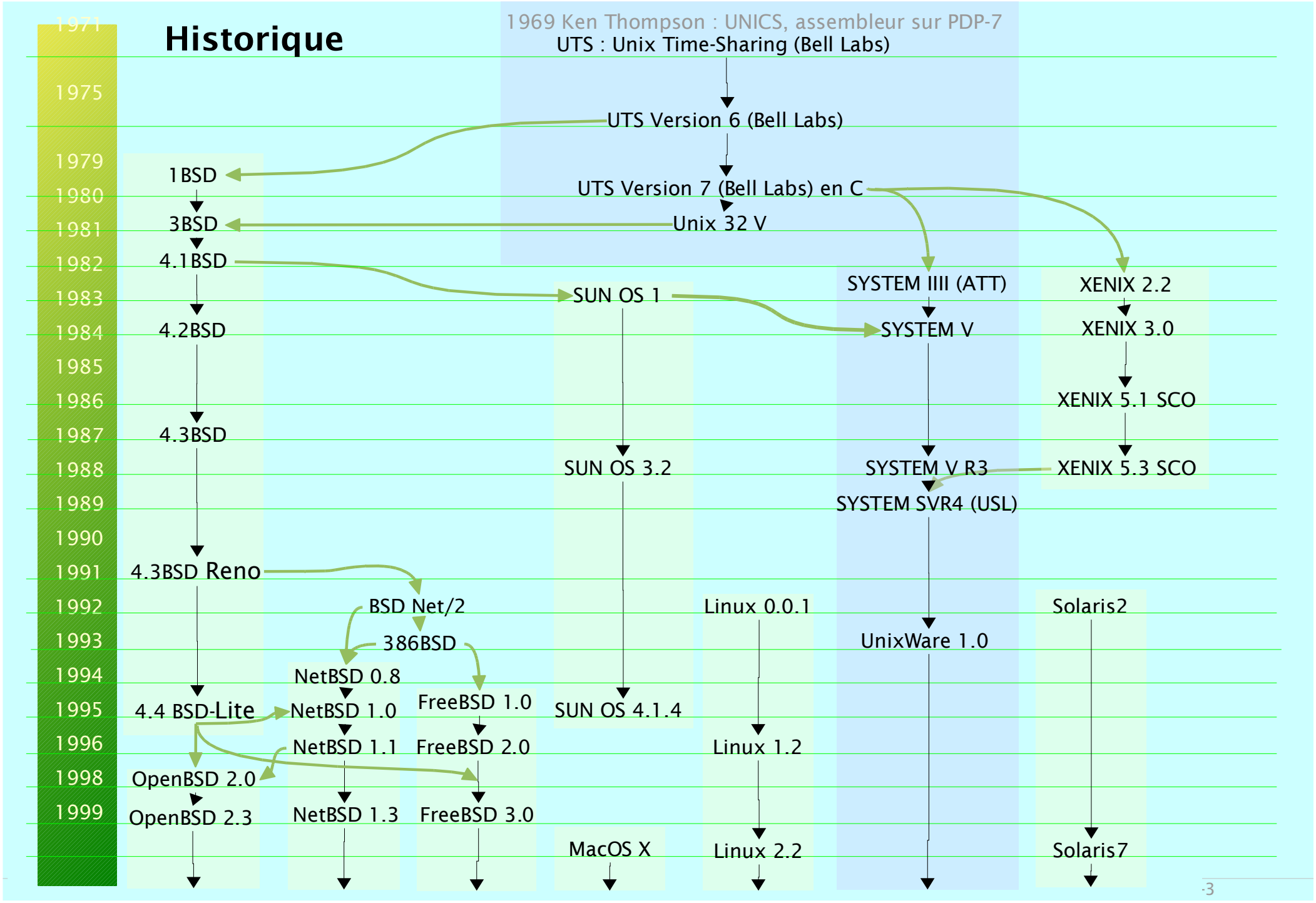
# Cours d'administration Unix

# 1 Présentation d'Unix

- Historique (AT&T, BSD, ... GNU/Linux)
- Unix propriétaire / Unix Libre
- Unix® en 2005 : The Open Group
- Concept de logiciel libre
- GNU et FSF
- Open Source
- GNU/Linux
- Les distributions de GNU/Linux

# Historique


1971
1975
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1998
1999
1969 Ken Thompson : UNICS, assembleur sur PDP-7
UTS : Unix Time-Sharing (Bell Labs)
UTS Version 6 (Bell Labs)
UTS Version 7 (Bell Labs) en C
Unix 32 V
1BSD
3BSD
4.1BSD
4.2BSD
4.3BSD
4.3BSD Reno
BSD Net/2
386BSD
NetBSD 0.8
NetBSD 1.0
NetBSD 1.1
NetBSD 1.3
4.4 BSD-Lite
OpenBSD 2.0
OpenBSD 2.3
FreeBSD 1.0
FreeBSD 2.0
FreeBSD 3.0
SUN OS 1
SUN OS 3.2
SUN OS 4.1.4
MacOS X
Linux 0.0.1
Linux 1.2
Linux 2.2
SYSTEM IIII (ATT)
SYSTEM V
SYSTEM V R3
SYSTEM SVR4 (USL)
UnixWare 1.0
XENIX 2.2
XENIX 3.0
XENIX 5.1 SCO
XENIX 5.3 SCO
Solaris2
Solaris7

# 1 Présentation d'Unix : Unix propriétaire/libre

## Les principaux Unix propriétaires

| | |
|---|---|
| IBM | AIX® |
| HP | HP-UX®<br>Tru64 UNIX® |
| SCO | UnixWare® |
| SGI | IRIX® |
| SUN | SOLARIS® |

## Les principaux Unix Libres

| | |
|---|---|
| OpenBSD | OpenBSD 3.7 Free, Functional & Secure |
| FreeBSD | http://www.freebsd.org FreeBSD FreeBSD:The Power To Serve |
| NetBSD | NetBSD |
| Mac-OS X | |
| les GNU/Linux | Linux Online! http://www.linux.org |


THE Open GROUP
Making standards work™
http://www.unix.org

### The Open Group

- Possède la marque déposée **UNIX®**
- Publie la norme « Single UNIX Specification »

(intègre les normes précédentes : X/Open Company's XPG4, IEEE's POSIX Standards et ISO C)

# 1 Présentation d'Unix : Concept de Logiciel Libre

## Origines du Logiciel Libre :

Richard Stallman

- **Richard Stallman** (chercheur au MIT, auteur de gcc, Emacs ...) énonce clairement le concept de **logiciel libre** (« free »)

  « **...** un savoir scientifique doit être partagé en le distribuant, ...
  les codes source doivent être libres d'accès ... »

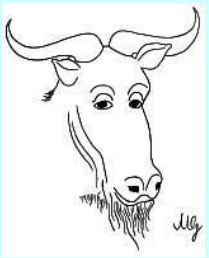
- Démarre le projet **GNU** (1984). But : re-créer un système d'exploitation complet (Unix-like), composé uniquement de logiciels libres.
- Créé la **FSF** (Free Software Fundation, 1985) pour gérer le projet GNU.
- Remarque : « Free » dans la culture hacker signifie « libre », pas nécessairement « gratuit » ou « non commercial »

# 1 Présentation d'Unix : le projet GNU

Le projet **GNU** www.gnu.org

- Principe de base : le libre accès au code source accélère le progrès en matière d'informatique car l'innovation dépend de la diffusion du code source
- La liberté au sens GNU est définie selon quatre principes :
  - liberté d'exécuter le programme, pour tous les usages
  - liberté d'étudier le fonctionnement du programme, de l'adapter à ses besoins
  - liberté de redistribuer des copies
  - liberté d'améliorer le programme et de publier ses améliorations, pour en faire profiter toute la communauté

# 1 Présentation d'Unix : la licence GPL



La licence **GPL** (General Public licence) www.gnu.orgcopyleft/gpl.html

- Autorise l'utilisateur à copier et distribuer à volonté le logiciel qu'elle protège, pourvu qu'il n'interdise pas à ses pairs de le faire aussi
- Requiert aussi que tout dérivé d'un travail placé sous sa protection soit lui aussi protégé par elle
- Quand la GPL évoque les logiciels libre, elle traite de liberté et non de gratuité (un logiciel GPL peut être vendu)
- Remarque : en anglais « free » mélange gratuité et liberté
  (R. Stallman précise bien : “Free as in speech, not as in beer")

# 1 Présentation d'Unix : le copyleft GPL

Le **copyleft** de la licence GPL www.gnu.org/copyleft

- Créé par Stallman en 1984
- Garantit les 4 libertés fondamentales pour tous les utilisateurs (artiste, informaticien, ou quiconque produit un travail soumis au droit d'auteur)
- Évite de mettre les logiciels GNU dans le domaine public (pas de protection)
- Spécifie que quiconque redistribue le logiciel, avec ou sans modifications, doit aussi transmettre la liberté de les copier et de les modifier
- Encourage et aide les programmeurs (entreprises, universités) qui veulent ajouter et/ou contribuer à des améliorations des logiciels libres.
- Un logiciel copyleft est d'abord déclaré sous copyright, puis on ajoute les conditions de distribution et les libertés légalement indissociables.

# 1 Présentation d'Unix : l'Open Source



L' **Open Source Initiative** OSI www.opensource.org

- En 1997, Eric Raymond (consulutant), Tim O'Reilly et Larry Augustin (président de VA Research), leaders de la communauté du logiciel libre, introduisent **Open Source**, pour labeliser les logiciels au code source ouvert
- **Open Source** est moins contraignant que la **GPL**
- **Open Source Definition** est un descendant direct du **Debian Social Contract**
- **Open Source** permet surtout une plus grande promiscuité lors d'un mélange de code propriétaire avec du code open source

# 1 Présentation d'Unix : GNU/Linux


**GNU/Linux :**

- C'est un **Unix** à part entière
- C'est un logiciel libre distribué sous licence GPL
- Les sources du noyau Linux sont disponibles sur **http://www.kernel.org/**
- Intègre :
  - le noyau **Linux** (clone du système Unix écrit par Linus Torvalds et toute une équipe de développeurs sur InterNet)
  - des composants logiciels libres issus du projet GNU (gcc, ...)
- conforme à la norme « Single UNIX »
- Disponible pour toutes les plateformes (PC, station,cluster, mainframe, ...)
- La plupart des éditeurs de solutions **UNIX** propriétaires intègrent **GNU/Linux** en remplacement (IBM, HP, SiliconGraphics ...)
- L'administration Linux est calquée sur UNIX System V (AT&T)

# 1 Présentation d'Unix : distributions de Linux



- www.redhat.fr
- société américaine fondée en 1994.
- célèbre pour avoir introduit le système Red hat Package Manager (RPM), de gestion de paquets logiciels


- fedora.redhat.com
- version grand publique gratuite de RedHat
- utilise le système de gestion de paquets RPM

# 1 Présentation d'Unix : distributions de Linux


- www.mandriva.com
- LA distribution française par excellence
- Très répandue dans le milieu éducatif, et en PME/PMI
- utilise le système de gestion de paquets RPM


- www.novell.com/linux/suse
- société allemande fondée à Nuremberg en 1993
- rachetée en 2003 par l'éditeur de logiciels américain Novell
- utilise le système de gestion de paquets RPM

# 1 Présentation d'Unix : distributions de Linux


- www.debian.org
- distribution issue d'un effort communautaire, le « projet Debian », et non d'une entreprise
- distribution très soignée et ingénieuse
- austère à installer et à administrer (déconseillée aux débutants ?)
- Utilise le système de gestion de paquets **debian**


- www.ubuntu-fr.org
- Distribution populaire à base débian lancée en 2004. Son nom provient d'un ancien mot bantou (langue d'Afrique), ubuntu, signifiant « Humanité aux autres», ou encore « Je suis ce que je suis grâce à ce que nous sommes tous »

# 1 Présentation d'Unix : distributions de Linux


- www.turbolinux.com
- La distribution de Linux majeure pour les pays d'Asie(particulièrement répandue en Chine)

# Cours d'administration Unix

# 2 Principes de fonctionnement d'UNIX

- Boot et lancement du noyau
- Processus ***init***
- *Services* et *démons*
- Les *runlevels*
- Scripts de lancement des services
- Exemple d'outil graphique (Mandriva)

# 2 Principes de fonctionnement d'UNIX : lancement du noyau

## Lancement du système : boot et chargement du noyau

- Au *boot* le **BIOS** exécute le **MBR** (Master Boot Reccord) situé sur le premier secteur (512 octets) du support bootable choisi (disque, CD, clef USB, ...)

- Le **MBR** :
  - scanne le disque pour trouver LA partition bootable (flag)
  - lance le ***boot loader*** (chargeur de démarrage) du secteur de boot (premier secteur) de la partition bootable
- Le ***bootloader*** :
  - charge le noyau en mémoire et l'exécute
  - charge le *ramdisk* `initrd.img` en mémoire

- 2 bootloader possibles: **Lilo** (Linux Loader)
  **Grub** (Grand Unified Bootloader)

# 2 Principes de fonctionnement d'UNIX : processus init

## Lancement du système : boot -> init

- Une fois le noyau chargé en mémoire, il lance le premier processus : `/bin/init`
- ***init*** est le père de tous les autres processus qui seront créés par l'appel system `fork()`
- ***init*** lit le fichier `/etc/inittab` pour savoir :

  - quel est le fichier à exécuter pour continuer le chargement du système
  - quel est le *runlevel* (niveau d'exécution) par défault
  - comment lancer les services pour un *runlevel* donné
  - ...

# 2 Principes de fonctionnement d'UNIX : processus init

## Exple de fichier `/etc/inittab` format des lignes `id:runlevels:action:process`


Le niveau d'exécution par défaut

Les niveaux d'exécution possibles

Action à faire sur l'évènement CTRL-ALT-DEL

Pour les niveaux 2 à 5, activer plusieurs consoles en mode caractère

```
#
# inittab    This file describes how the INIT process should set up
#        the system in a certain run-level.
#
# Default runlevel. The runlevels used by Mandrakelinux are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
... ...
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
... ...
```

# 2 Principes de fonctionnement d'UNIX : processus init

## Exple de fichier `/etc/inittab` format des lignes `id:runlevels:action:process`


Le niveau d'exécution par défaut

Les niveaux d'exécution possibles

Action à faire sur l'évènement CTRL-ALT-DEL

Pour les niveaux 2 et 3, activer plusieurs consoles en mode caractère

```
# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt, 1 is single-user, 2-5 are multi-user, 6 is reboot.
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
... ...
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

# What to do when CTRL-ALT-DEL is pressed.

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
... ...
```

# 2 Principes de fonctionnement d'UNIX : processus init

Exemples d'**actions** courantes pour le fichier `inittab` :

| | |
|---|---|
| **respawn** | Le processus est redémarré à chaque fois qu'il se termine |
| **wait** | Le processus n'est démarré qu'une seul fois |
| **boot** | Le processus est exécuté pendant le démarrage du système |
| **initdefault** | indique le mode de démarrage une fois le système opérationnel. Si aucun n'existe, init demandera un runlevel sur la console |
| **sysinit** | Le processus est exécuté pendant le démarrage du système. Il est exécuté avant les entrées **boot** ou **bootwait** |
| **ctrlaltdel** | Le processus est exécuté lorsque **init** reçoit le signal SIGINT. (appui simultané des touches CTRL+ALT+DEL) |

# 2 Principes de fonctionnement d'UNIX : services et démons

## Lancement du système : boot -> init -> modules/services

- Après le chargement du noyau, le script correspondant à **sysinit** dans fichier `inittab` est chargé :

  - Mandrake : `/etc/rc.d/rc.sysinit` (1 script ~ 1300 lignes !!)

  - Debian : `/etc/rc.d/rcS` lance les scripts `/etc/rcS.d/S??*`

- Ce script d'initialisation est chargé de 2 tâches fondamentales :

  - charger les **modules** dans le noyau (gestion des périphériques)

  - démarrer les **services** en exécutant les processus «Deferred Auxiliary Executive Monitor» (*daemons*) correspondant, en français : **démons** 

# 2 Principes de fonctionnement d'UNIX : services et démons

Exemple de **démons** :

| | |
|---|---|
| **kswapd** | swap mémoire <-> disque |
| **kflusd** | écriture physique des données |
| **nfsd** | serveur fichiers réseaux (NFS) |
| **portmap** | mise en correspondance numéro de ports TCP/IP <-> numéro de processus RPC |
| **xinetd** | super-serveur de gestion des services réseau |
| **ftpd** | serveur de transfert de fichiers (FTP) |
| **httpd** | serveur HTTP |

- Les démons peuvent implémenter :
  - un service noyau (**kswapd**, ...)
  - un service réseau (**httpd**, ...)

# 2 Principes de fonctionnement d'UNIX : les runlevels

## Lancement du système : boot -> init -> services

- Le mécanisme de démarrage des **services** est caractéristique d'une distribution (☹: incompatibilités entre distributions) :
  - Mandriva, Debian, RedHat, ... mécanisme dérivé d' «Unix System V»
  - Slackware, FreeBSD, NetBSD, ... mécanisme dérivée d' «Unix BSD»
- le répertoire `/etc/init.d` contient tous les scripts de gestion des services installés (1 service <-> 1 ou plusieurs **démon**(s))
- les lignes «`/etc/rc.d/rc x`» du fichier `/etc/inittab` déterminent le lancement des scripts pour le runlevel x
- Le *runlevel* de l'action *initdefault* est lancé par le ligne correspondante

Mandriva

```
id:5:initdefault:
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
... ...
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

# 2 Principes de fonctionnement d'UNIX : les runlevels

## Lancement du système : boot -> init -> services

- Le ***runlevel*** (numéro de 0 à 6) fixe le répertoire de démarrage des services :
  - Mandrake -> répertoires `/etc/rc.d/rc[0-6].d`
  - Debian -> répertoires `/etc/rc[0-6].d`
- `rcX.d` : contient des liens symboliques vers les scripts de gestions des services qui sont dans le répertoire :
  - `/etc/rc.d/init.d` (Mandrake, + lien symbolique vers `/etc/init.d`)
  - `/etc/init.d` (Debian)
- Les liens sont formés selon la syntaxe : `[S|K]XX<nom_du_script>`
  - **S** lance le script avec l'argument **start** (démarrage du service)
  - **K** lance le script avec l'argument **stop** (arrêt du service)
  - **XX** est un rang qui fixe l'ordre dans lequel les scripts sont lancés

## Extrait du fichier /etc/rc.d/rc

```
#! /bin/sh
... ...
# Now find out what the current and what the previous runlevel are.
Argv1="$1"
... ...
# Get first argument. Set new runlevel to this argument.
[ -n "$argv1" ] && runlevel="$argv1"
... ...

# First, run the KILL scripts.
for i in /etc/rc$runlevel.d/K* ; do
... ...
done

# Now run the START scripts.
for i in /etc/rc$runlevel.d/S* ; do
        check_runlevel "$i" || continue

        # Check if the subsystem is already up.
        subsys=${i#/etc/rc$runlevel.d/S??}
        [ -f /var/lock/subsys/$subsys -o -f /var/lock/subsys/$subsys.init ] && continue

        # If we're in confirmation mode, get user confirmation
        if [ -f /var/run/confirm ]; then
                if [ "$subsys" = dm ]; then
                    CONFIRM_DM=1
                    continue
                fi
                confirm $subsys
                case $? in
                    1) continue;;
                    2) rm -f /var/run/confirm;;
                esac
        fi
... ...
```

## Extrait du fichier /etc/init.d/rc

```
... ...
 # Get first argument. Set new runlevel to this argument.
  [ "$1" != "" ] && runlevel=$1
... ...
        # First, run the KILL scripts.
... ...
                for i in /etc/rc$runlevel.d/K[0-9][0-9]*
                do
                        # Check if the script is there.
                        [ ! -f $i ] && continue

                        # Stop the service.
                        startup $i stop
                done
        fi
        # Now run the START scripts for this runlevel.
        for i in /etc/rc$runlevel.d/S*
        do
                [ ! -f $i ] && continue
... ...
                        suffix=${i#/etc/rc$runlevel.d/S[0-9][0-9]}
                        stop=/etc/rc$runlevel.d/K[0-9][0-9]$suffix
                        previous_start=/etc/rc$previous.d/S[0-9][0-9]$suffix
                        #
                case "$runlevel" in
                        0|6)
                                startup $i stop
                                ;;
                        *)
                                startup $i start
                                ;;
                esac
        done
  fi
# eof /etc/init.d/rc
```

# 2 Principes de fonctionnement d'UNIX : Scripts de lancement

## Exemple de contenu du répertoire `/etc/rc5.d` :


```
lrwxrwxrwx 1 root root 14 Jul 20 21:46 K59dund -> ../init.d/dund*
lrwxrwxrwx 1 root root 14 Jul 20 21:46 K59hidd -> ../init.d/hidd*
lrwxrwxrwx 1 root root 14 Jul 20 21:46 K59pand -> ../init.d/pand*
lrwxrwxrwx 1 root root 14 Jul 20 21:43 S01udev -> ../init.d/udev*
lrwxrwxrwx 1 root root 19 Jul 20 21:46 S05harddrake -> ../init.d/harddrake*
lrwxrwxrwx 1 root root 17 Jul 20 22:02 S10network -> ../init.d/network*
lrwxrwxrwx 1 root root 17 Jul 20 22:02 S11portmap -> ../init.d/portmap*
lrwxrwxrwx 1 root root 16 Jul 20 22:02 S12syslog -> ../init.d/syslog*
lrwxrwxrwx 1 root root 17 Jul 20 21:43 S13partmon -> ../init.d/partmon*
lrwxrwxrwx 1 root root 17 Jul 20 21:46 S14nfslock -> ../init.d/nfslock*
lrwxrwxrwx 1 root root 14 Jul 20 21:42 S17alsa -> ../init.d/alsa*
lrwxrwxrwx 1 root root 16 Jul 20 22:03 S17ypbind -> ../init.d/ypbind*
lrwxrwxrwx 1 root root 15 Jul 20 22:02 S18sound -> ../init.d/sound*
lrwxrwxrwx 1 root root 13 Jul 20 21:44 S20xfs -> ../init.d/xfs*
lrwxrwxrwx 1 root root 20 Jul 20 21:43 S24messagebus -> ../init.d/messagebus*
lrwxrwxrwx 1 root root 19 Jul 20 21:46 S25bluetooth -> ../init.d/bluetooth*
lrwxrwxrwx 1 root root 19 Jul 20 21:44 S25haldaemon -> ../init.d/haldaemon*
lrwxrwxrwx 1 root root 15 Jul 20 22:02 S25netfs -> ../init.d/netfs*
lrwxrwxrwx 1 root root 17 Jul 20 21:47 S29numlock -> ../init.d/numlock*
lrwxrwxrwx 1 root root 12 Jul 20 21:43 S30dm -> ../init.d/dm*
lrwxrwxrwx 1 root root 14 Jul 20 21:44 S33nifd -> ../init.d/nifd*
...
lrwxrwxrwx 1 root root 13 Jul 20 21:46 S40atd -> ../init.d/atd*
lrwxrwxrwx 1 root root 14 Jul 20 21:47 S55sshd -> ../init.d/sshd*
lrwxrwxrwx 1 root root 20 Jul 20 22:02 S56rawdevices -> ../init.d/rawdevices*
lrwxrwxrwx 1 root root 18 Jul 20 22:02 S75keytable -> ../init.d/keytable*
lrwxrwxrwx 1 root root 15 Jul 20 17:52 S80spamd -> ../init.d/spamd*
lrwxrwxrwx 1 root root 17 Jul 20 21:50 S85proftpd -> ../init.d/proftpd*
lrwxrwxrwx 1 root root 15 Jul 20 22:02 S90crond -> ../init.d/crond*
lrwxrwxrwx 1 root root 14 Jul 20 21:44 S92lisa -> ../init.d/lisa*
lrwxrwxrwx 1 root root 17 Jul 20 22:02 S95kheader -> ../init.d/kheader*
lrwxrwxrwx 1 root root 11 Jul 20 21:43 S99local -> ../rc.local*
```

Démarrage du réseau

Démarrage du service Son

Démarrage de la bannière de connexion en mode graphique (**d**isplay **m**anager)

Démarrage du service anti-spam

Pour finir, Lancement du script de configuration locale

# 2 Principes de fonctionnement d'UNIX : Scripts de lancement

## Exemple de contenu du répertoire `/etc/rc5.d` :


Démarrage du réseau

Démarrage du service d'impression

Démarrage du serveur ssh

Démarrage de la bannière de connexion en mode graphique ( **g**nome **d**isplay **m**anager)

```
lrwxrwxrwx 1 root root 18 2005-09-18 20:04 S10sysklogd -> ../init.d/sysklogd
lrwxrwxrwx 1 root root 15 2005-09-18 20:04 S11klogd -> ../init.d/klogd
lrwxrwxrwx 1 root root 13 2005-09-18 20:03 S14ppp -> ../init.d/ppp
lrwxrwxrwx 1 root root 17 2005-09-18 19:10 S18portmap -> ../init.d/portmap
lrwxrwxrwx 1 root root 14 2005-09-19 09:42 S20apmd -> ../init.d/apmd
lrwxrwxrwx 1 root root 16 2005-09-18 19:11 S20dbus-1 -> ../init.d/dbus-1
lrwxrwxrwx 1 root root 17 2005-09-18 19:11 S20dirmngr -> ../init.d/dirmngr
lrwxrwxrwx 1 root root 15 2005-09-18 20:03 S20exim4 -> ../init.d/exim4
lrwxrwxrwx 1 root root 15 2005-09-18 20:03 S20inetd -> ../init.d/inetd
lrwxrwxrwx 1 root root 13 2005-09-18 19:10 S20lpd -> ../init.d/lpd
lrwxrwxrwx 1 root root 17 2005-09-18 20:02 S20makedev -> ../init.d/makedev
lrwxrwxrwx 1 root root 16 2005-09-18 20:05 S20pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 15 2005-09-18 15:16 S20rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 13 2005-09-18 19:11 S20ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 13 2005-09-18 14:57 S20vdr -> ../init.d/vdr
lrwxrwxrwx 1 root root 16 2005-09-18 22:01 S20webmin -> ../init.d/webmin
...
lrwxrwxrwx 1 root root 13 2005-09-18 21:00 S20xfs -> ../init.d/xfs
lrwxrwxrwx 1 root root 16 2005-09-18 21:00 S20xinetd -> ../init.d/xinetd
lrwxrwxrwx 1 root root 15 2005-09-18 21:07 S21aumix -> ../init.d/aumix
lrwxrwxrwx 1 root root 13 2005-09-18 19:12 S21fam -> ../init.d/fam
lrwxrwxrwx 1 root root 20 2005-09-18 19:10 S21nfs-common -> ../init.d/nfs-common
lrwxrwxrwx 1 root root 13 2005-09-18 20:04 S89atd -> ../init.d/atd
lrwxrwxrwx 1 root root 14 2005-09-18 20:03 S89cron -> ../init.d/cron
lrwxrwxrwx 1 root root 13 2005-09-18 19:17 S99gdm -> ../init.d/gdm
lrwxrwxrwx 1 root root 19 2005-09-18 20:02 S99rmnologin -> ../init.d/rmnologin
lrwxrwxrwx 1 root root 23 2005-09-18 20:02 S99stop-bootlogd -> ../init.d/stop-
bootlogd
```

# 2 Principes de fonctionnement d'UNIX : Scripts de lancement

**Lancement du système : boot -> init -> services**

- Utilitaires en mode console :

  - Debian : `update-rc.d`

  - Mandrake : `chkconfig, service`


- le script `/etc/rc.local` peut contenir des personnalisations locales qui seront lancées à la fin du processus **init**
- Pour démarrer un service sous mandrake, on peut taper :

  `service <nom_du_service> start`

  ou encore :

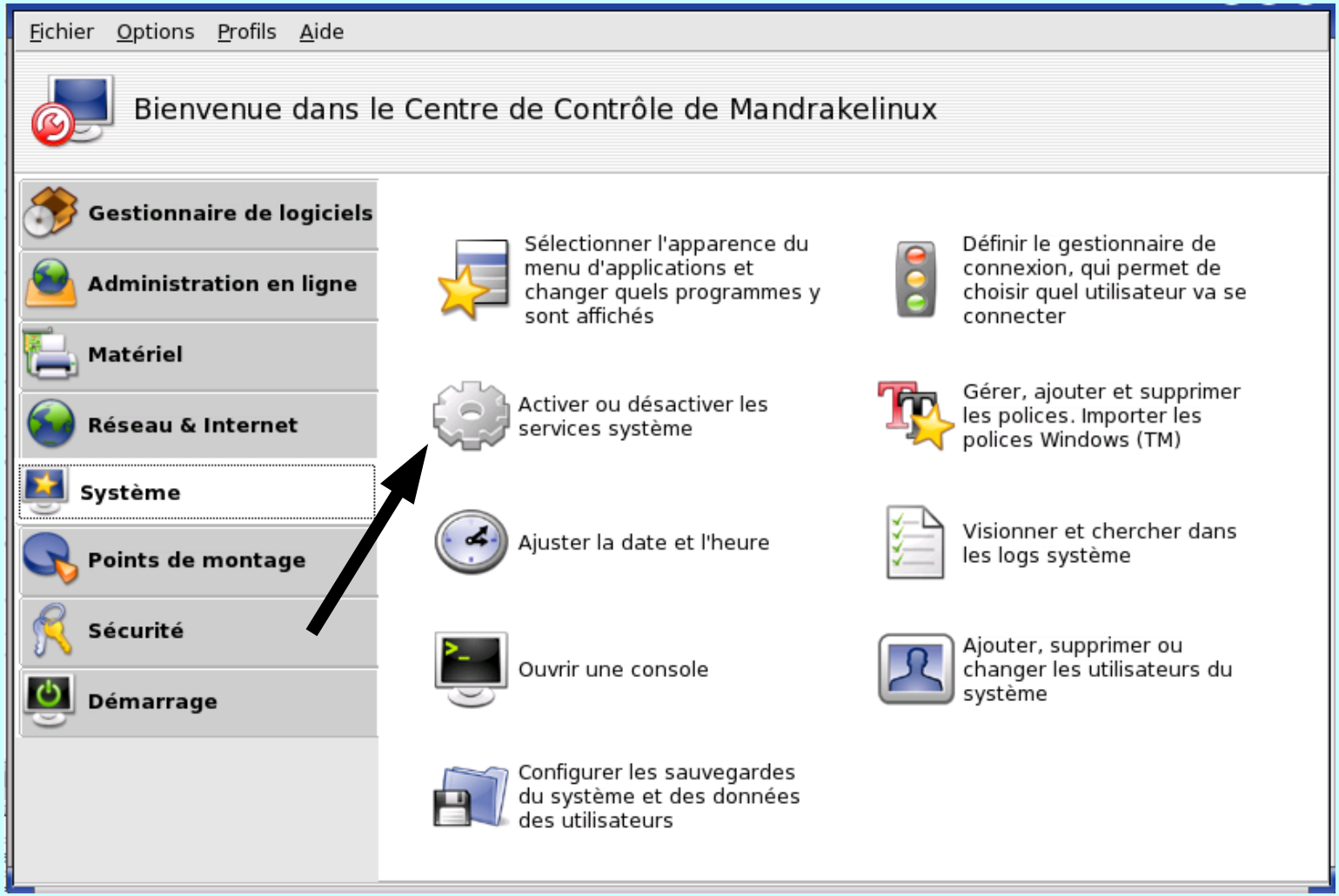
  `/etc/rc.d/init.d/<cript_correspondant_au_service>` **`action`**

  **action** : `starp | stop | restart | status | ...`

# 2 Principes de fonctionnement d'UNIX : Scripts de lancement

## Lancement du système : boot -> init -> services

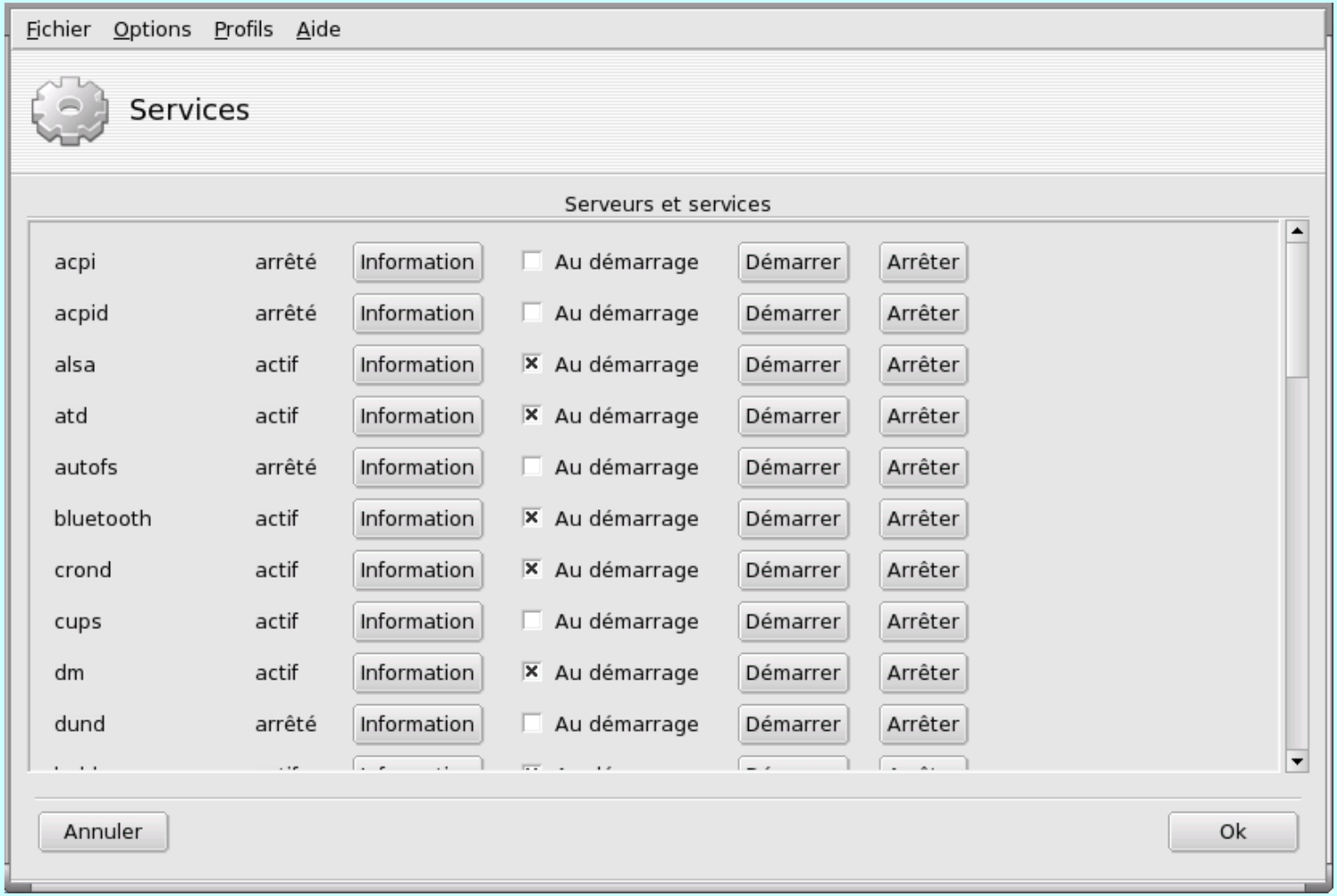
Outil graphique de gestion des services :



# 2 Principes de fonctionnement d'UNIX : Scripts de lancement

## Lancement du système : boot -> init -> services

Outil graphique de gestion des services :



# Cours d'administration Unix

# 3 Éléments d'administration Unix

Context -> architecture PC

- Les 2 modes d'administration
- Notion de « **fichier spécial** »
- **Nommage** des périphériques de boot
- **Partitionnement** des disques
- **Formatage** disque et *filesystem*
- Le « **montage** » des périphériques
- Les **gestionnaires** de paquets (**rpm** et **debian**)
- Configuration du ***bootloader*** (Lilo)

# 3 Éléments d'administration Unix : 2 modes d'administration

Unix supporte 2 modes d'administration :

« **À la main** » :

- Édition (manuelle) des fichiers de configuration
- Utilisation (manuelle) des commandes d'administration
- Utilisation (manuelle) des gestionnaires de paquets RPM ou DEBIAN
- Édition de scripts de commande (langage : shell, perl, awk, ...)

Avec des **logiciels** d'administration (graphique ou mode caractère) :

- Qui manipulent les fichiers de configuration
- Qui utilisent des commandes d'administration standard ou spécifiques
- Souvent incontournables, avec des Unix propriétaires (HP : SAM)
- Linux : **linuxconf**, **webmin**, **DrakConf**, ....

# 3 Éléments d'administration Unix : notion de « fichier Spécial »

Principe :

**sous Unix,**
**tout est fichier**

=> tous les périphériques
- disques,
- clavier, souris,
- carte son,
- ports d'E/S
- sockets réseau,
- mémoire ...

sont représentés par
un **fichier spécial**
dans le répertoire /dev

```
root@blaise cups]# ls -l /dev
total 0
crw-rw----  1 jlc  audio     14,  12 sep 28 11:36 adsp
crw-rw----  1 root video     10, 175 sep 28 11:34 agpgart
crw-rw----  1 jlc  audio     14,   4 sep 28 11:36 audio
lrwxrwxrwx  1 root root            3 sep 28 11:34 cdrom -> hdd
...
crw-rw----  1 jlc  audio     14,   3 sep 28 11:36 dsp
lrwxrwxrwx  1 root root            3 sep 28 11:34 dvd -> hdc
...
brw-rw----  1 jlc  floppy     2,   0 sep 28 11:34 fd0
brw-rw----  1 jlc  floppy     2,   1 sep 28 11:34 fd1
...
brw-rw----  1 root disk       3,   0 sep 28 11:33 hda
brw-rw----  1 root root       3,   1 sep 28 11:33 hda1
brw-rw----  1 root root       3,   2 sep 28 11:33 hda2
brw-rw----  1 root root       3,   5 sep 28 11:33 hda5
brw-rw----  1 root root       3,   6 sep 28 11:33 hda6
brw-rw----  1 jlc  cdrom     22,   0 sep 28 11:34 hdc
brw-rw----  1 jlc  cdrom     22,  64 sep 28 11:34 hdd
...
crw-r-----  1 root root       1,   1 sep 28 11:33 mem
crw-rw----  1 root root      13,  63 sep 28 11:33 mice
drwxr-xr-x  2 root root          120 sep 28 11:34 misc/
crw-rw----  1 jlc  audio     14,   0 sep 28 11:36 mixer
lrwxrwxrwx  1 root root            5 sep 28 11:34 mouse -> psaux
...
crw-rw----  1 jlc  usb       99,   0 sep 28 11:34 parport0
crw-rw----  1 root root      10,   1 sep 28 11:33 psaux
lrwxrwxrwx  1 root root            5 sep 28 11:34 psmouse -> psaux
...
brw-rw----  1 root disk       8,   0 sep 29 08:08 sda
brw-rw----  1 root disk       8,   1 sep 29 09:26 sda1
crw-rw----  1 jlc  audio     14,   1 sep 28 11:36 sequencer
crw-rw----  1 jlc  audio     14,   8 sep 28 11:36 sequencer2
...
[root@blaise cups]# _
```

Carte son

Disque dur

Lect. DVD

Lect. CD

# 3 Éléments d'administration Unix : notion de «fichier Spécial»

Attributs des fichiers spéciaux :

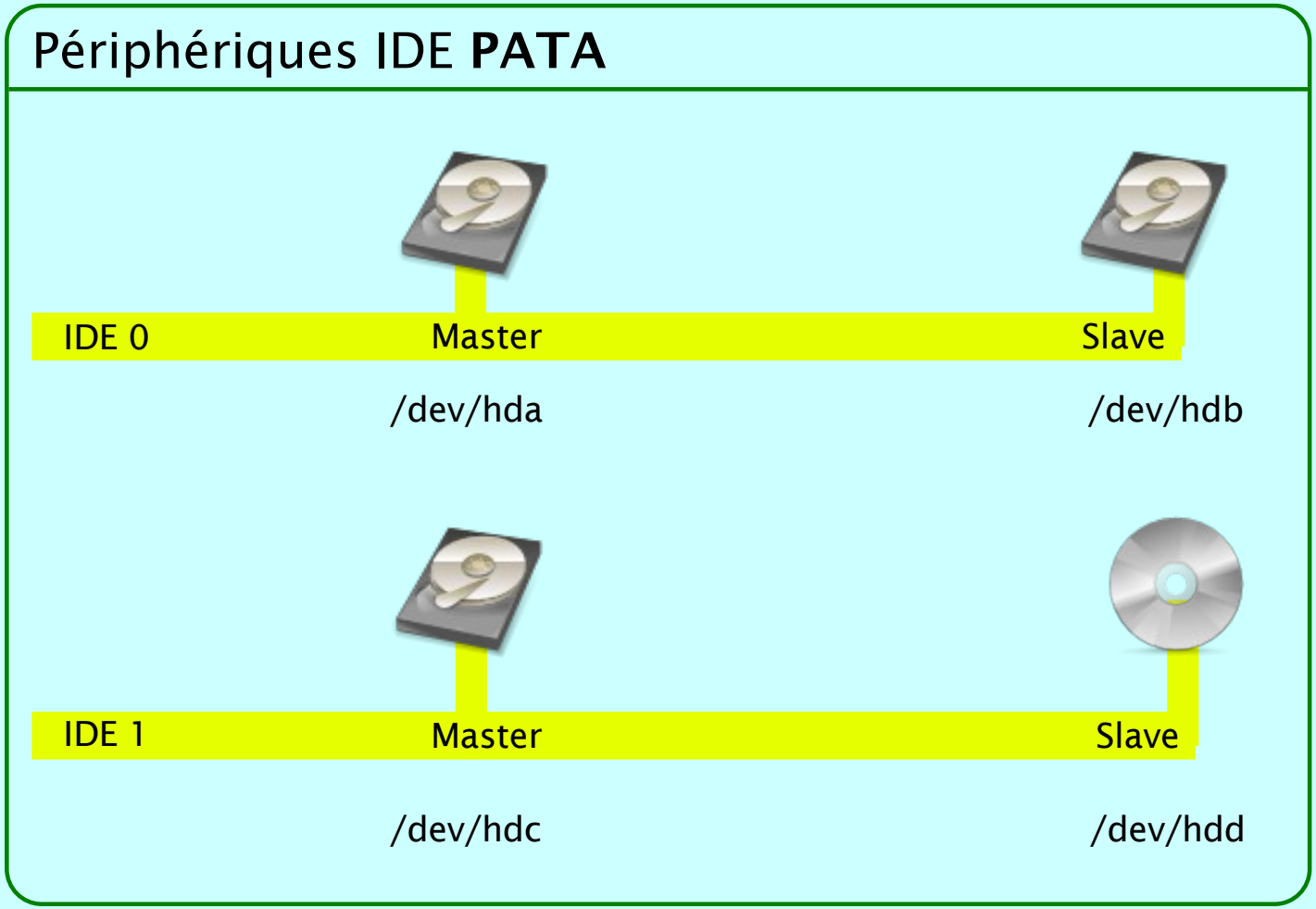
- Mode d'accès : **b**loc , ou **c**haracter
- Propriétaire, groupe
- Droits d'accès classique unix : rwxrwxrwx
- Au lieu de la taille en octet (???) :

  - **Majeur** (entier) : permet au noyau d'activer le driver du périophérique
    (-> indexe dans une table de pointeurs de fonctions)
  - **Mineur** (entier) : argument passé au driver

```
crw-rw----  1 jlc  usb      99,   0 sep 28 11:34 parport0
crw-rw----  1 root root     10,   1 sep 28 11:33 psaux

brw-rw----  1 root disk      3,   0 sep 28 11:33 hda
brw-rw----  1 root root      3,   1 sep 28 11:33 hda1
brw-rw----  1 root root      3,   2 sep 28 11:33 hda2
brw-rw----  1 root root      3,   5 sep 28 11:33 hda5
brw-rw----  1 root root      3,   6 sep 28 11:33 hda6
```

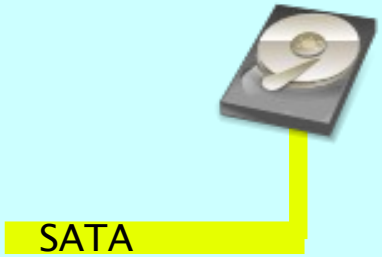
b|c majeur, mineur

## Nommage des périphériques :



## Nommage des périphériques :

### IDE SATA

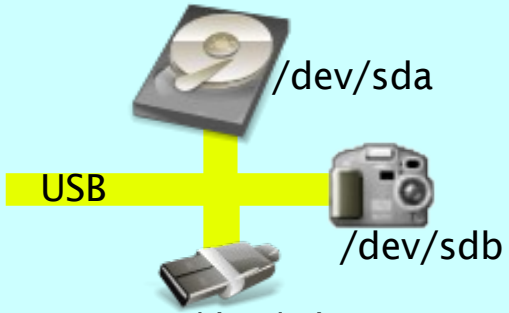



Linux <=2.4 /dev/hda
linux >= 2.6 /dev/sda

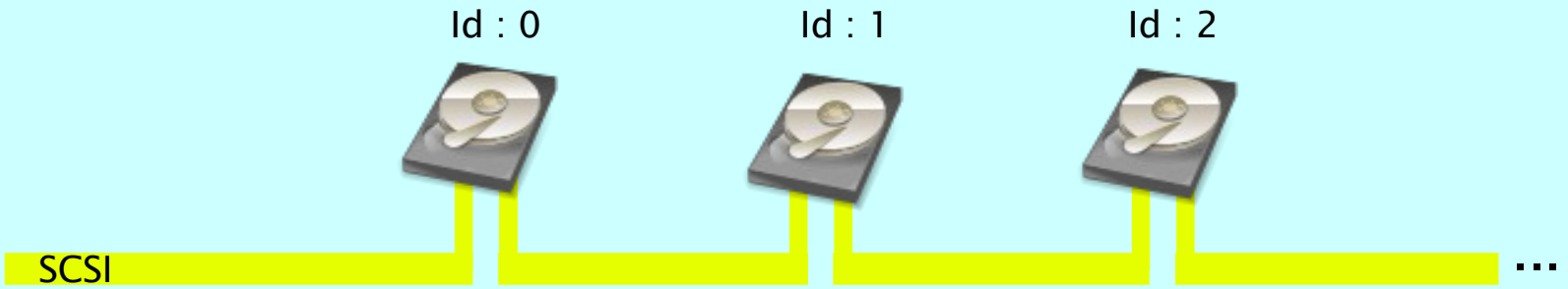
### disquette


/dev/fd0

### USB : Assimilé SCSI




### Périphériques SCSI



# 3 Éléments d'administration Unix : Partitionnement des disques

- La plupart des systèmes d'exploitation (fixes) « correctement » installés utilisent un disque plusieurs partitions:

  - partition « système » (fichiers systèmes, fichiers de configuration ...)
  - partition « utilisateurs » (données des utilisateurs)
  - ...

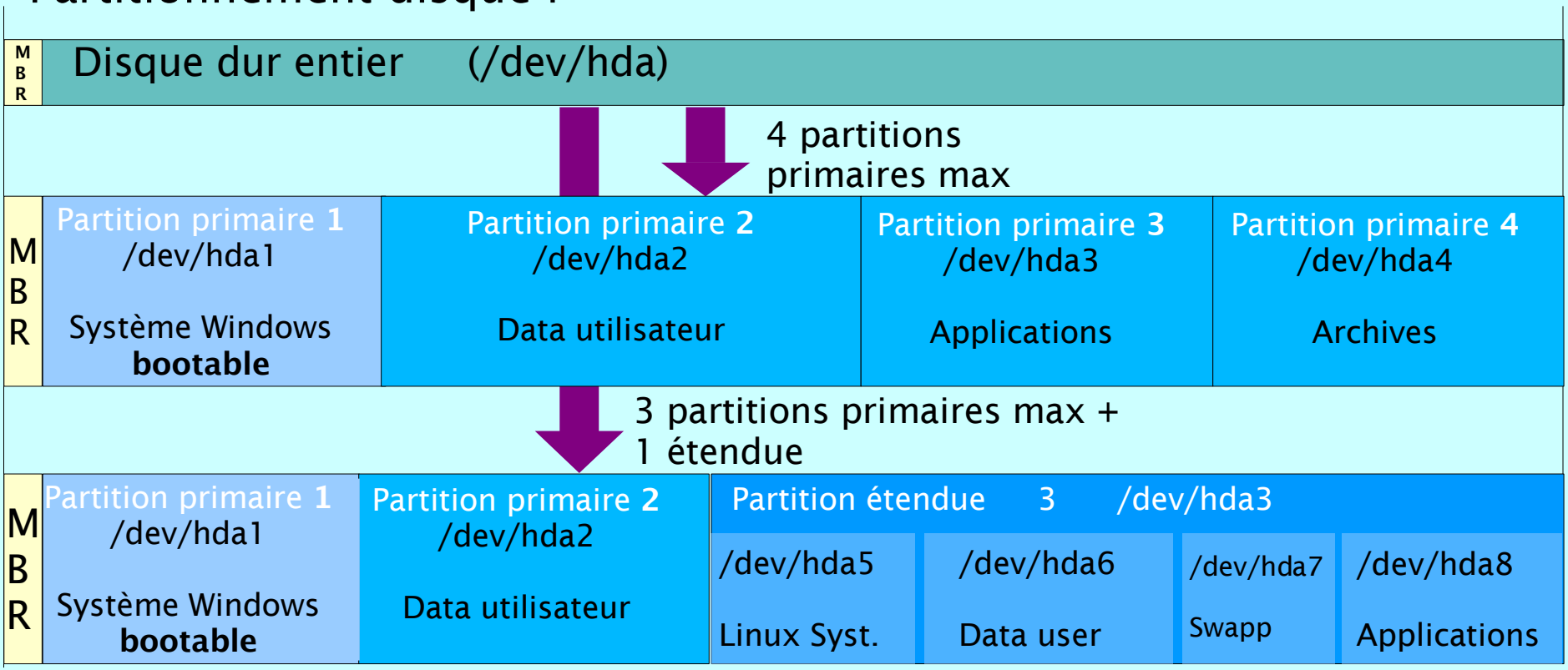
- exploitation plus sécurisée
- on peut formater une partition indépendamment des autres
- on peut utiliser une partition en lecture seule
- ...

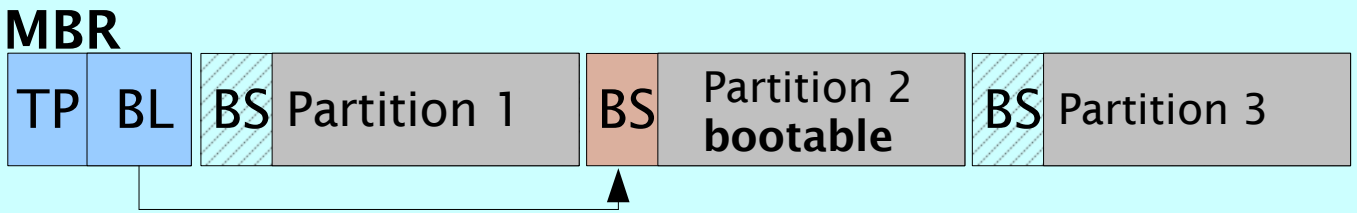
- partitionnement statique => planifier le partitionnement
- on ne peut pas modifier simplement un partitionnement statique
- partitionner est une opération « »low level », risquée !!
- ...

Pour bénéficier des avantage du partitionnement dynamique il faut passer à des solutions de type **RAID** (**R**edondant **A**rray of **I**ndependant **D**isks) ou **LVM** (**L**ogical **V**olume **M**anager)

# 3 Éléments d'administration Unix : Partitionnement des disques

## Partitionnement disque :




**MBR** : Master Boot Reccord
**TP** : Table Partition
**BL** : Boot Loader
**BS** : Boot Sector

Rq: correspondance Windows : C: <=> /dev/hda1
D: <=> /dev/hda2 ...

# 3 Éléments d'administration Unix : Partitionnement/Filesystem

Partitionnement et formatage du disque dur :

- Formatage « bas niveau » (physique, en usine)
- Partitionnement (à l'installation de l'OS)
  - ***fips***, *fdisk*, *PartitionMagic* (DOS)
  - *fdisk*, *parted* (linux)
  - à *l'installation* de Linux (menu caractère, menu graphique)
- « Formatage » « haut niveau » (logique, dépends de l'OS et du FileSystem cible)
  - *format* (Windows : créé un filesystem FAT ou NTFS)
  - *mkfs* (Unix : créé un filesystem Ext2, Ext3, FAT, ...)

    exple : `mkfs -t ext2 /dev/hda1`

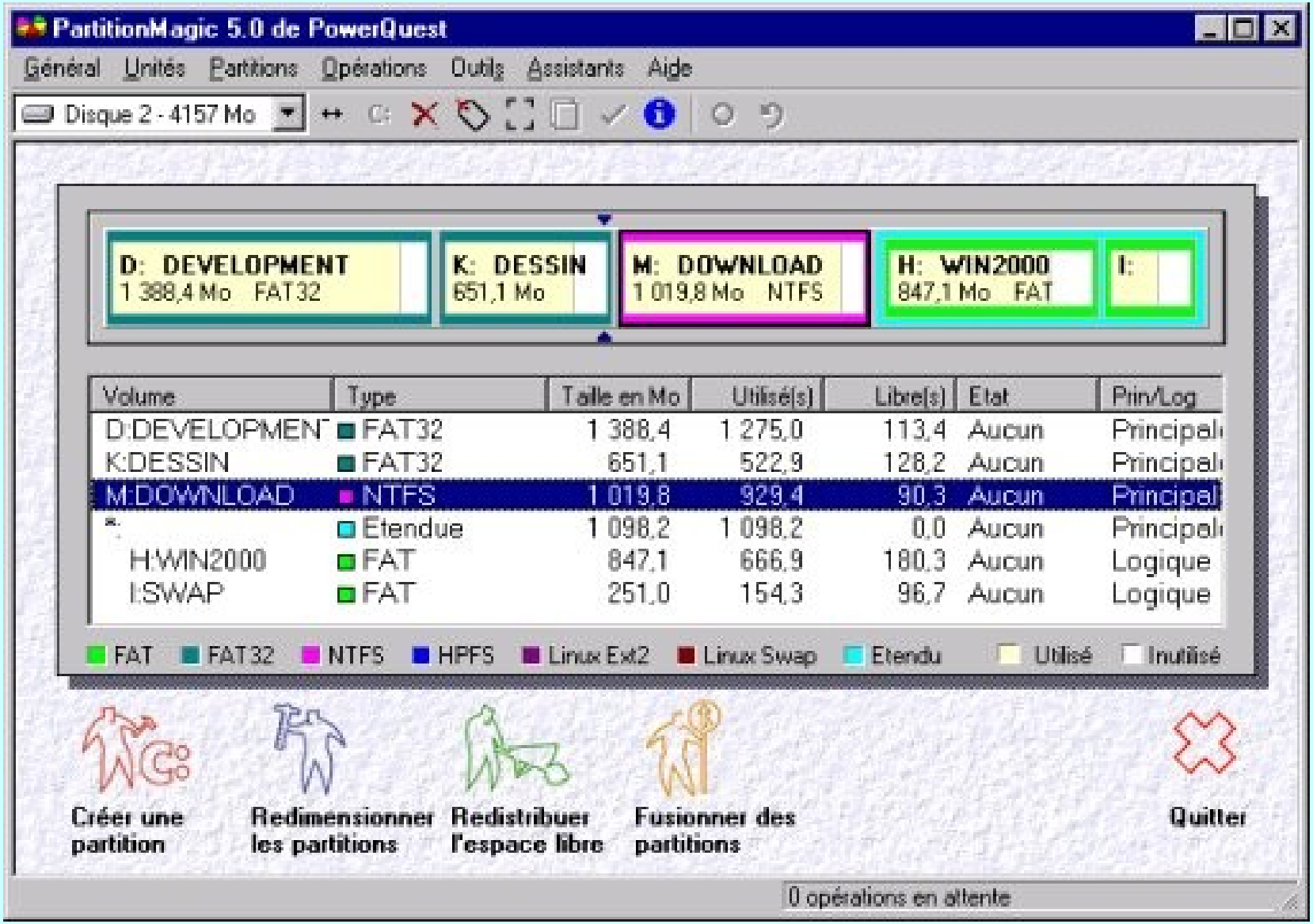
    `mkfs -t vfat /dev/fd0`
- Système de fichiers journalisés : plus robuste aux pannes secteurs

## FIPS

```
FIPS version 2.0, Copyright (C) 1993/94 Arno Schaefer
FAT32 Support Copyright (C) 1997 Gordon Chaffee

DO NOT use FIPS in a multitasking environment like Windows, OS/2, Desqview,
Novell Task manager or the Linux DOS emulator: boot from a DOS boot disk first.

If you use OS/2 or a disk compressor, read the relevant sections in FIPS.DOC.

FIPS comes with ABSOLUTELY NO WARRANTY, see file COPYING for details
This is free software, and you are welcome to redistribute it
under certain conditions; again see file COPYING for details.
[...]
Checking root sector ...
Partition table adapted to the current drive geometry:

     |         |      Start      |       |       End       | Start   |Number of|
Part.|bootable|Head Cyl. Sector|System|Head Cyl. Sector| Sector |Sectors  | MB
-----+--------+-----------------+------+-----------------+--------+---------+----
1    | yes    |   1    0      1|   06h| 254   25     63|      63|   417627| 203
2    | no     |   0   26      1|   0Fh| 254 1221     63|  417690| 19213740|9381
3    | no     |   0    0      0|   00h|   0    0      0|       0|        0| 0
4    | no     |   0    0      0|   00h|   0    0      0|       0|        0| 0
OK

Which Partition do you want to split (1/2)?
```

# 3 Éléments d'administration Unix : Partitionnement/Filesystem

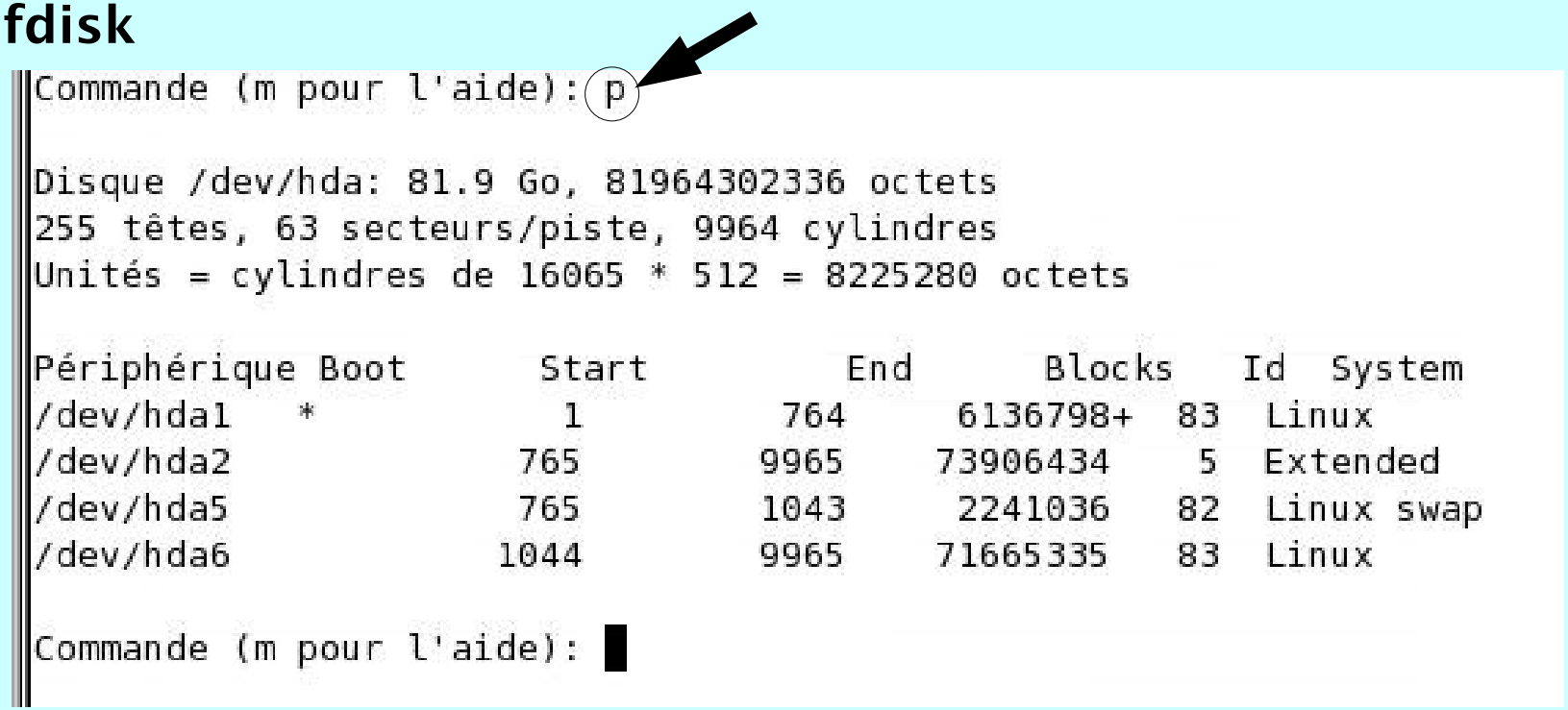
## PartitionMagic



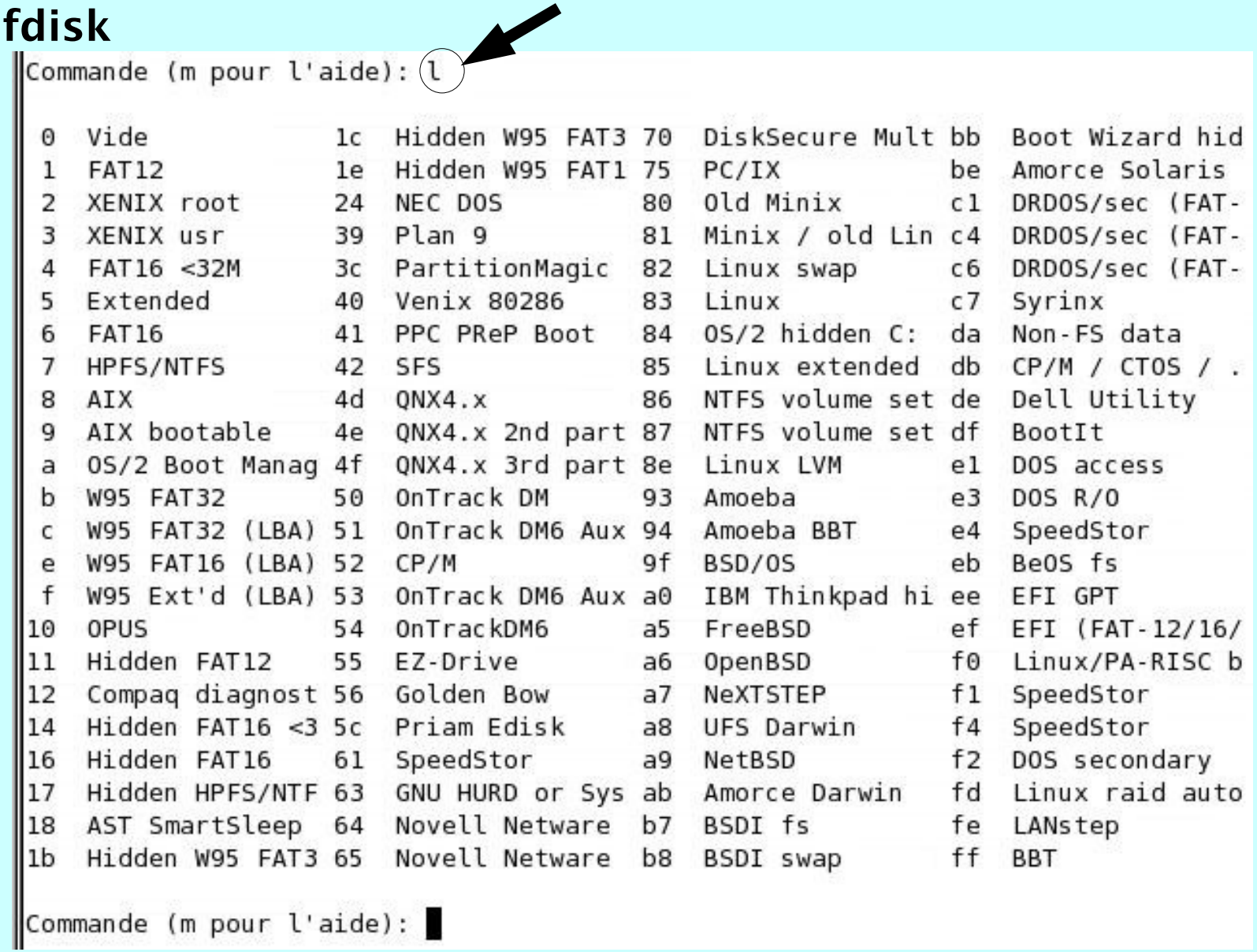
## fdisk

```
[root@blaise ~]# fdisk /dev/hda

Le nombre de cylindres pour ce disque est initialisé à 9964.
Il n'y a rien d'incorrect avec cela, mais c'est plus grand que 1024,
et cela pourrait causer des problèmes en fonction pour certaines configurations:
1) logiciels qui sont exécutés à l'amorçage (i.e., vieilles versions de LILO)
2) logiciels d'amorçage et de partitionnement pour d'autres SE
   (i.e., DOS FDISK, OS/2 FDISK)

Commande (m pour l'aide): m
Commande action
   a   bascule le fanion d'amorce
   b   éditer l'étiquette BSD du disque
   c   basculer le fanion de compatibilité DOS
   d   détruire la partition
   l   lister les types de partitions connues
   m   afficher ce menu
   n   ajouter une nouvelle partition
   o   créer une nouvelle table vide de partitions DOS
   p   afficher la table de partitions
   q   quitter sans faire de sauvegarde
   s   créer une nouvelle étiquette vide pour disque de type Sun
   t   modifier l'identificateur de la partition système
   u   modifier l'affichage et la saisie des unités
   v   vérifier la table de partitions
   w   écrire la table sur le disque et quitter
   x   fonctionnalité additionnelle (pour experts seulement)
```

## fdisk

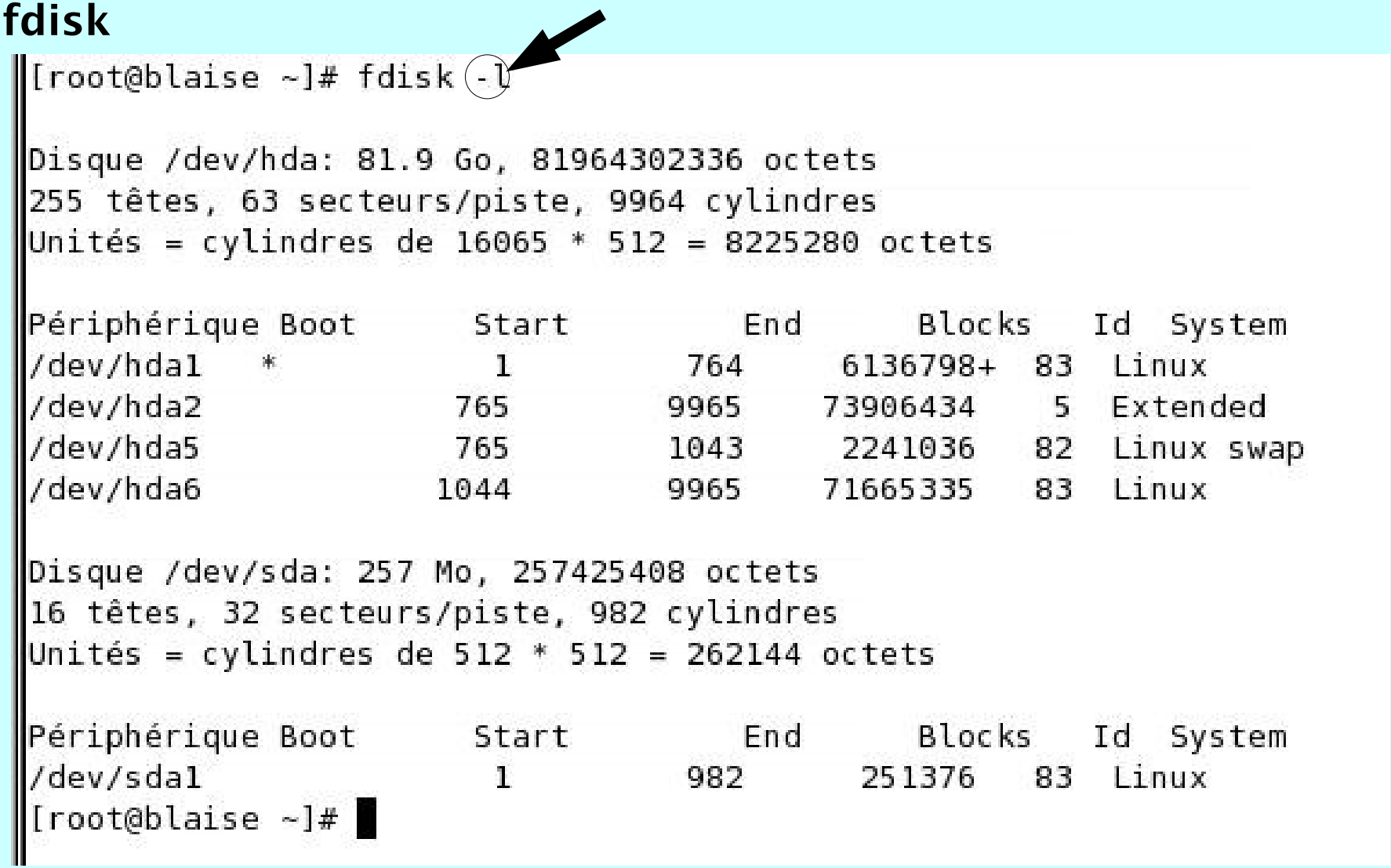

Commande (m pour l'aide): p
Disque /dev/hda: 81.9 Go, 81964302336 octets
255 têtes, 63 secteurs/piste, 9964 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets
Périphérique Boot Start End Blocks Id System
/dev/hda1 * 1 764 6136798+ 83 Linux
/dev/hda2 765 9965 73906434 5 Extended
/dev/hda5 765 1043 2241036 82 Linux swap
/dev/hda6 1044 9965 71665335 83 Linux
Commande (m pour l'aide):

## fdisk



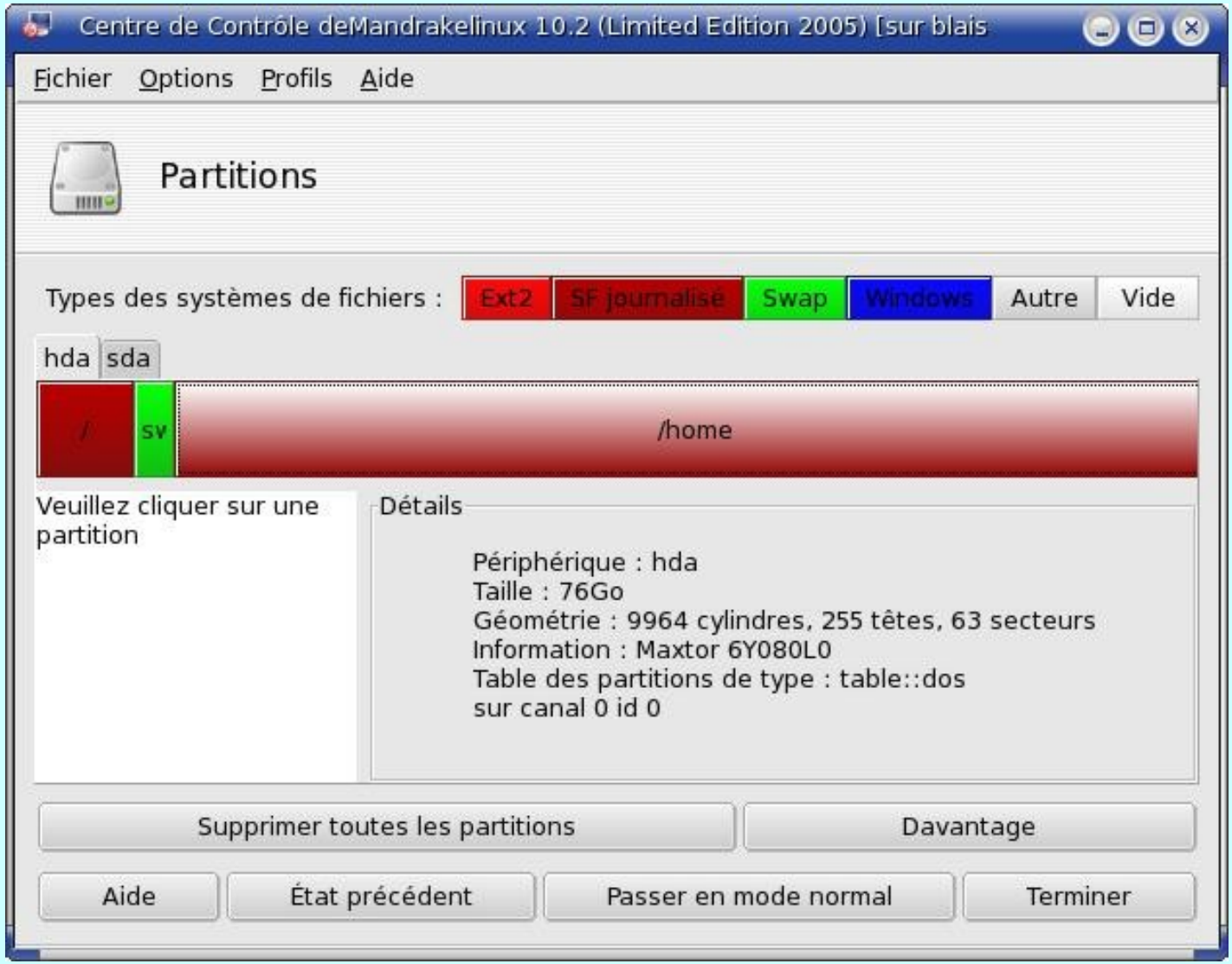
```
Commande (m pour l'aide): l

 0  Vide            1c  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid
 1  FAT12           1e  Hidden W95 FAT1 75  PC/IX           be  Amorce Solaris
 2  XENIX root      24  NEC DOS         80  Old Minix       c1  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          81  Minix / old Lin c4  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic  82  Linux swap      c6  DRDOS/sec (FAT-
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 6  FAT16           41  PPC PReP Boot   84  OS/2 hidden C:  da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux extended  db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  NTFS volume set de  Dell Utility
 9  AIX bootable    4e  QNX4.x 2nd part 87  NTFS volume set df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 b  W95 FAT32       50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  W95 FAT32 (LBA) 51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 e  W95 FAT16 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 f  W95 Ext'd (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
10  OPUS            54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
12  Compaq diagnost 56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
14  Hidden FAT16 <3 5c  Priam Edisk     a8  UFS Darwin      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor       a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Amorce Darwin   fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden W95 FAT3 65  Novell Netware  b8  BSDI swap       ff  BBT

Commande (m pour l'aide):
```

## fdisk



```
[root@blaise ~]# fdisk -l

Disque /dev/hda: 81.9 Go, 81964302336 octets
255 têtes, 63 secteurs/piste, 9964 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets

Périphérique Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1         764     6136798+  83  Linux
/dev/hda2             765        9965    73906434    5  Extended
/dev/hda5             765        1043     2241036   82  Linux swap
/dev/hda6            1044        9965    71665335   83  Linux

Disque /dev/sda: 257 Mo, 257425408 octets
16 têtes, 32 secteurs/piste, 982 cylindres
Unités = cylindres de 512 * 512 = 262144 octets

Périphérique Boot      Start         End      Blocks   Id  System
/dev/sda1               1         982      251376   83  Linux
[root@blaise ~]#
```

## Installation Mandriva / outil DrakConf



| Système d'exploitation | Types de système de fichiers supportés |
|---|---|
| Dos | FAT16 |
| Windows 95 | FAT16 |
| Windows 95 OSR2 | FAT16, FAT32 |
| Windows 98 | FAT16, FAT32 |
| Windows NT4 | FAT, NTFS (version 4) |
| Windows 2000/XP | FAT, FAT16, FAT32, NTFS (versions 4 et 5) |
| Linux | Ext2, Ext3, ReiserFS, Linux Swap, (FAT, NTFS, ...) |
| MacOS | HFS, MFS |
| SGI IRIX | XFS |
| FreeBSD, OpenBSD | UFS (Unix File System) |
| Sun Solaris | UFS (Unix File System) |
| IBM AIX | JFS (Journaled File System) |

# 3 Éléments d'administration Unix : Montage des périphériques

L'opération de **montage** des périphériques :

- DOS et Windows utilisent la notion d'**unité logique** pour fournir un accès aux ressources de stockage
  ( A: -> floppy,  C: -> disque dur, ... E: -> lecteur CD)

- Tous les Unix utilisent la notion de **montage** :
  - un périphérique est associé à un point de montage (répertoire) par une « opération de montage » (commande **mount**)
  - la commande **mount** peut être utilisée « à la main »

    exple : `mount /dev/hda1 /`

    `mount /dev/sda1 /mnt/removable`
  - tous les périphériques montés bénéficient du « cache disque »

# 3 Éléments d'administration Unix : Montage des périphériques

- avant d'extraire un périphérique amovible (disquette, clef USB, ...), on DOIT le démonter (**umount**), pour synchroniser les écritures (vidage du cache disk)
- tous les montages permanents sont indiqués dans le fichier `/etc/fstab`

```
[root@blaise ~]# cat /etc/fstab
/dev/hda1 / ext3 defaults 1 1
/dev/hda6 /home ext3 defaults 1 2
/dev/hdc /mnt/cdrom auto umask=0,user,iocharset=iso8859-15,codepage=850,noauto,ro,exec,users 0 0
/dev/hdd /mnt/cdrom2 iso9660 user,iocharset=iso8859-15,noauto,ro,exec 0 0
/dev/fd0 /mnt/floppy auto umask=0,user,codepage=850,iocharset=iso8859-15,noauto,sync 0 0
none /proc proc defaults 0 0
/dev/hda5 swap swap defaults 0 0
[root@blaise ~]# _
```

La commande **df** affiche la liste des périphériques montés + propriétés :

```
[root@julot ~]# df
Sys. de fich.         Tail. Occ. Disp. %Occ. Monté sur
/dev/hda1             9,9G  4,4G  5,0G  47% /
/dev/hdb2              63G   40G   20G  68% /home
/dev/hda6             9,7G 1010M  8,3G  11% /opt
none                  252M   81M  172M  32% /tmp
/dev/hda5             9,9G  496M  8,9G   6% /var
[root@julot ~]#
```

# 3 Éléments d'administration Unix : configuration de lilo

- **lilo** (**Li**nux **Lo**ader) permet d'installer un **bootloader** multi-systèmes
- Fichier de configuration : `/etc/lilo.conf`
- Après toute modification de `lilo.conf`, il faut ré-installer le bootloader modifié en tapant la commande **lilo**

Installation en MBR

Noyau à booter

Partition racine

Fichier ram-disque

Options du noyau

```
[root@blaise cups]# cat /etc/lilo.conf
# File generated by DrakX/drakboot
# WARNING: do not forget to run lilo after modifying this file

default="linux"
boot=/dev/hda
map=/boot/map
keytable=/boot/fr-latin1.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
        label="linux"
        root=/dev/hda1
        initrd=/boot/initrd.img
        append="resume=/dev/hda5 splash=silent idebus=66"
        vga=788
        read-only
```

# 3 Éléments d'administration Unix : Gestionnaires de paquets

GNU/Linux et les applications tournant sous ce système peuvent être livrés (source ou binaires) sous 3 formats principaux :

- Le **format Read Hat (RPM** : ReadHat Package Manager) :

  fichiers *.rpm

- Le **format DEBIAN** (**DPKG** : Debian PacKaGe) :

  fichiers *.deb

- Le **format tar** (archive) :

  - fichiers *.tar : archive créée par la commande Unix **tar**
  - fichiers *.tar.gz : archive tar comprimée par **gzip**
  - fichiers *.tar.bz2 : archive tar comprimée par **bzip**

# 3 Éléments d'administration Unix : Gestionnaires de paquets

Historiquement, avec les premières version d'Unix, tout était fait :

- Avec la commande **tar**
- Avec des utilitaires dédiés (Unix propriétaires)

Aujourd'hui, le format **tar** n'est plus utilisé que :

- pour des installations particulières (code « maison », archivage, ...)
- Pour déplacer une arborescence de fichiers (copie vers une autre répertoires, une autre machine, document attaché mail, ...)

->L'utilisation de la commande **tar** comme utilitaire d'archivage sera vue lors des séances de TP.

# 3 Éléments d'administration Unix : Gestionnaires de paquets

Les systèmes de gestions de paquets Read Hat ou Debian permettent une gestion très puissante des applications installées sous GNU/Linux :

- **Création** des paquets (compilation et création du paquet source ou du paquet binaire)
- **Installation** des paquets (gestion des composants nécessaires)
- **mise à jour** des paquets (gestions des dépendances, des conflits, ...)
- **Suppression** des paquets (gestion des dépendances, et de tout ce qui a été installé)
- **Interrogation** :

  - liste des paquets,
  - fichiers contenus dans un paquet donné,
  - quel paquet contient un fichier donné,
  - ...

# 3 Éléments d'administration Unix : Gestionnaires de paquets

**Mandriva** (comme **Debian**) propose 2 niveaux de commandes :

- **Niveau du paquet** : **rpm** (Debian : **dpkg**)

  la gestion des dépendances est purement informative, sans résolution automatique => il faut installer tous les paquets » à la main »

- **Niveau global** : **urpm[iqfe]** (Debian : **apt** (**a**dvanced **p**aquet **t**ool))

  la gestion des dépendances est totale : les paquets requis sont automatiquement rajoutés à la liste des paquets à installer

- Il existe des interface graphique pour le niveau global :

  - Mandriva : **drakrpm**

  - Debian : **synaptic**, **aptitude**

->L'utilisation détaillée des commandes **rpm**, **urpmq**, **urpmi**, ...
sera vue en TP.

# 3 Éléments d'administration Unix : Gestionnaire RPM

Commande **rpm** : infos sur les **paquets installés**

(options de confort : v -> verbose ; h -> barre d'avancement tracée avec des #)

- **Installation** (**i**nstall) : rpm -ivh nom_du_paquet

  Exple : `rpm -ivh acpid-1.0.4-4mdk.rpm`

- **mise à jour** (**u**pdate) : rpm -Uvh nom_du_paquet

  Exple : `rpm -Uvh acpid-1.0.4-4mdk.rpm`

- **Suppression** (**e**rase) : rpm -e nom_du_paquet

  Exple : `rpm -e acpid`

- **Interrogation** (**q**uery) :

  - Informations sur un paquet : rpm -q nom_paquet_court

    Exple : `rpm -q emacs (réponse : emacs-21.3-20mdk)`

# 3 Éléments d'administration Unix : Gestionnaire RPM

- **Interrogation** (**q**uery) suite :

  - informations sur un paquet : rpm -qi paquet
  - liste des fichiers d'un paquet installé :

    rpm -ql paquet_installé , ou  rpm -ql -p fichier_rpm

    Exple : `rpm -ql cups`
  - liste de tous les paquets installés  : rpm -qa
  - rechercher si un paquet est installé  : rpm -qa | grep paquet
  - recherche des paquets installés contenant un fichier :

    rpm -qf nom_de_fichier_absolu

    Exple : `rpm -qf /usr/sbin/cupsd`

# 3 Éléments d'administration Unix : Gestionnaire URPM

Commande **urpmi** (installer un paquet) :

- **urpmi** gère les installations à partir de divers médias :

  - ftp, http,

  - volumes nfs et locaux,

  - médias amovibles (CD, DVD, ...)

- La configuration des sources (origines) de l'installation est donnée par le fichier `/etc/urpmi/urpmi.cfg`

- On peut modifier les sources de la distribution :

  - **urpmi.removemedia** : permet de supprimer des entrées

  - **urpmi.addmedia** : permet d'ajouter de nouvells entrées

- Syntaxe de urpmi est très simple : **urpmi** nom_du_paquet

  Si nécessaire, **urpmi** demande d'insérer le médium nécessaire

->Des exemples d'utilisation de la commande **urpmi** seront vus en TP.

Commande **urpmq** : infos sur les **paquets « installables »**

- **urpmq** [option] nom_paquet|fichier_rpm

permet d'interroger la Base de Données (BD) urpmi (cf man urpmi);

Exemple d'options utiles :

--fuzzy : (ou -y) recherche les paquets « *nom_paquet* »

--liste : liste de tous les paquets connus de la BD urpmi

--liste-media : affiche les média de la distribution (CD, réseau, ...)

--liste-url : idem, avec en plus les URL d'accès

--dump-config : fournit la totalité des informations sur les média de la distrib, les URL d'accès et les fichiers d'info utilisés

-l paquet : liste les fichiers du paquets de la distribution

Exples :

```
urpmq cups
urpmq --fuzzy cups
```

Commande **urpmf** :

- **urpmf** [option] fichier

  trouve le paquet RPM de la distribution contenant le fichier mentionné

  - Si `fichier` est un nom absolu, il est cherché exactement,

  - Si `fichier` est un extrait de path absolu, il est cherché sous la forme

    « *fichier* »

  Exples :

  ```
  urpmf /usr/sbin/cupsd
  urpmq bin/cupsd
  ```

# 3 Éléments d'administration Unix : Gestionnaire DEBIAN

Commande **dpkg** : gestion d'un paquet isolé,

pas de gestion automatique des dépendances

- **dpkg** [option] [fichier.deb ...] ; Principales options :

-i : installe le(s) paquets contenus dans les fichiers(s)

-r : désinstalle le paquet sans supprimer ses fichiers de configuration

--purge : désinstallation complète (paquet + fichiers config)

-reconfigure paquet : rejouer la configuration du paquet

-l : liste de tous les paquets

-L : affiche la liste de tous les fichiers contenus dans le paquet

-S fichier : recherche de tous les paquets contenant fichier

Exples :

```
dpkg -i cups
dpkg -S /usr/bin/cupsd
```

# 3 Éléments d'administration Unix : Gestionnaire DEBIAN

Le système **APT** : **A**dvanced **P**ackage **T**ool

- Résoud automatiquement les problèmes de dépendances entre paquets
- Tient à jour les listes des installés et des paquets disponibles
- Gère la liste des sources d'installation (fichier `/etc/apt/sources.list`)

**apt-get install** `fichier.deb` ... : installer des paquets Debian

**apt-get remove** `[--purge] fichier.deb` ... : désinstaller des paquets

**apt-get update** : récupère la liste des paquets disponibles en utilisant les sources mentionnées dans le fichier `sources.list`

**apt-get upgrade** : mise à jour des paquets

**apt-get clean** : supprime les fichiers présents dans `/var/cache/apt/archives`

# 3 Éléments d'administration Unix : Gestionnaire DEBIAN

Récapitulatif incomplet et approximatif des équivalences RPM/DEBIAN !!!

| | |
|---|---|
| rpm -ql paquet | dpkg -L paquet |
| rpm -qa | dpkg -l<br>dpkg –get-selection |
| rpm -i paquet | dpkg -i paquet |
| rpm -e paquet | dpkg --purge paquet |
| ? | dpkg -r paquet |
| ? | dpkg -reconfigure paquet |
| rpm -qf fichier | dpkg -S |
| urpmi paquet | apt-get install paquet |
| urpme paquet | apt-get remove paquet |

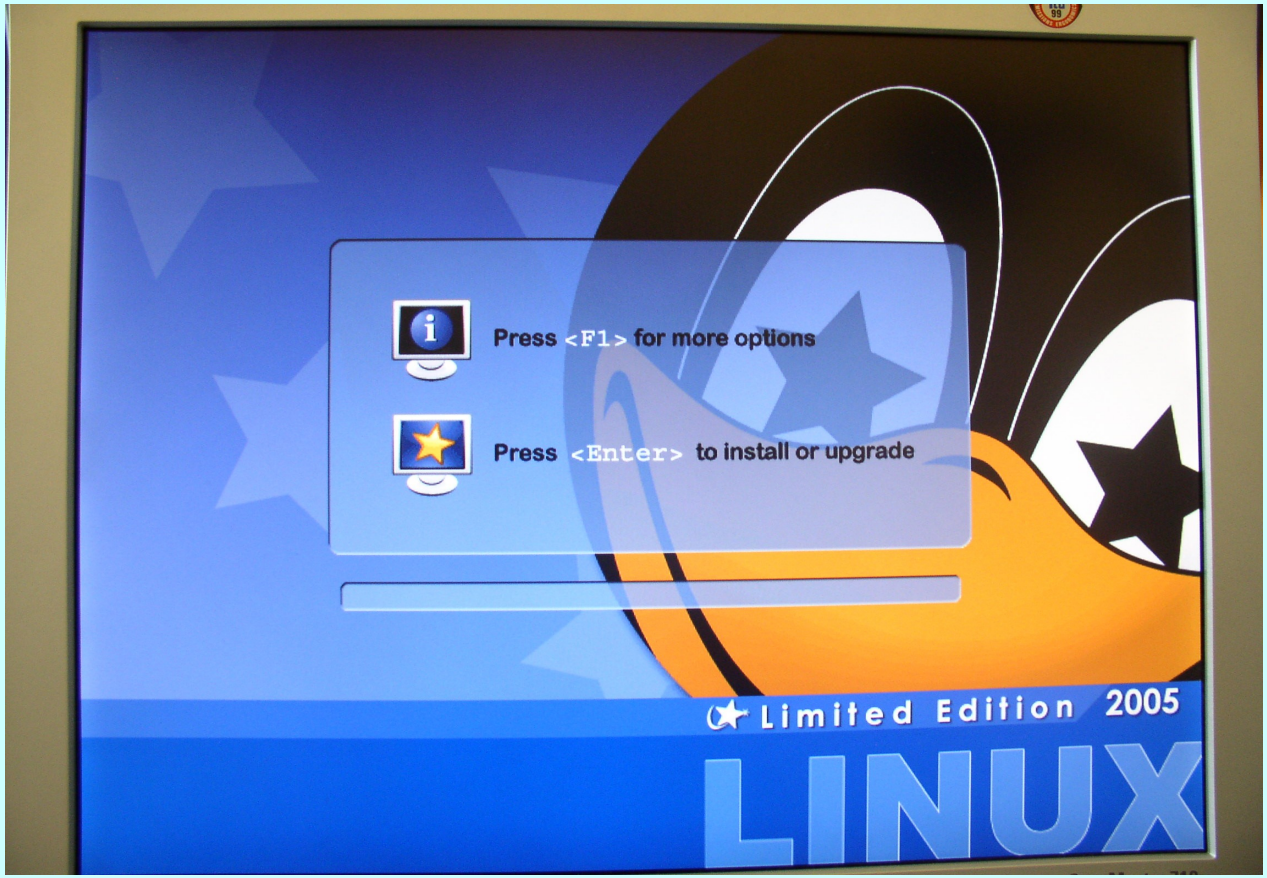
# Cours d'administration Unix

# 4 Installation de GNU/Linux : TP1

☞ **Voir le texte du TP1**

Objectifs :

- Installer un PC serveur GNU/Linux (installation réseau FTP ou HTTP n'utilisant que le premier CD).



# Cours d'administration Unix

# 5 Les fichiers de configuration : TP1

☞ **Voir le texte du TP1**

Objectifs :

- Installer un PC serveur GNU/Linux (installation réseau FTP ou HTTP n'utilisant que le premier CD).
- Visiter les principaux fichiers de configuration utiles à l'administration d'un serveur Linux.

# Cours d'administration Unix

# 6 Gestion des utilisateurs, des groupes : TP1

☞ **Voir le texte du TP1**

Objectifs :

- Installer un PC serveur GNU/Linux (installation réseau FTP ou HTTP n'utilisant que le premier CD).
- Visiter les principaux fichiers de configuration utiles à l'administration d'un serveur Linux.
- Réaliser des opérations élémentaires d'administration « à la main » :
- utiliser des commandes de base d'administration (`mount`, `mkfs`, `df` , ...)
- gérer les comptes utilisateurs, ...
- installer des serveurs usuels (`ssh`, `ftp`).
- Utiliser le shell (commandes, redirections, filtres) en interactif, et comme langage de programmation (scripts d'administration).

# Cours d'administration Unix

# 7 NIS : Présentation

- **NIS** : une base de données répartie
- Principes
- Installation
- Commandes utiles
- Configuration
- Lancement
- TP

# 7 NIS : Présentation

**NIS** (**N**etwork **I**nformation **C**enter, http://www.linux-nis.org ) :

- Protocole datant des années 1985
- Origine : les « Yellow Pages » (YP) de SUN
- Constitue une base de données répartie pour les fichiers de configuration Unix
- Permet de centraliser les fichiers de configuration sur un serveur **NIS** pour éviter de duppliquer tous ces fichiers en autant d'exemplaires que d'ordinateurs à gérer sur un réseau
  - ☞ Évite les copies multiples !
  - ☞ Évite les copies pas à jour !!
- Permet de n'avoir à maintenir les fichiers que sur une seule machine (serveur)
- **NIS** et **NFS** permettent de construire des systèmes informatiques répartis assurant un partage des fichiers et des données cohérent et transparent.

# 7 NIS : Présentation

Services/démons utilisés par le service **NIS** :

**portmap** mise en correspondance numéro de ports TCP/IP <-> numéro de processus RPC (voir `/etc/rpc` pour les numéros réservés)

côté serveur :

**ypserv** implémente le serveur NIS

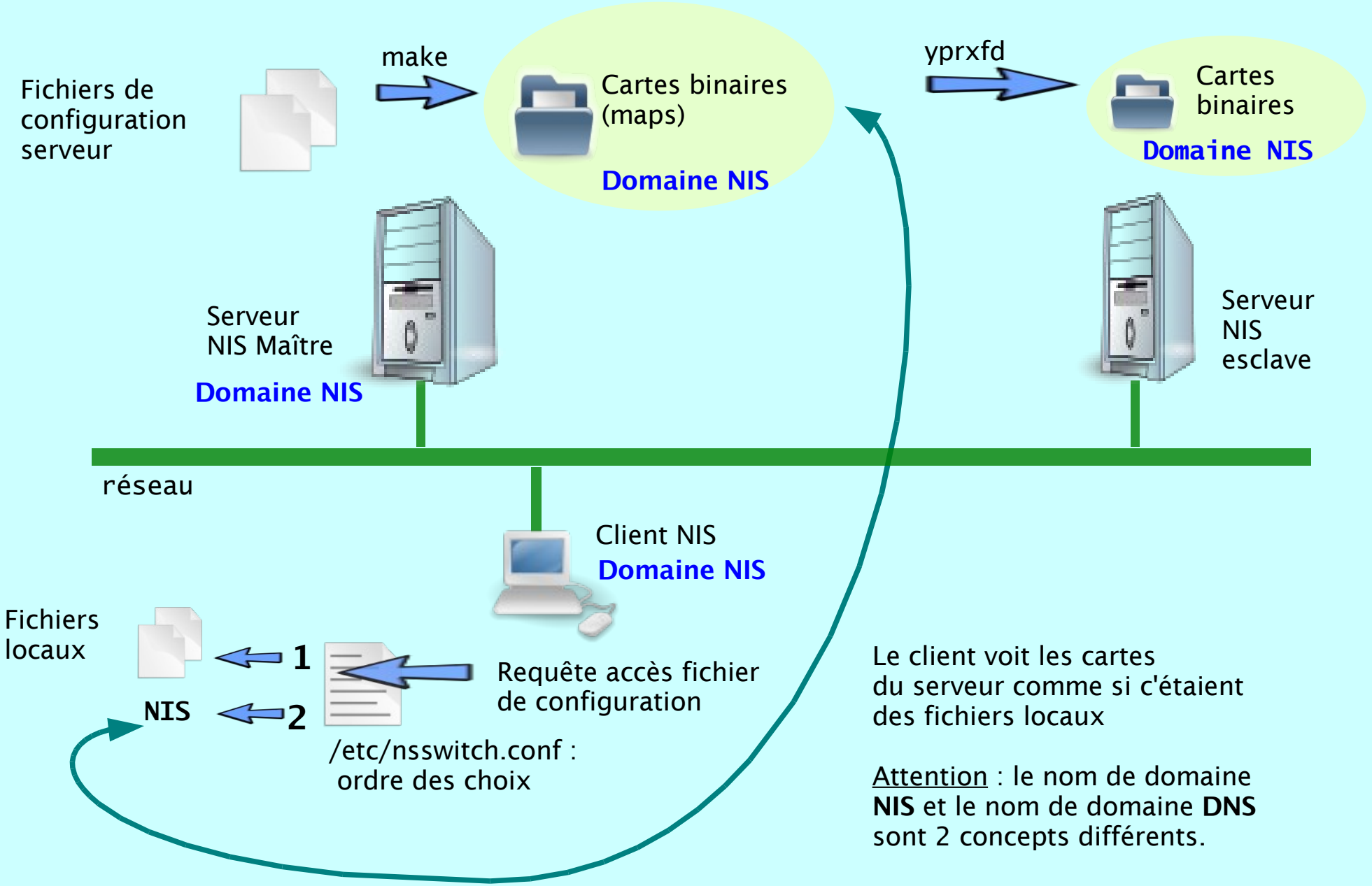
**yppasswd** permet de changer un mot de passe sur le serveur NIS depuis un client NIS (démon rpc.yppasswd)

**ypxfrd** accélère les transferts entre serveur maître et esclave (démon rpc.ypxfrd)

Côté client:

**ypbind** implémente le client NIS

# 7 NIS : Principe



# 7 NIS : Installation - serveur

**Installation** : **portmap** et **réseau** pré-requis,
serveur -> `urpmi ypserv`

Paquet serveur : **ypserv**

```
# rpm -ql ypserv
/etc/rc.d/init.d/yppasswdd
/etc/rc.d/init.d/ypserv
/etc/rc.d/init.d/ypxfrd
/etc/ypserv.conf
/usr/include/rpcsvc/ypxfrd.x
/usr/lib/yp/...
/usr/lib/yp/ypinit
/usr/lib/yp/ypxfr
...
/usr/sbin/rpc.yppasswdd
/usr/sbin/rpc.ypxfrd
/usr/sbin/yppush
/usr/sbin/ypserv
...
/usr/share/man/man8/ypxfrd.8.bz2
/var/yp/Makefile
```

Fichiers de lancement

Fichiers de configuration

Fichiers binaires (commande ou démon)

# 7 NIS : Installation - client

**Installation** : **portmap** et **réseau** pré-requis,

```
client    -> urpmi ypbind tp-tools
```

Paquets client : **ypbind** et **yp-tools**

```
# rpm -ql ypbind
/etc/rc.d/init.d/ypbind
/etc/yp.conf
/sbin/ypbind
/usr/share/doc/ypbind...
/usr/share/man/man5/...
/usr/share/man/man8/...
/var/yp/binding
```

```
# rpm -ql yp-tools
/usr/bin/ypcat
/usr/bin/ypchfn
/usr/bin/ypchsh
/usr/bin/ypmatch
/usr/bin/yppasswd
/usr/bin/ypwhich
/usr/sbin/yppoll
/usr/sbin/ypset
/usr/sbin/yptest
/usr/share/doc/...
/usr/share/man/man1/...
/var/yp/nicknames
```

Fichiers lancement

Fichiers configuration

Binaires (commandes, démons)

# 7 NIS : Commandes utiles

## Commandes utiles

| | |
|---|---|
| `nisdomainname` | |
| `domainname` | |
| `ypdomainname` | afficher le nom de domaine NIS |
| `rpcinfo -p` | afficher les services portmap |

### Serveur

| | |
|---|---|
| `/usr/lib/yp/ypinit -m` | initialiser le serveur maître |
| `cd /var/yp; make` | refaire les cartes du serveur maître, et les propager vers les serveurs esclaves |
| `/usr/lib/yp/ypinit -s` | initialiser un serveur esclave |

### Client

| | |
|---|---|
| `ypwhich` | afficher le nom du serveur NIS |
| `yppasswd` | changer son mot de passe sur le serveur NIS |
| `ypcat passwd` | afficher le fichier `passwd` du serveur |
| `ypcat group` | afficher le fichier `group` du serveur |
| `ypcat ...` | afficher le fichier `...` du serveur |

# 7 NIS : Configuration commune client & serveur : config réseau

`/etc/nsswitch.conf` Network Service Switch

ordre dans lequel utiliser les différents mode d'accès possibles (fichier, nis, dns) pour accéder au contenu des fichiers de configuration

```
...
passwd:      files nis nisplus
shadow:      files nis nisplus
group:       files nis nisplus
...
```

Fichiers locaux

NIS+ = NIS version 3 (sécurisé)

`/etc/host.conf`

ordre dans lequel utiliser les différents mode d'accès possibles (fichier, nis, dns) pour résoudre les correspondances adresse IP <=> FQDN
Il est préférable d'utiliser le DNS plutôt que NIS :

```
...
order hosts,bind
...
```

Mais si aucun serveur DNS n'est accessible, on pourrait écrire :

```
...
order hosts,nis
...
```

# 7 NIS : Configuration serveur : config réseau

**Un serveur a une adresse IP fixe** => on **doit** préciser le nom du **domaine NIS** dans un fichier de configuration


Tous les fichiers de configuration des services spécifiques aux distributions dérivées de RedHat (Fedora, Mandriva, ...) sont dans le répertoire : /etc/sysconfig



/etc/sysconfig/network

```
HOSTNAME=hostaaa.domainebbb.fr
NETWORKING=yes
GATEWAY=192.168.74.1
NISDOMAIN=tp
```


Fichier de configuration réseau : `/etc/network/interfaces`
Fichier de configuration domaine NIS : `/etc/defaultdomaine`

/etc/defaultdomain

```
tp
```

# 7 NIS : Configuration serveur

`/etc/ypserv.conf`

fichier ASCII qui contenant 2 types de ligne :

- des options pour le démon `ypserv` : « `option: [yes|no]` » Exemples :

  `dns yes` le serveur interrogera le DNS pour trouver ses clients qui n'apparaissent pas dans les maps hosts.

  `xfr_check_port yes` : pour faire tourner le serveur sur un port inférieur à 1024 (`yes` par défaut)

- des règles d'accès au serveur NIS ( -> qui peut voir quoi ...) au format :

  `host:domain:map:security`

  `security` : `none` -> accès autorisé
  `port` -> accès autorisé seulement si port client < 1024
  `deny` -> pas d'accès

# 7 NIS : Configuration serveur

`/etc/ypserv.conf` Exemple de fichier :

```
...
# Should we do DNS lookups for hosts not found in the hosts table ?
# This option is ignored in the moment.
dns: no

# xfr requests are only allowed from ports < 1024
xfr_check_port: yes

# Not everybody should see the shadow passwords, not secure, since
# under MSDOG everbody is root and can access ports < 1024 !!!
*:*:shadow.byname:port
...
```

`/var/yp/Makefile`

Contient les options importantes comme le choix des cartes du serveur qui seront exportés par NIS.
À personnaliser après toutes installation d'un paquet ypserv.

`/etc/nsswitch.conf`

`/etc/host.conf`

# 7 NIS : Initialisation, lancement de NIS sur un serveur

Après l'installation du paquet serveur **NIS** :

**Serveur maître**

- 1- Configurer le nom de domaine NIS
- 2- Éditer si besoin le fichier `/etc/hosts` pour déclarer tous les serveurs (maître ou esclaves) utilisés par NIS
- 3- Éditer si besoin le fichier `/var/yp/Makefile`
- 4- Éditer si besoin le fichier `/etc/host.conf` : choix de l'ordre des méthodes de résolution FQDN<->adresse IP (local, dns, nis ?)
- 5- Éditer si besoin le fichier /etc/nsswitch.conf : choix de l'ordre des méthodes d'accès aux fichiers sélectionnée en 2-
- 6- Éditer le fichier `/etc/ypserv.conf`

# 7 NIS : Initialisation, lancement de NIS sur un serveur

- 7- Vérifier la présence des scripts de démarrage
  `/etc/rc.d/init.d/yppasswdd, ypserv, ypxfrd`
- 8- Lancer les services `ypserv,ypserv` et `yppasswdd`
- 9- initialiser le serveur maître :`/usr/lib/yp/ypinit -m`

**Serveur esclave**

- 1- Installer la machine comme un client NIS
- 2- Installer la machine comme un serveur maître (étapes 1- à 8- )
- 3- initialiser le serveur esclave :

  `/usr/lib/yp/ypinit -s server_maitre`
- 4- sur le serveur maître :

  - mettre la liste des serveurs secondaires dans le fichier `/var/yp/ypserv`
  - éditer le fichier `/var/yp/Makefile`, mettre la variable `NOPUSH` à `false`
  - aller dans le répertoire `/var/yp` et taper `make`

# 7 NIS : Configuration client : réseau

**Client fixe** => on **doit** préciser le nom du **domaine NIS**

/etc/sysconfig/network

```
HOSTNAME=hostaaa.domainebbb.fr
NETWORKING=yes
GATEWAY=192.168.74.1
NISDOMAIN=tp
```

/etc/defaultdomain

```
tp
```

**Client nomade (portable)**

- on peut préciser le nom du domaine NIS dans le fichier `network`, mais PB quand on change de réseau
- Il est plus intéressant d'utiliser la configuration dynamique **DHCP** pour récupérer les informations « nom de domaine NIS » et « serveur NIS »

/etc/dhclient-eth0.conf

/etc/dhclient.conf

```
...
request subnet-mask, broadcast-address, time-offset,routers,
domain-name, domain-name-servers, host-name, nis-domain;
...
```

# 7 NIS : Configuration client

`/etc/yp.conf` (Permet d'éviter la recherche du serveur NIS par broadcast)

Liste des serveurs pour le domaine nis du client :

```
domain nom_domaine_nis server adresseIP|FQDN
```

`/etc/nsswitch.conf`

`/etc/host.conf`

# 7 NIS : Lancement du service NIS côté client

Après l'installation du paquet client **NIS** :

- 1- Configurer le nom de domaine NIS
- 2- Éditer si besoin le fichier `/etc/host.conf` : choix de l'ordre des méthodes de résolution FQDN<->adresse IP (local, dns, nis ?)
- 3- Éditer si besoin le fichier `/etc/nsswitch.conf` : choix de l'ordre des méthodes d'accès aux fichiers sélectionnée en 2-
- 4- Éditer le fichier `/etc/yp.conf` pour désigner le(s) serveur(s) NIS (utiliser des adresses IP plutôt que des FQDN ...)
- 5- Relancer le service network (prise en compte du nom de domaine NIS)
- 6- Lancer le service `ypbind`.

# 7 NIS : sécurité

NIS n'est pas un protocole très sécurisé.

- Préférer l'utilisation de NIS en **réseau local** (192.168.x.y) accédant à internet par une passerelle sécurisé (firewall)
- Utiliser les fichiers `/etc/ypserv.conf` et `/var/yp/securenet` pour limiter les accès non autorisés à NIS

# Cours d'administration Unix

# 8 NFS : Présentation

- **NFS** : un système de fichiers réseau
- Principes
- Installation
- Configuration
- Lancement
- TP

# 8 NFS : Présentation

**NFS** (Network File System) http://nfs.sourceforge.net/ :

- Fournit un accès réseau transparent aux fichiers d'un serveur (système de fichier réseau)
- Serveur sans état
- Protocole introduit par SUN en 1984, pris en main par l'IETF (Internet Engineering Task Force) depuis 1988
- Protocole ouvert, portable sur de nombreux environnements plateformes/ réseaux, utilise les **RPC** (Remote Procedure Calls) de SUN
- Serveur/Client NFS présent sur tous les Unix
- Clients **NFS** disponibles pour tous les systèmes d'exploitation (Ws,Ux,Mc)
- Version actuelle : 3 (RFC 1813), version 4 en cours (RFC 3530)

# 8 NFS : Présentation

Services/démons utilisés par le service NFS :

**portmap** mise en correspondance numéro de ports TCP/IP <-> numéro de processus RPC (voir /etc/rpc pour les numéros réservés)

côté serveur :

**rpc.nfsd** implémente la partie utilisateur du protocole

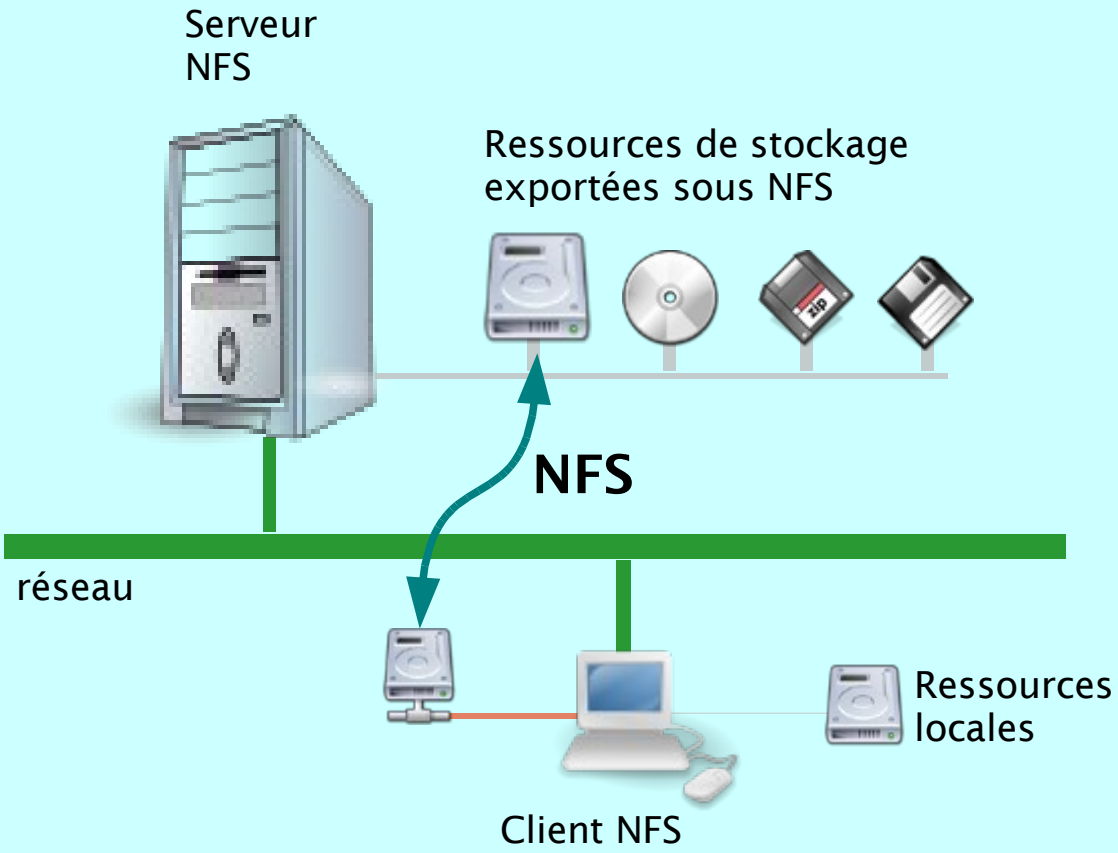
**rpc.statd** prise en compte du redémarrage des serveurs, pour une gestion correcte des vérrous
(implémente le protocole **RPC NSM** (Network Status Monitor))

**rpc.lockd** gestion du vérouillage des fichiers
(implémente le protocole **NLM** (Network Lock Manager))

**rpc.mountd** implémente la partie serveur du protocole, à l'écoute des demandes de montage

côté client : **rpc.statd rpc.lockd**

# 8 NFS : Principe




Le client accède à la ressource
de façon transparente

```
mount -t nfs serveur:path_distant  path_local
```

# 8 NFS : Installation - serveur

**Installation** : **portmap** et **réseau** pré-requis,
serveur -> `urpmi nfs-utils`

Paquet serveur : **nfs-utils**

```
# rpm -ql nfs-utils
/etc/rc.d/init.d/nfs
/etc/sysconfig/nfs
/sbin/nfsddebug
/sbin/nfsdebug
/sbin/rpcdebug
/usr/sbin/exportfs
/usr/sbin/nfsstat
/usr/sbin/nhfsgraph
/usr/sbin/nhfsnums
/usr/sbin/nhfsrun
/usr/sbin/nhfsstone
/usr/sbin/rpc.mountd
/usr/sbin/rpc.nfsd
/usr/share/doc/...
/usr/share/man/man5/...
/usr/share/man/man7/...
/usr/share/man/man8/...
```

Fichiers de lancement

Fichiers de configuration

Fichiers binaires (commande ou démon)

# 8 NFS : Installation – serveur et client

**Installation** : **portmap** et **réseau** pré-requis,

```
serveur ou client  ->   urpmi nfs-utils-client
```

Paquet client : **nfs-utils-client**

```
#rpm -ql nfs-utils-clients
/etc/rc.d/init.d/nfslock
/sbin/rpc.lockd
/sbin/rpc.statd
/usr/sbin/showmount
/usr/share/doc/...
/usr/share/man/...
/var/lib/nfs
/var/lib/nfs/statd
/var/lib/nfs/state
```

Fichiers de lancement

Fichiers de configuration

Fichiers binaires (commande ou démon)

# 8 NFS : Commandes utiles

(voir man pour les détails)

**Serveur**

| | | |
|---|---|---|
| `showmount` | `-a` | liste des clients/répertoire utilisé |
| | `-d` | liste des répertoires montés par des clients |
| | `-e` | liste des répertoires exportés |
| `exportfs` | `-a` | relit le fichier `/etc/exportfs` |
| | `-ra` | remet à jour la liste des répertoires exportés |
| `rpcinfo` | `-p` | afficher les services portmap |

**Client**

| | | |
|---|---|---|
| `rpcinfo` | `-p [host]` | afficher les services portmap |
| `showmount -e host` | | liste des répertoires exportés par le serveur «host» |

# 8 NFS : Configuration serveur


Tous les fichiers de configuration des services spécifiques aux distributions dérivées de RedHat (Fedora, Mandriva, ...)
sont dans le répertoire : `/etc/sysconfig`



`/etc/sysconfig/nfs`

options de lancement du serveur NFS

`/etc/exports`

liste des répertoires exportés par le serveur,
et des options (droits d'accès, ...)

Syntaxe :

```
directory1 host1(option,...) host2(option,...) ....
directory2 host1(option,...) host2(option,...) ....
...
```

Exemple :

```
/opt          192.168.1.0/255.255.255.0(rw,no_root_squash)
/home         192.168.1.0/255.255.255.0(rw)
/mnt/disk     192.168.1.45(rw) pc1(ro)
/usr/local    *(ro)
```

# 8 NFS : Configuration serveur

Principales options du fichier `exports` (serveur) :

| | |
|---|---|
| **ro** | read-only (accès en lecture seule au répertoire exporté) |
| **rw** | read-write : le client accède au répertoire en lecture/écriture |
| **root_squash** | transforme les accès UID root en UID nobody |
| **no_root_squash** | les accès *root* sur le client restent *root* sur le serveur ! |
| **all_squash** | Convertit tous les UID/GID en utilisateurs anonymes. Utile pour exporter avec NFS des répertoires publics. |

Principales options du fichier `fstab` (client) :

| | |
|---|---|
| **hard** | montage avec écriture bloquante (-> attente) |
| **fg, bg** | montage en **for**ground, back**ground** |
| **intr** | montage interruptible (utile avec **hard** en cas de blocage) |
| **soft** | montage sans écriture bloquante (-> pas d'attente) |

Attention : les options doivent être séparées par des virgules, SANS ESPACE.

# 8 NFS : Lancement de NFS sur le serveur

Le lancement des services se fait classiquement :

- Vérifier la présence des scripts de démarrage

  `/etc/init.d/nfs et /etc/init.d/nfslock`

- lancement automatique par lien symbolique (préfixé par `Sxx`) dans

  `/etc/rc3.d` ou `/etc/rc5.d`

- lancement manuel :

  `/etc/init.d/nfs start`, ou `service nfs start (Mandriva)`

- On peut vérifier le fonctionnement en interrogeant portmaper
- On peut visualiser « quelle machine cliente monte quelle ressource » :

  ```
  showmount -a
  cat /proc/fs/nfs/exports
  ```

# 8 NFS : Lancement de NFS sur un client

Coté client, il suffit d'effectuer le montage de la ressource exportée par un serveur NFS :Le lancement des services se fait classiquement :

- À la main : `mount -t nfs serveur:path_distant point_de_montage`
  - `path_distant` : est le nom du répertoire exporté par le serveur
  - `point_de_montage` : est le nom du répertoire local à partir duquel seront utilisable les fichiers du serveur NFS
- Pour automatiser le montage des ressources NFS sur un poste client, on utiliser fichier /etc/fstab.

  Voir « man mount » pour les options de montage supportées par NFS

`/etc/fstab`

```
...
serveur:/home/users /home/users nfs rsize=8192,wsize=8192,soft 0 0
...
```

# 8 NFS : Sécurité

NFS n'est pas un protocole très sécurisé («*No File Security*» !!!) :

- Par défaut l'UID root d'une machine cliente est mappée en l'UID nobody pour tous les accès NFS
- L'authentification des clients repose uniquement sur le nom de domaine ou l'adresse IP => spoofing possible
- L'identification des utilisateurs repose sur le « user id » sur le poste clients => usurpation possible
- Transactions non cryptées
- Utilisation recommandée en intranet isolé protégé de l'InterNet par un Firewall (-> on fait confiance aux utilisateurs locaux !!!)

# 8 NFS : Sécurité

NFS peut utiliser les mécanismes de contrôle d'accès offert par **tcpd**, basé sur les fichiers `/etc/hosts.deny` et `/etc/hosts.allow`

```
# hosts.deny  This file describes the names of the hosts which are
#             *not* allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
portmap:      ALL
lockd:        ALL
mountd:       ALL
rquotad:      ALL
statd:        ALL
```

```
# hosts.allow This file describes the names of the hosts which are
#             allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
portmap:      192.168.0.0/255.255.255.0
lockd:        192.168.0.0/255.255.255.0
mountd:       192.168.0.0/255.255.255.0
rquotad:      192.168.0.0/255.255.255.0
statd:        192.168.0.0/255.255.255.0
```

# Cours d'administration Unix

# 9 DNS : Présentation

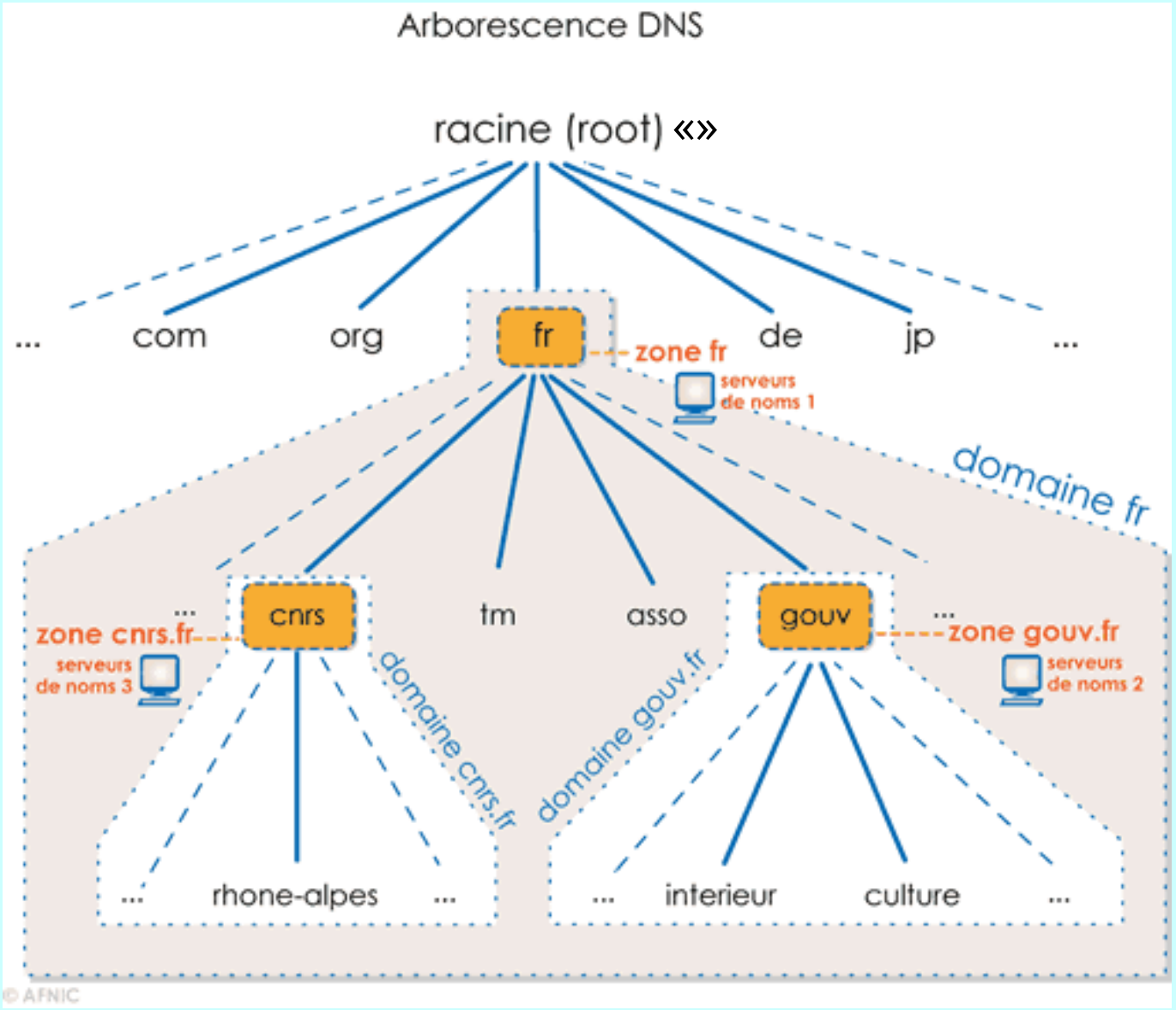
- **DNS** : Système de Nommage par base de données répartie
- Principes
- Installation
- Configuration
- Lancement
- TP

# 9 DNS : Présentation

**DNS** (Domain Name System)

- gère les correspondances entre les noms d'hôtes et les adresse IP :
  Par exple : www.free.fr correspond à l'adresse IP `213.228.0.42`
- À l'origine (années 70) la table des correspondances « nom d'hôte <-> adresse IP » était écrite dans un fichier unique HOSTS.TXT, géré par le Network Information Center (NIC) sur une machine située au Standford Research Institute (SRI) en Californie.
- les problèmes de taille du fichier HOSTS.TXT, de conflit de noms, de charge du réseau et du processeur lors de la diffusion du fichier ... ont montré que le mécanisme basé sur HOSTS.TXT ne pouvait pas suivre l'expansion du réseau
- En 1984 les premières RFC décrivant le « Système de noms de domaine » (DNS) sont publiées.

# 9 DNS : Principe




- les données de chaque noeud sont accessibles de partout (mécanisme de client-serveur)
- duplication (serveurs secondaires) et cache mémoire règlent les aspect de robustesse et performance
- à chaque noeud sont associés des serveurs de noms
- les clients sont des *resolvers* (librairie C)

# 9 DNS : Principe

- le principe arborescent des noms de domaine garantit l'unicité des noms
- nom absolu : www.free.fr.  équivalent à  www.free.fr

  (on parle aussi de **FQDN** : **F**ully **Q**ualified **D**omain **N**ame)
- un domaine est un sous-arbre de l'espace de nommage
- un mécanisme de délégation permet à un domaine de créer autant de sous-domaines qu'il veut ... et ainsi de suite ....
- il ya « délégation d'autorité » d'un domaine vers chacun de ses sous-domaines ... et ainsi de suite ...
- la résolution inverse (obtenir un nom d'hôte d'après une adresse IP) met en oeuvre une branche particulière de l'espace de nommage construite selon :

  d.c.b.a.**in-addr.arpa**  pour une machine d'adresse IP  a.b.c.d

  Par exemple la consultation de l'hôte correspondant à 42.0.228.213.in-addr.arpa donne « www1.free.fr »

# 9 DNS : Principe

- l'implémentation Linux est dérivée de **BIND** (Berkeley Internet Name Domain), écrit à l'origine pour Unix BSD 4.3
- BIND est aujourd'hui maintenu par ISC (Internet Software Consortium)
  http://www.isc.org/bind.html

Services/démons utilisés par le service DNS :

**named** implémentation du serveur DNS

# 9 DNS : Installation - serveur

**Installation** : **réseau** pré-requis, `urpmi bind`

Paquet serveur : bind-9.3.1

/etc/logrotate.d/named
/etc/rc.d/init.d/named
/etc/rndc.conf
/etc/rndc.key
/etc/sysconfig/named
/usr/sbin/bind-chroot.sh
/usr/sbin/dns-keygen
...
/usr/sbin/named
/usr/sbin/rndc
...
/usr/share/doc/bind-9.3.1...
/usr/share/man/man3/...
/usr/share/man/man8/...
...
/var/log/named
/var/named
/var/named/named.ca
/var/run/named

Fichiers de lancement

Fichiers de configuration

Fichiers binaires (commande ou démon)

# 9 DNS : Installation utilitaires – (serveur ou client)

**Installation** : **réseau** pré-requis, `urpmi bind-utils`

Paquet client : bind-utils-9.3.1

```
/usr/bin/dig
/usr/bin/host
/usr/bin/ldap2zone
/usr/bin/nslookup
/usr/bin/nsupdate
/usr/bin/zonetoldap
/usr/share/man/man1/...
/usr/share/man/man5/...
/usr/share/man/man8/...
/usr/share/doc/bind-utils-9.3.1/...
```

Fichiers de lancement

Fichiers de configuration

Fichiers binaires (commande ou démon)

# 9 DNS : Commandes utiles

(voir man pour les détails)

## Commandes utiles

### Client

Un client *resolver* peut être mis en oeuvre avec les commandes `dig`, `host`.
Exemple de syntaxe pour demander au serveur DNS `serv` les infos de type `typ` sur le domaine `dom` :

```
dig [@serv] dom [typ]
host [-a] [-t type] dom [serv]
```

`dnsdomainname` retourne le nom de domaine de l'hôte

### Serveur

On peut agir sur le démon **named** en cours de fonctionnement grâce à la commande `rndc`

| | |
|---|---|
| `rndc` | affiche les options possible |
| `rndc flush` | vide les caches du serveur de nom |
| `rndc status` | affiche l'état du serveur de noms |

# 9 DNS : Configuration commune client & serveur : config réseau

`/etc/nsswitch.conf` Network Service Switch

ordre dans lequel utiliser les différents mode d'accès possibles (fichier, nis, dns ...) pour accéder au contenu des fichiers de configuration

```
    ...
    passwd:      files nis nisplus
    shadow:      files nis nisplus
    group:       files nis nisplus
    ...
    hosts:       files dns nis
    ...
```

Fichiers locaux

DNS

`/etc/host.conf`

ordre dans lequel utiliser les différents mode d'accès possibles (fichier, nis, dns ...) pour résoudre les correspondances adresse IP <=> FQDN
Il est préférable d'utiliser le DNS plutôt que NIS :

```
   ...
   order hosts,bind
   ...
```

# 9 DNS : Configuration serveur : config réseau

on peut préciser le nom du **domaine DNS** dans un fichier de configuration


Tous les fichiers de configuration des services spécifiques aux distributions dérivées de RedHat (Fedora, Mandriva, ...)
sont dans le répertoire : /etc/sysconfig



`/etc/sysconfig/network`

```
HOSTNAME=hostaaa.mydomaine.fr
NETWORKING=yes
GATEWAY=192.168.74.1
```

Le fait d'utiliser un **nom FQDN** pour **HOSTNAME** permet de fixer le **domaine DNS** de la machine

`/etc/nsswitch.conf`

`/etc/host.conf`

# 9 DNS : Configuration serveur maître : le fichier named.conf

**/etc/named.conf** : fixe les options principales du serveur DNS et les domaines sur lesquels le serveur a autorité.

Exemple de fichier d'un serveur DNS primaire ayant autorité sur le domaine `mydomaine.fr` :

même clef que celle définie pour **rndc** dans /etc/rndc.conf

`/etc/named.conf` serveur DNS maître

```
key "key" {
   algorithm hmac-md5;
   secret "c3Ryb25nIGVub3Bmb3I...YnV0IG1hZGUgZm9yIGEgd29tYW4K";
};
controls {
   inet 127.0.0.1 allow { 127.0.0.1; } keys { "key"; };
};
options {
   pid-file "/var/run/named/named.pid";
   directory "/var/named";
};
...
```

# 9 DNS : Configuration serveur maître : fichier named.conf

/etc/named.conf serveur DNS maître : suite

```
zone "." {
        type hint;
        file "named.cache";
};

zone "mydomaine.fr"{
        type master;
        file "named.mydomaine";
};

zone "74.168.192.in-addr.arpa"{
        type master;
        file "named.74.168.192";
};

zone "0.0.127.in-addr.arpa"{
        type master;
        file "named.0.0.127";
};
```

définit la racine de type **hint** à partir d'un fichier zone

définit la zone mydomaine.fr de type **master** à partir d'un fichier zone

définit la zone inverse 192.168.74 de type **master** à partir d'un fichier zone

définit la correspondance : 127.0.0.1<->localhost

# 9 DNS : Configuration serveur : structure d'un fichier de zone

Principaux types d'enregistrement d'un fichier zone DNS

| | |
|---|---|
| **SOA** | start of authority : définit le serveur DNS fournissant l'information faisant autorité sur un domaine Internet |
| **NS** | name server : définit les serveurs DNS de ce domaine |
| **MX** | mail exchange : définit les serveurs mail pour ce domaine |
| **A** | address : associe une adresse IPv4 à un nom d'hôte |
| **AAAA** | IPv6 address : associe une adresse IPv6 à un nom d'hôte |
| **PTR** | pointer : associe un nom de domaine à une adresse IP |
| **CNAME** | canonical name : déclare un nom alias d'un autre |
| **SRV** | généralise la notion de MX record, standardisé dans la RFC 2782. |

# 9 DNS : Configuration serveur : structure fichier zone

```
@     IN      SOA   serv1.mydomain.fr. jl.mydomain.fr. (
              2005112501 ; numéro série (annee mois jour xy)
              21600      ; rafraichissement après 6h
              3600       ; nouvel essai après 1h
              604800     ; expiration après 7 jours
              86400 )    ; TimeToLive (TTL) de 1 jour
; serveurs (primaire et secondaires ...)
              IN NS      serv1.mydomaine.fr.
              IN NS      serv2.mydomaine.fr.
; échangeur de courier SMTP
              IN MX 0     mail.mydomaine.fr.
; machines du domaine
localhost     IN A       127.0.0.1
serv1         IN A       192.168.74.1
serv2         IN A       192.168.74.2
mail          IN A        192.168.74.10
asterix       IN A        192.168.74.100
...
; alias de machines
ftp           IN CNAME    serv2
```

ATTENTION : tous les noms d'hôtes DOIVENT être terminés par un « . »

# 9 DNS : Configuration serveur : structure fichier zone

- le symbole @ indique le domaine auquel s'applique le fichier (ici mydomaine.fr).

  => les noms partiellement qualifiés sont relatifs à ce domaine
- Structure du début d'autorité **SOA** :
  - rappel du nom du serveur-maître  (serveur1.mydomaine.fr)
  - adresse mail responsable du serveur (caractère « @ » remplacé par « . » exple : jl@mydomaine.fr -> jl.mydomaine.fr)
  - parenthèse ouvrante « («
    - numéro de série
    - intervalle entre 2 tentatives des serveurs secondaires de se resynchroniser sur le serveur primaire
    - attente avant une nouvelle tentative lorsque la resynchronisation d'un serveur secondaire a échoué
    - délai au bout duquel les données expires sur les serveurs secondaires
    - durée de vie sur les serveurs autres que les serveurs secondaires
  - parenthèse fermante « ) »

# 9 DNS : Configuration serveur : structure fichier zone inverse

La zone inverse **74.168.192.in-addr.arpa** décrit les machines du réseau 192.168.74.0:

```
@     IN      SOA   serv1.mydomain.fr. jl.mydomain.fr. (
              2005112501 ; numéro série (annee mois jour xy)
              21600      ; rafraichissement après 6h
              3600       ; nouvel essai après 1h
              604800     ; expiration après 7 jours
              86400 )    ; TimeToLive (TTL) de 1 jour
;
              IN NS      serv1.mydomaine.fr.
              IN NS      serv2.mydomaine.fr.

1     IN      PTR      serv1.mydomaine.fr.
2     IN      PTR      serv2.mydomaine.fr.
10    IN      PTR      mail.mydomaine.fr.
...
100   IN      PTR      asterix.mydomaine.fr.
...
```

# 9 DNS : Configuration serveur : structure fichier zone inverse

La zone inverse **0.0.127.in-addr.arpa** décrit les machines du réseau 127.0.0.0:

```
@     IN      SOA   localhost. root.localhost. (
              2005112501 ; numéro série (annee mois jour xy)
              21600      ; rafraichissement après 6h
              3600       ; nouvel essai après 1h
              604800     ; expiration après 7 jours
              86400 )    ; TimeToLive (TTL) de 1 jour
;
              IN NS      localhost.

; résolution inverse 127.0.0.1 -> localhost
1             IN PTR     localhost.
```

# 9 DNS : Configuration serveur : fichier de zone racine

On créé ce fichier par la commande :

```
dig @a.root-servers.net > named.cache
```

```
; <<>> DiG 9.3.0 <<>> @a.root-servers.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42818
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUESTION SECTION:
;.                              IN      NS
;; ANSWER SECTION:
                        518400  IN      NS      A.ROOT-SERVERS.NET.
                        518400  IN      NS      H.ROOT-SERVERS.NET.
...
                        518400  IN      NS      E.ROOT-SERVERS.NET.
                        518400  IN      NS      D.ROOT-SERVERS.NET.
;; ADDITIONAL SECTION:
A.ROOT-SERVERS.NET.     3600000 IN      A       198.41.0.4
H.ROOT-SERVERS.NET.     3600000 IN      A       128.63.2.53
...
D.ROOT-SERVERS.NET.     3600000 IN      A       128.8.10.90
...
```

# 9 DNS : Configuration serveur : le fichier rndc.conf

/etc/rndc.conf

```
options {
        default-server  localhost;
        default-key     "key";
};

server localhost {
        key     "key";
};

key "key" {
        algorithm       hmac-md5;
        secret "c3Ryb25nIGVub3VnaCB...nV0GUgZm9yIGEgd29tYW4K";
};
```

# 9 DNS : Initialisation, lancement du serveur

Après l'installation du paquet serveur **DNS** :

- 1- Configurer le nom de domaine DNS
- 2- Éditer si besoin le fichier `/etc/host.conf` : choix de l'ordre des méthodes de résolution FQDN<->adresse IP (local, dns, nis ?)
- 5- Éditer si besoin le fichier `/etc/nsswitch.conf` : choix de l'ordre des méthodes d'accès aux fichiers
- 6- Éditer le fichier `/etc/named.conf` : configurer au besoin
- 7- Créer le fichier de zone racine
- 8- Éditer les fichiers de zone
- 9- Lancer le serveur.
- 10 – Tester avec les commandes `host` ou `dig`

# 9 DNS : Configuration client : réseau

**Client fixe** => on peut préciser le nom du **domaine DNS**

`/etc/sysconfig/network` Mandriva

```
HOSTNAME=hostaaa.mydomaine.fr
NETWORKING=yes
GATEWAY=192.168.74.1
NISDOMAIN=tpAL
```

**Client nomade (portable)**

- on peut préciser le nom du domaine DNS dans le fichier `network`, mais PB quand on change de réseau !
- Il est plus intéressant d'utiliser la configuration dynamique **DHCP** pour récupérer les informations « nom de domaine DNS» et « serveur DNS »

`/etc/dhclient-eth0.conf` Mandriva

`/etc/dhclient.conf` debian

```
...
request subnet-mask, broadcast-address, time-offset,routers,
domain-name, domain-name-servers, host-name, nis-domain;
...
```

# 9 DNS : Configuration client (resolver)

`/etc/resolv.conf`

Contient les informations lues par les routines de la bibliothèque resolver :

| | |
|---|---|
| **namesever** | adresse IP d'un serveur DNS accessible |
| **search** | nom du domaine qui sera rajouté par défaut aux noms de domaine partiellement qualifiés |

`Exemple  /etc/resolv.conf`

```
search mydomaine.fr
nameserver 192.168.74.1
```

`/etc/nsswitch.conf`

`/etc/host.conf`

# 9 DNS : Configuration serveur esclave : le fichier named.conf

Il est « plus que conseillé » de configurer un ou plusieurs serveur DNS esclave par  zone.
Le serveur DNS esclave reçoit la copie des fichiers zones envoyés par le serveur DNS maître.

Le début du fichier named.conf est identique à celui du serveur maître

/etc/named.conf serveur DNS eclave

```
key "key" {
   algorithm hmac-md5;
   secret "c3Ryb25nIGVub3Bmb3I...YnV0IG1hZGUgZm9yIGEgd29tYW4K";
};
controls {
   inet 127.0.0.1 allow { any; } keys { "key"; };
};
options {
   pid-file "/var/run/named/named.pid";
   directory "/var/named";
};
...
```

# 9 DNS : Configuration serveur esclave : fichier named.conf

/etc/named.conf serveur DNS esclave : suite

```
zone "." {
        type hint;
        file "named.cache";
};
zone "mydomaine.fr"{
         type slave;
         masters { 192.168.74.1; }
         file "named.mydomaine";
};
zone "74.168.192.in-addr.arpa"{
         type slave;
         masters { 192.168.74.1; }
         file "named.74.168.192";
};
zone "0.0.127.in-addr.arpa"{
        type master;
        file "named.0.0.127";
};
```

définit la racine de type **hint** à partir d'un fichier zone

définit la zone mydomaine.fr de type **slave**

définit la zone inverse 192.168.74 de type **slave**

définit la correspondance : 127.0.0.1<->localhost

# Cours d'administration Unix

# 9 LDAP : Présentation

- **LDAP** : un annuaire centralisé
- Présentation
- Installation
- Configuration
- Lancement
- TP

# 9 LDAP : Présentation

**LDAP** (Lightweight Directory Access Protocole ; http://www.openldap.org)

- Protocole ouvert d'accès à un **Annuaire** (1993), résultant de l'adaptation de la norme X.500 à TCP/IP
- Les **Annuaires** sont des Bases de Données particulières spécialisées dans la recherche de l'information, pas dans le traitement
- L'organisation d'un annuaire LDAP est hiérachique et dynamique (structure et contenu)
- Évolutivité : on peut facilement ajouter des informations à un objet d'un annuaire (structure et contenu)
- La dernière version **LDAPv3** propose chiffrement (SSL, ...) et authentification (SASL) pour sécuriser l'accès aux informations
- **OpenLDAP** est un annuaire dérivé de LDAP de l'Université du Michigan

# 9 LDAP : Présentation

- Un annuaire permet de stocker des données typées,
  - organisées selon des classes particulières
  - présentées dans un arbre.
- L'exemple le plus commun est l'annuaire de personnes
- Mais on peut stocker bien d'autres choses :
  - comptes Unix,
  - données personnelles (carnet d'adresses, photos, etc.),
  - données d'identification,
  - certificats ...
- ... plus généralement tout ce qui peut être nommé et à qui on peut attacher des informations.

# 9 LDAP : Présentation – Principe d'une session

- Un client ouvre une session LDAP sur le port TCP 389 du serveur.
- Le client envoie des requêtes au serveur qui envoie des réponses en retour.
- Une fois la connexion au serveur établie, les opérations classiques sont :
  - **Bind** : indique la version du protocole utilisée, et authentifie l'utilisateur.
  - **Start TLS** : utilisation  Transport Layer Security pour sécuriser la connexion
    - o **Search** : recherche dans l'annuaire et rapatriement des données ;
    - o **Compare** : test si une entrée contient un attribut avec une valeur donnée
    - o **Add** : ajout d'une nouvelle entrée ;
    - o **Delete** : suppression d'une entrée ;
    - o **Modify** : modification d'une entrée ;
    - o **Modify DN** : déplacement ou renommage d'une entrée ;
    - o **Abandon** : annulation d'une requête précédente ;
    - o **Extended** : permet de définir d'autres opérations ;
  - **Unbind** : clôture la connexion.

# 9 LDAP : Présentation

**Exemples de serveurs LDAP :**

- Serveurs LDAP
- Apache Directory Server
- Fedora Directory Server
- Red Hat Directory Server
- OpenLDAP
- Novell eDirectory
- Sun Directory Server Enterprise Edition
- IBM SecureWay Directory
- IBM Tivoli Directory Server (formerly IBM Directory Server)
- IBM Lotus Domino
- Windows Server 2003 Active Directory
- Oracle Internet Directory
- ...

# 9 LDAP : Présentation – L'arborescence d'informations (DIT)

- Les informations d'un annuaire sont organisées selon une arborescence hiérarchique (le **DIT** : Directory Information Tree)
- les informations sont des entrées (**DSE** : Directory Service Entry)
- Au sein du **DIT**, l'identification d'une entrée se fait à l'aide d'un nom,le **Distinguish Name** (DN).
  - **Relative Distinguished Name** (RDN)

    Exemple : `mail=Audrey.Tautou@inpg.fr, uid=tautoua, etc.`
  - **Distinguished Name** (DN)
    - RDN + chemin dans l'arborescence en remontant
    - attribut du RDN à choisir pour que tout DN soit unique

    Exemple : `uid=tautoua,ou=people,ou=inpg,dc=agalan,dc=org`

# 9 LDAP : Présentation – L'arborescence d'informations (DIT)

- Les **entrées** de l'annuaire

  - sont des **objets** (correspond à un objet abstrait ou réel ),

  - appartiennent à des **classes**, définissant des attributs

- les attributs, les syntaxes et les classes d'objets sont identifiés à l'aide d'un **numéro unique**, **OID** (Object Identifier).
- L'ensemble des attributs, de leur syntaxe, des règles de comparaison et des classes d'objets, constitue le **schéma** de l'annuaire.

# 9 LDAP : Présentation – Les Classes

## Une classe

- est constituées d'attributs obligatoires (MUST) ou optionnels (MAY)
- est de type structurelle, auxiliaire ou abstraite
- s'inscrit dans un arbre d'héritage de classes (SUP : classe mère)
- LDAP définit des classes d'après X.500
- On peut en fabriquer de nouvelles !

## Exemple de classe

```
inetorgPerson ( 2.16.840.1.113730.3.2.2
    NAME 'inetOrgPerson'
    SUP organizationalPerson
    STRUCTURAL
    MAY ( audio $ businessCategory $ carLicense $ departmentNumber $
          displayName $ employeeNumber $ employeeType $ givenName $
          homePhone $ homePostalAddress $ initials $ jpegPhoto $...
          x500uniqueIdentifier $ preferredLanguage $
          userSMIMECertificate $ userPKCS12
        )
)
```

# 9 DNS : Installation - serveur

**Installation** : **réseau** pré-requis, serveur -> `urpmi openldap-servers-2.3.6`

Paquet serveur : **`openldap-servers-2.3.6`**

```
/etc/logrotate.d/ldap
/etc/openldap
/etc/openldap/DB_CONFIG.example
/etc/openldap/schema
/etc/openldap/schema/local.schema
/etc/openldap/slapd.access.conf
/etc/openldap/slapd.conf
/etc/rc.d/init.d/ldap
/etc/ssl/openldap
/etc/sysconfig/ldap
/usr/lib/openldap
/usr/lib/openldap/accesslog-2.3.so.0...
/usr/sbin/slapacl
/usr/sbin/...
/usr/sbin/slapd...
/usr/sbin/slappasswd...
/usr/share/doc/openldap-servers-2.3.6 ...
/usr/share/man/man5/slapd-bdb.5.bz2    ...
/usr/share/openldap/schema...
/usr/share/openldap/schema/autofs.schema
...
```

Fichiers de configuration

Fichiers de lancement

Fichiers binaires (commande ou démon)

nom du binaire ldap (Stand-alone LDAP Daemon)

# 9 DNS : Installation - serveur

**Autres paquets à installer :**

- **openldap-clients**

  fournit les commandes : ldapadd, ldapdelete, ldapmodify, ldapsearch, ...
- **nss_ldap**

  fournit l'interfaçage entre **LDAP** et `le` **`Name Service Switch`** `(fichier /etc/nsswitch)`
- **pam_ldap**

  fournit l'interfaçage entre **LDAP** et le **Linux-PAM** - (Module d'authentification pour Linux)

# 9 LDAP : Configuration - serveur

## Configuration du serveur **ldap stand-alone**

`/etc/openldap/slapd.conf`

fichier ASCII contenant différents types de ligne :

- des include de schémas pré-établies :

```
include         /usr/share/openldap/schema/nis.schema
```

- des définitions d'ACL (Access Control List) complémentaires

```
include         /etc/openldap/slapd.access.conf
```

- des définitions de base :

```
database        bdb
suffix          "dc=world-company,dc=com"
```

Préconisations de l'IETF :

-> du nom de domaine DNS comme suffixe de son annuaire

-> l'utilisation de l'attribut Domain Component (dc) :

dc=world-company, dc=com

# 9 LDAP : Configuration - serveur

`/etc/openldap/slapd.conf` suite

- le répertoire ou seront stockés les fichiers de données ldpap :

```
directory       /var/lib/ldap
```

- Le nom du super-utilisateur pour cette base :

```
rootdn      "cn=admin,dc=example,dc=com"
```

- et son mot de passe :

```
rootpw      {SSHA}/egZlrlv21XhkwRsU2cjzZ6F0Upd31ar
```

Le mot de passe crypté peut être généré par la commande `slappasswd` :

```
[root@mars ~]# slappasswd -h '{SSHA}' -s secret -v
[root@mars ~]# {SSHA}/egZlrlv21XhkwRsU2cjzZ6F0Upd31ar
```

# 9 LDAP : le langage textuel LDIF

- **LDAP Data Interchange Format** (LDIF) est le standard de représentation des entrées sous forme texte.
- Utilisé pour afficher/modifier les données de la base suivant deux modes :
  - faire des imports/exports de base,
  - faire des modifications sur des entrées.
- Le format utilisé est l'ASCII.
- Toute valeur d'attribut ou tout DN qui n'est pas ASCII, est codé en base 64.

Des exemples de fichiers LDIF seront vus en TP.

CRU (Comité Réseau des Universités):
http://www.cru.fr/ldap/