

# Administration Linux

Esial 2A  
(2004-2005)

- Martin QUINSON -  
d'après Cyril Lamy

# Plan de la présentation

- Introduction
- Présentation de Linux
- Démarrage de Linux
- Administration du système

# Introduction

- Rôle de l'administrateur système
- Choix d'un système d'exploitation

# Rôle de l'administrateur système

- Installation des systèmes d'exploitation
- Gestion des comptes utilisateurs
- Gérer les fichiers et les disques
- Organiser les sauvegardes
- Ajout / Suppression de périphériques
- Gestion du parc informatique
- Installation des nouveaux produits
- Veille à la sécurité du système

# Critères de choix d'un système d'exploitation

- On choisit un système d'exploitation principalement en fonction des logiciels que l'on compte utiliser, du type de personnes qui auront à effectuer le suivi du système (taches d'administration courantes), de l'environnement informatique existant.
- On choisit aussi le système en fonction du coût (humain et financier) des différentes solutions envisageables.

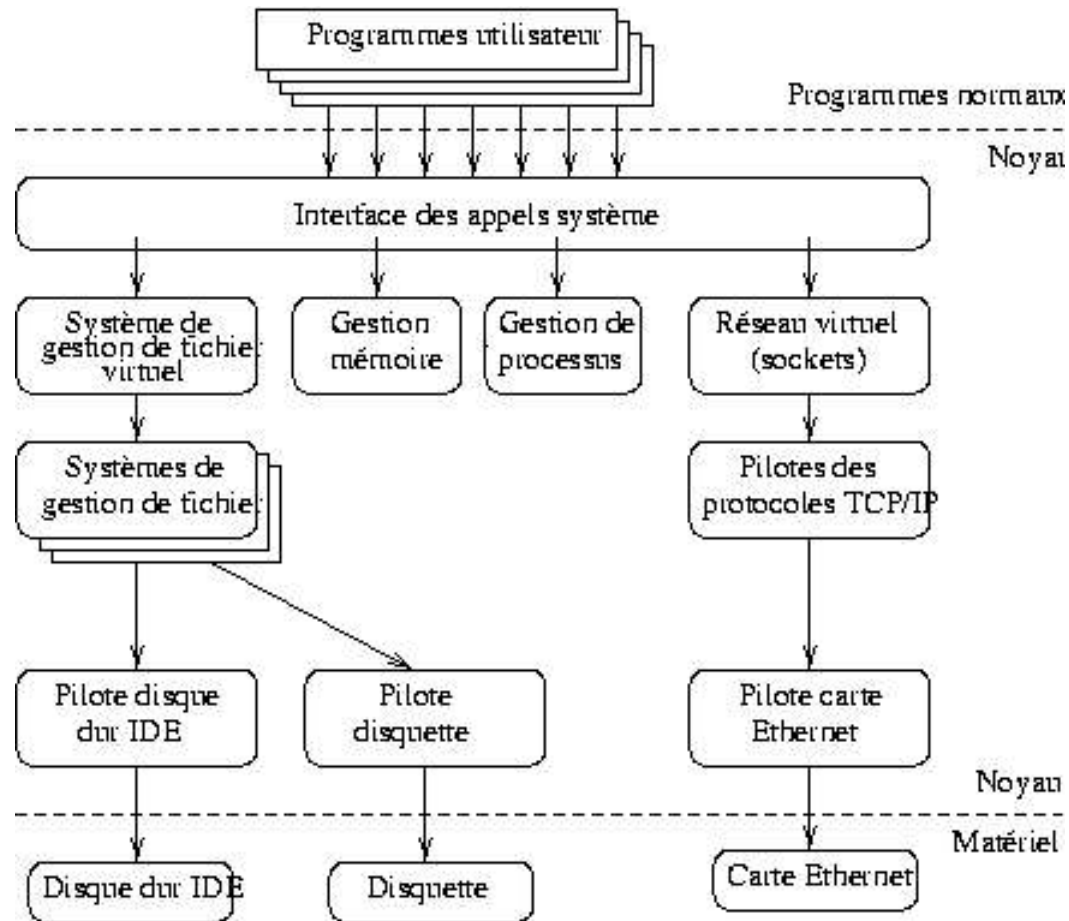
# Présentation de Linux

- Système Unix initialement créé par un étudiant finlandais Linus Torvalds puis aidé par de nombreux bénévoles
- Première version opérationnelle sortie fin 1991 (v 0.9.9)
- Distribué sous licence GPL (Gnu Public License)
- Version 1.0 sortie en 1994

# Notions sur Unix

- Noyau et modules
- Fichiers
- Disques et autres média

# Composants principaux d'un système d'exploitation





# Le noyau

- Ensemble logiciel assurant l'interface entre le matériel et les processus (systèmes ou utilisateurs)
  - Chargé au démarrage du système, reste en mémoire
  - Une fois initialisé, il détecte les périphériques et lance le premier processus (init)
- Il peut être monolithique ou à modules dynamiques
- Entièrement configurable par l'administrateur



# Modules du noyau

- Module = extension du noyau à charger dynamiquement  
  .o placé dans /lib/modules/version\_de\_noyau
- Pilotes matériels sont les modules les plus courants
- Commandes :
  - lsmod : liste les modules
  - insmod : charge un module
  - rmmod : retire un module
  - depmod : établit les dépendances inter-modules
  - modprobe : charge ou retire modules et dépendances
- Fichiers :
  - /etc/modules.conf : configuration de modprobe
  - /lib/modules/??\*/modules.dep : liste des dépendances

# Notions sur Unix

- Noyau et modules
- ▶ **Fichiers**
- Disques et autres média

# Unix et les fichiers

- Sous UNIX, tout est représenté par un fichier données (bien sûr), mais aussi périphérique, mémoire...
- Différents types de « fichiers »  
Les plus courants : normal (-), répertoire (d)  
Mais aussi : lien symbolique (l), périphérique (c), etc.
- Commande pour voir le détail des fichiers : `ls -l`  

```
-rw-r--r--  1 mquinson  users  1055740 Jun 13 16:42 toto
```

# Les droits des fichiers

## Les droits d'accès aux fichiers sont limités

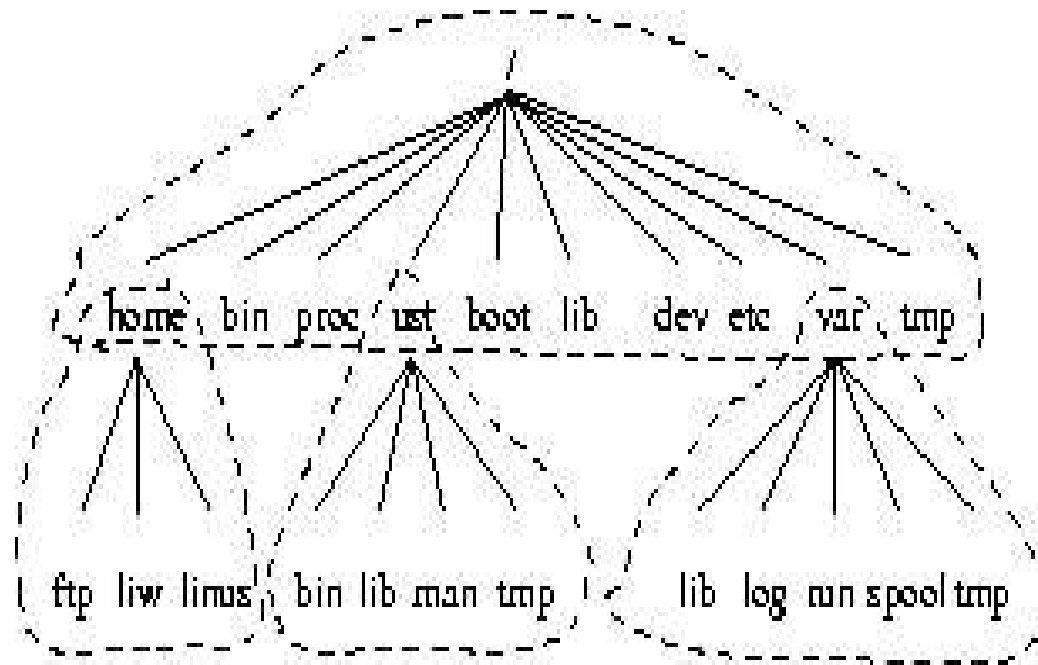
- On distingue trois classes d'utilisateurs par fichier :  
Propriétaire ; Membres de son groupe ; Autres
- Pour chaque classe, trois droits d'accès de base :  
Lecture (r) ; Écriture (w) ; Exécution (x)  
*Question* : que veut dire exécution d'un répertoire?
- Changer les droits (des fichiers dont on est propriétaire)
  - `chmod o+x toto` (donne le droit x aux autres – other)  
[ugo][+][rwx]
  - `chmod 755 toto` (fixe les droits à `rw-r-xr-x`)  
 $r = 4$  ;  $w = 2$  ;  $x = 1$

# Les droits particuliers des fichiers

- **Sticky bit** (t – 1000)  
Reste en mémoire après exécution (chargement rapide)  
(Seul le super-utilisateur peut donner ce droit)
- **Droits d'endossement** (s)  
S'exécutent sous l'identité du propriétaire
  - setuid (user identity) : 4000
  - setgid (group identity) : 2000

# L'arborescence des fichiers

- Les fichiers sont rangés sous forme d'arbre  
Racine unique → différence avec windows (a:, c:, d:)
- Le standard FHS précise la place des fichiers  
Filesystem Hierarchy Standard ; plus ou moins suivi





# L'arborescence des fichiers

/etc : emplacement de la configuration

/home : fichiers des utilisateurs

  /home/mquinson et /home/toto etc.

/dev : fichiers spéciaux des périphériques

/proc (ou /sys) : fichiers spéciaux interface avec le noyau

/usr : programmes classiques

  /usr/bin : les exécutable eux-mêmes

  /usr/lib : bibliothèques

  /usr/sbin : exécutable réservés au super-utilisateur

  /usr/share/doc et /usr/share/man : documentation

/bin et /sbin : programmes nécessaires au démarrage

/tmp : fichiers temporaires (effaçables)

/var : ce qui change au cours du temps (comme DB)

# Les fichiers de configuration (/etc)

- Configuration du système en général

/etc/rc : scripts de démarrage

/etc/passwd : DB utilisateurs, mots de passe, etc.

/etc/fstab : configuration des montages disques

/etc/motd : mot du jour affiché au login

/etc/profile : configuration des shells

- Configuration de chacun des programmes  
réglages pour tout le système (non modifiable par les utilisateurs)

Un fichier ou un répertoire par programme.

# Les programmes (/usr et /)

/usr/bin : presque tous les programmes du système

/usr/sbin : les programmes réservés au super-utilisateur

/usr/share : ce qu'on peut partager entre architecture

  /usr/share/doc : la documentation (format libre)

  /usr/share/man : documentation au format man

/usr/lib : les bibliothèques et autre

  (dépend de l'architecture sans être exécutable)

/usr/include : les entêtes (.h du langage C)

/usr/local : fourre tout pour les installations manuelles

  /usr/local/bin, /usr/local/lib etc.

/bin et /sbin : programmes nécessaires au démarrage



# Interface avec le noyau (/proc ou /sys)

## Points d'interaction avec le noyau

/proc/123 : informations sur le processus numéro 123

/proc/self : idem pour le processus appelant

/proc/cpuinfo : informations sur le(s) processeur(s)

/proc/kcore : la mémoire physique (pour accès direct)

/proc/stat : diverses infos sur le système

/proc/modules : listes des modules chargés actuellement

/proc/loadavg : charge actuelle

/proc/net : interactions avec les modules réseau

# Les fichiers de périphériques (/dev)

- Points d'interaction avec les pilotes du noyau (et donc avec le matériel)

- Ils sont créés avec MAKEDEV ou mknod :

```
mknod /dev/ttyS0 c 4 64
```

```
chown root.dialout /dev/ttyS0
```

```
chmod 0644 /dev/ttyS0
```

```
ls -l /dev/ttyS0
```

```
crw-rw---- 1 root dialout 4, 64 Oct 23 18:23 /dev/ttyS0
```

**/dev/dsp** : carte son

**/dev/fd0** : premier lecteur de disquette

**/dev/psaux** : port souris PS/2

**/dev/null** : écritures ignorées (poubelle)

**/dev/zero** : lecture donne que des zéros

**/dev/random** : lecture donne des données aléatoires

# Le reste (/)

/mnt : point de montage des (autres) disques

/tmp : fichiers temporaires

/boot : fichiers nécessaire au démarreur

# Notions sur Unix

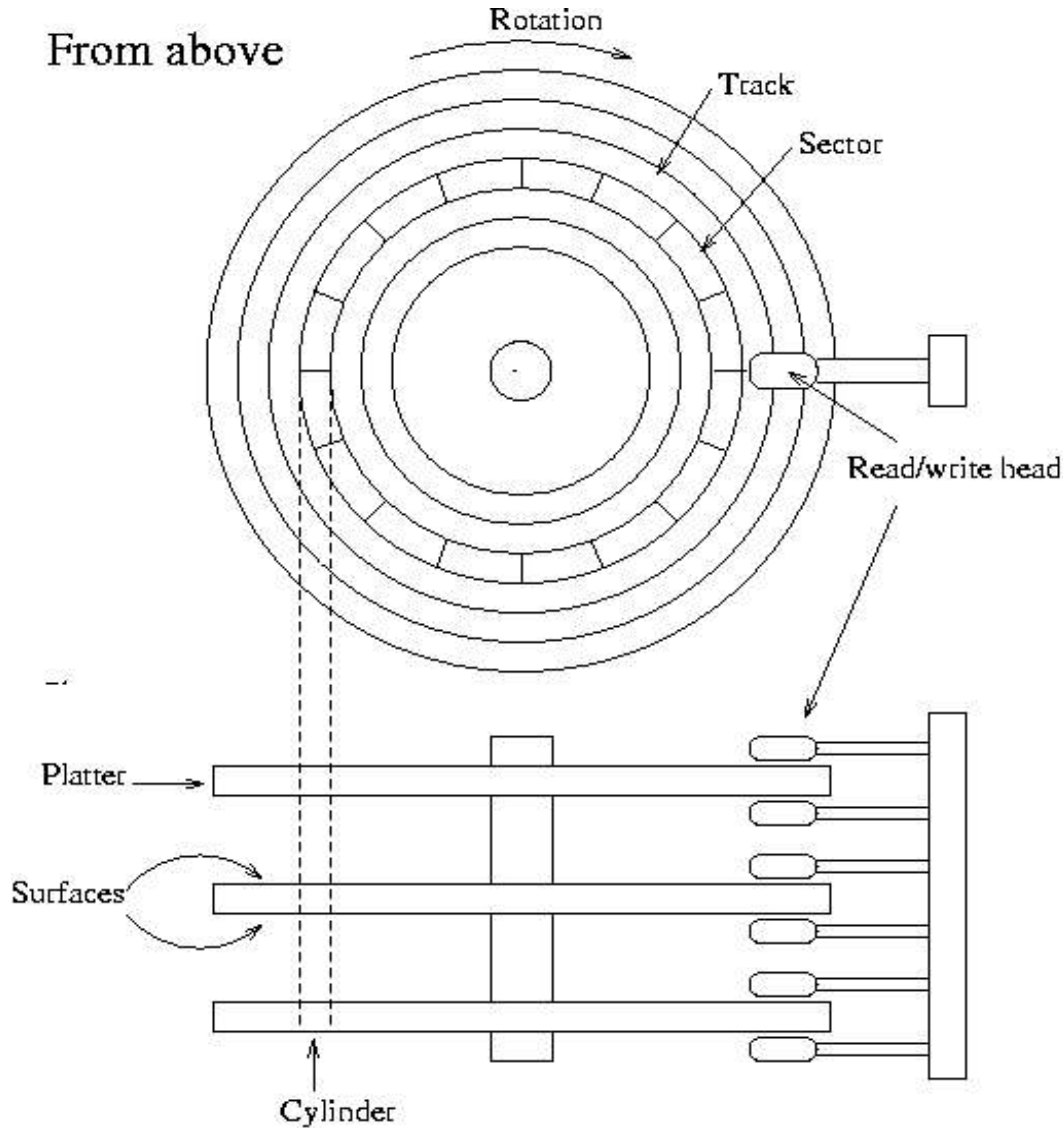
- Noyau et modules
- Fichiers
- ▶ **Disques et autres média**



# Disques et autres média

- Deux types de média :
  - Accès direct par bloc (block device) : disques durs
  - Accès par caractère (character device) : bandes
- Ce qu'on peut faire avec les disques :
  - Formater (préparation – marque pistes et secteurs)
  - Partitionner (découper en disques durs logiques)
  - Créer un système de fichiers (définir la représentation physique des données logiques)
  - Monter les disques (attribuer une place dans l'arbre)

# Schéma physique d'un disque dur

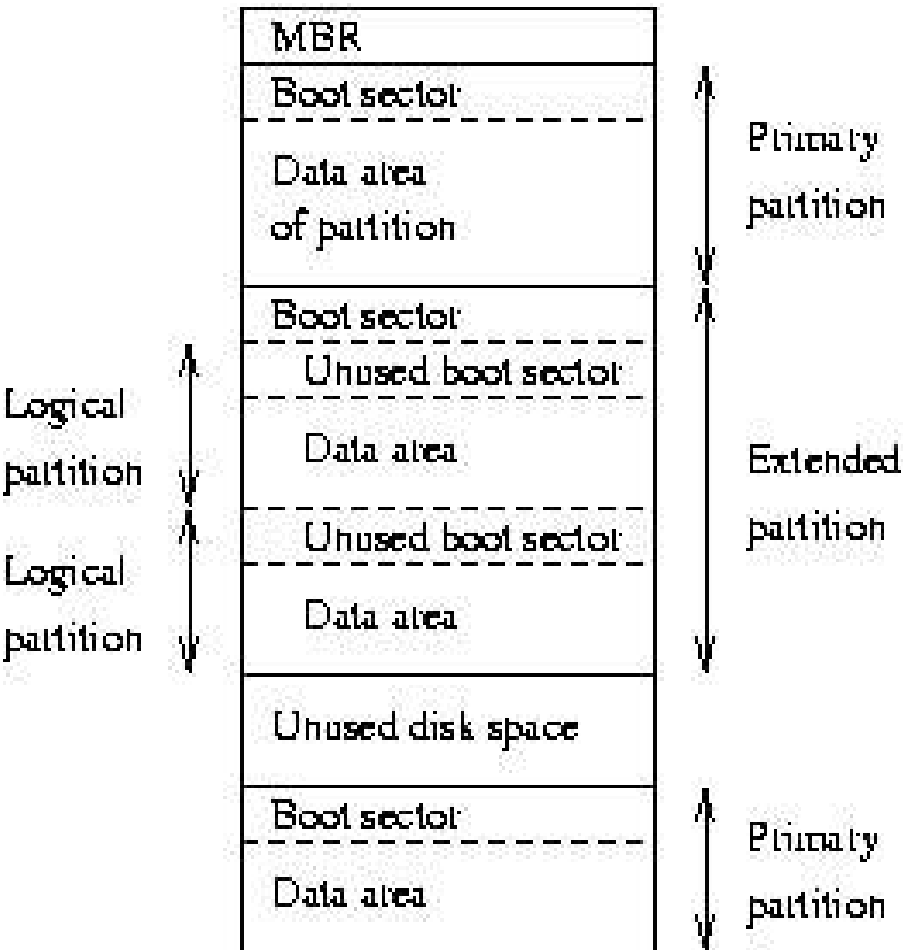


**Formater :**

Marquer les pistes et secteurs sur le support magnétique.

**Outils :** fdformat(8) et badblocks(8)

# Schéma logique d'un disque dur



## Partitionner :

Couper le disque physique en disques logiques (un disque, deux OS).

**MBR** : master boot record

- table des partitions
- chargeur de démarrage

## Partitions primaires, étendues et logiques

4 partitions primaires par disque au plus.  
 Partition *étendue* = une *primaire* contenant des « sous-partitions » (*logiques*)

## Partitions et fichiers de périphérique

Ex : /dev/hda0 1er disque, 1ère partition

# Le système de fichiers

- Représentation physique des données sur disque
- Moyen d'accès aux données (logiciel)  
Interface logique d'accès aux données  
Personne n'accède directement au disque
- Type de FS existants :
  - Linux : ext2, ext3, reiserfs, nfs
  - Dos/windows : vfat, ntfs, smbfs
  - CD-ROM : iso9660
- Journalisation : FS valide à tout instant  
=> ok même si on coupe le courant (ext3, reiserfs, ntfs).
- Outil de création de système de fichier : mkfs(8)

# Le système de fichiers

- Tous les systèmes de fichiers comportent au moins trois tables systèmes :
  - **Superbloc**  
liste des emplacements libres et occupés
  - **Table des inodes**  
informations sur chaque fichier (ls -l) + pointeur vers données
  - **Les répertoires**  
chainage des fichiers depuis la racine
- Le reste du disque est utilisé pour les blocs de données



# Le montage des systèmes de fichiers

- Montage = placement du contenu d'un disque dans l'arbre
- **Commande** : `mount(8)`
- **Exemple** : `mount -t vfat /dev/fd0 /mnt/floppy`

Place le contenu (répertoires et fichiers) de la disquette (/dev/fd0) sous le répertoire /mnt/floppy. Le système de fichier à utiliser est vfat (dos, très courant).

# Démarrage de Linux

matériel, bios, chargeur,  
noyau, init et utilisateur



# Le démarrage de Linux

## matériel vers bios

- Lorsque un ordinateur démarre, en premier lieu, le BIOS prend le contrôle de la machine : il vérifie l'intégrité de la machine, recense les périphériques et en teste certains.
- Une fois cette phase terminée, il faut démarrer le système d'exploitation. Pour cela, le BIOS va examiner (dans un ordre défini dans sa configuration (boot sequence)) les périphériques disponibles pour tenter de trouver un système d'exploitation. Celui-ci peut être stocké sur une disquette, disque dur, cdrom, réseau, ...

# Le démarrage de Linux

## bios vers MBR et chargeur

- Dans le cas le plus courant, le système d'exploitation se trouve sur une partition du disque dur.
- Le bios va donc lire les 512 premiers octets du périphérique d'amorçage, les stocker en RAM puis l'exécuter. Ces 512 premiers octets s'appellent le Master Boot Record (MBR).
- Comme 512 octets n'est pas suffisant pour stocker le noyau Linux (la taille du noyau dépend des options avec lesquelles il a été compilé mais un noyau «moyen» a une taille de 800 Ko), le MBR contient un chargeur (LILO) qui a pour rôle d'aller chercher le noyau sur le disque et de le charger en mémoire.

# Le démarrage de Linux

## chargeurs

- Suivant l'installation, deux cas sont possibles : LILO est installé dans le MBR ou alors il est installé sur le 1er secteur (secteur de boot) de la partition Linux.
- Dans le premier cas, c'est LILO qui sera directement exécuté, sinon le MBR contiendra un programme qui ira charger en mémoire les 512 premiers secteurs (secteur de boot) de la partition active et qui les exécutera ensuite. Si la partition Linux est active, ce programme chargera donc LILO.

# Le démarrage de Linux

## chargeur vers OS

- Ensuite, si LILO est bien chargé, il affiche un prompt. En appuyant sur 'Tab', il affiche la liste de tous les noyaux qu'il peut booter. Si l'on appuie sur Entrée, il chargera le noyau par défaut sinon on peut lui entrer le nom du noyau à booter.
- Étant donné que Lilo a une taille extrêmement réduite, il lui est impossible d'accéder aux systèmes de fichiers via les pilotes de périphérique. C'est pourquoi LILO stocke dans le fichier /boot/map la position physique du ou des noyaux sur le disque dur afin d'accéder directement à ceux-ci secteur par secteur.
- Lorsqu'on charge le noyau, LILO affiche «Loading Linux ...»

# Le démarrage de Linux

## noyau de l'OS

- Le noyau Linux (/boot/vmlinuz-???) est auto-extractible et est constitué d'une entête contenant le code nécessaire au décompactage et du noyau compressé.
- Durant la décompression en RAM, l'écran affiche «Uncompressing Linux».
- Le message «OK, booting the kernel» est ensuite affiché et la main est passée au noyau. Une longue liste de messages défile ensuite détaillant l'initialisation du noyau ainsi que la découverte et la configuration des périphériques.
- A ce stade, le noyau libère l'espace occupé par le code destiné à la détection et la configuration de périphérique. Le message «Freeing unused kernel memory : ..k freed» est alors affiché.

# Le démarrage de Linux

## noyau vers init

- Le système est maintenant initialisé, et le noyau lance le programme `/sbin/init`, chargé de terminer la séquence d'amorçage.
- La configuration d'init se trouve dans `/etc/inittab`. Ce fichier varie énormément d'une distribution à l'autre mais globalement, les actions effectuées sont à peu près les mêmes.

# Le démarrage de Linux

## configuration de init

Sur une distribution compatible RedHat, il existe 7 runlevels :

0 - halt

1 - mode mono-utilisateur (dépannage)

2 - mode multi-utilisateurs sans les montages NFS

3 - mode multi-utilisateurs (texte)

4 - non utilisé

5 - mode multi-utilisateurs avec lancement de X11  
(graphique)

6 - reboot





# Le démarrage de Linux

## init vers utilisateurs

- Lorsque toutes les initialisations sont faites, il faut permettre à l'utilisateur de se connecter. Ceci est configuré dans `/etc/inittab` :  
*1:2345:respawn:/sbin/mingetty tty1*
- Chaque ligne indique les programmes qui doivent être lancés sur chacune des consoles virtuelles. Le rôle du programme `mingetty` est d'afficher une invite (d'après `/etc/issue`) :  
*Debian GNU/Linux 3.1 papagos tty0*  
  
*login :*
- Ensuite, il exécute la commande `/bin/login` chargé d'authentifier l'utilisateur et le connecter (afficher le contenu de `/etc/motd`, vérifie la mailbox, puis lancer un shell).
- Lorsque l'utilisateur termine sa connexion, `init` relancera un nouveau processus `mingetty` sur la console virtuelle correspondante

# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- Gestion des processus
- Configuration des montages
- Configuration du réseau
- Automatisation de tâches
- Sauvegarde

# La gestion des utilisateurs

- Fichier /etc/passwd est la base de données sur les utilisateurs.
- Exemple de ligne :

```
mquinson:DcoJmNQ:1003:1003:Martin Quinson,,,:/home/mquinson:/bin/bash
```

Suite de champs séparés par des ':'

- **login** de l'utilisateur
- **mot de passe** crypté. Il peut être remplacé par x et déporté dans /etc/shadow pour le cacher (seul root peut lire) et améliorer la sécurité.
- **uid** : numéro d'utilisateur utilisé dans les inodes, entre autres. l'uid du super-utilisateur est 0
- **gid** : numéro de groupe principal.
- **commentaire** : en général nom et prénom
- **répertoire personnel**
- **commande de connexion** : shell à lancer lors de la connexion

# La gestion des utilisateurs

- Fichier `/etc/group` est la base de données sur les groupes  
nomme les groupes et défini les autres groupes de l'utilisateur

- Exemple de ligne :

```
audio:DcoJmNQ:29:plateau,mquinson
```

Suite de champs séparés par des ':'

- **Nom du groupe**
- **Mot de passe** crypté : demandé aux utilisateurs n'étant pas dans le groupe voulant s'y connecter avec `newgrp(1)` ou `sg(1)`
- **Liste des utilisateurs** du groupe (connexion sans mot de passe)

# La gestion des utilisateurs

- /etc/passwd et /etc/group sont lisibles par tous les utilisateurs
- ⇒ attaque des mots de passe en force brute sur autre machine
- ⇒ les mots de passe placés dans /etc/shadow (lisible que par root)

```
mquinson:$1$wflguEva$6hednbGEedlNqCuqKVwQN1:12663:0:99999:7:::
```

- login
- mot de passe crypté
- date dernière modification de mot de passe (jours depuis 1/1/70)
- jours à attendre avant de pouvoir changer le mot de passe
- jours avant avertissement de fin de validité du mdp
- jours entre avertissement et désactivation
- date de désactivation
- champs réservés

# La gestion des utilisateurs

## La commande *passwd* :

- Rôle pour l'utilisateur : créer ou modifier son mot de passe
- Rôles pour l'administrateur :

modifier le mot de passe d'un utilisateur	<i>passwd nom_utilisateur</i>
supprimer le mot de passe	<i>passwd -d nom_utilisateur</i>
verrouiller le compte d'un utilisateur	<i>passwd -l nom_utilisateur</i>
déverrouiller le compte d'un utilisateur	<i>passwd -u nom_utilisateur</i>

## Autres commandes de gestion des utilisateurs et groupes :

*useradd, usermod, userdel ; groupadd, groupmod, groupdel ;*

*finger ;*

*passwd ; chfn, chsh ;*

*su, sudo, sg, newgrp ;*

*id, groups*

# Administration système

- Gestion des utilisateurs
- ▶ **Gestion des fichiers**
- Gestion des processus
- Configuration des montages
- Configuration du réseau
- Automatisation de tâches
- Sauvegarde

# La gestion des fichiers

- Changer les droits d'un fichier  
`chmod [-R] mode_octal fichier`  
`chmod [-R] [ugoa][+ -=][rwxXstugo] fichier`
- Changer le propriétaire d'un fichier  
`chown [-R] propriétaire fichier`
- Changer le groupe d'un fichier  
`chgrp [-R] groupe fichier`
- Changer les attributs d'un fichier (ext2/ext3)  
`chattr [-R] [+ -attributs] fichier`
- Visualiser les attributs  
`lsattr fichier`



# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- ▶ **Gestion des processus**
- Configuration des montages
- Configuration du réseau
- Automatisation de tâches
- Sauvegarde



# La gestion des processus

**ps(1)** : affiche des informations sur les processus en cours.

Principales options :

- u** Affiche le nom du propriétaire
- a** Affiche les processus des autres utilisateurs
- x** Affiche les processus sans terminaux
- e** Affiche les informations sur tous les processus
- f** Affiche des informations complémentaires
- u utilisateur** Affiche les processus de cet utilisateur

**top(1)** : affichage interactif d'informations similaires

**lsuf(8)** : liste les fichiers ouverts pour chaque processus

# La gestion des processus

**kill(1)** : envoie un signal à un ou plusieurs processus

`kill -signal PID`

Les signaux les plus courants sont :

<b>HUP (1)</b>	demande au démon de relire ses fichiers de configuration
<b>KILL (9)</b>	tue un processus
<b>TERM (15)</b>	demande à un démon de se terminer

`kill -KILL -1` : façon rapide et dangereuse de se délogger

**killall(1)** : envoie un signal à tous les processus de ce nom

`killall netscape`

# La gestion des processus

Les programmes sont exécutés par le processeur à tour de rôle. Le partage n'est pas équitable, mais tient compte des priorités de chacun.

Valeur numérique de chaque processus : son nice (sa « gentillesse »).

0 : normal ; 20 : priorité minimale ; -20 : priorité maximale.

Processus critiques à -10 (X, noyau). Les autres doivent être au dessus.

**nice(1)** : lance un programme à une autre priorité

*nice [-n valeur] [commande [arguments ...]]*

Valeur = incrément du nice du père. Seul root peut utiliser valeur négative.

**renice(1)** : modifie la valeur après le lancement

*renice priorité [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]*

# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- Gestion des processus
- ▶ **Configuration des montages**
- Configuration du réseau
- Automatisation de tâches
- Sauvegarde

# Configuration des montages

Fichier `/etc/fstab` : liste des systèmes de fichier à monter automatiquement.

Chaque ligne de ce fichier correspond à un montage.

Les différents champs d'une ligne sont :

**Périphérique** **Point de montage** **Type** **Options** **Dump** **fsck**

```
/dev/hda5 /          ext3  defaults,errors=remount-ro          0  1
/dev/fd0  /floppy  vfat  defaults,user,noauto,showexec,umask=022  0  0
/dev/sda  /mnt/usb auto   noauto,users,sync,codepage=850      0  0
sysfs     /sys     sysfs defaults                              0  0
/dev/cdrom /mnt/cdrom iso9660 defaults,ro,user,noexec,noauto      0  0
nfs-id.imag.fr:/e3/mquinson /mnt/nfs/mquinson nfs
noauto,user,nosuid,rw,soft,nolock,nosuid,noroot 0 0
```

# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- Gestion des processus
- Configuration des montages
- ▶ **Configuration du réseau**
- Automatisation de tâches
- Sauvegarde



# La configuration du réseau

## chargement des pilotes

- Nommage des interfaces :  
*eth0* : première carte ethernet, *eth1* : deuxième  
*lo* : boucle locale; *ppp0* : modem ; *irlan0* : infrarouge
- Pour savoir si les pilotes nécessaires sont chargés :  
`ifconfig eth0`  
Si message « Périphérique non trouvé », il faut alors charger le pilote manuellement.  
Obtenir la liste : `ls /lib/modules/$(uname -r)/net`  
Charger le module : `modprobe pilote`

# La configuration du réseau manuelle

**ifconfig(8)** : affiche (et modifie) la configuration réseau actuelle

```
/sbin/ifconfig
```

```
/sbin/ifconfig eth0 192.168.0.3 netmask 255.255.255.0
```

**route(8)** : affiche (et modifie) la table de routage

```
/sbin/route -n
```

Table de routage IP du noyau

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	eth1

```
/sbin/route add default gw 192.168.0.1
```

# La configuration du réseau

## à la redhat/mdk

Le fichier `/etc/sysconfig/network` contient les variables :

**NETWORKING** : initialisée à "yes" pour valider l'utilisation du réseau

**FORWARD\_IPV4** : initialisée à "no" pour empêcher le transfert automatique des paquets

**HOSTNAME** : contient le nom pleinement qualifié (avec le domaine) de la machine

**GATEWAY** : adresse IP de la passerelle

**DOMAINNAME** : le domaine de la machine

Ce fichier est utilisé dans les scripts d'initialisation pour positionner les variables d'environnement.



# La configuration du réseau à la debian

Pour chaque interface, un morceau de /etc/network/interfaces

```
auto lo
```

```
iface lo inet loopback
```

```
iface eth0 inet dhcp
```

```
iface eth1 inet dhcp
```

```
wireless_essid IMAG-visiteurs
```

```
wireless_ssid IMAG-visiteurs
```

```
wireless_key s:devinez
```

```
iface eth2 inet static
```

```
address 129.88.103.44
```

```
netmask 255.255.0.0
```

```
network 129.88.0.0
```

```
broadcast 129.88.103.255
```

```
gateway 129.88.103.1
```

Initialisation :

```
/etc/init.d/network restart
```

Prise en compte des changements :

```
ifdown interface ; ifup interface
```

# La configuration du réseau à la slackware (et BSD)

Appels d'**ifconfig** dans un script de démarrage

`/etc/rc.d/rc.inet1` qui utilise `/etc/rc.d/rc.inet1.conf`

# La configuration du réseau

## résolution de noms

Contenu de /etc/resolv.conf :

```
search mondomaine.com  
nameserver 192.168.0.1
```

Ce contenu peut soit être indiqué manuellement soit fixé automatiquement par dhcp

# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- Gestion des processus
- Configuration des montages
- Configuration du réseau
- ▶ **Automatisation de tâches**
- Sauvegarde



# Automatisation de tâches

Deux types de tâches automatiques :

- retardées (*delayed*)
- répétitives

**at(1)** : programme une exécution retardée

at 10am Jul 31 ; at 1am tomorrow ; at now ; at 4pm

(ouvre un éditeur interactif)

**atq, atrm** : liste les exécutions prévues, en annule une

# Automatisation de tâches

**crontab(1)** : programme une tâche répétitive

Une table par utilisateur (édition: crontab -e) dans /var/spool/cron

Le démon crond se charge d'appliquer ces configurations.

```
PATH=/usr/bin:/bin:/home/mquinson/SCRIPTS
```

```
# Minutes Heures Jours du mois Mois Jours de la semaine Commande
2 0,4,8,12,16,20 * * * p=`pidof sauvegarde`; [ -n "$p" ] || sauvegarde
49 10 * * 1 linux-counter-machine-update -m
```

**# Le caractère \* permet d'indiquer le plus grand intervalle possible.**

Certaines distributions comportent les répertoires :

**/etc/cron.hourly /etc/cron.daily /etc/cron.monthly**

/etc/crontab est alors configuré pour exécuter les scripts se trouvant dans ces répertoires respectivement toutes les heures, tous les jours et tous les mois.

# Administration système

- Gestion des utilisateurs
- Gestion des fichiers
- Gestion des processus
- Configuration des montages
- Configuration du réseau
- Automatisation de tâches
- ▶ **Sauvegarde**

# La sauvegarde

**cpio(1)** : copie de fichiers

Exemple 1 : sauvegarde sur disquette

```
cpio -o > /dev/fd0
```

```
/etc/passwd
```

```
/etc/group
```

```
Ctrl-D
```

Exemple 2 : sauvegarde sur bande

```
find /home -print | cpio -o > /dev/rft0
```

# La sauvegarde

**tar(1)** : sauvegarde d'arborescence ou de fichiers  
Création et utilisation de fichiers d'archives

Exemples :

```
tar cfz etc.tar.gz /etc
```

```
tar xvj toto.tar.bz2
```

Possibilité d'archives incrémentales



# Résumé et conclusion

# Résumé

- **Notions sur Unix**
  - Noyau et modules
  - Fichiers et systèmes de fichiers
  - Disques (formater, partitionner, créer un FS, monter)
- **Démarrage de Linux**

matériel, bios, chargeur, noyau, init, services et utilisateur
- **Administration système**
  - Gestion des utilisateurs, fichiers et processus
  - Configuration des montages disques et du réseau
  - Automatisation de tâches
  - Sauvegarde



# Conclusion

- Linux est un UNIX complet et moderne  
Solution viable dans un contexte professionnel
- Ce n'est pas la seule solution  
Windows, mais aussi Solaris, AIX, QNX, ...

Enfin, le meilleur système d'exploitation,  
c'est celui que l'on connaît le mieux...

