

# Chapitre 1: Démarage de Linux

## & Gestionnaire d'amorçage

### Démarage de l'ordinateur:

- **démarrage du système**: bios / boot monitor

↳ Basic Input/Output System  
(Prog. servant d'interface entre le système d'exploitation et la carte mère): détermine quels périphériques sont candidats pour démarrer.

- exécution du chargeur de démarrage;

- . TBR: Master Boot Record:

- détermine la partition active et son adresse.

- changement d'une partie du bloc de boot de la partition.

512 octets:

Programme « loader » 446 octets	Table des partitions 64 octets	« magic number » 2 octets
---------------------------------------	-----------------------------------	------------------------------

Lancement du chargeur secondaire

Description des partitions:  
taille, localisation, types et statuts.

⚠ ne peut contenir que 4 partitions dites primaires

Vérification de la signature du secteur.

! La taille de ce TBR rend impossible le stockage d'un chargeur de démarrage complet.

→ le chargeur initial appelle un chargeur de démarrage secondaire.

Rôle du chargeur de démarrage: localiser le noyau du SE, le charger et l'exécuter

Sous Linux, les chargeurs de démarrage sont généralement:

- GRUB: GRand Unified Bootloader.

- LILO: LInux LOader.

Chacun peut être installé :

- soit en tant que chargeur secondaire (si le TBR pointe vers eux).
- ou en tant que chargeur en 2 parties (initial + secondaire)

## le boot loader:

Lorsqu'il est exécuté, un splash-screen est affiché proposant les images d'OS disponibles → sélectionner l'image à partir de laquelle on veut démarrer.

GRUB est le gestionnaire par défaut de nombreuses distributions Linux.

### GRUB

- + possède une invite de commandes
- + plus sécurisé
- + conserve les infos sur le Boot dans le syst. de fichiers → possibilité d'extension
- ne supporte pas le changement depuis des syst. de fichiers spécifiques.

. Fichier de configuration:

/boot/grub/menu.lst ou /boot/grub/grub.conf

. Installer GRUB dans le MBR:

permet d'écrire dans le MBR les éléments contenus dans menu.lst ou grub.conf

. Les noms des périphériques se trouvent entre ()

Ex: (hd0,1):

hd signifie que c'est un disque dur.

0 indique le numéro du disque

1 indique le numéro de la partition.

⚠ Les numéros des partitions sont déterminés à partir de 0

Ex: (hd0,4)

4 indique le numéro du lecteur logique.

⚠ Les numéros des lecteurs logiques sont déterminés à partir de 4.

. Protection de GRUB avec un mdp crypté:

#grub

grub > md5crypt

password : \*to be\*  
Encrypted:

⚠ On doit copier le code crypté et l'insérer dans /etc/grub.conf avant la déf. des images.

⚠ Pour protéger une entrée en particulier,

### LILO

- conserve les infos sur le Boot dans le MBR → impossible d'ajouter de nouvelles fonctionnalités

. Fichier de configuration: /etc/lilo.conf

. Installer LILO dans le MBR : commande lilo

. lilo permet d'écrire dans le MBR les éléments contenus dans le fichier /etc/lilo.conf.

. Ajouter un mot de passe à lilo.conf : password = mot-de-passe

↳ le mdp est écrit en clair et peu être lu par n'importe quel individu qui se connecte au PC.

→ Retirer le droit de lecture à tout autre utilisateur que le root :

# chosen root=root /etc/lilo.conf

# chmod 000 /etc/lilo.conf

## - le processus init :

Lorsque la machine démarre, le noyau du syst. est changé et décompressé en RAM, il s'exécute et s'initialise :

- Réservation de la mémoire.
- Prise en compte de la zone d'échange.
- Détection du matériel.
- Montage du système de fichiers.
- Lancement du 1<sup>er</sup> processus init.

Rôle du processus Init : lancer chacun des processus, démons, sessions de login et de gérer l'arrêt du système.

## Les RunLevel :

- Linux offre plusieurs modes d'exploitation appelés Runlevel.
- A chaque Runlevel correspond à un état dans lequel se trouve le système.
- init 0 : mode d'arrêt du système.
- init 1 : mode simple utilisateur.
- init 2 : mode multi-utilisateurs.
- init 3 : " " " + réseau : services de partage à distance.
- init 4 : Indéterminé
- init 5 : Init 3 + interface graphique
- init 6 : Redémarrer le système.

Fichier de configuration : /etc/inittab

Le niveau d'exécution par défaut y est défini dans la ligne :

Ex: id:5:initdefault:

\* La commande init : changer le niveau d'exécution courant.

runlevel : connaître " " " " .

init q : prendre en compte immédiatement les modifications

. un champs vide est équivalent à tous les niveaux sauf 's' .

sysinit : exécuté une seule fois lors du démarrage à froid.

Le processus 'init' lance en premier lieu le script '/etc/rc.d/rc.sysinit' ensuite le script '/etc/rc.d/rc'

\* La commande : ls -l /etc/rc?.d [0-6]

Ex: S18sound ou K08lisa

- S : indique que le script doit être lancé avec l'opération start.
- K : indique que le script doit être lancé avec l'opération Stop.
- XX : numéro qui détermine l'ordre de lancement ou d'arrêt.

Ex: #chkconfig -list NetworkManager

\* La commande **chkconfig** : affiche l'état d'un service pour chacun des runlevel.

Ex: #chkconfig --level 3 httpd on :

démarrer le service httpd dans le niveau d'exécution 3..

\* La commande : **# /etc/init.d/prog1 start** : démarrer le programme.  
**# /etc/init.d/prog1 stop** : arrêter le programme.  
**# /etc/init.d/prog1 status** : arrêter le programme.

## Chapitre 2 : Gestion des utilisateurs & des groupes

### \* 3 types d'utilisateurs :

- Administrateur : contrôle total.
- Utilisateurs applicatifs : faciliter la gestion des droits d'accès des applications et des démons.
- Simples utilisateurs : se connecter et utiliser les ressources de la machine.

### \* Le login comporte entre 8 et 32 caractères.

- **UID : User ID (<sup>valeur</sup>>0)** : identification de l'utilisateur.
- Les UID < 100 sont associés à des comptes spéciaux.
- UID 0 et GID 0 sont standardisés et identifient le root.
- **GID : User ID** :
  - Chaque utilisateur fait partie d'au moins un groupe.
  - deux types de groupe pour un utilisateur : le gr. primaire et les gr. secondaires.  
Si le user "esprit" a pour groupe primaire users alors les fichiers créés par "esprit" auront comme groupe d'appartenance users.
  - un utilisateur dispose de tous les droits associés à ses groupes secondaires.

### \* La commande id - : connaître les infos essentielles sur un utilisateur.

- GID par défaut pour un user : le groupe "users" avec le GID = 100.

### - Les mots de passe :

- cryptage par md5
- un utilisateur devrait changer régulièrement son mdp.
- Fichier /etc/login.defs : fixer la longueur et la complexité.  
modifier les infos relatives aux utilisateurs.
- Le fichier /etc/passwd : contient la liste des users du syst. local.  
• Lisible par tout le monde.
- Chaque ligne représente un utilisateur et comporte 7 champs séparés par ":".
- Le fichier /etc/group : contient la liste des groupes et pour chacun la liste des users dont il est groupe secondaire. (4 champs)
- \* La commande ~~newgrp~~ : changer de grp principal.
- Le fichier /etc/shadow : contient les mdp cryptés des users.  
chaque ligne est composée de 9 champs séparés par ":".

## - Gestion des utilisateurs :

### \* Ajout d'un utilisateur :

\$ useradd <options> nom-utilisateur

→ Ajout d'un nouveau compte.

Création d'un utilisateur et remplissage des fichiers.

Création d'un groupe privé (du même nom que l'utilisateur).

Création du répertoire personnel, remplissage et modification des droits.

# cat /etc/default/useradd : fichiers contenant les infos.

### \* Modification du mot de passe :

L'utilisateur root a le droit de modifier les mdp de tous les users sans avoir à connaître leurs anciens mdps.

\$ passwd <options> login

→ Gestion des infos de validité.

Modification de tous les champs de /etc/shadow.

\$ chage <options> login

→ Il choisit que passwd.

Accessible que par le root.

### \* Modification d'un utilisateur :

\$ usermod <options> login

→ Il choisit que useradd + syntaxe complémentaire.

### \* Suppression d'un utilisateur :

\$ userdel <x>

⚠ Le répertoire personnel n'est pas supprimé, il faut utiliser userdel.

## - Gestion des groupes :

\$ groupadd <x> : Création

\$ groupadd -g : Préciser un GID

\$ groups nom-user : Lister les groupes d'un user.

### \* Modification d'un groupe :

\$ groupmod <options> nom-groupe

### \* Suppression d'un groupe :

\$ groupdel <x>

⚠ Si c'est le groupe principal d'un utilisateur, il ne peut pas être supprimé.

## \* Contrôle du propriétaire d'un fichier:

- # chfn : indiquer différentes infos sur un user dans le fichier password
- # chown nom-user / nom-fich : changer le propriétaire
- # chgrp nom-group / nom-fich : modifier le groupe

## - Les droits étendus :

S'il on applique ces droits à une commande, elle s'exécutera avec les droits du propriétaire ou du groupe.

Ex: la commande passwd ne peut pas modifier le fichier pour un simple user.

Le droit 's' : droit de root : appelé SUID-bit sur l'utilisateur.      SGID-bit sur le groupe.

# chmod u+s commande

ou chmod 4755 commande

→ pour un utilisateur.

Seul le propriétaire  
ou le root peuvent  
donner ce droit.

# chmod g+s commande

ou chmod 2755 commande

→ pour un groupe.

\* Le sticky bit : il permet d'affecter une protection contre l'effacement du contenu d'un répertoire.

# chmod o+t X

ou chmod 1777 X

Seul le propriétaire  
et le root peuvent  
le supprimer.

# ls -l : pour visualiser les droits.

# Chapitre 3: Installation et gestion des paquetages

## - Archivage et compression:

Archive: ensemble de fichiers qui ont une relation et sont stockés dans un fichier unique.

\* La commande tar : rassembler fichiers et sous répertoires dans une archive: **tar <options> <options> fichiers**

## Compression:

- une fois l'archive créée, elle peut être compressée.
- Algos de compression: Gzip, Gunzip / bzip2, bunzip2.

Efficacité	Bonne	Excellent
Vitesse	Moyenne	Lente

## - Compiler et installer des programmes à partir des sources:

Sous Linux, gcc compile les programmes écrits en C.

g++ " " " " " C++.

Les bibliothèques: fournissent un ensemble de fonctions  
→ Faciliter l'écriture du programme.

- Le programmeur introduit un lien dans le fichier source.
- Le lien peut être statique ou dynamique

lien statique: le programme contiendra les fonctions de la biblio dans ses propres fichiers. (- Augmentation de la taille).

lien dynamique: la biblio sera chargée en RAM quand le prog aura besoin d'exécuter les fonctions.

**/etc/ld.so.conf**: localisation des biblio. → il utilise un cache binaire avec les infos des biblios: cache stocké dans **/etc/ld.so.cache**.  
**/lib**: pour les progs de /bin.  
**/usr/lib**: pour les progs de /usr/bin.

\* La commande **ldd X**: afficher les biblios d'un programme.

Pour que les biblios soient intégrées au cache, il faut utiliser la commande **ldconfig**.

## Changeur de liens:

**ld.so**: cherche la biblio dont un prog a besoin.

## Odre de recherche:

- 1 - LD\_LIBRARY\_PATH
- 2 - /etc/ld.so.cache
- 3 - /lib et /usr/lib.

## Outils GNU:

Installation d'un prog à partir des sources:

```
# ./configure  
# make  
# make install (droits root)
```

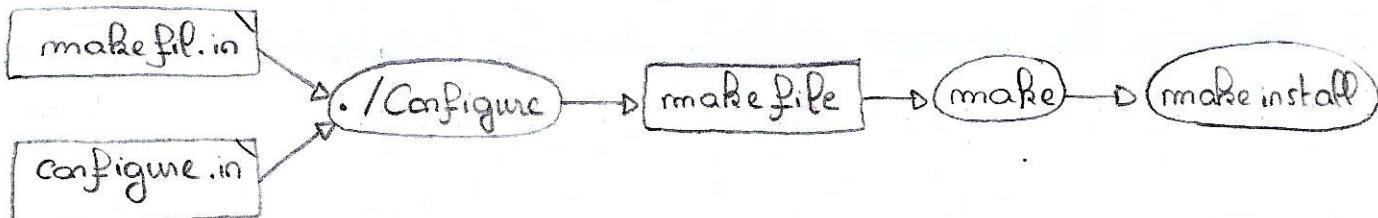
## • ./Configuring:

- Vérifier la présence de toutes les dépendances.
- Ecrire et configurer un fichier `Makefile` (ordres de compilation).
- `./Configure --prefix`: spécifie le répertoire (par défaut `/usr/local`).

## make :

- Effectue la compilation.

`make install`: doit être tapée en tant que root.



## - Gestion de paquetages et installateurs en ligne:

Package: similaire à une archive mais doit permettre de:

- Désinstaller/Installer automatiquement une application.
- Connaitre les packages installés et leurs dépendances.

Familles de paquetage: RPM et DEB,

CentOS, RedHat, Debian, Ubuntu

## • RPM: RedHat Package Manager: Outil de gestion des paquets.

Commande: `RPM <options> <package>`.

Télécharger: `wget <lien du package>`

- La gestion des packages nécessitent des droits de root.

`rpm -ivh nom-paquet.rpm`: installer un nouveau paquetage.

- h: afficher une barre de progression.
- U: mettre à jour.
- v: installation bavarde.
- i: installer.

`rpm -e nom-paquet`: désinstaller.

BD RPM: `/usr/lib/rpm`.

## • Installateurs en ligne RedHat: Yum: gestion des packages RPM en ligne

c'est une sur-couche à RPM gérant les dépendances.

Cherche des paquets sur différents dépôts (sources).

Les dépôts sont placés dans `/etc/yum.conf` ou `/etc/yum.repos.d`

## • Debian Packages:

utilitaire de gestion de paquets sur Debian: `dpkg`

autres: `apt` (surcouche sur dpkg)

similaire à RPM (installer, supprimer, fournir des infos à propos des paquets .deb)

interface graphique `Gdebi`

Autres différences

• A la différence de la commande apt-get, dpkg ne gère pas les dépendances et ne gère pas les conflits.

BD DPKG : var/lib/dpkg

## • Installateur en ligne Debian : Apt

Système avancé et complet de gestion de paquets.

- Prend en charge directement le téléchargement et l'installation des dépendances.
- Interroge les serveurs déclarés dans les fichiers /etc/apt/sources.list

Interfaces graphiques : Aptitude / Synaptic

mode texte plein écran

mode graphique

## • Conversion entre les formats :

Alien : commande de conversion entre différents formats de paquets.  
(entre RPM, Deb et dhag)

⚠ Installez le paquet de Alien s'il n'existe pas et accéder au shell en mode root.

# apt-get install alien

alien --to-rpm < paquet.deb > : .DEB en .RPM

alien --to-deb < paquet.rpm > : .RPM en .DEB

alien --to-tgz < paquet.rpm > : .RPM en .TGZ

## Chap 4: FHS et Gestion de disques

### Plan de partitionnement :

Lors de l'installation, il faut créer 2 partitions :

- / (syst. de fichiers racine) : partition contenant la distribution Linux.
- Espace de Swap : partition permettant au noyau de faire tourner plus de processus que la RAM.
  - ↳ Pas besoin d'un syst. de fichiers.
  - Le noyau y accède en ~~lecture~~ écrit.
  - Eviter les surcoups dus aux appels.

### Comment décider du plan de partitionnement ?

- utiliser le disque le plus rapide.
- /var / log sur le disque le plus rapide.

- Applications qui font tourner le système :
- Serveur de mails
- Serveur web
- App. graphiques

### Pour distribuer les e-mails :

**Sendmail** écrit dans /var/spool/mqueue et /var/spool/mail

### - FHS : Filesystem Hierarchy Standard :

Propose une organisation du système de fichiers.

bin : programmes utilisateurs

boot : fichiers nécessaires au chargement de Linux.

dev : fichiers offrant l'accès aux périphériques.

etc : configuration du système et des services.

home : répertoires principaux des utilisateurs.

lib : bibliothèques.

tmp : fichiers temporaires.

sbin : exécutable système essentiels.

usr : la plupart des fichiers des applications.

var : données vivantes du syst. et des applications.

commande : **fdisk [disque]** création d'une partition.

**partprobe** : informe le noyau des nouvelles modifications sur la table de partitions.

Le fichier **/etc/fstab** contient tous les fichiers système et les infos nécessaires pour monter un lecteur au démarrage du système.

2 catégories indépendantes de fichiers :



données partagées / non partagées :  
entre les hôtes ↔ spécifiques à un hôte.



données variables / non variables :  
(statique) :  
(modifiables ou non)  
sauf via le fichier de configuration

Les inodes : structures de données contenant les informations des fichiers stockés. A chaque fichier correspond un num d'inode (**i-number**) unique au périphérique sur lequel il est situé.

- ils sont créés lors de la création du syst. de fichiers.
- indique le nb max de fichiers que le système peut contenir.

**ls -i tot** : affichage du num d'inode d'un fichier.

Norme POSIX : norme qui impose que les fichiers aient ces attributs :

- Taille du fichier en octets.
- Id du périphérique contenant le fichier.
- Id du propriétaire du fichier.
- Id du groupe.
- Le num d'inode.
- compteur du nb de liens physiques.
- Horodatage.

Les systèmes de fichiers :

Pour stocker, lire et manipuler des données sur un lecteur block.  
il maintient la structure interne des données.

La journalisation :

Partie d'un syst. de fichiers qui trace les opérations d'écriture tant qu'elles ne sont pas terminées

→ garantir l'intégrité des données en cas d'arrêt brutal.

les types :

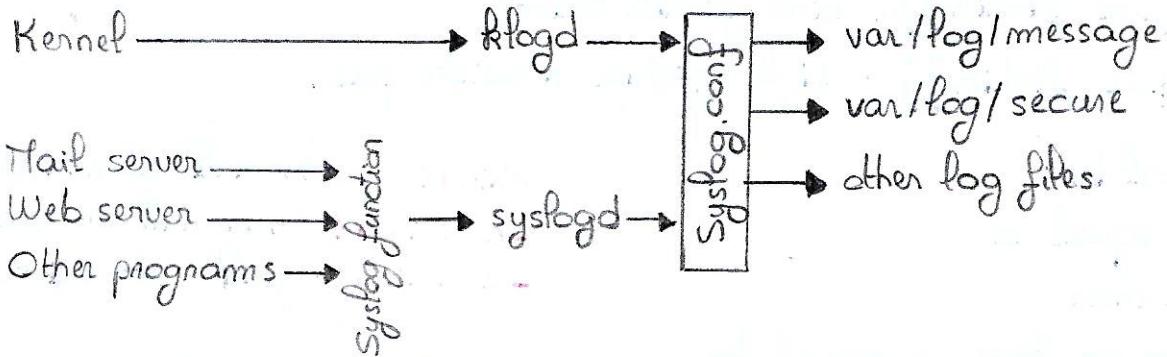
- ext 2 : ancien et très stable.
- ext 3 : extension pour ext2.
- reiserfs : plus performant pour les fichiers de petite taille (internet).
- XFS : fichiers de plus de 9 Exabytes.
- JFS : Crée par IBM.
- FAT32 et NTFS : Syst. de MS-Windows.

## Chapitre 5 : Journalisation avec Syslog

### Syslog : System Logging

Système de journalisation pour transmettre des informations sur des événements survenus au cours du fonctionnement de logiciels.

- Les fichiers syslog se trouvent dans /var/log
- Les fichiers de configuration : /etc/syslog.conf
- Il est préférable de mettre le répertoire /var/log dans une partition.
- /var/log/messages : fichier qui récupère tout.
- /var/log/secure : informations de connexion.
- /var/log/maillog : trafic de courrier entrant et sortant.
- /var/log/spooler : messages d'erreur des daemons uucp et innd.



`rpm -ivh syslogd`: installation.

`# etc/rc.d/init.d/syslog status` : lancement (s'il n'est pas lancé au démarrage)

- Le fichier de configuration est organisé en 2 colonnes :

- Sélecteur : définit l'origine (service + priorité)
- Action : définit la destination (fichier, console, machine ...)

`<service.priorité> </path/fichier>`

Ex: mail.crit /dev/ttys0 : récupérer les logs critiques (crit) des programmes de gestion de mail (mail) et les rediriger dans la console /dev/ttys0

Syslog dispose de :

- 12 services possibles :

KERN msgs du noyau  
USER msgs du prog. en cours d'un user  
MAIL msgs du serveur mail  
DAEMON msgs syst. générés par daemon  
AUTH msg de sécurité et authentification  
SYSLOG msgs de syslog lui même  
LPR msgs du serveur d'impression  
NEWS msgs du serveur des news  
UUCP msgs unix to unix copy protocol  
CRON messages de Cron et At  
AUTHPRIV m chose que auth mais plus privés  
FTP msgs du serveur FTP

- 8 niveaux de priorités :

7 DEBUG	msg de débogage
6 INFO	msg d'info
5 NOTICE	" "
4 WARNING	msg d'avertissement
3 ERR	msg d'erreur
2 CRIT	situation critique
1 ALERT	" " nécessitant intervention immédiate
0 EMERG ou PANIC	système inutilisable.

\*.info; mail.none; authpriv.none /var/log/messages

## Chapitre 6 : Programmation Shell

### - Les tests if :

- if [[ -f fichier]] : si fichier existe et n'est pas un répertoire.
- if [[ -d fichier]] : " " " " et est un répertoire.
- if [[ -r fichier]] : s'il est accessible en lecture.
- if [[ -w fichier]] : " " " " écriture.
- if [[ \$ word == "bonjour" ]]
- if [[ \$ var != "chaîne" ]]
- if [[ \$ var ]] : si la variable var n'est pas la chaîne vide.

for i in \$(seq 1 15) ↳ i allant de 1 à 15

- if [[ -a fichier]] : si le fichier existe.
- if [[ !-a fichier]] : si le fichier n'existe pas.

-eq equal to

string1 non null ou non défini

-ne not equal to

-n string1 non null et existe

-lt less than

-z string1 null et existe

-le less or equal to

-gt greater than

-ge greater than or equal

## Chapitre 7 : Les quotas

Les **Quotas** permettent de poser des limites à l'utilisation du syst. de fichier.

Il existe 2 limites :

- sur le nombre d'inode (donc sur le nb de fichiers créés)
- sur le nombre de blocks (donc sur la taille du disque).

Pour chaque limite, il y a deux types :

- Limite **materielle** (hard) : spécifie la limite sur le disque au delà de laquelle l'utilisateur ne peut pas aller.
- Limite **logicielle** (soft) : l'utilisateur peut dépasser cette limite mais le système affiche warning.

Mise en place des quotas utilisateurs :

`rpm -q acl quota` pour vérifier l'existence.  
puis `grep -i quota /etc/fstab`

Ajouter dans chaque syst. de fichiers le fichier `aquota.user` et/ou `aquota.group`  
il faut installer le paquetage `quota` s'il n'existe pas.

Ajouter `usrquota` et/ou `usrgroup` dans le fichier `/etc/fstab`

`quotacheck -acmugv` : vérifier et maj la configuration.

`quotactl -av` : mettre en place les quotas disque.

`edquota -u user` : configurer les limites.

`edquota -g group`

**quota [options] [user | group]**

- `u` : affiche les quotas utilisateurs.
- `g` : affiche les quotas groupes / du groupe du user courant.

`repquota` : afficher un rapport de quota.

- `a` : rapport sur tous les systèmes de fichiers indiqués dans `/etc/mntab`
- `g` : rapport pour le groupe.

## Chapitre 8: Planification avec Cron

Cron est un daemon utilisé pour programmer des tâches.

- chaque utilisateur a un fichier crontab contenant les actions et à quelles périodes elles vont être exécutées.
- Le fichier existe également pour le système.

Répartis en fichiers:

/etc/cron.hourly : les scripts se lancent chaque heure.

/etc/cron.daily : " " " " " jour.

/etc/cron.weekly : " " " " " semaine.

/etc/cron.monthly : " " " " " mois.

Ajouter des entrées au fichier /etc/crontab

mm hh jj MM JJ user commande > Dog  
0-59 00-23 1-31 1-12 0-7

Formalisme :

Ex: la valeur 15 : exécution à la 15<sup>e</sup> minute.

Ex: les valeurs 1,4,7,10 dans le champs mois : janvier, avril, juillet, octobre

Ex: les valeurs 1-5 dans le champs JJ : du lundi au vendredi

Ex: \* dans le champs jj : tous les jours du mois.

1 2 3 4 \* root /usr/bin/apt-get update  
or \* 2h 3 Avril.

\* 2 \* 4 5 /usr/root usr/bin/apt-get update

→ Tous les vendredis d'Avril à 2h toutes les minutes.

→ Si \*/10 → toutes les 10 minutes.

option: run-parts lancer un ensemble de commandes dans un répertoire.

logrotate : effectuer des sauvegardes et renommer des fichiers logs et des journaux

- La sécurité dans CRON :

- . Si le fichier /etc/cron.allow existe → vous devez être présents dedans pour être autorisé à utiliser cron.
- . Si le fichier /etc/cron.deny existe et /etc/cron.allow n'existe : → vous ne devez pas être mentionnés dedans.
- . Si les deux fichiers sont absents, seul le root peut utiliser cron.

Outil KCRON : éditer une crontab de manière visuelle.

- ANACRON jobs: anachronistic command scheduler

⇒ CRON: si le syst. est stoppé au milieu de la tâche, celle-ci est arrêtée.  
Anacron est né pour permettre l'exécution des tâches sans laisser la machine allumée.

- il utilise des indications de temps relatives (une fois par jour / mois ... ) au lieu de références temporales.
- il est lancé :
  - \* au démarrage via un script d'init (/etc/init.d/anacron)
  - \* via le crontab système

fichier de configuration: /etc/anacrontab

- Si la machine reboot plusieurs fois dans la journée, anacron est exécuté plusieurs fois.

Pour cela, il stocke la date de sa dernière exécution dans un fichier log /var/spool/anacron

- At jobs :

Atd est un daemon qui s'occupe des commandes à exécuter une seule fois à un instant précis.

Exemples:

dans 2 minutes:

```
# at now + 2 minutes
at > ls /bin > fich.log
Ctrl + D
```

à 13h37:

```
# at 13:37
at > ls/bin > /fich.log
Ctrl + D
```

atq : affiche la liste de tâches.

atrm num-tâche : supprimer une tâche.

# Liste des commandes

## Le fichier /etc/passwd :

# cat /etc/passwd

esprit : x : 500 : 100 : commentaire : /home/esprit: bin/bash

login	{ mdp }	UID	{ GID }	commentaire	{ rep. personnel }	{ shell (commande exécutée au lancement) }
-------	---------	-----	---------	-------------	--------------------	--

## Le fichier /etc/group :

# cat /etc/group

bin : x : 1 : root, bin, daemon

nom	{ mdp }	GID	{ liste des users dont il est secondaire }
-----	---------	-----	--

## Le fichier /etc/shadow :

# cat /etc/shadow

root : \$1\$3\$53eGV4e:15741:0:99999:7:::

login	{ mdp crypté }	{ nb de jours depuis le dernier changement de mdp. }	{ nb de jours avant lesquels le mdp peut être changé }	{ nb de jours après lesquels il doit être changé }	{ " " " avant l'expiration }	{ Réserve }
-------	----------------	--	--	--	------------------------------	-------------

## Ajout d'un utilisateur: useradd :

-u : Afficher l'UID.

-g : Afficher le GID.

-G : " les groupes secondaires.

-d : chemin du rep. personnel.

-c : commentaire.

-k : chemin du rep contenant le squelette.

-s : shell par défaut.

-p : mdp.

## Modification du mdp: passwd :

-l : verrouille le compte.

-u : déverrouille le compte.

-d : supprime le mdp.

-n <j> : durée de vie min en jours du mdp.

-w <j> : " " " max " " "

-i <j> : " " " jours avant avertissement.

-S : statut du compte.

## charge:

- E: fixer une date d'expiration sous la forme YYYY/MM/DD.
- H: changer la valeur du nb max avant le changement du mdp.
- m: " " " " " min " " " " " "
- W: changer le nb de jours de jours d'alerte avant un changement obligatoire.
- d: changer la date du dernier changement.

## Modification d'un utilisateur: usermod :

- L: vérification du compte.
- U: dévérification.
- e <n>: le mdp expire n jours après le 1/1/1970.
- u <UID>: modifie UID.
- p <login>: modifie le nom.
- m: nouveau répertoire personnel.

## Ajout d'un groupe : groupadd:

- n <nom>: renommer
- g <GID>: modifie le GID
- A <user>: ajoute le user spécifié dans le groupe secondaire.
- R <user>: supprime le user du groupe.

## Archivage : tar <fonctions> <options> fichiers

fonctions: -c : créer une archive

- t : liste les fichiers rassemblés dans un fichier tar.
- x: extrait le contenu d'un fichier tar.

options: -f : spécifie le nom d'un fichier tar.

tar -cvzf backup.tgz : Backup du répertoire utilisateur avec gzip

tar -cvjf backup.tbz : " " " " " bzip2

tar -xvf backup.tgz : Décompression et extraction de backup.tgz avec

tar -xvjf backup.tbz : " " " " avec bunzip2.

## RPM:

- rpm -qa : lister tous les paquets installés.
- qi <fichier.rpm> : afficher une page d'info.
- ql <paquet> : lister les fichiers d'un paquet.
- q --requires <paquet> : afficher les dépendances.

## Yum: yum

# yum list: liste tous les paquets.

yum list <paquet>: recherche un paquet.

yum install <paquets>: installe un paquet.  
update  
remove

search <paquets>: trouve le paquet sur le dépôt rpm.

## Debian Package: dpkg

dpkg -i nomfich.deb : installation

dpkg -r nomfich.deb : désinstallation

dpkg -L <paquet> : composants du paquet

dpkg -l: paquets installés

dpkg -P <paquet> : désinstalle un paquet et supprime ses fichiers de configuration.

dpkg -p <paquet> : affiche des infos sur un paquet installé.

dpkg -S <fichier> : donne le paquet auquel appartient le fichier.

## Installateur en ligne Debian: Apt

apt-get update: construit la base / met à jour la liste des paquets.

apt-get upgrade: met à jour tous les paquets installés.

apt-get autoclean: supprime les infos uniquement sur les paquets qui ne peuvent plus être téléchargés.

apt-cache: obtenir des infos au sujet d'un paquet.

apt-cache search <noms>: liste les paquets contenant le <noms>

apt-cache show <paquet>: fournit une description rapide

apt-cache depends <paquet>: affiche les dépendances.

## Les partitions: fdisk

-l: afficher la table des partitions.

-n: ajouter une partition.

-m: afficher le help.

-w: enregistrer les modifications.

## Système de fichiers: mkfs

-t: type du syst. de fichiers: formatage.

-c: contrôle des ~~BB~~ badblocks du lecteur avant la construction d'un syst. de fichiers.

## Système de fichiers étendu : `mke2fs`

- b : type du syst. de fichiers.
- c : contrôle.
- j : crée le syst. avec un journal ext3.
- L : donne un label au syst.

`dumpfs -b [nom]` : affiche la liste des blocs défectueux.

## Modification d'un système de fichiers : `tunefs`

- l : liste le contenu du superblock.
- L : donne un label au volume d'un syst. de fichiers.
- j : passer de ext2 à ext3.

`fsck -i` : corriger les fichiers corrompus.

## Monter et démonter un syst. de fichiers : `mount`

-a : monter tous les syst. de fichiers déclarés dans /etc/fstab.

`mount -t [syst.de fichiers] -o option [point de montage]`

Ex: `mount -t vfat /dev/hda1 -o ro /DOS`

ro : pour lecture seule.

rw : pour lecture-écriture.

noauto : pour que le syst. ne soit pas monté au démarrage (+ auto)

user : pour qu'un simple user puisse monter et démonter un syst. de f.

uid, gid, umask : définir les permissions.

defaults : options par défaut (rw, auto, nouser).

sw : systèmes de swap.

## User CRON Jobs : `crontab`

-e : éditer les commandes à planifier.

-l : affiche la liste des tâches programmées.

-r : supprimer la table cron. Ex: `crontab -r salah -r`