 'a.b.c')

`uname -r` permet de connaître votre version du noyau

# Kernel

## Modules

- Les modules sont des « morceaux » de noyau
- Ils peuvent être chargés/déchargés pendant l'exécution du kernel
- Ils sont compilés à partir de sources du kernel...  
...et peuvent aussi provenir de fournisseurs commerciaux (« drivers »)
- Un dictionnaire (modules.dep) permet de savoir quel module est nécessaire au fonctionnement de tel autre module
- On les trouve dans `/lib/modules/`uname -r`/`

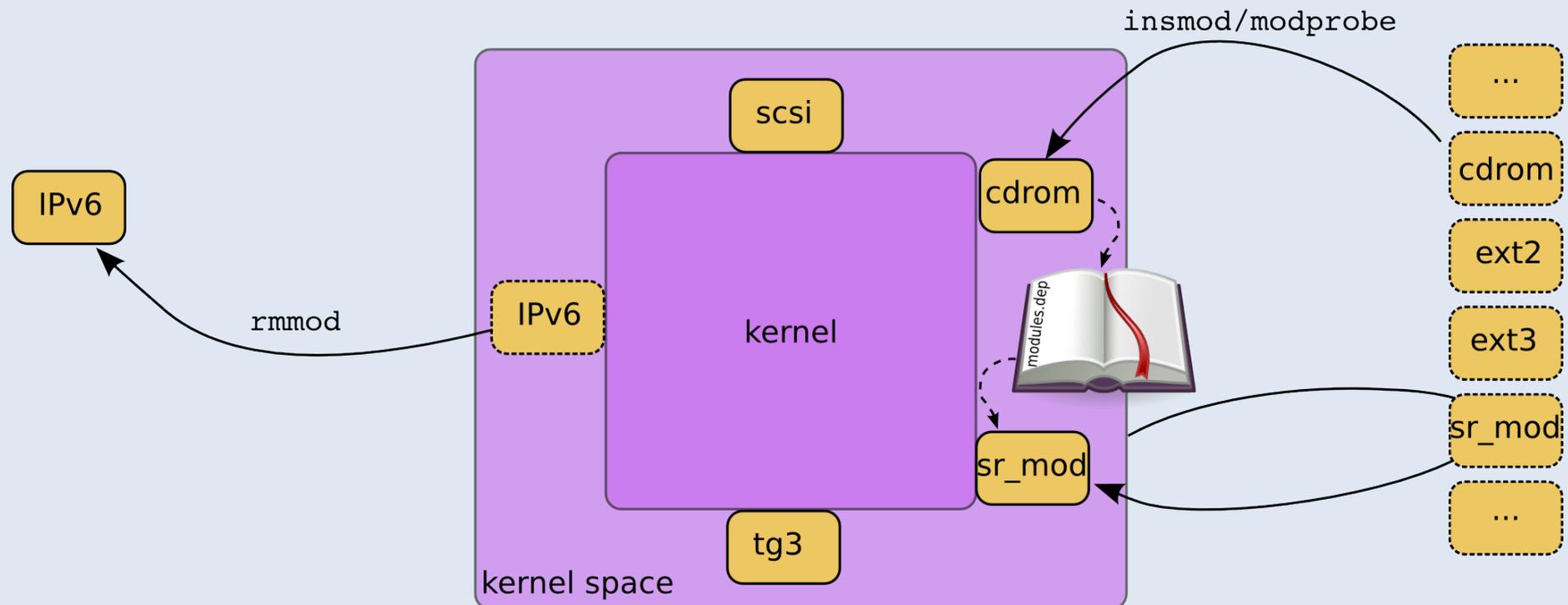
# Kernel Modules

## ...intérêt ?

- kernel plus petit (barre des 1,44Mo !)
- Empreinte mémoire optimisée
- Déboguage plus facile
- Modules binaires provenant de tiers
- Plusieurs version peuvent «cohabiter»
- Attention à la sécurité !

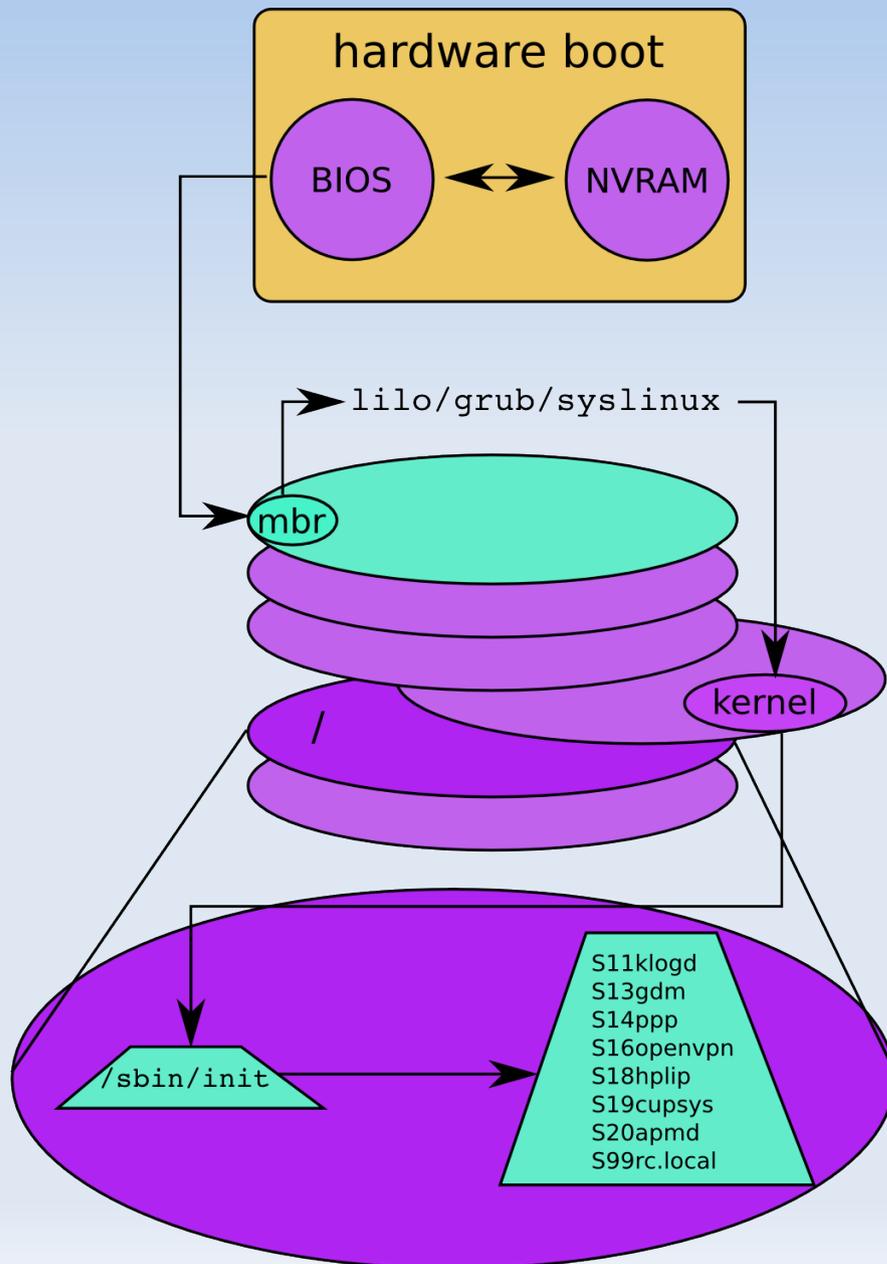
# Kernel Modules

- **Insertion** d'un module dans le kernel  
`insmod /chemin/complet/vers/module`  
`modprobe module`
- **Retrait** d'un module du kernel  
`rmmmod module`



# Boot

## Les 5 étapes



- (1) Boot hardware**  
exécution du bios  
paramètres dans la NVRAM
- (2) Chargeur d'OS**  
exécuté depuis le MBR par le BIOS  
limité à 512 bytes
- (3) Boot kernel**  
montage du root fs (/)  
chargement des drivers  
exécution d'*init*
- (4) *init***  
passage au runlevel demandé  
exécution séquencée des scripts  
console
- (5) Scripts SysV**  
démarrage des services

# Runlevels

## `init`

- `init` est le premier et le seul processus exécuté directement par le kernel (PID 1)
- il est le père de tous les autres
- du point de vue d'`init`, le système est dans des états discrets : les runlevels
- chaque runlevel définit un démarrage spécifique du système
- les runlevels possibles sont [0-6sS]
- le comportement d'`init` en fonction du runlevel est défini dans `/etc/inittab`
- `init` a donc pour vocation principale de mettre le système en état de marche

# Runlevels

`/etc/inittab`



`id:levels:action:process`

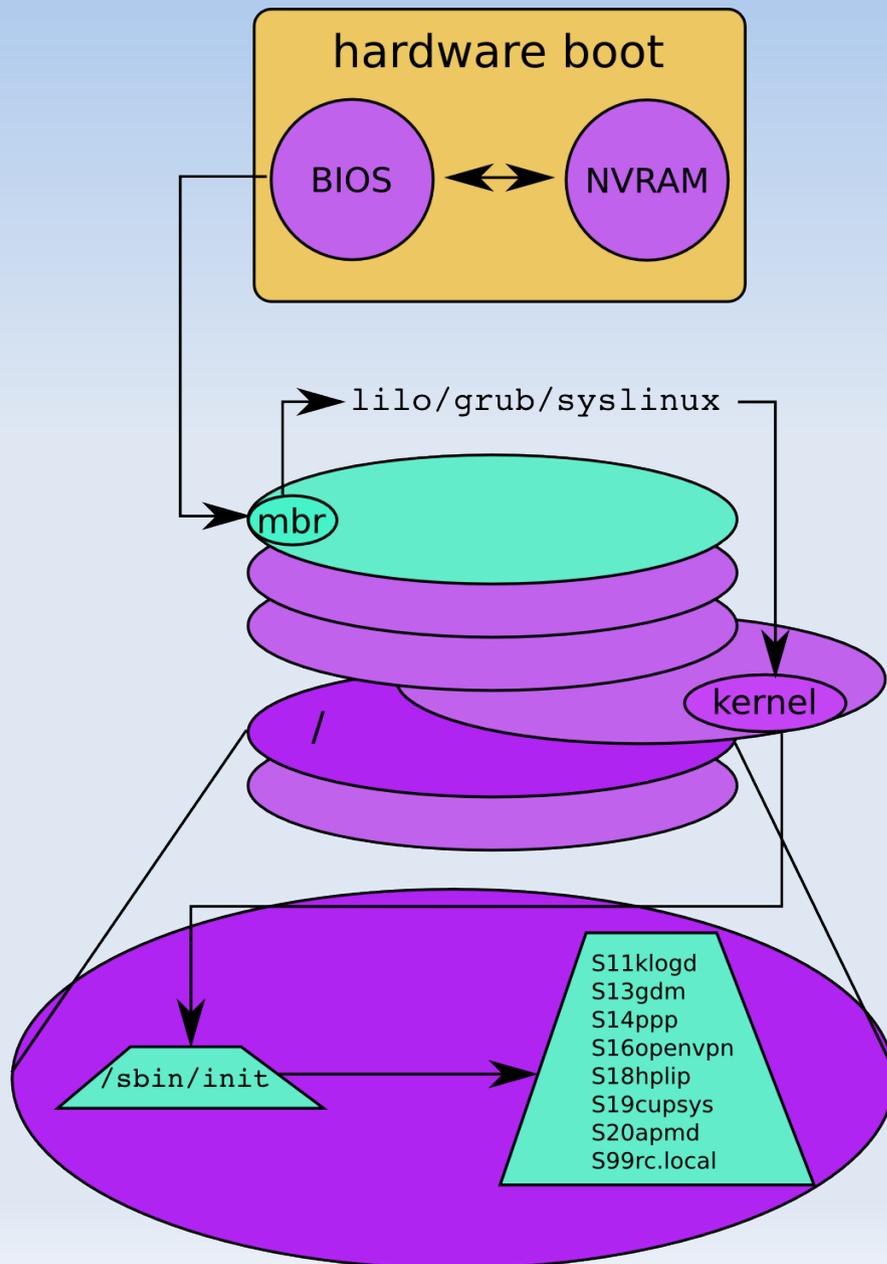
- `id` : identifiant unique
- `levels` : les runlevels pour lesquels cette action s'applique
- `action` : action ou déclencheur
- `process` : le(s) processus à lancer

```
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
```

```
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

# Boot

## Les 5 étapes



- (1) Boot hardware  
exécution du bios  
paramètres dans la NVRAM
- (2) Chargeur d'OS  
exécuté depuis le MBR par le BIOS  
limité à 512 bytes
- (3) Boot kernel  
montage du root fs (/)  
chargement des drivers  
exécution d'*init*
- (4) *init*  
passage au runlevel demandé  
exécution séquencée des scripts  
console
- (5) **Scripts SysV**  
démarrage des services

# SysV init

## Startup scripts

- init met le système en route en invoquant (indirectement) des scripts de démarrage
- Ces scripts permettent d'exécuter des applications persistantes (par exemple, un serveur de messagerie) appelées démons
- Ces scripts peuvent aussi exécuter des actions ponctuelles (montage des différents systèmes de fichiers, démarrage du réseau, ...)
- Ces scripts sont situés dans des répertoires `/etc/rcN.d` (N : runlevel)
- Ces scripts sont en fait des liens symboliques vers de vrais scripts existant dans `/etc/init.d`
- Ils sont exécutés dans l'ordre de leur numéros de séquence
- Ce type d'initialisation est appelé « SysV style init » (N.d.T : *initialisation à la sauce système cinq*)

# SysV init

## Startup scripts

- Les scripts situés dans les différents répertoires correspondant aux runlevels suivent une convention de nommage précise :

[SK][0-9][0-9]label

- **S/K** : le script sera appelé avec le paramètre **start** ou **stop** (**kill**).  
*C'est la responsabilité du script d'accepter et argument et de le traiter convenablement.*
- **00-99** : nombre à deux chiffres donnant le numéro de séquence.  
*Les scripts sont exécutés du plus petit numéro au plus grand. L'unicité n'est pas requise.*
- **label** : un label unique pour le script.  
*Correspond en général au nom entier du script original*

# SysV init

## Runlevels



**0** : arrêt de la machine

**S/s/1** : single user

mode mono-utilisateur utilisé pour la maintenance  
pas de réseau, personne ne se loggue

**2** : mode multi-utilisateur variant selon la distribution  
mode par défaut sous Ubuntu  
mode multi-utilisateur sans réseau sous RedHat

**3** : mode multi-utilisateur variant selon la distribution  
mode texte par défaut sous Redhat/Fedora

**4** : inutilisé sous la plupart des distributions

**5** : mode multi-utilisateur  
mode graphique par défaut sous Redhat/Fedora

**6** : reboot de la machine

*A l'exception des niveaux 0 et 6, ces conventions varient d'un unix à l'autre*

La commande `runlevel` permet de connaître le runlevel précédent et le runlevel actuel

# SysV init

## Startup scripts



```
$ ls /etc/rc2.d/
```

```
K77ntp-server          S20nbd-server          S99rc.local
S01appport             S20nvidia-kernel      S89anacron
S05vbesave             S20postfix            S89atd
S10acpid               S20powernowd          S89cron
S10powernowd.early    S20rsync              S90binfmt-support
S10sysklogd           S20ssh                S95preload
S11klogd               S20sysstat            S98usplash
S13gdm                 S20tftpd-hpa          S99acpi-support
S14ppp                 S20virtualbox         S99rc.local
S16openvpn            S20xinetd             S99rmnologin
S18hplip               S25bluetooth          S99stop-readahead
S19cupsys              S25mdadm
S20apmd                S89anacron
S20dbus                S89atd
S20festival            S89cron
S20hotkey-setup        S90binfmt-support
S20laptop-mode         S95preload
S20makedev             S98usplash
S20nbd-client          S99acpi-support
```

# SysV init



## Démarrer un script 'à la main'

- Les applications qui doivent s'exécuter au boot installent un script dans `/etc/init.d`
- Il est ensuite possible de contrôler cette application grâce à ce script
- **Démarrer** l'application  
`/etc/init.d/application start`
- **Stopper** l'application  
`/etc/init.d/application stop`
- **Redémarrer** l'applications  
`/etc/init.d/application restart`
- Demander à l'application de **recharger** sa configuration  
`/etc/init.d/application reload`

# Runlevels

## Changer de runlevel



- Au boot, il est possible de demander un démarrage mono-utilisateur à GRUB

```
root (hd0,0)
kernel /vmlinuz-2.6.17-10-generic root=/dev/mapper/Ubuntu-root ro
initrd /initrd.img-2.6.17-10-generic
quiet
savedefault
boot
```

Use the ↑ and ↓ keys to select which entry is highlighted.  
Press 'b' to boot, 'e' to edit the selected command in the boot sequence, 'c' for a command-line, 'o' to open a new line after ('O' for before) the selected line, 'd' to remove the selected line, or escape to go back to the main menu.

- En ajoutant 'S' ou 'single' à la ligne kernel, le noyau va demander à init de démarrer en mode 'single user'

# Runlevels

## Changer de runlevel



- En cours de fonctionnement, la commande telinit demande à init de modifier le runlevel  
`telinit level`

```
[17179584.648000] eth0: Using PHY number 0.
[17179584.648000] eth0: link up, 100Mbps, full-duplex
* Filesystem type 'usbfs' is not supported. Skipping mount.
* Loading manual drivers...
[17179585.060000] lp: driver loaded but no devices found
* Activating swap... [ ok ]
* Checking root file system...
fsck 1.39 (29-May-2006)
/dev/mapper/Ubuntu-root: clean, 22345/348160 files, 361354/696320 blocks [ ok ]
* Assembling RAID arrays...
* Setting up LVM Volume Groups... [ ok ]
* Checking file systems...
fsck 1.39 (29-May-2006)
/dev/hda1: clean, 30/124496 files, 28654/248976 blocks [ ok ]
* Mounting local filesystems...
* Configuring network interfaces...
* Setting up console font and keymap...
root@edgyserver:~# _
```

- `shutdown -h now` (ou `halt`) est l'équivalent de `telinit 0`
- `shutdown -r now` (ou `reboot`) est l'équivalent de `telinit 6`

# Cron

## Fonctionnement

- cron permet de programmer des tâches récurrentes sur le système
- ces tâches sont listées dans des «crontabs»
  - chaque utilisateur possède sa propre crontab
  - il y a une crontab système
- ces fichiers sont scrutés par le démon cron/crond/anacron chaque minute (pas besoin de redémarrer le démon)
- cron exécute une tâche si son heure est venue
- grâce à cron, on peut automatiser la rotation des logs, la mise à jour de bases de données, les backups, la génération d'index...

# Cron

## Configuration

**cron** peut être configuré de multiples manières

- */etc/crontab*  
crontab système; contient les programmations globales du système
- */var/spool/cron/crontabs/\$USER*  
contient la crontab de \$USER
- */etc/cron.d/*  
contient des 'mini-crontabs' ajoutées par les packages à l'installation
- */etc/cron.{hourly,daily,weekly,monthly}*  
contient des **scripts** exécutés respectivement toutes les heures/jours/semaines/mois

# Cron

## Format des crontabs

min heure jourmois mois joursemaine user command  
(le champ user n'existe que dans la crontab système)

**min** : à quelle minute de l'heure [0-59]

**heure** : à quelle heure [0-23]

**jourmois** : quel jour du mois [1-31]

**mois** : quel mois de l'année [1-12]

**joursemaine** : quel jour de la semaine [0-7], 0=7=dimanche

**user** : sous quel utilisateur

**command** : commande à exécuter

Pour chaque champ chronologique, on peut avoir :

- une valeur, l'exécution aura lieu à cette valeur
- une plage de valeurs A-B, l'exécution aura lieu à chaque valeur
- une combinaison de plages séparée par des ',' (A-B,C,D-E,...)
- une '\*', correspondant à la plage entière
- un modificateur '/n' pour une plage, provoquant l'exécution toutes les *n* fois

# Cron

## Format des crontabs

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other
# crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h jdm moi jds user  command
17 * * * * root    commande1
25 6 * * * root    commande2
47 6 * * 7 root    commande3
52 6 1 * * root    commande4
* * * * * root    commande5
* */2 * * * root    commande6
0-29/3 * * * * root    commande7
#
```

# Cron

## Editer sa crontab

- crontab permet d'éditer un fichier cron  
`crontab -u user -e`  
édition de la crontab de user
- L'édition se fait avec l'éditeur spécifié dans `$VISUAL` OU `$EDITOR`
- La variable `$MAILTO` dans une crontab permet de spécifier l'adresse à laquelle envoyer le résultat des commandes

`MAILTO=` → envoi au propriétaire de la crontab

`MAILTO=tux@linux.org` → envoi a `tux@linux.org`

`MAILTO=""` → aucun envoi

# At

## Execution différée unique

- **at** permet d'exécuter une commande à une heure donnée
- contrairement à cron, cette exécution est unique
- **at** lit les commandes à exécuter sur STDIN
- **at** renvoie par mail STDOUT et STDERR renvoyé par les commandes
- **atq** et **atrm** permettent respectivement de voir la liste des jobs et d'en supprimer

```
echo "echo Anniversaire Tux" | at Aug 25
echo "echo fais chauffer l'eau" | at teatime
- 5 minutes
echo "md5sum /bin/*" | at now + 1 hour
```

# Logs

## Syslog

- **syslog** permet d'unifier la gestion des logs sur un ou plusieurs systèmes
- syslog :
  - un démon
  - un protocole permettant de transmettre des logs via un réseau IP
- syslog permet d'envoyer les logs vers
  - la console
  - un tty
  - des fichiers, des pipes
  - des usager loggués localement
  - un démon syslog distant

# Logs

## Syslog : concepts

**syslog** repose sur 2 concepts fondamentaux

- **facility**

la 'facility' est la catégorie de l'évènement prise parmi : *auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, syslog, user, uucp, local[0-7]*

- **level**

level est le niveau d'un événement, pris parmi : *debug, info, notice, warning, err, crit, alert, emerg*

→ chaque événement envoyé au démon syslog possède une *facility* et un *level*

→ syslog va ensuite traiter ce message en fonction de ces deux paramètres et de sa configuration

# Logs

## Syslog : /etc/syslog.conf

<u>facility.level</u>	<u>destination</u>
auth	debug
authpriv	info
cron	notice
daemon	warning
ftp	err
kern	crit
lpr	alert
mail	emerg
news	
syslog	
user	
uucp	
local[0-7]	
	@machine
	-/var/log/fichier
	/var/log/fichier
	/dev/console
	root
#	
# Exemples de logs	
#	
auth,authpriv.*	/var/log/auth.log
*.*;auth,authpriv.none	-/var/log/syslog
daemon.*	-/var/log/daemon.log
kern.*	-/var/log/kern.log
uucp.*	/var/log/uucp.log
mail.info	-/var/log/mail.info
mail.warn	-/var/log/mail.warn
mail.err	/var/log/mail.err

# Logs

## logrotate : rotation des logs

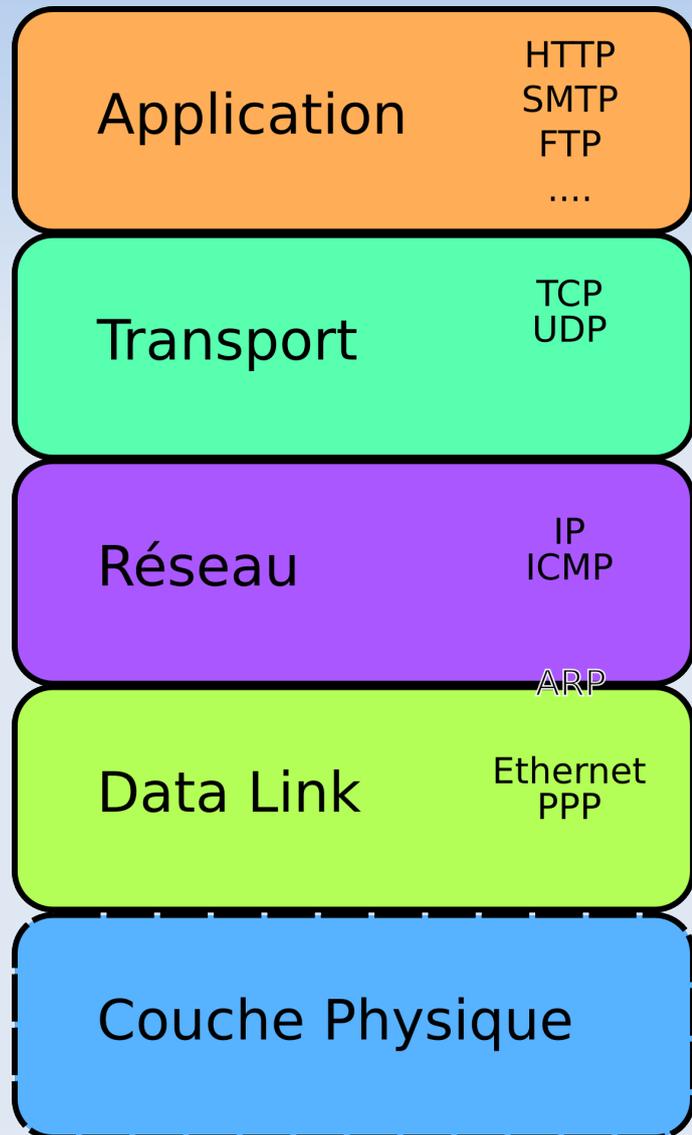
- Sans intervention, les logs vont devenir de plus en plus gros
  - problème, le filesystem n'est pas illimité !
- logrotate est appelé périodiquement par cron et permet :
  - d'archiver un log
  - de compresser un log
  - de redémarrer un démon afin qu'il réouvre son fichier de log
- Il est très configurable et peut fonctionner selon une période programmable et conserver autant d'archives que nécessaire

# Réseau TCP/IP



# Réseau TCP/IP

## Théorie

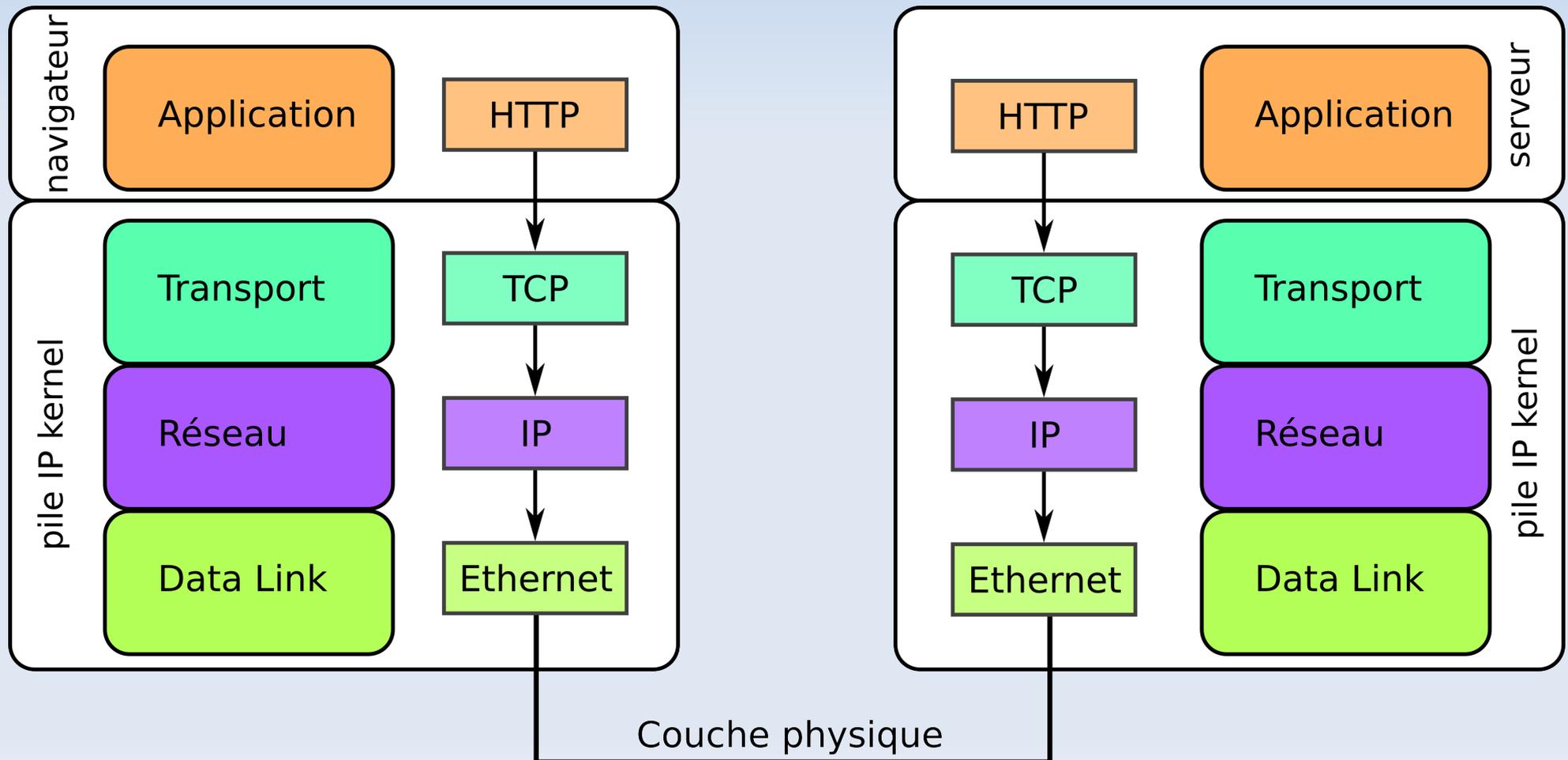


- Les réseaux sont généralement organisés en "piles protocolaires"
- chaque couche de la pile offre un niveau d'abstraction supplémentaire à la couche supérieure
- chaque couche offre un service supplémentaire par rapport à la couche inférieure

# Réseau TCP/IP

## Piles

### Communication navigateur $\Leftrightarrow$ serveur web

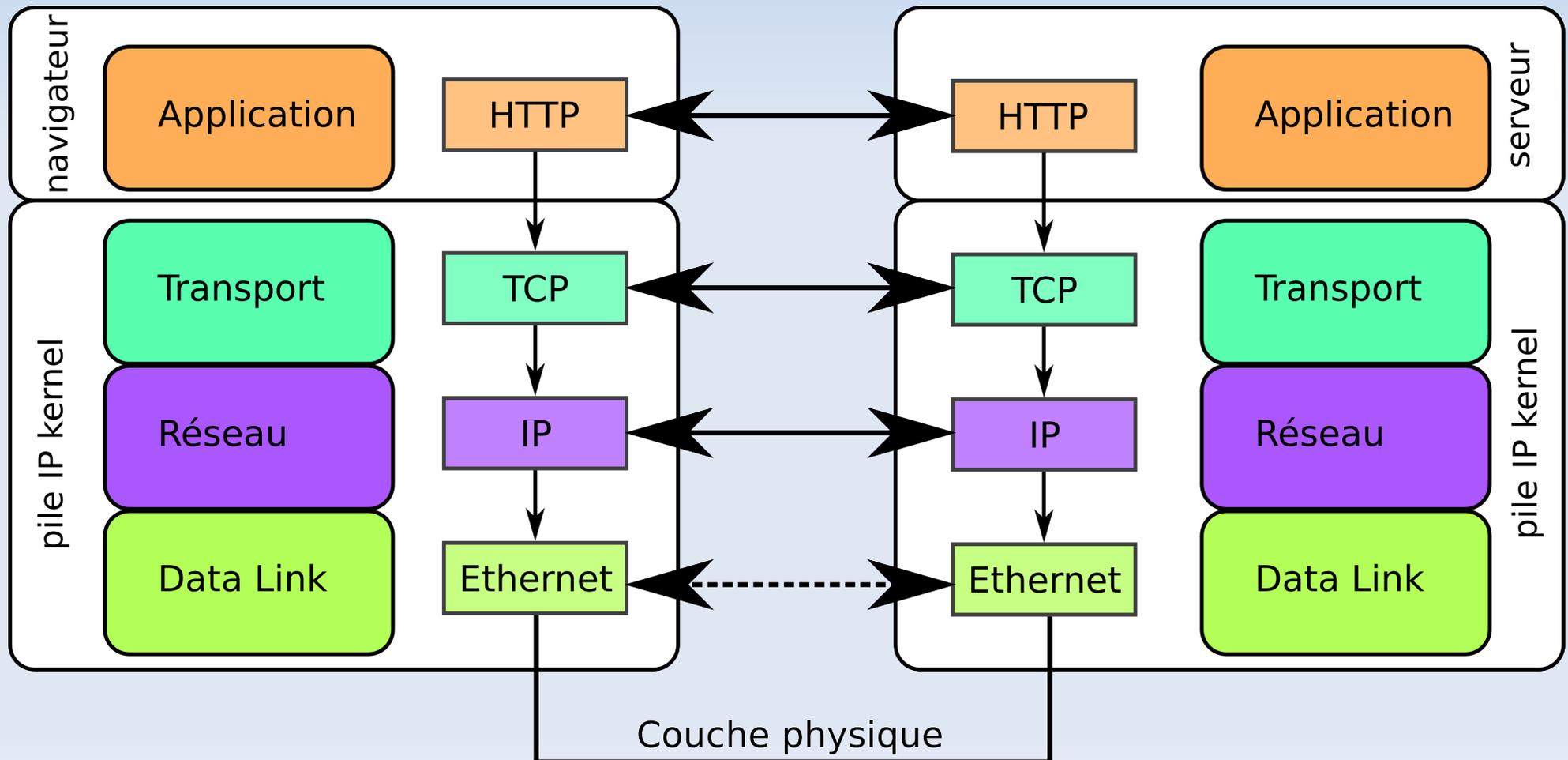


# Réseau TCP/IP

## Piles

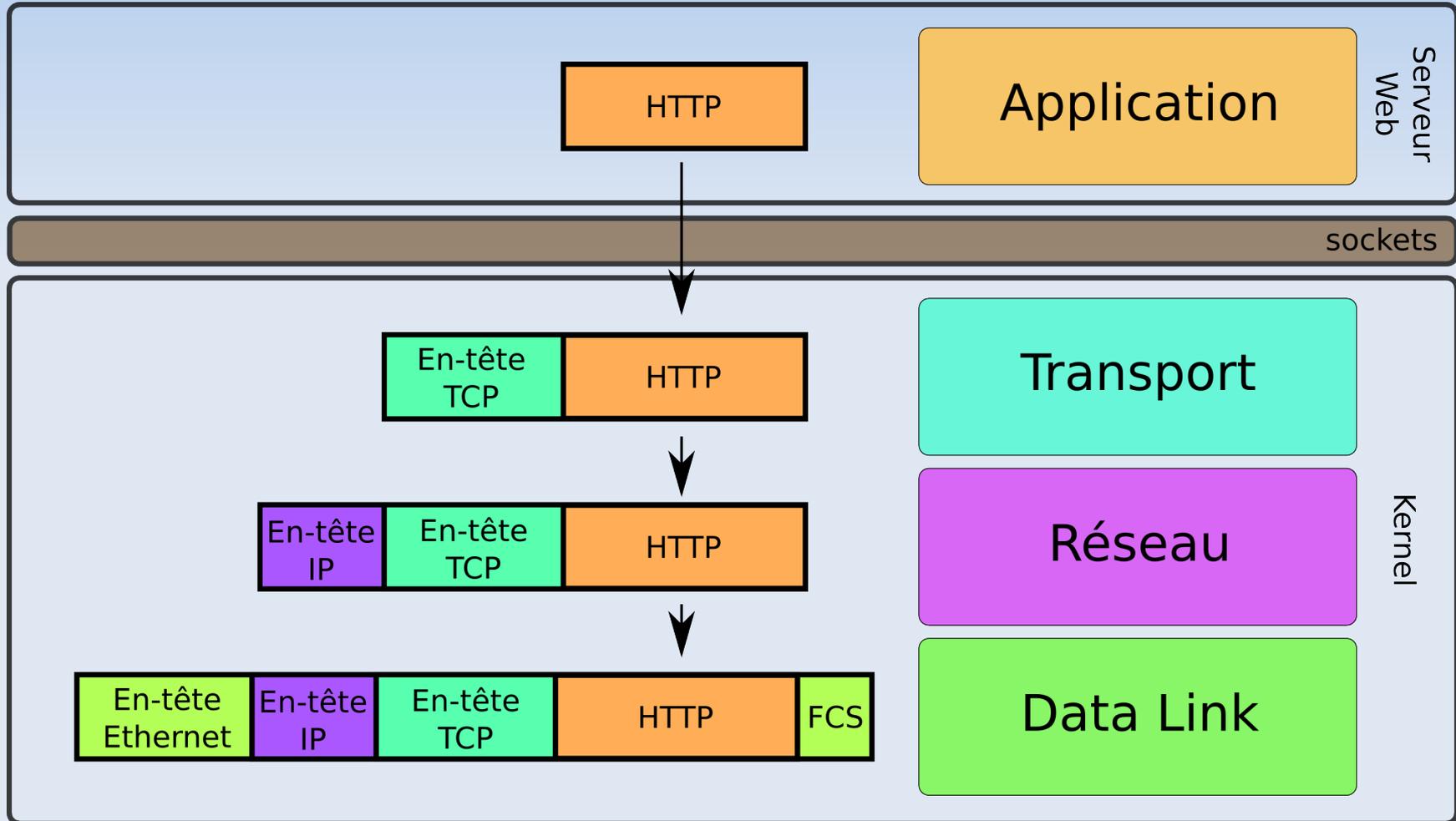
### Communication navigateur $\Leftrightarrow$ serveur web

*c'est aussi une communication entre chacune des couches*



# Réseau TCP/IP

## Encapsulation



# Réseau TCP/IP

## Adressage IP

- 4 octets (32 bits)
- 0.0.0.0  $\Leftrightarrow$  255.255.255.255  
soient  $2^{32}$  adresses possibles ( $\sim$ 4.3 milliards)
- $n$  bits pour l'adresse réseau
- $(32-n)$  bits pour l'adresse de l'hôte
- Adresses publiques attribuées par l'ICANN via différents RIRs (RIPE, ARIN, APNIC, LACNIC, AfriNIC)
- Plages d'adresses privées d'usage libre :
  - 10.0.0.0/8 (10.0.0.0  $\Leftrightarrow$  10.255.255.255)
  - 172.16.0.0/12 (172.16.0.0  $\Leftrightarrow$  172.31.255.255)
  - 192.168.0.0/16 (192.168.0.0  $\Leftrightarrow$  192.168.255.255)

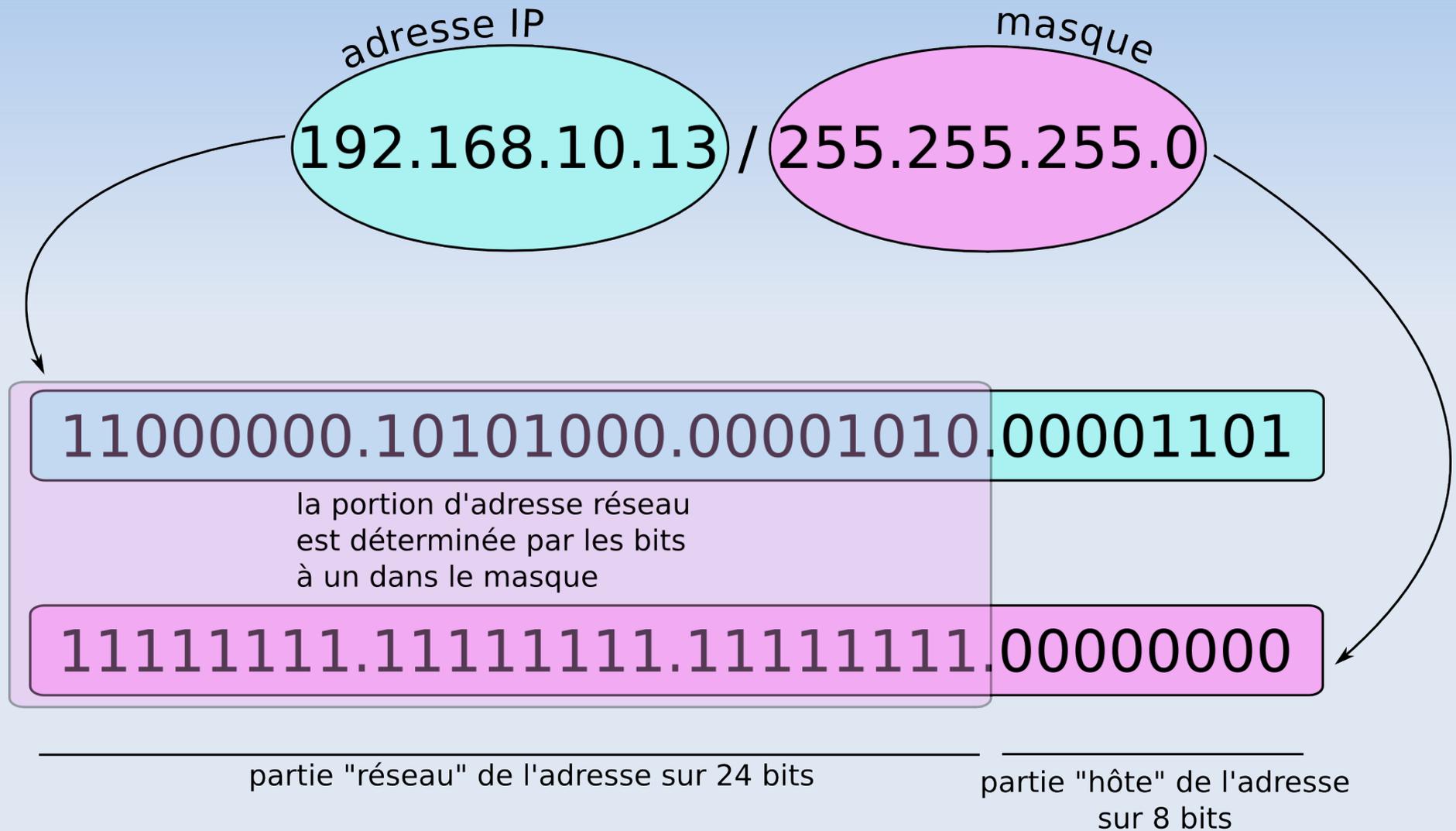
# Réseau TCP/IP

## Adressage IP : masques

- Le masque détermine quelle est la proportion d'adresse réseau et d'adresse d'hôte dans une adresse IP
- Ce concept n'a de sens que pour prendre des décisions de routage
- La première adresse d'une plage désigne le réseau lui-même; la dernière est l'adresse de broadcast (multi-diffusion)
- < 1994, les adresses étaient séparées par classes ("classfull addressing") :
  - classe A : 8 bits adresse réseau, 24 bits adresse hôte
  - classe B : 16 bits adresse réseau, 16 bits adresse hôte
  - classe C : 24 bits adresse réseau, 8 bits adresse hôte
- > 1994, les adresse IP se font rares : les frontières ne sont plus forcément alignées aux classes (8, 16, 24 bits)
  - ➔ naissance de CIDR (Classless InterDomain Routing)

# Réseau TCP/IP

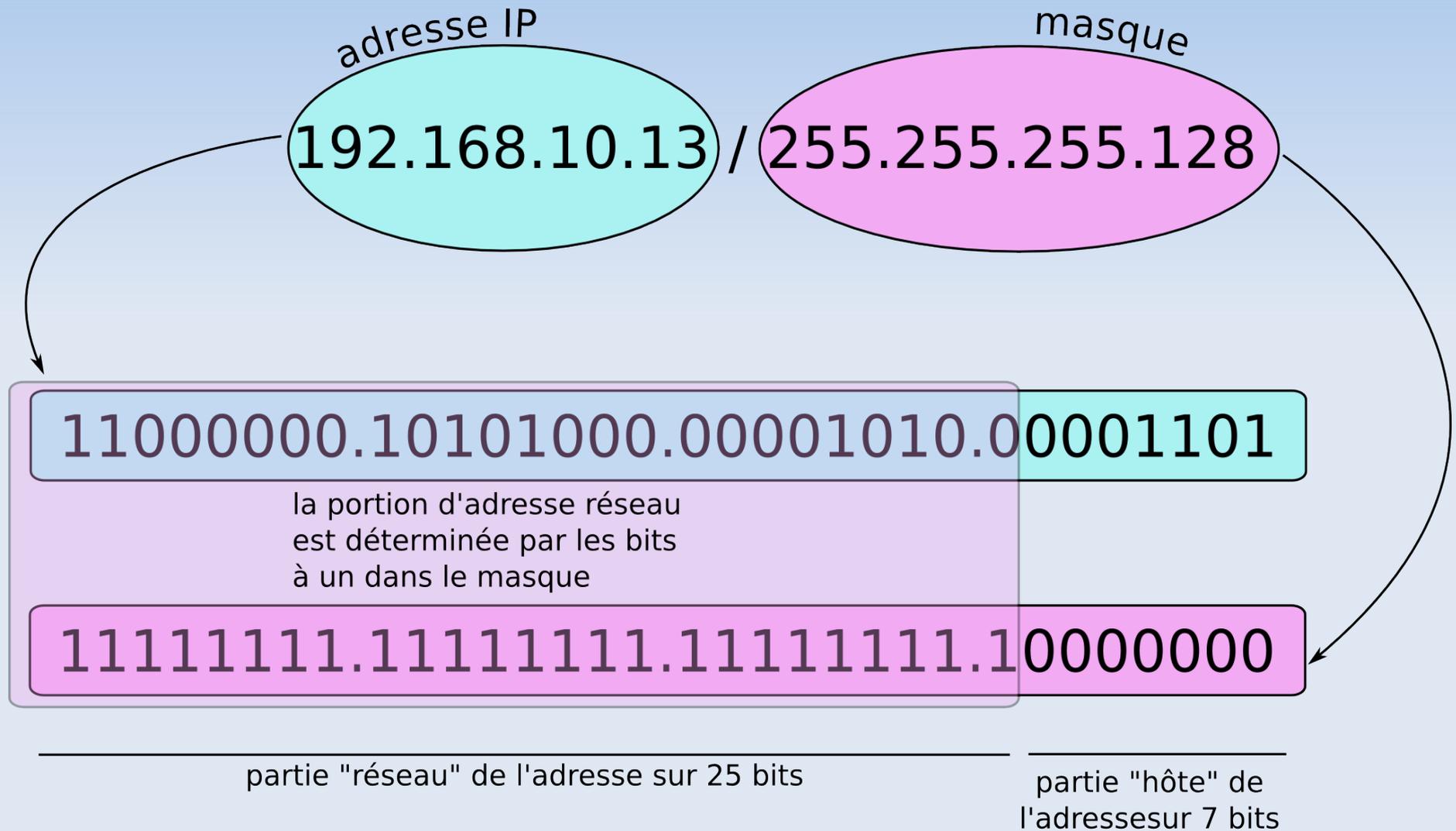
## Adressage IP : masques



→ masque CIDR : /24

# Réseau TCP/IP

## Adressage IP : masques



→ masque CIDR : /25

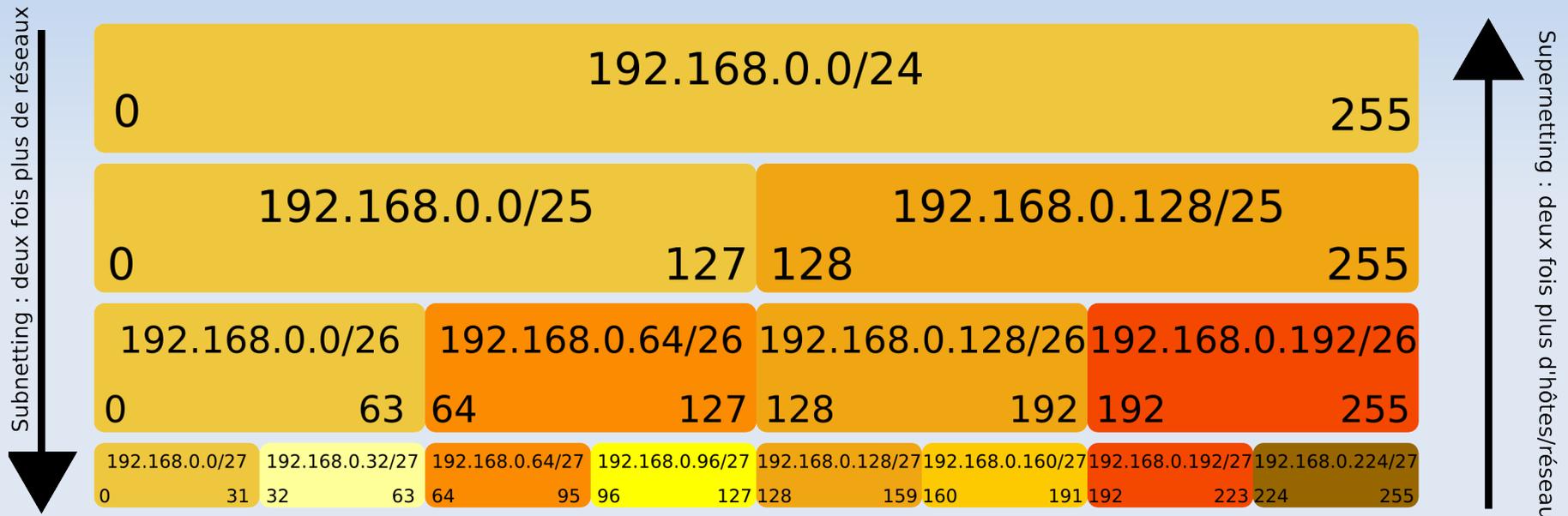
# Réseau TCP/IP

## Adressage IP : CIDR

- Les masques CIDR donnent directement le nombre de bits réseau dans l'adresse
- CIDR permet une (plus) grande souplesse dans l'usage des adresses
- CIDR permet d'agréger des blocs ensemble, afin de réduire la taille tables de routage
- L'opération de division d'un bloc s'appelle le "subnetting"
- L'opération d'aggregation de blocs s'appelle le "supernetting"

# Réseau TCP/IP

## Adressage IP : CIDR





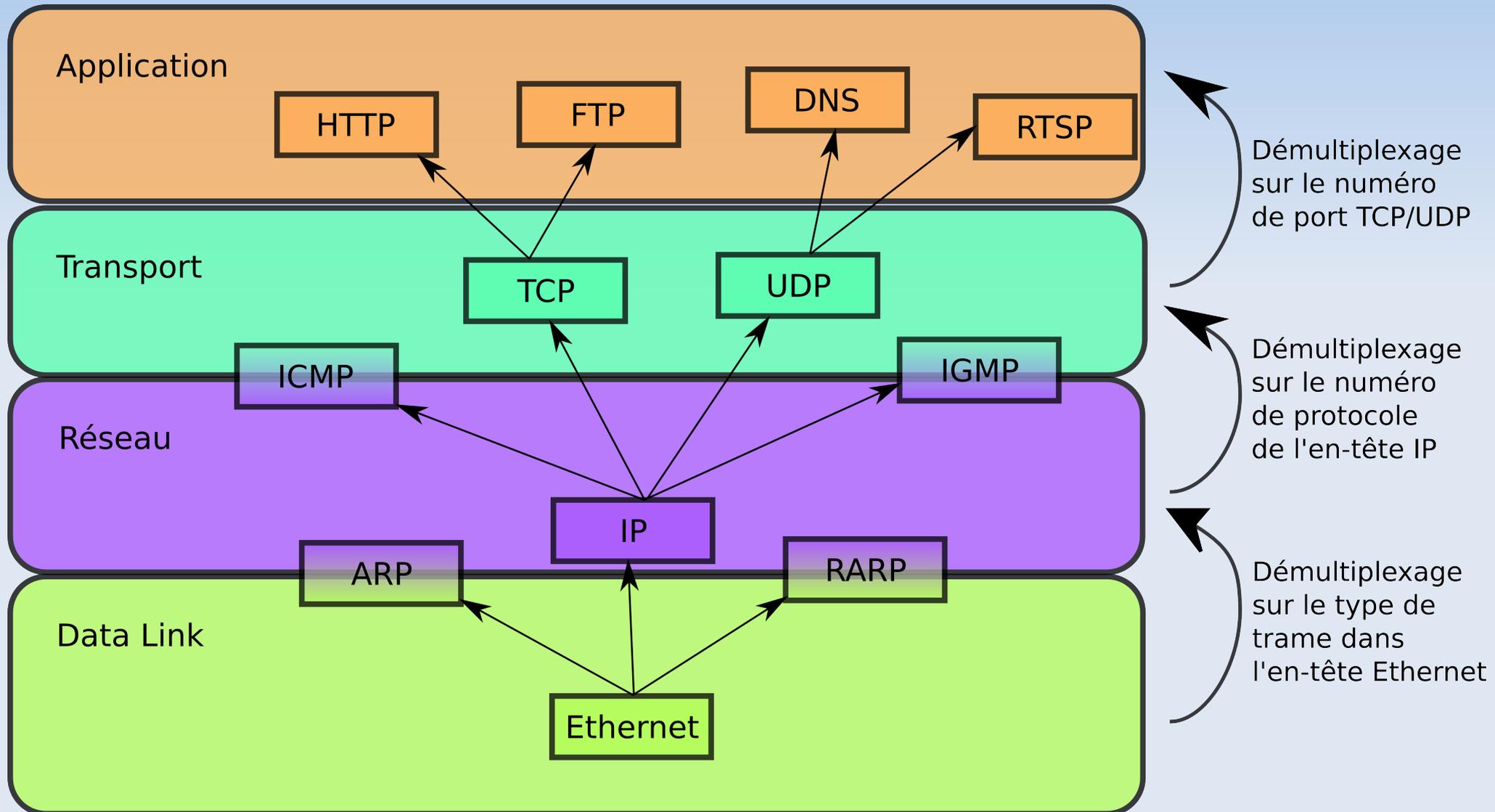
# Réseau TCP/IP

## Ports UDP/TCP

- Codés sur 16 bits (0-65535)
- Ports standards attribuées aux protocoles par l'ICANN
- Pas d'obligation d'utiliser le port standard (séparation des couches de la pile)
- Numéros de port  $> 30000$  rarement utilisés (masquerading, traceroute)
- Sous Linux (\*nix), les ports  $< 1024$  sont réservés à root

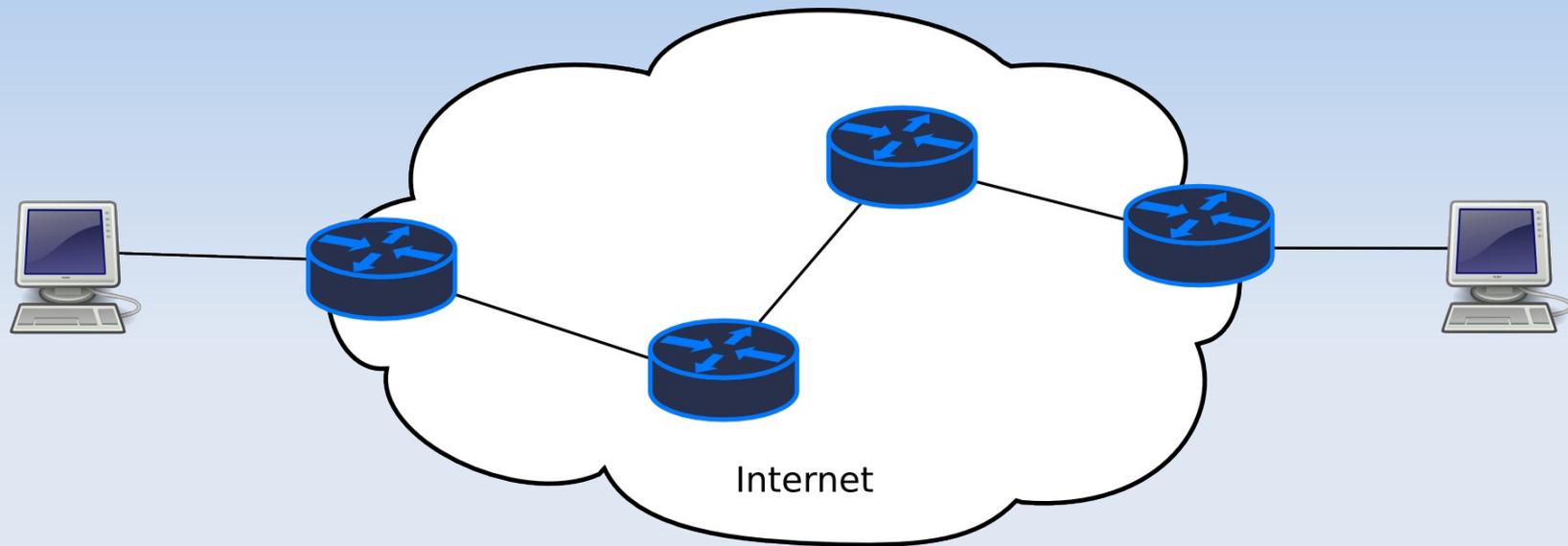
# Réseau TCP/IP

## Démultiplexage



# Réseau TCP/IP

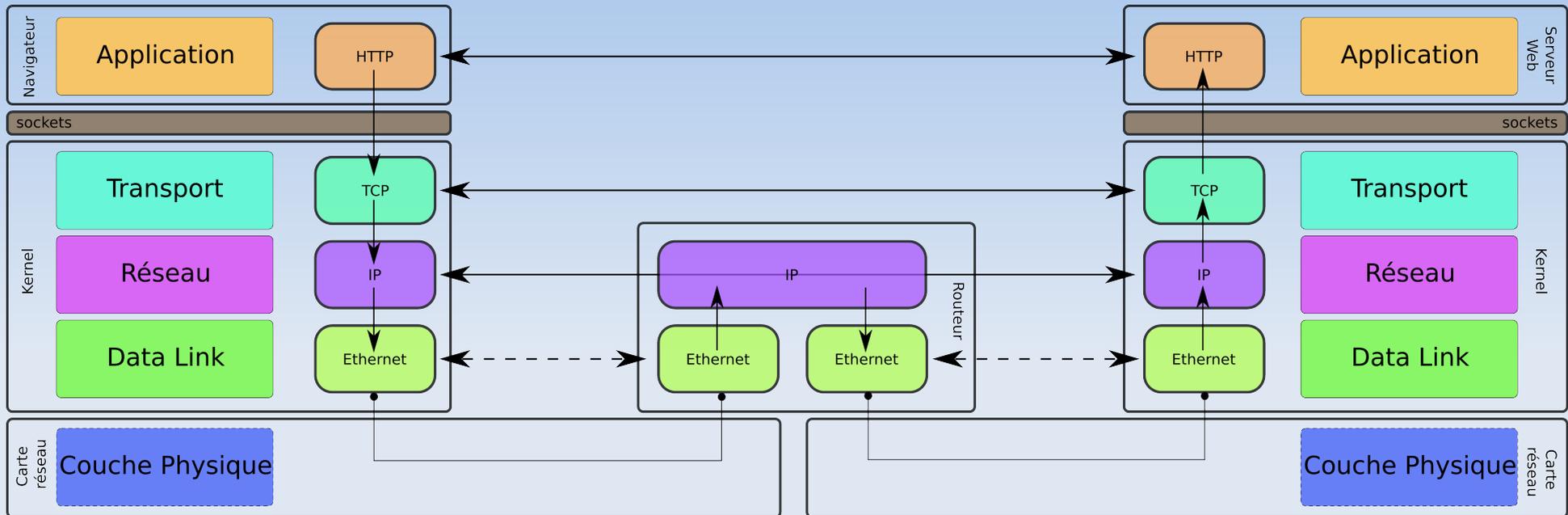
## Routage des datagrammes IP



- Pour communiquer, les hôtes transmettent leur paquets IP à des "routeurs"
- Les "routeurs" sont aussi des hôtes au sens TCP/IP, avec la particularité
  - d'avoir plusieurs interfaces
  - de faire passer des paquets d'une interface à l'autre

# Réseau TCP/IP

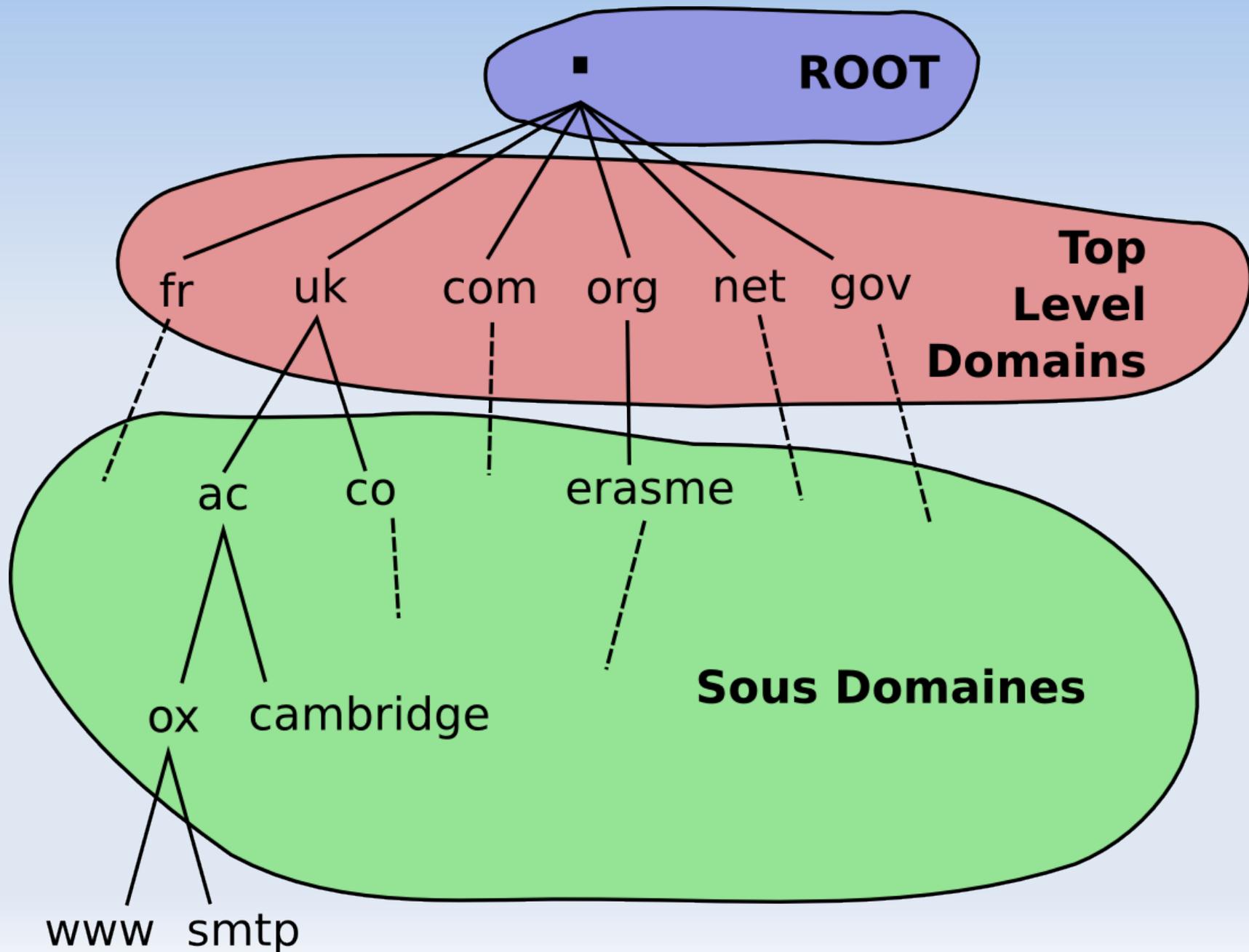
## Routage des datagrammes IP



- Un datagramme voyage dans plusieurs dimensions
  - entre les équipements
  - dans la pile IP de ces équipements
- Décapsulation/encapsulation à chaque noeud selon "niveau" du "switch"
- Le routage IP s'effectue par prise de décision en fonction de l'IP de destination

# Réseau TCP/IP

## DNS : Domain Name System



# Réseau TCP/IP

## Configuration

- Pour être fonctionnelle, la pile IP n'a besoin que d'une chose : une adresse (avec son masque)
- Pour sortir du LAN, il faudra aussi une passerelle (gateway, next-hop)
- Pour résoudre les noms en adresse, il faut un serveur DNS

**Accès Internet = adresse + next-hop (+ DNS)**

# Réseau TCP/IP



## Adresse IP : `ifconfig`

- `ifconfig` permet d'affecter des adresses IP aux interfaces de la machine
- `ifconfig` existe sur tous les Unix BSD

```
ifconfig [interface] [paramètres]
```

```
ifconfig
```

affiche les informations sur toutes les interfaces

```
ifconfig eth0
```

affiche les informations sur l'interface eth0

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

affecte l'adresse 192.168.0.1/24 à l'interface eth0

# Réseau TCP/IP



## Table de routage : route

- route permet d'affecter des adresses IP aux interfaces de la machine
- route existe sur tous les Unix BSD

```
route [add|del] réseau [gw passerelle]
```

```
route
```

affiche les informations sur toutes les interfaces

```
route add default gw 192.168.0.254
```

ajoute une route par défaut via 192.168.0.254

```
route del default
```

supprime la route par défaut

# Réseau TCP/IP

## Adresse IP : `ip`

- Il faut savoir qu'`ifconfig` et `route` existent...  
... mais il *faut* utiliser la commande `ip`
- `ip` gère :
  - l'adressage IP (comme `ifconfig`)
  - la table de routage (comme `route`)
  - la table arp (comme `arp`)
  - les devices (comme `ifconfig`)
- mais apporte aussi du nouveau :
  - le policy routing
  - gestion des autres tables de routage
  - syntaxe cohérente et concision
- `ip` est la commande à utiliser pour bénéficier du routage avancé (`iproute2`) sous linux

# Réseau TCP/IP



## Adresse IP : `ip link/addr`

- Voir les adresses  
`ip addr show [dev device]`
- Affecter une adresse avec ip  
`ip addr add adresse/msk dev device`  
`ip addr add adresse netmask msk dev device`  
(en cas d'appels multiples, ip ajoute des alias à l'interface)
- Supprimer une adresse  
`ip addr del adresse/msk dev device`  
`ip addr del adresse netmask msk dev device`
- Supprimer toutes les adresses  
`ip addr flush dev device`
- Mettre en route/couper une l'interface  
`ip link set device état`  
(état : *up* pour mettre en marche, *down* pour couper)

# Réseau TCP/IP



## Adresse IP : `ip route`

- Voir la table de routage principale  
`ip route show`
- Ajouter une route  
`ip route add adresse/msk via passerelle`
- Supprimer une route  
`ip route del adresse/msk`
- Supprimer toutes les routes  
`ip route flush table main`
- Obtenir la route pour un subnet  
`ip route get adresse/msk`
- Couper le routage vers un subnet  
`ip route add blackhole adresse/msk`

# Réseau TCP/IP

## DNS : le resolver

- La résolution d'adresses est l'affaire du *resolver*
- Le resolver n'est pas un démon mais une librairie qui est utilisée par les applications (pas de cache)
- Le resolver se configure via `/etc/resolv.conf`
- Dans une configuration standard, le resolver consulte dans l'ordre :
  - le fichier `/etc/hosts`
  - le 1<sup>er</sup> serveur de noms de `/etc/resolv.conf`
  - le 2<sup>ème</sup> serveur de noms de `/etc/resolv.conf`
  - le 3<sup>ème</sup> serveur de noms de `/etc/resolv.conf`
  - parfois DNS multicast (Ubuntu)

# Réseau TCP/IP



## DNS : /etc/resolv.conf

- la directive 'nameserver' permet de spécifier les adresses IP de serveurs de noms (3 maximum)

```
nameserver a1.b1.c1.d1  
nameserver a2.b2.c2.d2
```

⇒ le résolveur tentera une résolution sur a1.b1.c1.d1 puis sur a2.b2.c2.d2 si infructueux

- la directive 'search' permet d'ajouter automatiquement un domaine à un nom sans point ('.') :

```
search domaine1.org domaine2.org
```

⇒ une résolution de 'serveur' se traduira par la résolution de serveur.domaine1.org puis serveur.domaine2.org si infructueux

# Réseau TCP/IP

## DNS : /etc/hosts



**/etc/hosts** peut contenir des associations  
IP/hostname, permettant :

- de définir des noms personnalisés (plus courts, pour du test, ...)
- de surcharger (recouvrir) des noms d'hôtes existants (interception de l'accès à un hostname, utilisation d'un miroir plus proche)
- de résoudre plus rapidement des noms fréquemment demandés
- de résoudre des noms en l'absence ou indisponibilité des DNS

```
a.b.c.d          nom.domaine.org nom nombis ...
```

# Réseau TCP/IP

## Configuration via DHCP



- DHCP se présente sous forme d'un démon
- Deux variantes occupent le terrain :
  - dhcpcd (Fedora)
  - dhclient (Debian)
- Démarrer dhcp sur l'interface :  

```
dhcpcd eth0
```

```
dhclient eth0
```
- Stopper le démon dhcp :  

```
dhclient -r
```

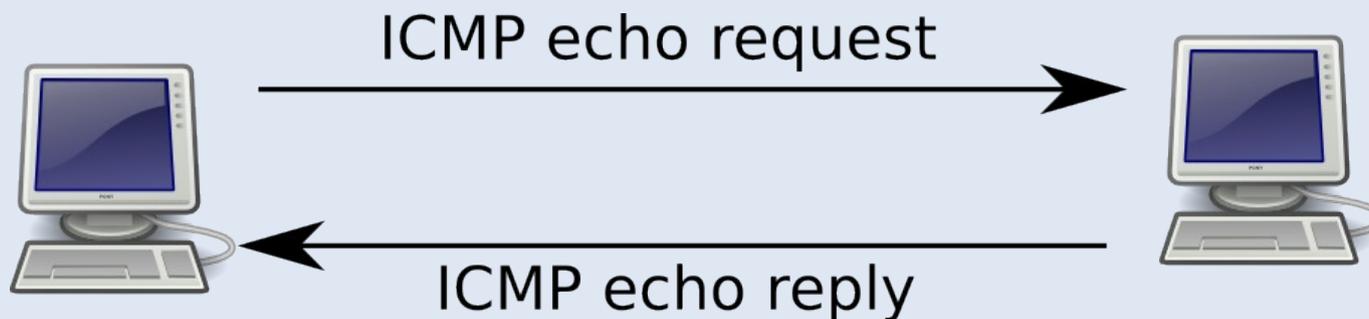
  

```
dhcpcd -k
```

# Réseau TCP/IP

## Diagnostic : connectivité IP

- ping utilise le protocole ICMP pour vérifier la connectivité entre deux hôtes
  - par défaut, ping envoie un paquet par seconde indéfiniment
  - l'option -f (*flood*) permet d'envoyer au moins 100 'echo request' par seconde



- `fping` est une variante qui permet de pinger plusieurs cibles à la fois

# Réseau TCP/IP



## Diagnostic : connectivité IP

`ping [-c nombre] [-f] [-s] adresse`

*nombre* : nombre de paquets à envoyer

*adresse* : adresse IP destinataire

-f : flood (au moins 1 paquet toutes les 10ms)

-s : taille du paquet en bytes

`fping [-c nombre] [-g] adresse`

*nombre* : nombre de paquets à envoyer

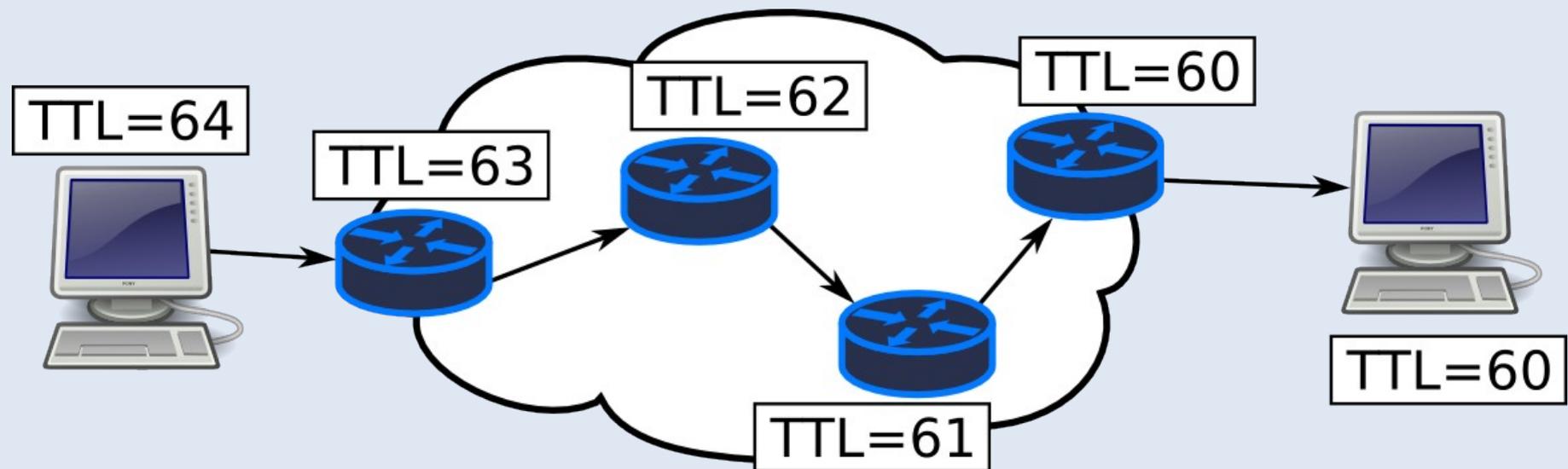
*adresse* : adresse IP destinataire

-g : teste un subnet complet

# Réseau TCP/IP

## Diagnostic : routage

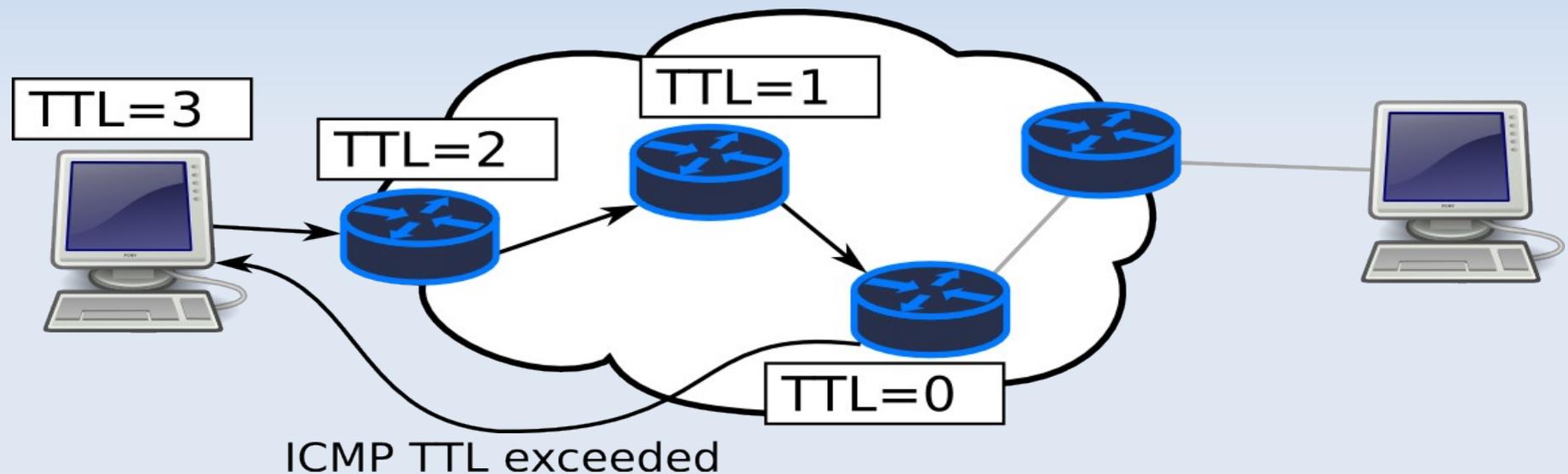
- Lorsqu'un routeur transmet un paquet d'une interface vers une autre, il décrémente le champ TTL



# Réseau TCP/IP

## Diagnostic : routage

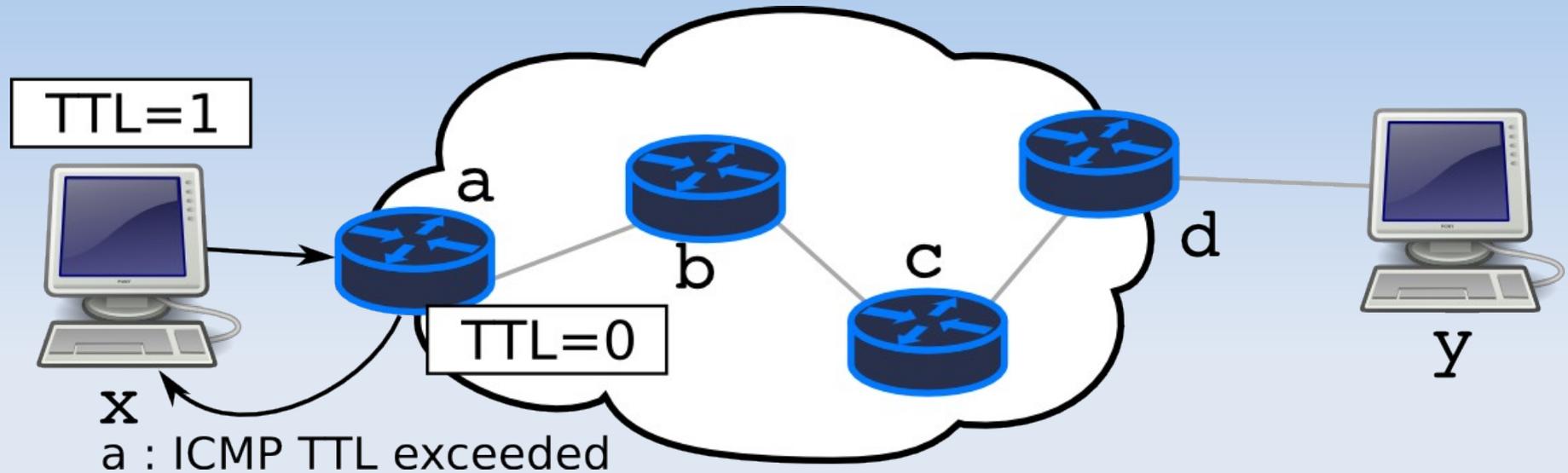
- Si après la décrémentation, le TTL est à zéro, le routeur détruit le paquet avertit l'expéditeur



- Les outils de la famille de traceroute exploitent cette particularité du protocole

# Réseau TCP/IP

## Diagnostic : routage



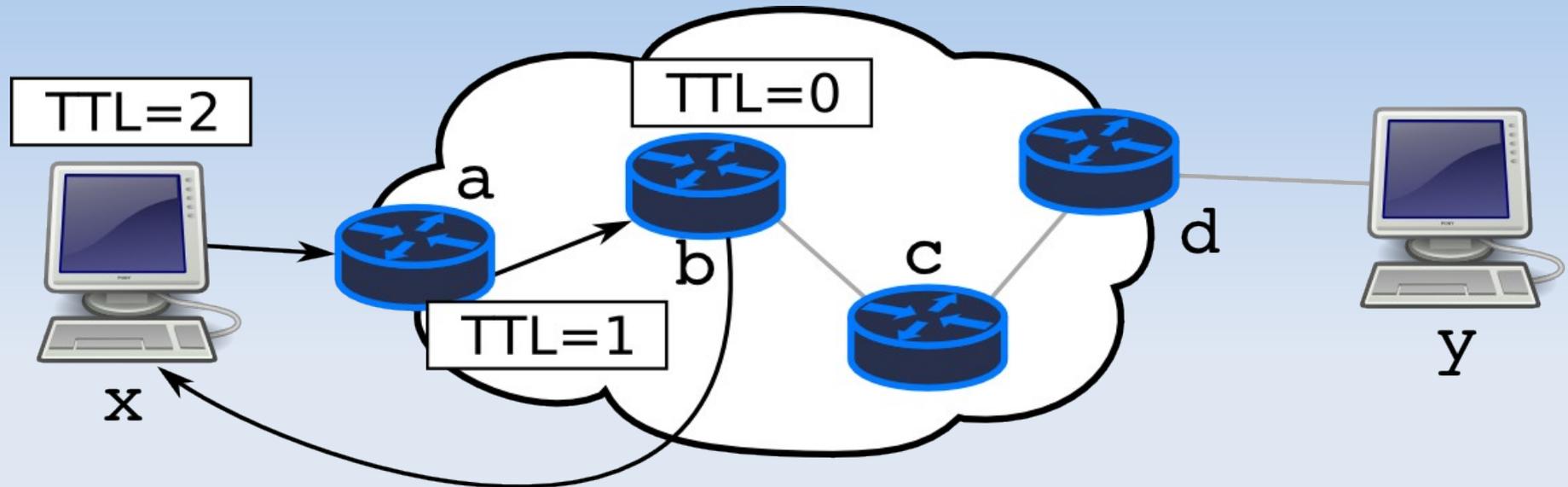
```
user@x:~$ tracepath y
```

```
1:  x (192.168.0.1)           0.188ms
```

```
1:  a (192.168.0.254)        22.844ms
```

# Réseau TCP/IP

## Diagnostic : routage

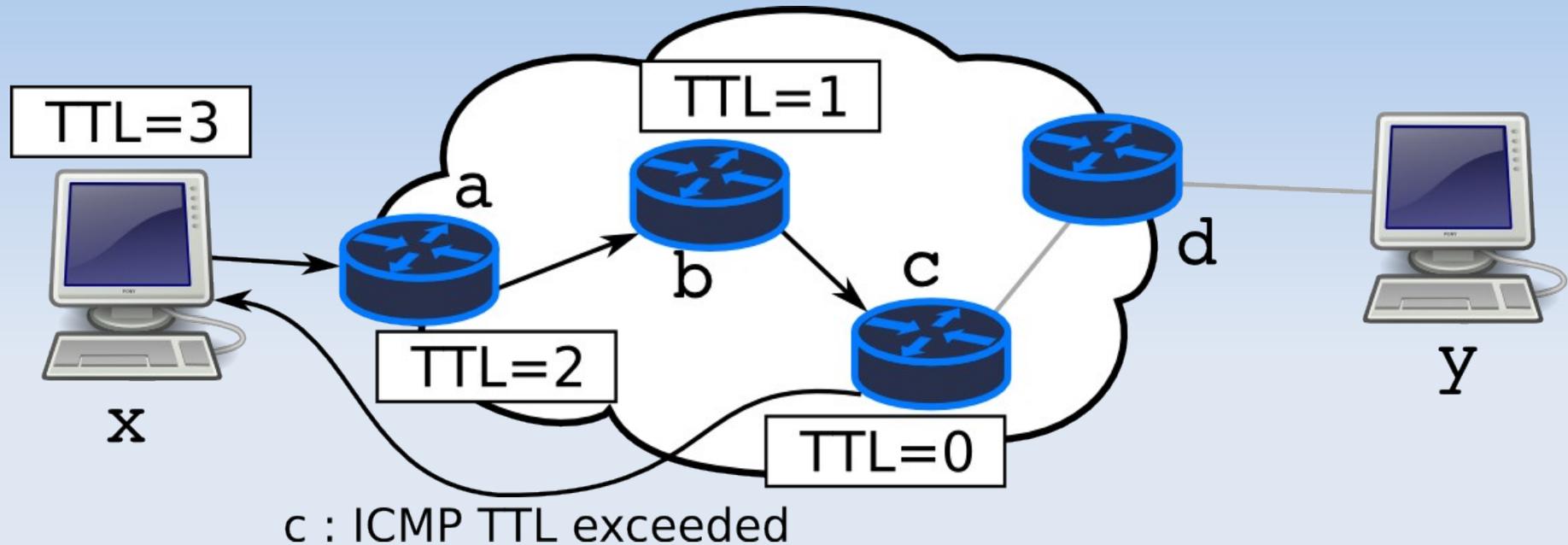


```
user@x:~$ traceroute y
```

1:	x (192.168.0.1)	0.188ms
1:	a (192.168.0.254)	2.84ms
2:	b (84.31.2.45)	22.74ms

# Réseau TCP/IP

## Diagnostic : routage

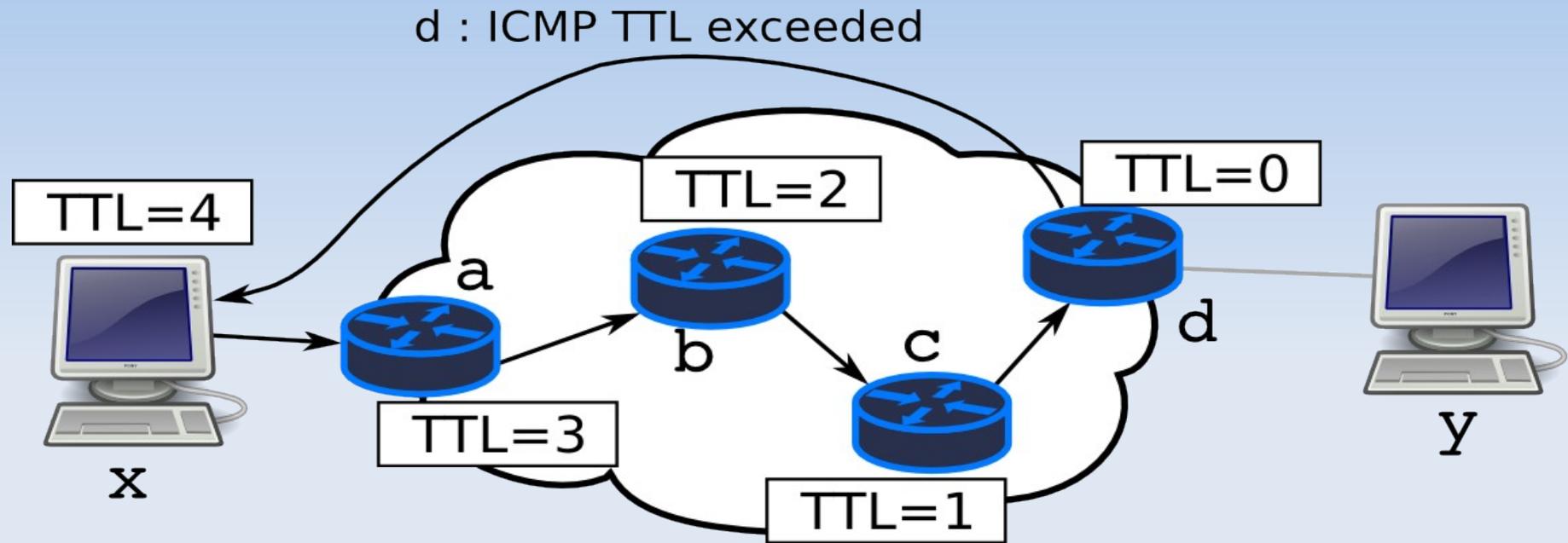


```
user@x:~$ traceroute y
```

1:	x (192.168.0.1)	0.188ms
1:	a (192.168.0.254)	2.84ms
2:	b (84.31.2.45)	22.74ms
3:	c (212.27.55.126)	28.41ms

# Réseau TCP/IP

## Diagnostic : routage



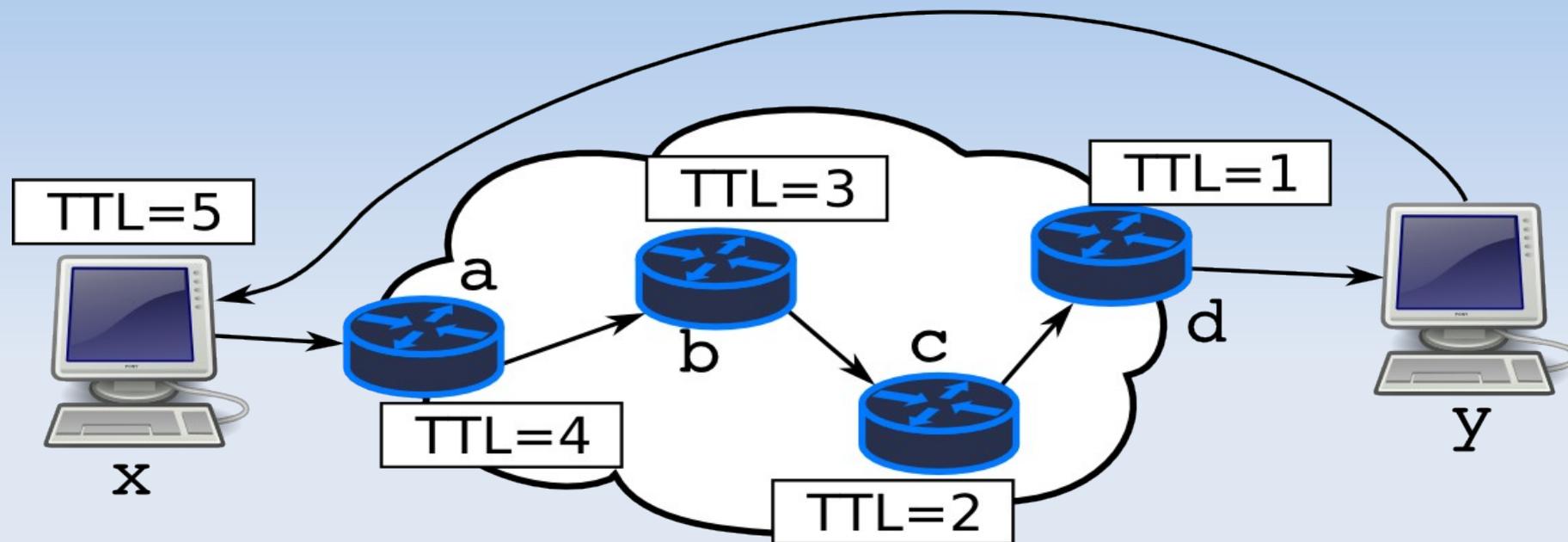
```
user@x:~$ tracepath y
```

1:	x (192.168.0.1)	0.188ms
1:	a (192.168.0.254)	2.84ms
2:	b (84.31.2.45)	22.74ms
3:	c (212.27.55.126)	28.41ms
4:	d (193.251.240.45)	31.44ms

# Réseau TCP/IP

## Diagnostic : routage

y : ICMP port unreachable



```
user@x:~$ traceroute y
```

```
1:  x (192.168.0.1)           0.188ms
1:  a (192.168.0.254)        2.84ms
2:  b (84.31.2.45)          22.74ms
3:  c (212.27.55.126)       28.41ms
4:  d (193.251.240.45)     31.44ms
5:  y (194.199.76.199)    79.319ms  reached
```

# Réseau TCP/IP

## Diagnostic : DNS

- nslookup et dig sont en compétition sur les diagnostics DNS
- nslookup est très largement disponible et possède un mode interactif
- dig a une syntaxe plus souple, et permet d'effectuer plusieurs requêtes simultanées
- Les deux outils permettent d'obtenir des réponses sur des enregistrements précis :

**a** (Address) : adresse IP de l'hôte

**mx** (Mail eXchanger) : serveur de messagerie entrant pour le domaine

**ns** (Name Server) : serveur de nom pour le domaine

**ptr** (Pointer) : résolution du nom depuis l'IP

# Réseau TCP/IP

## Diagnostic : DNS



- nslookup en ligne de commande  
`nslookup [-type=type] nom`  
(le type est **a** par défaut)
- nslookup sans argument passe en mode interactif
  - `server serveur` : change le serveur DNS utilisé
  - `set type=type` : sélectionne le type d'enregistrement voulu

**a** (Address) : adresse IP de l'hôte

**mx** (Mail eXchanger) : serveur de messagerie entrant pour le domaine

**ns** (Name Server) : serveur de nom pour le domaine

**ptr** (Pointer) : résolution du nom depuis l'IP

# Réseau TCP/IP

## Diagnostic : DNS



`dig [@server] type nom [type nom]`

(le type est **a** par défaut)

- `type` : le type d'enregistrement voulu
- `@server` : serveur de nom à interroger
- `dig` est assez verbeux : `grep -v '^;'` s'impose
- l'option `-x` permet d'effectuer une résolution inverse

**a** (Address) : adresse IP de l'hôte

**mx** (Mail eXchanger) : serveur de messagerie entrant pour le domaine

**ns** (Name Server) : serveur de nom pour le domaine

**ptr** (Pointer) : résolution du nom depuis l'IP



## Diagnostiquer les connexions

- netstat permet d'obtenir la liste de toutes les "sockets" ouvertes
  - une socket est un descripteur de fichier utilisé pour une communication entre processus (IPC)
  - les connexions TCP ainsi que les sockets d'écoute UDP sont listés par netstat

```
netstat -tunap
```

```
| | | | |__liste les process  
| | | |__liste même les sockets en écoute(all)  
| | |__n'effectue pas la résolution de noms  
| |__liste les sockets UDP  
|__liste les sockets TCP
```

# Réseau TCP/IP

## Internet texte



- Linux dispose d'un nombre conséquent d'outils internet 'texte'
  - HTTP/FTP : `wget`, `curl`, `lynx`, `elinks`, `links`
  - FTP : `ftp`, `ncftp`
  - SMTP : `mail(x)`
  - POP/IMAP : `pine`, `mutt`
  - MMS : `mimms`
  - IRC : `irc`
  - ... et `talk`, `finger`, `whois`, ...

# Réseau TCP/IP

## Internet texte



Ces outils permettent une grande automatisation des tâches liées au web :

- obtention de données, éventuellement filtrées  
`elinks -dump http://slashdot.org/ | grep -i tux`
- suivi de modifications de pages web (  
`elinks -dump www.kernel.org | diff index.html -`
- téléchargements avec reprise de transfert  
`wget -nd -c ftp://ftp.iso.fr/pub/ubuntu-6.10.iso`
- créations de miroirs  
`wget -m` (ou mieux : `rsync`)
- envoi de mails en ligne de commande (`cat | mail`)  
`elinks -dump www.site.org || logger -plocal0.err "site web hs" && echo "site web hs" | mail -s "etat du site" me@me.org`

Exemple : Faire sa propre newsletter avec les dernières nouvelles du kernel ?  
Simple comme :

```
lynx -dump http://www.kernel.org | mail -s "News" me@me.org
```

# Réseau TCP/IP

## Configuration au boot

- La configuration réseau n'est pas standard
  - chaque distribution possède une implémentation particulière
  - comme toujours, vous êtes libre de développer vos propres scripts si l'existant n'est pas adapté
- Configuration commune
  - `/etc/resolv.conf` : configuration du DNS
  - `/etc/hostname` : définition du nom d'hôte (parfois `/etc/HOSTNAME`)
  - `/etc/hosts` : fichier de correspondance ip/hôte
- La lecture des scripts de boot SysV est le meilleur moyen de déterminer le processus de configuration

# Réseau TCP/IP

## Configuration Debian/Ubuntu



- /etc/network/interfaces : configuration des interfaces

```
auto eth0 eth1
```

```
iface eth0 inet dhcp
```

```
iface eth1 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.254
```

- /etc/iftab : correspondance adresse mac/device

```
eth0 mac 00:15:c5:ab:dc:3b arp 1
```

```
eth1 mac 00:15:c5:ab:dc:3c arp 1
```

# Réseau TCP/IP

## Configuration RedHat/Fedora

- */etc/sysconfig/network* : paramètres généraux, passerelle par défaut

```
NETWORKING=yes  
HOSTNAME=server.home.lan  
GATEWAY=192.168.0.254
```

- */etc/sysconfig/network-scripts/ifcfg-<int>* : configuration ip de l'interface

```
DEVICE=eth0  
BOOTPROTO=static  
HWADDR=00:08:02:47:17:71  
NETMASK=255.255.255.0  
ONBOOT=yes  
TYPE=Ethernet
```

- */etc/sysconfig/network-scripts/route-<int>* : routes associées à l'interface

```
192.168.100.0/24 via 192.168.0.250
```

# Réseau TCP/IP

## Super-serveur xinetd

- xinetd (et tous les démons type "inetd") est un "super serveur" :
  - il s'intercale entre le client et le serveur
  - il permet de recevoir les connexions pour des serveurs TCP et UDP
  - une fois la connexion reçue, il va démarrer le serveur et lui passer la connexion
- une instance de xinetd permet de recevoir les connexions destinées à plusieurs serveurs
  - économie de ressources / latence accrue
- xinetd offre en plus des fonctionnalités de contrôle de ressources et de filtrage

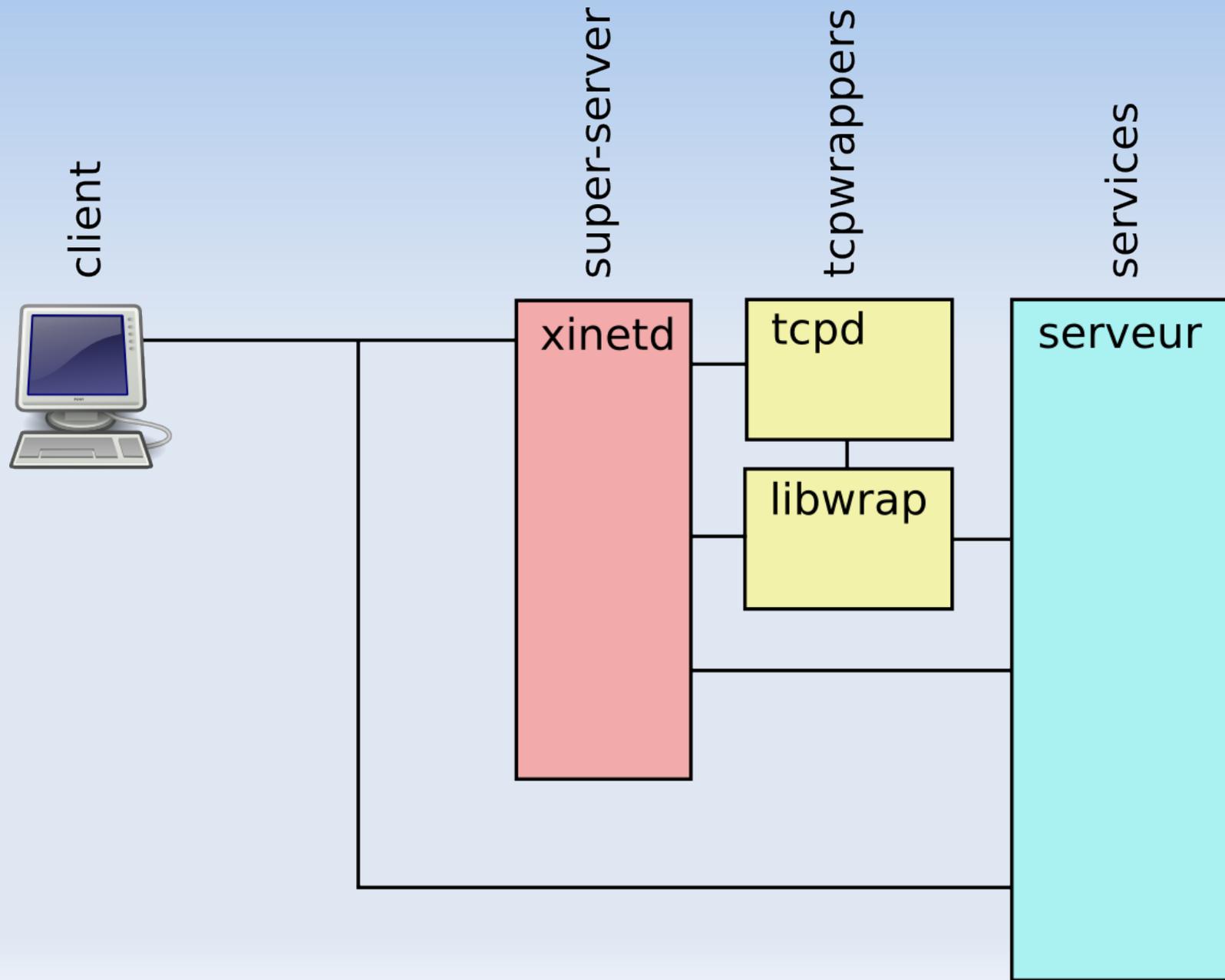
# Réseau TCP/IP

## tcpwrappers

- Mécanisme de filtrage pour l'accès aux serveurs
- Permet de réagir aux tentatives frauduleuses en lançant des scripts
- Les serveurs doivent explicitement supporter ce mécanisme (compilés avec libwrap), ou être lancés via xinetd (en utilisant tcpd)
- configuration dans */etc/hosts.allow* (couples serveur/client acceptés) et */etc/hosts.deny* (couples serveurs rejetés)
  - ➔ tombe en désuétude : de moins en moins de serveurs sont compilés avec libwrap et xinetd offre plus de possibilités

# Réseau TCP/IP

## inetd & tcpwrappers



# Réseau TCP/IP

## netfilter : filtrage avec iptables

- netfilter est le sous ensemble du kernel responsable du filtrage de paquets
  - filtrage selon caractéristiques niveau 2 (interface, mac)
  - filtrage selon les champs transport (ports, flags tcp)
  - filtrage selon des champs réseau (source, dest, tos, ttl, protocole transport, ...)
  - filtrage selon taux d'arrivée (rate limiting)
- grâce à netfilter, il est possible de faire subir les sévices les plus extrêmes aux paquets circulant dans le kernel
  - réécriture d'une partie du paquet (adresses IP, ports, ttl, ...)
  - rejet
  - destruction
  - notification (log)
  - traitement en espace utilisateur
- netfilter est configuré par des règles, constituées de chaînes et conditions et de cibles, et regroupées dans des tables

# Réseau TCP/IP

## netfilter : tables

- les *tables* sont des ensembles regroupant des règles ayant une vocation particulière :
  - **mangle** : table dans laquelle les paquets sont généralement "triturerés" : modification de ttl, marquage pour l'application de QoS, etc...
  - **nat** : table dans laquelle les translations d'adresse sont opérées (nat source, nat destination)
  - **filter** : table dédiée au filtrage de paquet, décide du sort de chaque paquet transitant dans le kernel
- la table filter est la table utilisée par défaut lorsque la commande `iptables` est invoquée

# Réseau TCP/IP

## netfilter : chaînes

- les chaînes sont des lieux "virtuels" traversés par les paquets :
  - **INPUT** : chaîne traversée par les paquets entrants et à destination de la machine
  - **OUTPUT** : chaîne traversée par les paquets sortant, émis par cette machine
  - **FORWARD** : chaîne traversée par les paquets routés par la machine (i.e. entrant par une interface et sortant par un autre)
  - **PREROUTING** : chaîne traversée par les paquets reçus du réseau
  - **POSTROUTING** : chaîne traversée par les paquets émis sur le réseau après la décision de routage
  - *chaines utilisateurs* : vous êtes libres de créer vos propres chaînes dans une table

# Réseau TCP/IP

## netfilter : cibles

- les cibles déterminent ce que l'on fait du paquet
- certaines cibles sont non-terminales (*returning targets*) : le traitement du paquet continue après cette cible
- d'autres sont terminales (*non-returning targets*) : le traitement du paquet s'arrête avec la dernière cible
  - **ACCEPT** : accepte le paquet
  - **DROP** : détruire le paquet en silence
  - **REJECT** : rejeter le paquet avec un message à l'expéditeur (code ICMP ou TCP reset)
  - **SNAT** : effectue une translation d'adresse source
  - **MIRROR** : retour à l'expéditeur !
  - **MARK** : mettre une marque sur le paquet pour utilisation ultérieure
  - **LOG** : écrire une ligne via syslog
  - **QUEUE** : traitement du paquet en espace utilisateur
  - ... quelques autres dizaines...

# Réseau TCP/IP

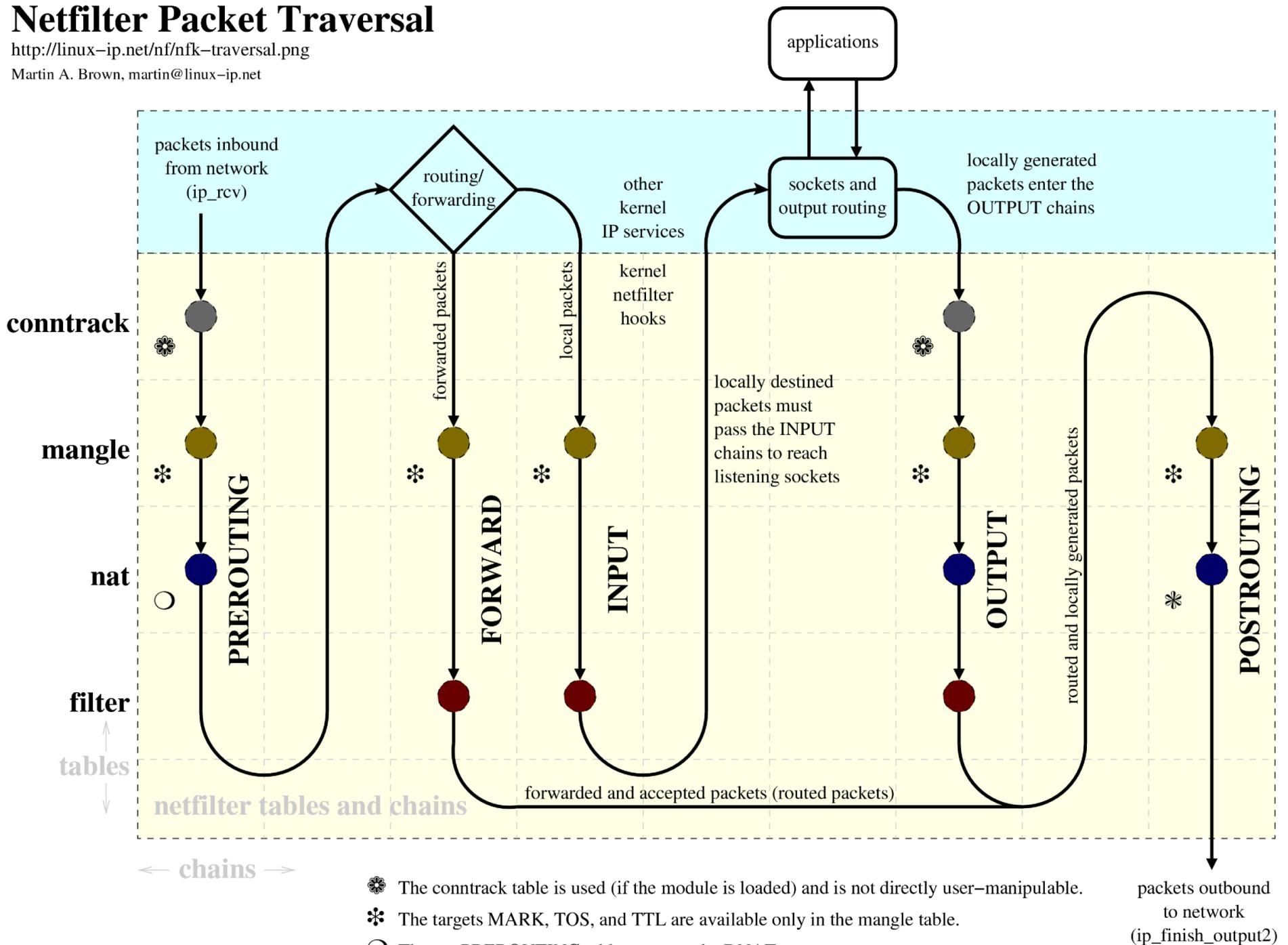
## netfilter : politiques

- chaque chaîne a une politique par défaut
- cette politique sera appliquée si aucune règle n'a attrapé un paquet auparavant
- les politiques possibles sont :
  - ACCEPT : tout est accepté
  - DROP : tout est silencieusement rejeté
- il est **très** fortement conseillé d'utiliser DROP comme politique par défaut
  - tout fermer par défaut
  - n'ouvrir que le nécessaire

# Netfilter Packet Traversal

<http://linux-ip.net/nf/nfk-traversal.png>

Martin A. Brown, martin@linux-ip.net



cf. <http://www.docum.org/qos/kptd/>

cf. [http://open-source.arkoon.net/kernel/kernel\\_net.png](http://open-source.arkoon.net/kernel/kernel_net.png)

cf. <http://iptables-tutorial.frozentux.net/>

# Réseau TCP/IP



## iptables : usage

- Déterminer la politique par défaut  
`iptables -P <CHAINE> <POLITIQUE>`  
`iptables -P INPUT DROP`
- Insérer une règle de filtrage (début de chaîne)  
`iptables -I <CHAINE> ..règle.. -j <CIBLE>`
- Ajouter une règle de filtrage (en fin de chaîne)  
`iptables -A <CHAINE> ..règle.. -j <CIBLE>`
- Supprimer une règle  
`iptables -D <CHAINE> ..règle.. -j <CIBLE>`
- Lister les règles  
`iptables -L -v`
- Vider les règles d'une chaîne  
`iptables -F <CHAINE>`
- Pour les opérations sur d'autres tables que *filter*, il faut spécifier '-t table'  
`ip -t nat -A FORWARD ...`

# Réseau TCP/IP



## iptables : spécifications de règles

- La règle permet de tester des caractéristiques du paquet
- Les options prennent la forme négative si elles sont précédées par ' ! '
- Quand ces options apparaissent dans une même règle, un ET logique est effectué
  - s : IP source
  - d : IP destination
  - p : protocol
  - i : interface d'entrée du paquet
  - o : interface de sortie du paquet
  - m *MOD* : charge le module *MOD*
  - sport : port source (modules tcp, udp)
  - dport : port destination (modules tcp, udp)
  - ...

# Réseau TCP/IP

## iptables : exemples

```
iptables -P INPUT DROP
```

→ politique par défaut : on jette

```
iptables -A INPUT -p tcp -m tcp -s 10.1.1.0/24 --dport 22 -j  
ACCEPT
```

→ on accepte les connexions tcp sur le port 22 (ssh) si elles viennent du réseau 10.1.1.0/24

```
iptables -A INPUT -p udp -m udp -s ! 10.1.1.23 --dport 53 -j  
ACCEPT
```

→ on accepte les connexions udp sur le port 53 (dns) si elles ne viennent *pas* de la machine 10.1.1.23

```
iptables -A INPUT -p tcp -m tcp -i lo --dport 3306 -j ACCEPT
```

→ on accepte les connexions tcp vers le port 3306 (mysql) uniquement si elles arrivent de l'interface loopback (cette machine)

```
iptables -A INPUT -p tcp -m tcp -i ! lo --dport 3306 -j LOG
```

→ on se plaint à syslog si une connexion arrive sur notre serveur MySQL de l'extérieur

```
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --  
limit 5/second --limit-burst 5 -j ACCEPT
```

→ on limite le traitement des echo requests icmp

```
iptables -A INPUT -m multiport -p tcp -s 10.1.1.1 --dports  
smtp,imap,pop3 -j ACCEPT
```

→ le module multiport permet de 'matcher' plusieurs ports en une règle

# Réseau TCP/IP

## Paramètres particuliers

- Activer le routage :  
`echo 1 > /proc/sys/net/ipv4/ip_forward`
- Logguer les martiens  
`echo 1 > /proc/sys/net/ipv4/conf/default/log_martians`
- Refuser les icmp redirects  
`echo 0 > /proc/sys/net/ipv4/conf/default/accept_redirects`
- Rejeter les paquets spoofés (avec des adresses locales)  
`echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter`
- Tous les paramètres réseau sont décrits dans  
`/usr/src/linux/Documentation/networking/ip-sysctl.txt`
- `sysctl` peut aussi être utilisé pour modifier ces valeurs  
`/sbin/sysctl -w net.ipv4.ip_forward=1`
- Au boot, le système applique les valeurs spécifiées dans  
`/etc/sysctl.conf`

# Réseau TCP/IP

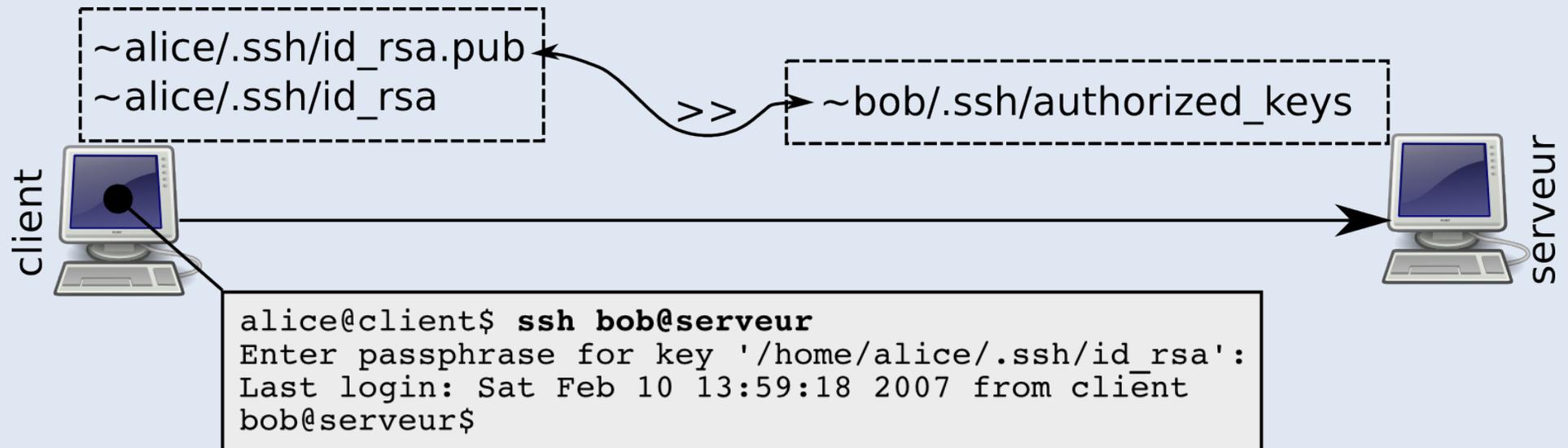
## telnet & ssh

- `telnet` permet de se connecter à un port TCP distant
  - pour utiliser un shell (protocole telnet)
  - pour "parler" un autre protocole (http, smtp, ...)
- `ssh` chiffre tout le trafic entre deux hôtes, et permet :
  - d'ouvrir un shell sur une machine distante
  - de copier des fichiers entre machines (`scp`)
  - de rediriger des ports
  - ...
- pour les connexions distantes, il faut préférer systématiquement `ssh` à `telnet` si possible

# Réseau TCP/IP

## ssh : authentication

- ssh a une multitude de possibilités pour authentifier les utilisateurs
- il est généralement plus sûr et plus simple d'utiliser l'*authentication à clef publique*



# Réseau TCP/IP

## Sécurité réseau élémentaire

- Services ouverts
  - netstat permet de connaître la liste des services ouverts
  - ➔ **ne démarrez que les services (démons) nécessaires**
- Services ouverts sur des interfaces
  - des services peuvent n'être utiles qu'aux serveurs locaux (exemple : MySQL pour apache)
  - netstat permet de lister les interfaces sur lesquelles un service est disponible (un port est ouvert)
  - ➔ **n'ouvrez les services que sur les interfaces adéquates**
- Filtrez toujours
  - il est toujours important de filtrer, même si un service n'est pas ouvert sur une interface
  - filtrer peut permettre de limiter les dénis de service
  - le filtrage est un filet de sécurité en cas d'erreur de configuration
  - ➔ **utilisez iptables, et utilisez des politiques DROP par défaut**

# Interfaces graphiques



# Interfaces graphiques

## Xwindow, XFree86, X.org

- Xwindow est l'infrastructure de base des interfaces graphiques sous unix depuis 1984
- L'interface graphique est complètement optionnelle, et parfaitement inutile sur des serveurs
- Xwindow est un protocole client serveur :
  - l'affichage d'une application est indépendant de son exécution
  - il peut être déporté à travers un réseau
  - il peut aussi être transporté via SSH
- Aujourd'hui l'implémentation la plus répandue sous linux est x.org, issue de Xfree86

# Interfaces graphiques

Gnome, KDE, Xfce, e, ...

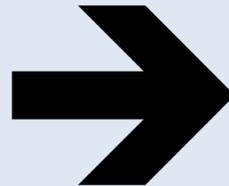
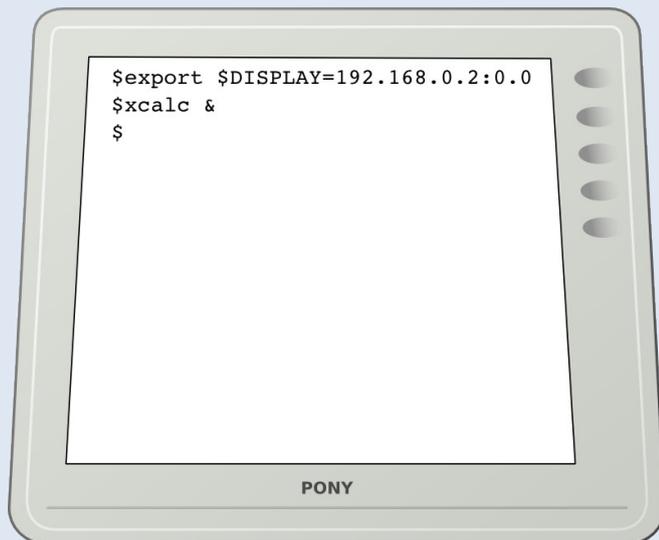
- Les interfaces graphiques sont implémentées par dessus le serveur X (x.org)
  - Gestionnaire de fenêtres (le "look") : gèrent l'aspect purement mécanique : iconification, bordures, décorations, etc...
    - blackbox, fvwm, windowMaker, fluxbox, twm
  - Gestionnaires de bureau (le "feel") : gèrent la session, les associations mime, les bus entre applications, les dialogues standard, les icônes, ...
    - Gnome, KDE, Xfce,

# Interfaces graphiques

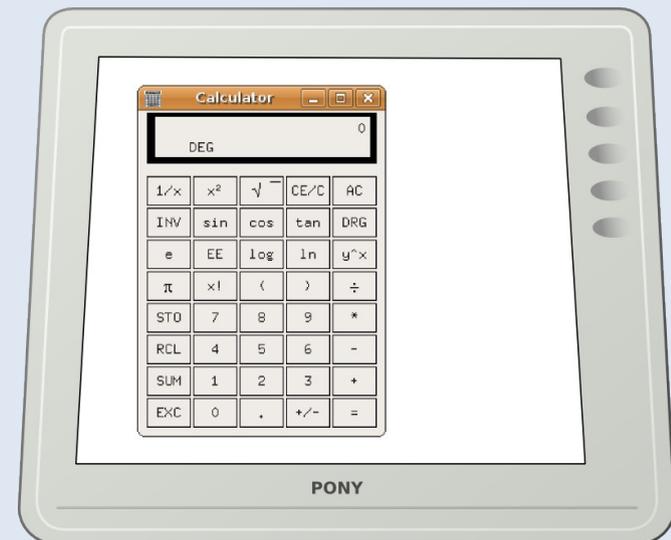
## \$DISPLAY

- La variable d'environnement \$DISPLAY permet définir le serveur X utilisé pour l'affichage
- Des terminaux spécialisés (terminaux X) sans disques durs ni CPU utilisent exclusivement cette fonctionnalité
- Xwindow livre par défaut la prise en main à distance depuis 1984...

192.168.0.1



192.168.0.2



# Questions ?

```
#!/bin/sh  
  
read Questions  
test $Questions || exit
```

*Unless you have a hundred unanswered questions in your mind  
you haven't read enough...*

*djb*

# Etudes de cas



# Etudes de cas

## A la carte

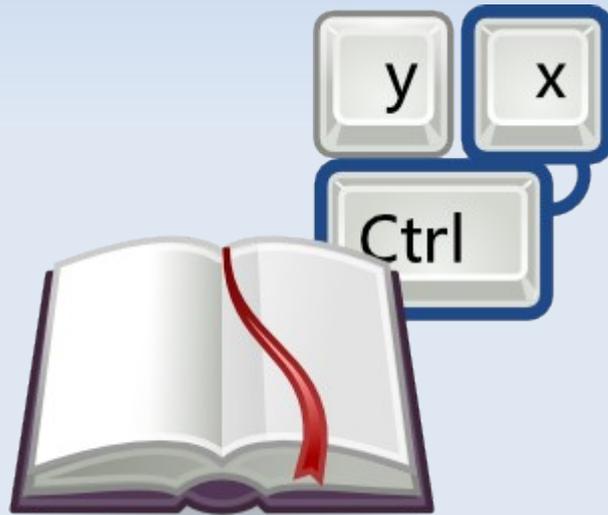
### Configuration

- **Filtrage niveau 7**
  - compiler un kernel et iptables et mettre en place du filtrage niveau 7
- **xinetd**
  - mettre en place xinetd
  - ouvrir les services echo et chargen
- **Filtrage**
  - filtrer tous les services
  - ouvrir chargen
- **Déploiement de ssh**
  - déterminer les causes d'une surcharge en processus
  - remédier au problème
- **Looping back**
  - mettre en place un filesystem chiffré en loopback
  - mettre en place un filesystem sur RAID en loopback
  - mettre en place un filesystem LVM2 en loopback
  - mettre en place une combinaison (deux ou trois ensemble)
- **L'attaque des clones**
  - cloner un système avec dd
- **Synchroniser des répertoires distants**
  - scp
  - tar & ssh
  - rsync
- **fuser : tout monter**
  - compilation de curlftps
  - monter un filesystem depuis un serveur ftp
  - faire unscript type SysV pour le montage

### Urgences

- **Out of swap space**
  - rajouter du swap à la volée par un swapfile
  - rajouter du swap à la volée depuis une partition
- **Too much processes**
  - déterminer les causes d'une surcharge en processus
  - remédier au problème
- **Out of memory**
  - déterminer la cause du manque de mémoire
  - remédier au problème
- **Filesystem full**
  - trouver d'ou vient le problème
  - mettre en place des quotas pour remédier au problème
- **Disque HS**
  - merci le RAID1
  - changer le disque défaillant
  - tester le disque défaillant
- **Au voleur, mon init !**
  - init à disparu : booter le système et réparer le problème
  - init à disparu : booter sur un CD live et réparer le problème
  - trouver, si possible, le coupable
- **Pirates !**
  - vérifier l'intégrité d'un système suspect
  - trouver la faille
  - quels enseignements ?

# Références



# Référence

## Raccourcis bash



**CTRL** + **r** : recherche dans l'historique

**CTRL** + **a** : début de ligne

**CTRL** + **e** : début de ligne

**CTRL** +  : mot précédent

**CTRL** +  : mot suivant

**CTRL** + **k** : coupe de la position courante jusqu'à la fin de la ligne

**CTRL** + **y** : colle ce qui a été précédemment coupé

**CTRL** + **d** : efface le caractère à droite du curseur

**CTRL** + **t** : transpose le caractère sous le curseur avec le caractère suivant

**ESC** + **d** : efface le mot à droite du curseur

**ESC** + **t** : transpose le mot précédent avec le mot suivant

# Référence

## Raccourcis less



**ESPACE** : avancer d'une page

 : avancer d'une ligne

**n** +  : avancer de *n* lignes

**b** : reculer d'une page

**n** + **b** : reculer de *n* lignes

**q** : quitter

**/** : rechercher

**n** : occurrence de recherche suivante

**>** : va à la fin du fichier

**<** : va au début du fichier

**!** + *cmd* : exécute la commande *cmd* dans un shell

**v** : edite le fichier en cours dans `$VISUAL` ou `$EDITOR`

# Référence

## Complétion et historique bash



### Complétion

TAB

: complète la saisie en cours

TAB

+

TAB

: affiche une liste des complétion possibles

### Historique

CTRL

+

r

*texte*

: recherche *texte* dans l'historique

↑

: commande précédente

↓

: commande suivante

!

*nombre* : exécute la commande numéro *nombre*

!

*texte* : exécute la dernière commande débutant par *texte*

\$HISTFILE : fichier historique (par défaut ~/.bash\_history)

\$HISTFILESIZE : *taille du* fichier historique (par défaut 500)

# Référence

## Job control



**CTRL** + **z** : suspension de la tâche en cours

**CTRL** + **c** : termine le processus à l'avant plan

`commande &` : exécute commande directement en arrière plan

`nohup commande &` : exécute commande directement en arrière plan et détache le terminal

`jobs` : affiche la liste des jobs en cours

`fg [%n]` : met le dernier job [ou job n] en avant plan

`bg [%n]` : met le dernier job [ou job n] en arrière plan

`kill [-SIGX] %n` : envoie le signal SIGTERM (ou SIGX) au job n

# Conventions typographiques

noms de commande ou de fichiers en courrier-10

**saisies utilisateur en courrier-10 gras**

```
console dans une fenêtre grise en courrier-10
```

*paramètres en italique*

STDIN en bleu (parties concernant les descripteurs de fichiers)

STDOUT en vert (parties concernant les descripteurs de fichiers)

STDERR en rouge (parties concernant les descripteurs de fichiers)



Manipulations



Raccourcis clavier

# Crédits, version, licence

## Crédits

**Tux** © Larry Ewing, <http://www.isc.tamu.edu/~lewing/linux/>

**GNU Head** © Etienne Suvasa, <http://www.gnu.org/graphics/agnuhead.html>

**Icons** : Tango Project (<http://tango.freedesktop.org>)

**Baby Pingouin** : <http://www.webweaver.nu/clipart>

**Systèmes** (p13) :

- ACME Systems <http://www.webweaver.nu/clipart>
- NASA Ames Research Center
- Cray Inc.
- IBM Corporation



# Crédits, version, licence

## Version & historique

Version 20070404\_01

20070404\_01 MB : création p245, mise à jour de graphes p250, p256 et p261

20070217\_01 MB : typos, coquilles, schémas partitionnement, notes

20070211\_01 MB : cosmétique

## Durée

Cette formation s'effectue normalement sur 4 jours.

## Licence

© Michel Blanc, Avril 2007

Ce document peut être distribué librement, selon les termes de la version 2.0 de la licence Creative Commons : Paternité - Partage sous conditions identiques (<http://creativecommons.org/licenses/by-sa/2.0/fr/deed.fr>)

### **Vous êtes libres :**

- de reproduire, distribuer et communiquer ce document au public
- de modifier ce document

### **Selon les conditions suivantes :**

- *Paternité.* Vous devez citer le nom de l'auteur original.
- *Partage des Conditions Initiales à l'Identique.* Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.
- A chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création.
- Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.

