# Synthèse
# Système d'exploitation
# Open Source Th.

Deuxième Bloc
Sécurité des systèmes
Année académique 2020-2021

*Rédigé par Sénéchal Julien*

Fin de rédaction le 18 Décembre 2020

**A savoir avant de commencer à lire la synthèse :**
Vous trouverez beaucoup d'infos que vous jugerez inutiles étant donné que si vous avez suivi les 3 premiers quadrimestres, une bonne partie de la matière est assimilée.
Néanmoins, j'ai préféré reprendre toutes les informations les plus pertinentes et les résumer. (Tables des matières en fin)
A vous de supprimer ce qui ne vous paraît pas utile (genre à l'ancienne vous prenez un fluo quoi xD). Alors c'est vrai que la synthèse ressemble plus a un gros résumé qu'à une synthèse mais si ça ne vous plaît pas, vous pouvez toujours la faire vous même xD.
Bonne lecture et bon courage

# 1 A savoir sur Linux

## 1.1 Qu'est-ce que c'est ?

- Officiellement le nom de cet OS est *GNU-Linux*, le nom du kernel est *Linux*.
- C'est un système multi-tâches & multi-utilisateur préemptif.
- Le kernel *Linux* fut créé sur un noyau de type *UNIX* et basé sur Minix.

## 1.2 Les distributions

### 1.2.1 Qu'est-ce que c'est ?

Une distribution Linux n'est pas une version différente du système Linux, il s'agit juste d'un regroupement d'applications et une présentation différente par rapport aux autres distributions (Ex : Kali Linux, Ubuntu, Debian).

Difference entre les distributions :

- Liste d'applications pré-installées différente
- Méthode de distribution (gratuit, payant, CD-ROM, DVD-ROM, etc...)
- Chaque distributions fourni des outils pour son intallation
- Ajout de ses propres utilitaires pour faciliter la vie de l'utilisateur

Une applications qui peut s'éxécuter sur une distribution peut donc s'éxécuter sur chaque distribution Linux. Si elle ne s'éxécute pas, c'est simplement qu'il manque une libraire.

### 1.2.2 Quelques exemples

**Red-Hat**

- La plus connue et une des plus distribuées
- A l'origine du système de distribution de packages *RMP*
- Evolue vite
- Mise à jour et maintenance difficiles (En cause : RPM)

**Mandrake**

- Basée sur Red-Hat
- Orientée utilisateur graphique (optimisée pentium)
- Origine française
- Facile à installer et "user-friendly"

**SuSE**

- Utilise aussi RPM
- Connue pour son utilitaire de configuration et d'administration du système *Yast* (système propriétaire)
  - ➤ Permet de faire de nombreuses actions qui ne sont pas possible autrement qu'en ligne de commande

- Distribuable qu'en CD.

**Debian**

- Maintenue par des bénévoles
- Une des distribution les plus stable et facile à maintenir (Ouais je sais, j'arrive pas à y croire moi même)
  - ➤ Grâce à son système de packages *APT*
- Plus complexe et destiné aux utilisateurs plus avancés

# 2 Principes de base

## 2.1 Comptes utilisateurs

- Système multi-utilisateurs → Plusieurs users peuvent travailler simultanément sur la même machine (grâce à une connexion à distance)
- Un compte utilisateur est un ensemble de droits de l'utilisateur et il possède en général son répertoire personnel
- Un utilisateur peut faire partie de plusieurs groupes
  - ➤ Un groupe permet de définir des droits et des profils pour un ensemble d'utilisateurs
  - ➤ Par défaut, chaque utilisateur appartient à un groupe à son propre nom
- Chaque fichier appartient à un utilisateur et un groupe.
- 3 niveaux d'accès pour chaque fichier attribuable au propriétaire, au groupe du propriétaire, et aux autres (groupes/users)
  - ➤ lecture
  - ➤ écriture
  - ➤ exécution
- root → super-utilisateur
  - ➤ Seul a pouvoir administrer
  - ➤ Accès illimité aux fichiers, ressources, périphériques

## 2.2 Login

Une fois l'utilisateur authentifié, il arrive sur le shell qui exécute le fichier de configuration personnel (par exemple : *.bash_profile* & *.bashrc*)

## 2.3 Arrêt/Démarrage

En général, seul *root* peut arrêter le système grâce à

- shutdown <options> [<minutes>|now]
- halt
  - ➤ Correspond à passer l'option -h à shutdown et donc arrêter le système
- poweroff
  - ➤ Correspond à passer l'option -h à shutdown et donc arrêter le système
- reboot
  - ➤ Correspond à passer l'option -r à shutdown et donc redémarrer le système

## 2.4 Shell

C'est la "coquille" qui entoure l'OS et lui permet de communiquer avec l'utilisateur.

### 2.5 Commandes

- Emploi de commandes très répandu sur un système de type UNIX.
- Plus rapide que de naviguer à travers des fenêtres, onglets, etc...
- Permet une automatisation avancée (scripts)

### 2.6 All -> Fichier

**TOUT** est considéré comme étant un fichier sous Unix.
Reste la façon de l'utiliser.

## 3 Interface graphique

### 3.1 Introduction à X

- La couche graphique n'est qu'une suite d'applications sous Linux
- Sous UNIX, l'interface graphique est appelée *X Window*.
  - ➤ Fonctions de bases comme le déplacement de la souris, réaction clic souris, affichage, etc...
  - ➤ Ne gère pas les boutons, menus, etc.. qui sont délégués au *Window Manager* (autre application)

### 3.2 Serveur X

- Fonctionne selon une architecture client-serveur
  - ➤ Le client X est l'application qui va demander d'effectuer diverses opération d'affichage
  - ➤ Le serveur X renvoie les informations clavier et souris de l'utilisateur et répond a la demande des clients.

### 3.3 Les gestionnaires de fenêtres

- Gère les fenêtres affichées à l'écran par le serveur X
  - ➤ Placement
  - ➤ Redimensionnement
  - ➤ Fermeture
  - ➤ Décorations des fenêtres
  - ➤ etc...
- C'est un client X qui joue le rôle d'intermédiaire entre d'autres client et le serveur X
- Exemples : AfterStep, Beryl Fusion, FluxBox, WindowMaker, etc...

### 3.4 Gestionnaires de bureau

- Les plus célèbres :
  - ➤ GNOME
  - ➤ KDE
- Couche logicielle permettant d'avoir un bureau (menus avec la liste des programmes, etc...)

## 4 Installation

**Il est conseillé de prendre notes de chaque étapes lors de sa réalisation pour se simplifier la vie en cas de crash ou autre problème dans l'avenir.**

### 4.1 Mise en place du/des FS

Il faut au minimum une partition sur le disque pour installer Linux mais il est conseillé d'en avoir au moins une $2^{eme}$ pour le *swap. fdisk, cfdisk, Diskdruid, etc...* sont des utilitaires permettant de gérer les partitions.

## 4.2 Partitions

La définition des partitions se trouve au niveau du secteur maître d'amorçage (Master Boot Record = MBR).

- Celui-ci est utilisé pour démarrer le système (contient une partie du code exécuté pour démarrer la machine)
- On y trouve aussi *la table des partitions* qui contient toutes les informations de celles-ci (localisatio, taille, etc...)
- Le *MBR* occupe les 512 premiers *Bytes* du disque
- L'autre partie du MBR est occupée éventuellement par un boot manager (grub, lilo,...) qui gère le démarrage système
- Si une partition est marquée comme *active*, le système demarre directement sur celle-ci qui doit alors contenir son propre gestionnaire de démarrage.

Il existe 3 types de partitions :

- primaire
- étendue
  - ➤ "Conteneur" a partition logique
- logique

En général on utilise un ou plusieurs partitions primaire. Mais si on désire avoir plus de 4 partitions, on peut définir la dernière partition comme étant une partition étendue afin de pouvoir contenir plusieurs partitions logique.
Les partitions primaires sont définies par le *MBR* et les partitions logiques sont chaînées (par manque de place).

## 4.3 Disques & partitions sous Linux

- Noms des disques et partitions sous forme de caractères alphanumérique (contrairement a Windows)
  - ➤ Disques IDE commencent par "hd"
    1. hda
    2. hdb
    3. ...
  - ➤ Disques SCSI & Data commencent par "sd"
    1. sda
    2. sdb
    3. ...
  - ➤ Les disques ou lecteurs branchés sur un port parrallèle (ZIP) ou sur un port USB ou FireWire apparaissent généralement comme des disque SCSI.
- Le nom des partitions se composent du noms du disque suivi du numéro de la partition.
  - ➤ De 1 à 4 pour les partitions primaires
  - ➤ A partir de 5 pour les partitions logiques
  - ➤ Exemple avec le premier disque SCSI contenant une parition étendue :
    1. sda1 - Primaire
    2. sda2 - Primaire
    3. sda3 - Etendue
    4. sda5 - Logique
    5. ...

# 5 Arborescence & File System

## 5.1 Introduction

- Linux ne distingue pas les disques les uns des autres.
  - ➤ Tout le FS possède une racine unique "/", même si il est réparti sur plusieurs disques
  - ➤ Pas de "C :", "D :", "E :" comme sur Windows
- Le FS par défaut de Linux est le *ext3* mais de plus en plus le *ext4*

## 5.2 Règles des FS UNIX

- Chemin d'accès séparés par des "/"
  - ➤ Contrairement aux "\" sur Windows
- Nom de fichier et de repertoire sensible a la casse
- Le seul caractère interdit sous Linux est le "/"
- Tous les fichiers commençant par un "." sont cachés par défaut (Ex : .bashrc)
- Les extensions (.exe, .txt, etc...) jouent un rôle moindre que sur Windows
- Les fichiers executable ne sont pas reconnu par leur noms
  - ➤ L'execution est un droit du fichier

## 5.3 Arborescence

- Seul root peut modifier l'Arborescence
  - ➤ Sauf les utilisateurs dans leur répertoire personnel
- */bin*
  - ➤ Contient les executables binaires servant à l'initialisation du système ainsi que de quelques commandes essentielles.
- */boot*
  - ➤ Noyau du système (vmlinuz), les fichiers nécessaires à ce noyau (Ramdisk initrd, etc), et une partie du Boot Loader.
  - ➤ Conseillé de placer ce dossier sur une partition dédié en début de disque
- */dev*
  - ➤ Répertoire de fichier spéciaux qui servent de canal de communication avec les périphériques
- */etc*
  - ➤ Contient les fichiers de configuration du système et les principaux scripts de démarrage
- */home*
  - ➤ Contient tous les répertoires personnels des utilisateurs
- */lib*
  - ➤ Contient les bibliothèques et les modules du noyau Linux
- */mnt*
  - ➤ racine des points de montage des fichiers périphériques et extérieurs
- */opt*
  - ➤ Répertoire d'installation d'application supplémentaire (StarOffice, Java, etc.)
- */proc*
  - ➤ pseudo-répertoire créé au boot de la machine qui contient une image du système et permet de dialoguer avec le noyau
- */root*
  - ➤ Repertoire personnel du root
- */sbin*
  - ➤ Contient les fichiers executables utiles pour l'administration du système
- */tmp*
  - ➤ lieu de stockage des fichiers temporaires
- */usr*
  - ➤ Programmes accessibles à tous les utilisateurs
  - ➤ Structure reproduisant celle de la racine
- */var*
  - ➤ données **var**iable liées à la machine (logs, spool, etc.)

## 5.4 Référence relative & Absolue

- Référence relative
  - ➤ Donne le chemin à partir du répertoire de travail courant
  - ➤ Exemple : usr@debian:/home$ cd user → usr@debian:/home/user$
- Référence Absolue
  - ➤ Commence par un "/"
  - ➤ Donne le chemin d'accès complet
  - ➤ Exemple : usr@debian:/home$ cd /etc/network → usr@debian:/etc/network$

## 5.5 Manipuler des fichiers

- touch
  - ➤ Modifier l'heure de dernier accès et dernière modification (si le fichier n'existe pas, il le crée)
- cp
  - ➤ utilisé pour copier un ou plusieurs fichiers ou répertoires
  - ➤ -p permet de copier les droits d'accès ainsi que le user et groupe propriétaire
  - ➤ -r permet d'appliquer a tout le répertoire et ses sous-répertoire
  - ➤ -i permet d'éviter d'écraser un fichier ayant le même nom que la destination
  - ➤ -u permet d'écraser que si le fichier écrasé est plus ancien
- mv
  - ➤ permet de déplacer un/des fichier.s/répertoire.s
  - ➤ -u permet de déplacer que les fichiers les plus récents que les fichiers existants à la destination
  - ➤ -i demande confirmation avant d'écraser
- rm
  - ➤ permet d'effacer des fichiers
  - ➤ -r va permettre d'effacer récursivement un répertoire (tout supprimer dans un dossier ou fichier)
  - ➤ -f va supprimer sans confirmer
  - ➤ -rf efface le contenu d'un répertoire sans confirmation
  - ➤ -i demande confirmation

## 5.6 Afficher le contenu d'un fichier

- cat
  - ➤ affiche le dossier sur la sortie standart
  - ➤ possibilité d'afficher plusieurs fichiers
- head
  - ➤ affiche les première lignes d'un fichier
  - ➤ -xx pour afficher les xx premières lignes
  - ➤ -v pour imprimer que l'en-tête
  - ➤ possibilité d'afficher les premières lignes de plusieurs fichiers
- tail
  - ➤ affiche par défaut les 10 dernières lignes d'un fichier
  - ➤ même option que pour head
- more
  - ➤ affiche un fichier long écran par écran
  - ➤ *enter* permet d'avancer ligne par ligne

    - *space* permet d'avancer page par page
- LESS
    - version GNU de MORE
    - beaucoup de fonctionnalités
    - permet de voyager dans le texte
    - permet de faire des recherches en tapant le caractère "/[mot a chercher]"
    - q pour quitter
- SORT
    - affiche le contenu par ordre alphabétique
    - -r = reverse
    - -u = n'affiche pas les doublons
    - -b = n'affiche pas les blancs
    - -f = ignorer les majs

## 5.7 Autres commandes utiles

- FIND
    - FIND [CHEMIN] EXPRESSION
    - -d affiche les répertoires et pas les fichiers
    - -f inverse de -d
    - -name <nom> cherche les fichiers qui portent le nom <nom>. Pour ne pas tenir compte des majs, on utilise -iname <nom>
    - -size[+/-]<size>
- LOCATE
    - fait une recherche grâce a une base de données qui est remise à jour toutes les nuit
    - plus rapide que find
- GREP
    - recherche de chaînes de caractères dans des fichiers
    - -n affiche le numéro de la ligne du résultat
    - -v affiche les lignes où la chaînes de caractère n'est pas présente
    - -i ne tient pas compte des maj
    - -l n'affiche pas les lignes mais le nom du fichier du résultat
    - -r fait une recherche récursive sur tout le répertoire et les sous-répertoires
- WHICH
    - pour trouver le fichier executable d'une commande
- DF
    - affiche l'espace libre sur le système
    - -h pour human permet de remplacer la notation en octet par une autre unitée (Go, Mo, etc...)
- FILE
    - affiche le type de données que contient un fichier (image, texte, etc...)
- SPLIT
    - permet de découper un fichier en plusieurs fichiers séparés
    - -b permet de choisir la taille max des fichiers générés
- WC
    - C'est drôle
    - permet de compter le nombre de caractères
    - -c le nombre d'octets
    - -w de mots
    - -l de lignes

# 6 Le Shell : BASH

## 6.1 Shell

### 6.1.1 Introduction au Shell

- Interface entre le noyau et l'utilisateur
- BASH = Bourne Again Shell, c'est le Shell par défaut sur Linux (lancé par la commande /bin/bash)
- Peut être changé selon l'utilisateur
  - ➤ Pour savoir le Shell affecté a un user → /etc/passwd

Ordre d'analyse d'une commande :

1. Lors de l'entrée d'une commande, le shell va vérifier si celle-ci est une commande interne (voir *man bash*)
2. Ensuite il vérifie les alias
   - ➤ Pour définir un *alias* on utilise la commande *alias* (Cette méthode crée un alias temporaire)
3. Enfin, il cherche un programme exécutable qui porte le même nom que la commande sur le FS en se servant des chemin d'accès listés dans **$PATH** (variable d'environnement)
4. Si il trouve, il execute en lui passant les arguments spécifiés sur la ligne de commande
   - ➤ Par sécurité, le répertoire courant est toujours exclu du Path

### 6.1.2 Lancer des commandes dans le Shell

- Les commandes sont la plus part du temps des commandes externes, ce sont donc de simples programmes exécutables
- Pour lancer une commande, on tape simplement son nom
- L'espace entre le nom de la commande et le premier argument est obligatoire
- Différents type d'arguments
  - ➤ Les Options : modifient le comportement d'une commande
    Elle sont généralement non obligatoire, et sous Linux elles sont souvent précédées par un "-"
  - ➤ Les aides : -h ou –help

### 6.1.3 L'environnement des commandes

L'environnement de la ligne de commande est initialisé par une série de script :

- /etc
  - ➤ fichiers dont les noms sont du genre bash, profiles, etc. qui sont des scripts pour tous les utilisateurs
- /home/[répertoire personnel]
- ➤ fichiers dont les noms sont du genre bash, profiles, etc. qui sont des scripts pour un utilisateur en particulier

### 6.1.4 Les caractères spéciaux

Certains caractères sont interprétés de façon particulière par le shell. Tableau 4-1 donne la liste de ces caractères. Si l'on veut représenter ces caractères littéralement, il faut les échapper avec \, "" ou ''.

| Caractère(s) | Signification | Échappement |
|---|---|---|
| *, ?, [] | Ces caractères sont des jokers (*wildcards*). Ils servent à sélectionner plusieurs fichiers ou répertoires. | \, "", '' |
| ~ | La tilde est remplacée par le répertoire personnel de l'utilisateur. | \, "", '' |
| >, >>, <, \| | Ces caractères servent à rediriger la sortie ou l'entrée d'une commande. | \, "", '' |
| $ | Ce caractère sert à substituer la valeur d'une variable d'environnement. | \, '' |

FIGURE 1 – caractères spéciaux

### 6.1.5 Les wildcars ou Jokers

- * → n'importe quelle suite d'un nombre quelconque de caractères
- ? → n'importe quel caractère unique
- ~ → raccourci pour le répertoire personnel
  - ~ JEAN → renvoie vers le répertoire personnel de "jean"
- [] → un caractère parmi ceux étant entre les crochets
  - $[aY]$ → a ou Y
  - $[a - d]$ → un caractère entre a et d (a et d compris)
- \ → permet d'échapper à un caractère spécial
  - Par exemple \ ? permet de prendre le caractère "?" et de le traiter comme un caractère normal
- $ → correspond au début d'une variable d'environnement
  - Propre à l'environnement en cours

Pour plus d'infos sur les jokers : MAN 7 GLOB

### 6.1.6 Variables d'environnement

- Les variables d'environnement peuvent être modifiés et paramétrées par l'utilisateur
- Pour afficher les variables d'environnement, commande : SET
- Pour définir une variable d'environnement : EXPORT GLOBALVAR
- Pour supprimer une variable d'environnement : UNSET GLOBALVAR

## 6.2 Les commandes de BASH

### 6.2.1 Historique

- Maintient d'un Historique
  - ➤ Pour revenir sur une commande précédemment tapée on utilise les flèche directionnelles
- Le point d'exclamation placé en début de commande indique à Bash de compléter la ligne en utilisant la dernière commande qui commence par ce qui à été tapé après le "!"
- La commande HISTORY permet d'affichier l'historique des commandes

### 6.2.2 Complétion des commandes

- Permet de compléter un mot (commande ou nom de fichier) grâce à *TAB*
  - ➤ Sur certaines distributions est fourni un jeux de scripts qui permet une auto complétion avancée. Elle peux par ex. afficher les options d"une commande, ou les fichiers reconnus par cette commande (ex : les fichiers zip avec unzip). C'est le cas p.ex. De la debian. Il s'agit d'un option non activée par défaut, il faut décommenter certaines lignes pour pourvoir l'employer

### 6.2.3 Touches reconnues par le shell

- *ENTER*
- *BACKSPACE*
- *CTRL-U* (efface la ligne)
- *CTRL-C*

## 6.3 Toutes les infos sur Linux

### 6.3.1 L'aide

- –HELP
- -?
- -H

### 6.3.2 man

- MAN [N° SECTION] <COMMANDE> -> affiche le manuel de la commande
- Les man pages sont séparées en sections, numérotées de 1 à 9
  - ➤ Section 1 : Documentation sur les commandes accessibles aux utilisateurs normaux
  - ➤ Section 2 à 7 : Informations utiles aux développeurs
  - ➤ Section 8 : Documentation sur les commandes utiles à l'administration système
  - ➤ D'autres sections existent, mais elles ne contiennent généralement pas d'informations utiles à un utilisateur moyen.
- Stocké dans /USR/SHARE/MAN

### 6.3.3 info

- Autre système d'aide en ligne
- Toutes les commandes n'ont pas de page info
- Utilise un système hypertexte qui facilite la navigation

### 6.3.4 apropos

Cette commande permet de voir toute la documentation se rapportant à un terme. Cela permet de s'y retrouver une commande par exemple

#### 6.3.5 La documentation

Se trouve dans /USR/SHARE/DOC

### 6.4 Entrée & sortie standart

#### 6.4.1 Redirections

- >
  - ➤ redirige la sortie standard vers un fichier (écrase le contenu du fichier)
  - ➤ sortie standard = écran
- »
  - ➤ redirige la sortie en ajoutant les données au fichier (n'écrase donc pas)
- <
  - ➤ redirige l'entrée standard depuis un fichier
  - ➤ rmdir < RépertoireQuiDoitEtreEffacé
- 2> ou 2»
  - ➤ redirige l'erreur standart
  - ➤ erreur standart = moniteur par défaut
- > fichier 2> &1
  - ➤ redirige la sortie standard et l'erreur vers un fichier
- » fichier 2> &1
  - ➤ ajoute la sortie standard et l'erreur a la suite du fichier

#### 6.4.2 Pipes

- |
  - ➤ redirige la sortie d'une commande vers une autre

#### 6.4.3 Tee

— redirige vers la sortie standard et vers tous les fichiers donné en argument

#### 6.4.4 Editeur de texte

*VIM* mais pas plus d'info ici :-)

## 7 Gestion des paquets

- RPM
  - ➤ Système de gestion de paquet pour RedHat
  - ➤ Détecte les dépendances sans les gérer
  - ➤ Installe/Désinstalle/Mets à jour a partir d'un fichier "rpm"
- yum
  - ➤ Surcouche a RPM
  - ➤ Apporte des fonctionnalités similaire à APT
- Dpkg
  - ➤ Equivalent "RPM" pour Debian
  - ➤ Système de gestion des paquets
  - ➤ Détecte les dépendances sans les gérer
  - ➤ Installe/Désinstalle/Mets à jour a partir d'un fichier "deb"

- APT
  - ➤ Gestionnaire de paquets de distribution chez Debian
  - ➤ Gère les dépendances
  - ➤ Possibilité de mélanger les versions de distribution (ex : version unstable chez Samba pour avoir l'accès à certaines fonctionnalités)
  - ➤ Possibilité de créer un fichier "preferences" (/etc/apt/preferences ou créer un fichier dans /etc/apt/preferences.d)
  - ➤ Syntaxe (à connaître) :
    Package : *
    Pin : release a=*[version]*
    Pin-Priority : [Priorité]
    - " * " : Tous les paquets (peut être remplacé par un paquet spécifique)
    - version : stable/stretch/testing
      (stable/lors du passage à la nouvelle version de Debian, la version des paquets de stretch sera préférée à la nouvelle stable/en test)
    - Priorité :

| Priorité | Installé si * |
|---|---|
| 0-100 | aucune version ne l'est déjà |
| 100-500 | pas de version plus récente installé ou disponible |
| 500-990 | pas de version plus récente installée ou disponible dans la version choisie (stable/testing/etc...) |
| 990-1000 | pas de version plus récente installée |
| > 1000 | priorité d'un paquet déjà installé |
| valeurs égale | prend le paquet le plus récent |

Pas d'info sur les paquets binaire et paquets sources

# 8 Users and groups

Plusieurs élément ne seront pas cité dans ce chapitre parce que ces points devraient déjà être connu.

## 8.1 La base

- UID
  - ➤ Identificateur numérique d'un utilisateur (seul manière de reconnaître les utilisateurs pour le kernel)
  - ➤ Possède un nom d'utilisateur (login)
  - ➤ UID 0 = Root
  - ➤ Seul root peut passer outre les droits des fichiers et ce même si les utilisateurs ont des droits étendu
- GID
  - ➤ Identificateur numérique pour les groupes
  - ➤ Même particularité que les UID
  - ➤ Un utilisateur appartient à un groupe primaire et peut appartenir des groupes secondaires
- La commande id permet d'obtenir le UID et les GID de l'utilisateur courant. id [nom d'utilisateur] permet d'avoir les infos de l'utilisateur en question
- Création de nombreux comptes et groupes lors de l'installation
  - ➤ Permet d'attribuer un utilisateurs à un deamon (parfois ce n'est pas le cas et le processus reçoit un propriétaire, le processus est lancé en SUID ou SGID (genre de su))
  - ➤ Permet de limiter les risques de sécurité étant donné que chaque compte est restreint

## 8.2 Fichiers concernés

- /ETC/PASSWD
  - Login
  - MDP
  - UID
  - GID
  - Nom complet + coordonnées
  - Emplacement du "home"
  - Shell utilisé (Optionnel)
- /ETC/SHADOW
  - login
  - Hash du mdp + salt (si = à * ou !, alors le compte est désactivé)
  - n jours depuis le dernier changement de mdp (compté à partir du 1er Janvier 1970 (WTF))
  - n jours à attendre avant de de pouvoir changer de mdp
  - n jours avant la fin de validité du mdp (et pendant lesquels il sera averti)
  - n jours après la fin de validité provoquant la désactivation du compte
  - n jours depuis que le compte est désactivé (compté à partir du 1er Janvier 1970 (Je piges par pourquoi autant d'exagération))
  - Champ Réservé
- /ETC/GROUP
  - Nom du groupe
  - mdp chiffré éventuel
  - GID
  - Liste des logins appartenant au groupe

## 8.3 Changer le mot de passe

- commande PASSWD
- Bonne stratégie de mdp
  - > 10 caractères
  - éviter les mot du dictionnaire (même inversé ou avec certains caractères en maj)
  - éviter les mots de passes évidents
  - Mélanger les différents types de caractères
  - Ne pas réutiliser de mot de passe
  - Le changer régulièrement

## 8.4 Limiter l'usage d'un compte

Respectivement les options -l et -u à PASSWD. Et pour désactiver le shell il faut le modifier par défaut dans le /ETC/PASSWD.

## 8.5 Modifier les propriétés d'un compte

- USERMOD
- Exemple :
  - usermod -G liste, des, groupes, secondaires
  - usermod -g groupe_primaire

### 8.6 Informations sur les utilisateurs

- who & users & w
  - ➤ Commandes permettant d'avoir des infos sur les utilisateurs connectés (w donnant plus d'infos)
- whoami
  - ➤ Commandes permettrant de savoir qui on est
- last [-n x]
  - ➤ Historique de connexions (ou des x dernières connexions)
- lastlog
  - ➤ le nom parle de lui même (sans argument liste tous les users)
- ac
  - ➤ Temps de connexion des utilisateurs
- fail & faillog
  - ➤ Comme pour last et lastlog mais pour les tentatives de connexion échouées

Le système retrouve toutes ces informations grâces aux divers fichiers :

- Pour les connexions réussies
  - ➤ /var/run/utmp
    - — Une entrée est créé lors de chaque connexion et supprimée à la déconnexion
  - ➤ /var/log/wtmp
    - — Contient toutes les connexions et déconnexions
  - ➤ /var/log/lastlog
    - — Date de dernière connexion classé par UID
- Pour les connexions échouées
  - ➤ /var/log/btmp
  - ➤ /var/log/faillog

## 9 Gestion avancées des fichiers

### 9.1 Droits spéciaux

- s
  - ➤ permet d'exécuter un fichier avec les droits du propriétaire (Set-UID ou SUID / Set-GUID ou SGUID)
- t
  - ➤ ne sert que sur les répertoires et donne le droit de supprimer le fichier qu'a son propriétaire (sticky-bit)

Je passe tout ce qui est chmod, chown, et l'umask étant donné que je connais déjà très bien et que flemme.

### 9.2 Inodes

Pour vérifier les inodes : ls -i et df -i

### 9.3 Types de fichiers

- fichier normal
  - ➤ texte, executable, etc..
- fichier répertoire
- fichier lien symbolique
  - ➤ fichier qui contient un pointeur vers un autre fichier (comme un raccourci windows)
- fichier spécial (de périphériques)
  - ➤ fichier qui permet l'accès à un périphériques, une partition (ex : /dev/sdb).
  - ➤ Les fichiers de ce type sont dans /dev

## 9.4 Lien

### 9.4.1 Liens durs

Un lien dur associe un ou plusieurs fichiers à un même espace disque sur un système de fichier. Ils permettent :

- d'avoir un fichier dans un répertoire
- d'avoir plusieurs nom pour un programme et de réagir en fonction de son nom d'appel
- à un inode de se trouver dans plusieurs répertoires différents
- à ne pas avoir un fichier en "double" sur le disque

### 9.4.2 Liens symboliques

- Chemin symbolique pointant vers un autre fichier ou répertoire
- Indépendant du FS
- Agis comme une sorte de "raccourci"

## 9.5 Liste des fichiers périphériques

| Chemin d'accès | Type de périphérique |
|---|---|
| /dev/hd/... | Périphériques IDE selon l'odre des bus |
| /dev/cdrom... | Raccourcis vers le lecteur de CD-ROM |
| /dev/sd... | Périphériques SCSI, USB, SATA |
| /dev/scd… | CD-ROM SCSI |
| /dev/ttyS… | Ports séries |
| /dev/tty… | Consoles virtuelles |
| /dev/psaux | Port PS/2 |
| /dev/mouse | Lien vers la souris |
| /dev/fd… | Lecteurs disquettes |
| /dev/par.. | Ports parallèles |
| /dev/lp… | Imprimantes parallèles |
| /dev/random | Périphérique aléatoire : génére des valeurs pseudo aléatoires |
| /dev/zero | Périphérique qui génére des zéros |
| /dev/null | Périphérique vide. Tout ce qui y rentre disparaît |
| /dev/loop | Périphérique de boucle de montage |

## 9.6 Numéro de périphériques

- C'est UDEVD qui gère les numéros de périphériques.
- C'est par ce numéro que le système reconnaît le type du fichier de périphérique et sait comment y accéder.
- Le numéro est interprété par le "driver" du périphérique

# 10 Gestion des systèmes de fichiers

## 10.1 Outils de partitionnement

- fdisk
  - ➤ commande traditionnelle de partitionnement
  - ➤ -l pour lister les partitions
- cfdisk
  - ➤ plus convivial
- sfdisk
  - ➤ permet de tout faire (d'après le prof c'est le couteau suisse du partitionnement)

Le kernel doit être mis au courant des modifications apportées et cela grâce à un reboot ou via les commandes partprobe ou sfdisk.

## 10.2 Systèmes de fichiers

- mkfs
  - ➤ Outil permettant de créer des systèmes de fichiers
  - ➤ Existe des versions différents pour chaque FS (ex : mk2fs ou mkfs.ext2)
  - ➤ -c pour rechercher les blocs défectueux et les marquer comme non utilisables

## 10.3 Montages

- mount
  - ➤ permet d'accéder a un système de fichier (partition, périphériques, etc...) en l'intégrant dans l'arborescence (ex : /dev/sdb)
- umount
  - ➤ démonte un FS monté
  - ➤ Pour ça il ne doit plus être occupé
  - ➤ La commande "lsof" permet de lister tous les fichiers ouvert
  - ➤ Pour les fichiers ouvets dans le home
    - — lsfo | grep /home
- /etc/fstab
  - ➤ fichier où sont précisé les fichiers à monter au démarrage
- /etc/mtab
  - ➤ liste des systèmes de fichiers actuellement montés

## 10.4 Taille des partitions

- /boot : 100MB
- /var : 2GB (permet de séparer les logs et ainsi assurer la stabilité système)
- /tmp : 500MB (like /var)
- /usr : pas d'info apparremment. Pourquoi faire au fond :/
- /home : 90GB selon le nombre d'utilisateur
- SWAP :
  - ➤ Si - de 2GB de ram : 2 fois la taille de la ram
  - ➤ Entre 2 et 4 GB de ram : 2G en + de la ram
  - ➤ Entre 6 et 8 GB de ram : taille de la ram
  - ➤ Si + de 8 GB de ram : 1/2 de la ram (si vous avez une explication je prends)

## 10.5 Gestion des quotas

### 10.5.1 Différentes limites de quota

- Limite dure par utilisateur (hard limit)
  - ➤ Max qui ne peut jamais être dépassé
- Limite douce par utilisateur (soft limit)
  - ➤ Limite a partir de laquelle un utilisateur pourra continuer a écrire sur le disque et recevra des avertissements
- Limite dure par groupe
  - ➤ Limite max du groupe, et ce même si un utilisateur n'as pas remplit son quota
- Limite douce par groupe
  - ➤ same as soft limit
- Temps de grâce (grace period)
  - ➤ Une fois le temps dépassé dans la limite douce, celle-ci devient une limite dure
  - ➤ De la seconde au mois

### 10.5.2 Mise en place des quotas

1. Activer les quotas sur le FS (via /etc/fstab avec les options de montage de usrquota & grpquota)
2. Remonter le FS pour pouvoir prendre en compte les changements
3. Créer la/les bases de données de gestion des quotas via la commande QUOTACHECK [OPTIONS] FILESYSTEM
   - ➤ -a : véfifie les options des quotas sur tout le FS
   - ➤ -g [group] : uniquement les quotas groupes
   - ➤ -u [user] : uniquement les quotas utilisateurs (par défaut)
   - ➤ -v : mode verbeux
4. Les fichiers aquota.user et aquota.grp seront créé et le FS sera inspecté
5. Activer les quotas via QUOTAON
6. QUATOCHECK doit être exécuté à chaque modification (ou de manière régulière grâce par exemple à "cron")
7. QUOTAOFF permet d'arrêter la gestion des quotas

### 10.5.3 Modification des quotas

- EDQUOTA [-P UTILISATEUR-PROTOTYPE] [OPTIONS] NOMS
  - ➤ -g : modification des quotas des groupes (si combiné avec -u, les noms sont supposé être des noms de groupe)
  - ➤ -u : modification des quotas utilisateurs
  - ➤ -t : modifier les périodes de grâces
  - ➤ -p : dupliquer les quotas à partir d'un utilisateur-prototype
- La commande QUOTA sert a vérifier les quotas
- WARNQUOTA melée a CRON permet d'envoyer automatiquement un courriel local aux users/groupes qui franchiraient les limites. (Utile pour les utilisateur qui utilisent l'interface graphique)

## 10.6 RAID - Redundant Array of Independent Disks

- Mode Linéaire
  - ➤ Pas vraiment un RAID
  - ➤ Il se contente de mettre bout à bout des disques
  - ➤ Mieux vaut utiliser LVM
- Raid 0 / Stripe
  - ➤ Augmente les perfs
  - ➤ Données réparties sur les disques
  - ➤ La perte d'un disque équivaut a tout perdre
- Raid 1 / Mirror
  - ➤ Les données sont écrites en parrallèle sur 2 disques
- Raid 3
  - ➤ C'est un Raid 0 avec un disque de parité en +
  - ➤ En cas de perte d'un disque, les données peuvent être reconstruite grâce au disque de parité
  - ➤ SI disque de parité perdu, pas de chute de performance
- Raid 4
  - ➤ Identique à Raid 3
  - ➤ Travaille par bloc au lieu que par byte
  - ➤ Plus performant que Raid 3
- Raid 5
  - ➤ Plus répandu
  - ➤ Bon compromis coût, fiabilité, performances

    - ➤ Utilise des données de parités réparties sur les différents diques (empêche le problème du goulot d'étranglement présent sur RAID 3 et 4)
    - ➤ Raid plus lent si il n'est pas remis en état
    - ➤ Si + d'un disque est perdu, les données sont perdues
    - ➤ + le nombre de disque est important, - il y a d'espace perdu
- Raid 10
    - ➤ Raid 1 de Raid 0
- Raid 50
    - ➤ Raid 5 de Raid 0
- Raid 0+5 et 0+1
    - ➤ Inverse de raid 50 et 10
- Raid 6
    - ➤ Identique a Raid 5 mais en doublant les infos de parité
    - ➤ Possibilité de perdre 2 disques
    - ➤ Perfs diminué
    - ➤ Avantageux sur de grandes tailles
- Souvent des spare disks sur les RAID, pour qu'en cas de panne, celui-ci prenne la main
- Le raid logiciel ne peux pas faire de hotswap pour remplacer les disques défectueux
- Raid $\neq$ BackUp (tous les 2 sont nécessaires)

## 10.7 LVM

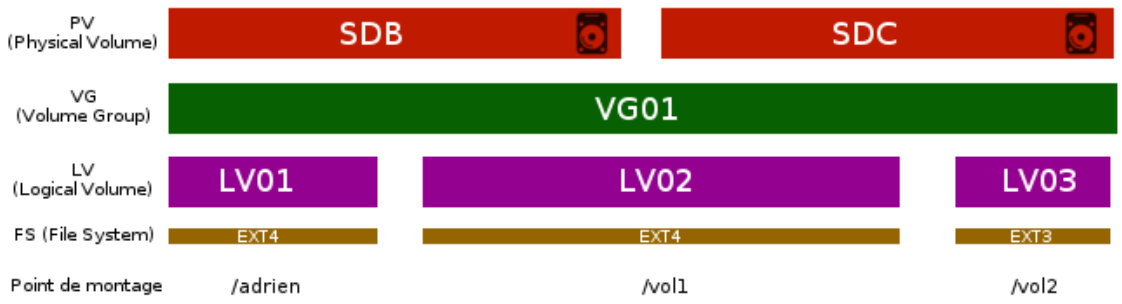


FIGURE 2 – Représentation visuelle du LVM

- PV : Volumes physiques
    - ➤ Les disques durs, partitions de disques durs, volumes RAID ou unités logiques provenant d'un SAN
- VG : Groupes de Volumes
    - ➤ Comme son nom l'indique, on crée un groupe comprenant plusieurs "PV" (Volumes physiques)
- LV : Volumes logiques
    - ➤ C'est comme une partition de VG
    - ➤ On utilise le VG qui regroupe déjà plusieurs PV pour en faire le nombre de "partitions" (LV) que l'on souhaite.
- FS : File System
    - ➤ Bien entendu, chaque LV doit comme pour une partition classique être formatée en un système de fichier pour être monté.
- Le LVM (Logical Volume Manager) permet d'optimiser l'espace de nos différents disques (Voir Figure 2)
- Il apporte le stripping qui améliore les performances
- Les snapshots permettent de faciliter les sauvegardes

# 11 Gestion Processus

Un processus a :

- Un PID (Process ID, numéro du processus)
- Une priorité
- Un user propriétaire
- Un groupe propriétaire

Celui-ci a les même droits que son propriétaire (comme vu précédemment).

## 11.1 Afficher les processus

- PS
  - ➤ Sans option, affiche les processus de l'utilisateur courant
  - ➤ -u [user]
  - ➤ -a all
  - ➤ -x affiche les processus non rattaché à une console
  - ➤ PS AUX affiche donc tous les processus du système
- Pstree
  - ➤ Permet de visualiser la hiérarchie parent/enfant sous forme d'un "arbre" (ps**tree**)
  - ➤ Init est le 1er processus lancé
- Top
  - ➤ Permet de voir quel processus consomme le plus une certaine ressource
- Pgrep
  - ➤ Permet d'obtenir le PID (Process ID) d'un processus en mettant son nom comme argument

## 11.2 Interaction Processus

- Kill
  - ➤ Envoie un signal à un ou plusieurs processus
  - ➤ On peut utiliser le nom du processus ou son PID
  - ➤ Les différents signaux :
    - — sighup (1) : relire les fichier de config
    - — sigint (2) : Interruption (Ctrl+C)
    - — sigquit (3) : Quit (Ctrl+D)
    - — sigkill (9) : Tue le processus immédiatement sans conditions
    - — sigterm (15) : Terminate (Demande de se terminer normalement)
    - — D'autres signaux dans la man page
- Killall
  - ➤ Permet d'envoyer un signal a un processus via son nom
  - ➤ Par défaut, il s'agira du sigterm (15)
- Pkill
  - ➤ Comme Pgrep mais pour la gestion
- Xkill
  - ➤ Fait apparaître une tête de mort a la place de la souris
  - ➤ Tue l'application dont la fenêtre est cliquée

## 11.3 Priorité

- Plus le chiffre est petit, plus il est important
- De -20 à 19
- 0 par défaut
- Seul root peut définir une priorité entre -20 et -1
- NICE -N [] permet de définir la priorité d'une application et de l'ouvrir
- RENICE change la priorité d'un processus (Seul le propriétaire peut le faire)

## 11.4 Contrôle des processus

- CTRL+C
  - ➤ Arrêt
- CTRL+z
  - ➤ Pause
  - ➤ Affiche la commande avec le numéro de job entre crochet
- FG
  - ➤ Permet de repasser en avant plan un processus
  - ➤ Si on précise le numéro de job, c'est le processus correspondant au numéro qui repassera en avant plan
- BG
  - ➤ Laisse le "job" tourner en arrière-plan
- &
  - ➤ Si placé lors de l'execution d'un job, celui-ci démarrera directement en arrière-plan
- ;
  - ➤ Permet de séparer plusieurs commandes les unes à la suite des autres
- &&
  - ➤ Executer une commande que si la précédente à réussi (a mettre entre les 2 commandes)

## 11.5 Planification

- cron
  - ➤ composé de crond et crontab
  - ➤ crond est le service qui va exécuter régulièrement les commandes voulues
  - ➤ /ETC/CRONTAB est le fichier qui contient les commandes. (cron doit reboot si on modifie crontab)
  - ➤ possibilité de placer des fichier crontab dans /etc/cron.d
  - ➤ /etc/cron.hourly cron.daily cron.monthly contiennent les commandes que l'on veut exécuter toutes les heures, jours, mois
  - ➤ cron ne lance pas les commandes dans un environnement complet. Faire attention a utiliser les chemins complets et initialiser ses propres variables si besoin
  - ➤ Les scripts doivent commencer par un "#"
- crontab
  - ➤ Commande qui permet a chaque utilisateur d'éditer son fichier "crontab"
  - ➤ Ouvre le crontab d'un utilisateur
  - ➤ Configuration identique a /ETC/CRONTAB
  - ➤ Cron vérifie le fichier toutes les minutes
  - ➤ Les fichiers cron.allow et cron.deny permettent de gérer les autorisations (gère l'accès a crontab mais pas les tâches déjà planifiées)
  - ➤ -e editer

    - -l lister
    - -r remove
    - -u d'un utilisateur
- Anacron
    - Si le système est éteint à l'heure de l'exécution d'une tâche, elle ne sera pas exécutée. Anacron s'occupe de lancer ces actions
    - Anacron est exécuté à intervalles réguliers par cron
- At
    - Passe au service atd des commandes à exécuter une seule fois à un moment précis
    - Les accès peuvent être géré de la même manière que crontab

# 12 Etat du système (/proc)

- /PROC
    - Système de fichier virtuel (pas d'existence réelle)
    - Créé et monté à chaque démarrage
    - Permet de communiquer avec le kernel
    - Donne la possibilté d'obtenir des informations mais aussi de modifier la configuration du noyau
    - (Voir la synthèse du prof page 85-88 pour les différents fichiers de /proc)

# 13 Démarrage Linux

- Boot
    - Chercher, charger et executer le code de bootstrap (ex : BIOS)
    - Pareil pour le kernel
    - Executer les instructions de démarrage et les démons
    - Gérer les processus et les états du stystème
- UEFI remplace BIOS

## 13.1 UEFI - BIOS

- Lors de la mise sous tension, le processeur est physiquement câblé de manière a charger et executer le code de démarrage situé a une adresse prédéterminé dans le NVRAM
- Pendant le bootstrapping, le firware va faire l'inventaire du matériel, l'initialiser et le vérifier
- Choisit un périphérique de démarrage (a partir de l'ordre de préférence pré-configuré)
- Démarre le système en chargeant le fichier de démarrage

### 13.1.1 UEFI

- Remplace le bios
- Peut se mettre en mode legacy pour être un supporté par un OS qui ne le supporte normalement pas
- Utilise des partitions GPT
- Peut interpréter le FS FAT
    1. Le firmware cherche la partition "ESP" en FAT dans la GPT
    2. EFI est lu pour charger n'importe quel executable conçu pour être chargé pour l'UEFI (boot manager, GRUB, etc...)
- N'importe quel OS qui support le FAT peut editer l'ESP
- Possibilité de consulter et de modifier UEFI sous linux grâce a l'API de UEFI (EFIBOOTMGR)

### 13.1.2 BIOS

- Firmware historique
- Souvent employé par défaut par les hyperviseurs
- Cherche les infos dans le BOOT LOADER (incapable de lire un FS)

## 13.2 Boot Manager

- Anciennement appellé BootLoader
- Plus indispensables avec l'UEFI (capable de charger le kernel seul)
- Différents boot manager pour Linux
  - ➤ GRUB 2 - Le plus répandu et celui utilisé par défaut
  - ➤ Isolinux - démarrage sur CD, live usb, etc...
  - ➤ pxelinux - démarrage réseau

### 13.2.1 Configuration

- En général dans /boot/grub/
- Utiliser une partition séparée permet de d'utiliser un FS compréhensible pour GRUB et un autre FS pour le reste de l'arborescence
- Pour faciliter le multi-boot, la commande "os-prober" détecte les différents OS et les ajoute a GRUB

## 13.3 Kernel Linux

- Initialisé avec les options passées en paramètres par le boot manager
- Réserve la mémoire et initialise les périphériques qu'il détecte
- Monte le FS racine et libère le ramdisk
- Lance le 1er processus : Systemd (gestion du système)
- Options passées au noyau
  - ➤ debug
  - ➤ racine : root=/dev/sda2
  - ➤ mode single user : single
- Possibilité de voir les messages du kernel quand il démarre grâce au logs et à DMESG
- Ne reste plus que lancer les processus qui tournent dans l'espace utilisateur

## 13.4 Gestionnaire d'initialisation du système - Systemd

- Systemd remplace init et upstart
- Rôle : Faire tourner l'OS avec les bon daemons et paramètres de configuration
- Différents modes :
  - ➤ single-user : sert pour le dépannage, fait le stricte minimum (et le seul utilisateur est root)
  - ➤ multiuser : montages et services, multi-utilisateur et interface graphique
  - ➤ server : comme multiuser mais en CLI
  - ➤ reboot : redémarrage système
  - ➤ etc...
- Systemd est composé de nombreux executables, librairies et composants du kernel ; ce qui le rend beaucoup plus intégré a l'OS que init qui n'est qu'un simple service.
  - ➤ Comme dépendant au kernel Linux, il n'est pas portable sur d'autre "nixes" (comme MACOS)
  - ➤ Peut ne pas supporter certaines version du kernel
- Différences init / systemd

    - ➤ runlevels remplacés par des "targets"
    - ➤ Commande init sans intérêt
    - ➤ inittab non existant
    - ➤ Daemons géré par des fichiers de configuration ("units") et plus des scripts
    - ➤ Systemd est conscient de ce qu'il se passe sur le système contrairement à init qui ne fait qu'executer des scripts
- Units
    - ➤ N'importe quel élément géré par systemd
    - ➤ daemon, socket, timer, etc...
- Systemctl
    - ➤ Gestion de systemd (vérifier le status ou modifier sa configuration)

## 13.5 Init

- Encore utilisé sur les distributions devant être légères
- 1er processus lancé : /sbin/init (lance le système grâce au fichier de configuration /etc/inittab)
- rcS est un script lancé par init qui va lancer les différents scripts dans "rcS.d" qui vont gérer les éléments de base du système (initialisation clavier, hostname, monter les partitions etc...)
- Fini par initialiser le runlevel par défaut puis démarre les consoles

### 13.5.1 RunLevels

- Peut varier selon la distribution
- Par convention :
    - ➤ 1 : mode single user
    - ➤ 0 : éteindre le système
    - ➤ 6 : redémarrage systèmes
- Sous Debian :
    - ➤ 2 : multiuser (par défaut)
    - ➤ 3-4-5 : Non utilisé
- RedHat :
    - ➤ 2 : multiuser sans NFS
    - ➤ 3 : multiuser
    - ➤ 4 : non utilisé
    - ➤ 5 : multiuser + GUI (par défaut)
- Pour changer le runlevel par défaut : "init" et "telinit" ou lors du démarrage (grâce au Boot Manager)

### 13.5.2 /etc/init.d

- Contient tous les scripts de démarrage de service
- Pour s'adapter au runlevel, différents liens symboliques sont crées dans les répertoires rc0.d → rc6.d
    - ➤ Commencent par un "S" si service à démarrer
    - ➤ Commencent par un "K" si service à arrêter
    - ➤ Suivi du numéro d'ordre de démarrage
    - ➤ ex : S20networking
- Les fichiers de configuration des scripts de démarrage de init.d sont dans /etc/default (/etc/sysconfig sur redHat)

### 13.5.3 Problème d'init

- Scripts RC pas tous compatibles avec certaines modules
- execute des scripts sans se préoccuper du système
- Nombreux scripts sur chaque runlevel
- Difficile de lancer un service suite à une action ou de relander automatiquement un service
- Travail dupliqué d'une distribution à l'autre (doit y être intégré)
- Difficilement modifiable par l'utilisateur
- Pas de gestion de dépendances
- Pas de parallèlisme (pas compris ce point)

# 14 Gestion du kernel

## 14.1 Rôle du kernel

- gérer les processus
- gérer les périphériques
- gérer les INPUT/OUTPUT
- gérer les interruptions
- gérer les ressources
- gérer les FS

## 14.2 Type de kernel

- monolithique
  - ➤ Le kernel forme un gros bloc de code qui gère tous les rôles
- microkernel
  - ➤ Le noyau est minimal et le maximum des features sont déportés sous dorme de serives dans l'espace utilisateur
- hybride
  - ➤ microkernel dont toutes les features ne sont pas externalisé (gain de perfs)
- Linux est monolithique modulaire (les modules peuvent être chargé/déchargé)

## 14.3 Version du noyau

- Commande UNAME
  - ➤ Forme : (NuméroMajeur).(NuméroMineur).(NiveauDePatch)
  - ➤ Numéro mineur :
    - — Impair → version de développement
    - — Pair → version stable

## 14.4 Modules

Morceau de code pouvant être chargé/déchargé du noyau

- /lib/modules/version_kernel/ (extension .ko)
- Risque perte de performance et de sécurité
- LSMOD liste les modules actifs
- MODINFO info sur un module
- INSMOD charger un module
- MODPROBE charger un module en gérant ses dépendances
- RMMOD désinstaller un module
- Possibilité de spécifier les modules à charger lors du démarrage grâce au boot manager

## 14.5 Compilation

- gcc
  - ➤ Compilateur C
- make
  - ➤ outils pour gérer la compilation

# 15 PAM

**Merci a Thibaut Watrisse pour ses notes**
**P**luggable **A**uthentification **M**odules

- Ensemble des librairies qui servent à fournir une authentification flexible sur un systèmme UNIX
- Réponse au soucis des développeur devant mettre en place un mécanisme d'authentification pour leur application
- L'administrateur va pouvoir changer comme il l'entend de mécanisme d'authentification, du moment que le module PAM nécessaire existe
- But de PAM : Séparer le dev. des applications de celui du mécanisme d'authentification sécurisé

## 15.1 Fonctionnement de PAM

- Login
  - ➤ vérifie que l'utilisateur est bien celui qu'il prétend être
    - — Login appelle PAM, puis PAM va lire ses fichiers de config et charge ensuit les modules appropriés
  - ➤ L'échange de messages avec l'utilisateur (envoyés ou reçus) est effectué à travers l'application via la fonction conversation()
  - ➤ fournit le shell tournant avec l'identité de l'utilisateur
- Gère l'authentification, les comptes, les sessions, les mdp

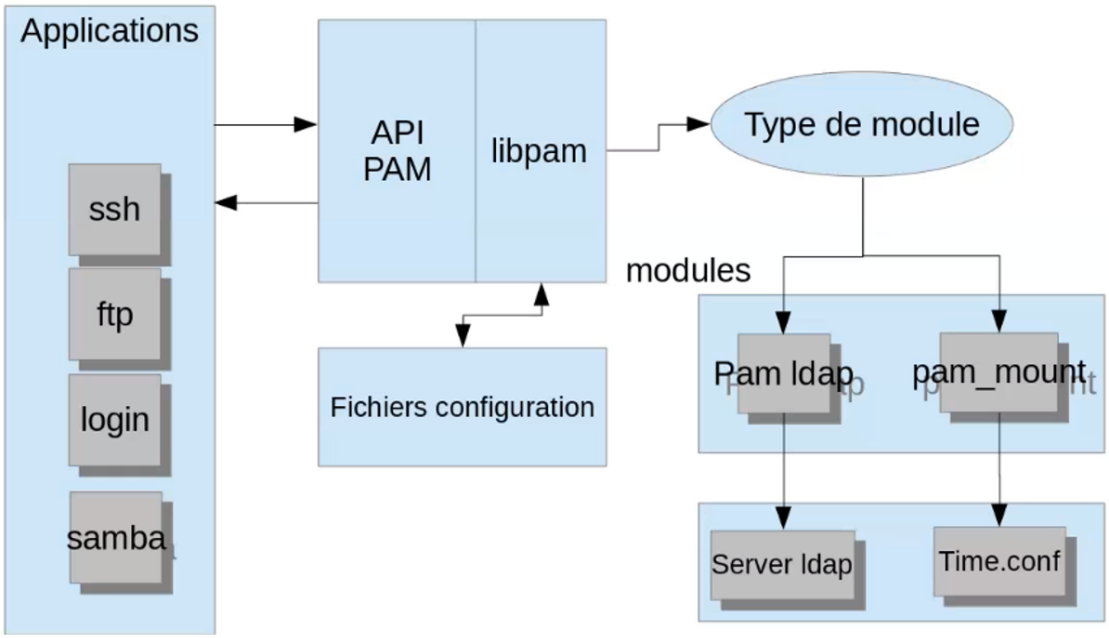


Figure 3 – Représentation visuelle du LVM

## 15.2 Config.

- Librairies
  - /lib/security
  - /lib64/security
- Fichiers de config.
  - /etc/pam.d
  - /etc/pam.conf
- Fichiers de config. des modules
  - /etc/security
- Chaque application utilisant PAM correspond un fichier dans pam.d portant le nom du service concerné
- Structure fichier pam

  service type_module contrôle_chemin_module arguments_module
- Conseil :
  - Si erreur dans la configuration → échec de PAM
  - Prendre ses précautions lors de modifications dans un fichier de configuration
- Différents types de modules
  - auth
    - sert à authentifier l'utilisateur
    - donne l'appartenance aux groupes (ou autres privilèges)
  - account
    - Tâche d'un compte (sauf authentification)
    - ex : restreindre l'accès a certaines heures, les ressources, etc...
  - session
    - environnement
    - monter des répertoires
    - création d'un répertoire personnel
    - (Tout ce qui est lié a une session)
  - password
    - ce module est requis pour mettre à jour les données d'authentification de l'utilisateur
    - Possibilité de mettre en place des stratégies de mdp
- Flags :
  - requisite
    - En cas d'échec, l'information est retournée directement, et les modules suivants ne sont pas parcourus
  - sufficient
    - Si réussite, envoie directement le message de réussite à l'application sans tester les modules suivants
    - Si échec, pas d'importance, sauf s'il s'agit du dernier module de la pile.
  - optionnal
    - PAM passera au module suivant dans la pile que ce soit un échec ou une réussite

### 15.3 Modules

| Module | Description |
|---|---|
| pam_console | Permet de donner à des user des droits qu'ils n'auraient pas autrement |
| pam_access | Limite l'accèes sur base de critères supplémentaires que le critère d'authentification |
| pam_cracklib | Vérification des mdp (imposer une taille, caractères spéciaux,…) |
| pam_deny | Sert à retourner un échec |
| pam_ftp | Permet l'accès anonyme sur FTP |
| pam_issue | Pour afficher un petit message avant de demander le login |
| pam_limits | Donne des limites dans les ressources utilisateur |
| pam_listfile | Permet d'autoriser(ou pas) sur base d'un fichier |
| pam_mount | Permet de monter un système de fichier distant |
| pam_mkhomedir | Permet de créer un répertoire personnel |
| pam_nologin | Vérifie qu'un fichier nologin existe dans etc ou pas, si c'est le cas, il n'y a que root qui peut se connecter |
| pam_permit | Renvoie automatiquement « succes » |
| pam_rootok | Permet à root d'avoir accès à certains services sans taper le mdp |

# 16 Radius

**Merci a Thibaut Watrisse pour l'aide des notes**
**R**emote **A**uthentification **D**ial-**I**n **U**ser **S**service (Dial-in = composer)

Exemple de mise en place d'une authentification radius qui m'a aidé à mieux comprendre :
https ://www.youtube.com/watch ?v=abIOD4O4AJg

- UDP Port 1812
- But de radius
  - ➤ Va permettre l'authentification à un service de manière individualisé
  - ➤ Exemple : Au lieu d'utiliser une seule clé WiFi, chaque employé à son propre accès au service WiFi
- Serveur
  - ➤ Gère l'authentification
  - ➤ Utilise une base de donnée d'Identification (.txt, SQL, LDAP, etc..)
    - — Peut être propre au serveur Radius ou bien la base de donnée d'un Active Directory par exemple
- Client
  - ➤ NAS (Network Acces Server)
  - ➤ Pour n'importe quel service demandant une authentification
  - ➤ Par exemple : Un routeur Wifi
- Fonctionnement
  - ➤ Le client radius va s'authentifier auprès du serveur Radius avec une clé PSK (clée partagée)
    - — La sécurité entre le serveur et le client est assurée par cette clée qui crypte notamment le mot de passe de l'utilisateur
  - ➤ L'utilisateur contacte le client Radius pour une demande d'authentification
  - ➤ Le client radius traite la demande d'authentification auprès du serveur Radius
- Protocoles
  - — EAP
    - ➤ Extensible Authentication Protocol
    - ➤ 802.1x = intégration de EAP dans la couche 2
- MDP

    - PAP
        - Simple authentification par mot de passe
        - Password Authentication Protocol
        - Le user envoie son login et mdp en clair
        - Le serveur renvoie soit un AUTHENTICATION-ACK si OK, sinon un AUTHENTICATION-NACK
    - CHAP
        - Challenge-Handshake Authentication Protocol
        - 4 types de paquets : Challenge, Reponse, Succès, Echec
            1. L'authentificateur envoie un challenge au client (Access-Challenge)
            2. Celui-ci répond avec une valeur calculée à l'aide d'une fonction de hashage (Access-Request)
            3. L'authentificateur vérifie la réponse par rapport à son propre calcul
            4. Si les hash correspondent, alors l'authentificateur reconnaît l'authentification
            5. Le challenge est renouvelé à des intervales random par sécurité
- Extensions Microsoft
    - Mschap
    - Mschapv2
- Problèmes : Sécurité des échanges faible
    - Solution :
        - Protéger les flux de données via VPN
        - Privilégier des méthodes sûre entre user et client

# 17 Kerberos

**Merci a Thibaut Watrisse pour l'aide des notes**

Rôle : Fournir un système d'authentification unifié en milieu hétérogène (pour n'importe quel client, serveur,...)

- Authentication
    - De manière sûre
    - De manière centralisée (Single Sign On - SSO)
- CAS
    - Central Authentication Service
    - Mdp centralisé dans un AD (ou autre système)
    - Permet de se connecter sur divers services avec nos mêmes identifiants
- SSO - Single Sign On
    - Plus avancé que "CAS"
    - Lors de la connexion, on reçoit un jeton pendant un certain temps nous permettant de nous connecter a nos différents services sans passer par la case "authentification"

## 17.1 Protocole Needham et Schroeder

- A = Alice
- B = Bob
- S = Tiers parti de confiance
- K = Key
- $K_{BS}$ = Key partagée entre Bob et le tiers parti de confiance
- $N_x$ = Nombre aléatoire généré par $x$

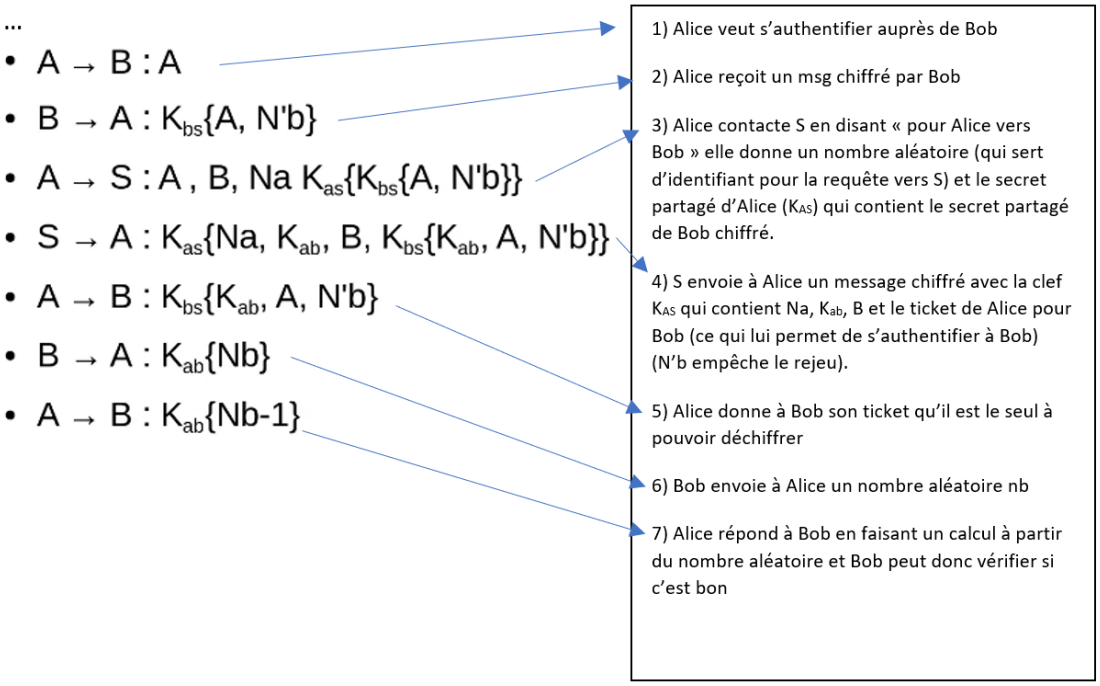


FIGURE 4 – Etapes (Image de Watrisse Thibaut)

Etant donné qu'il est difficile de générer des vrais nombres aléatoires avec Kerberos, on utilise donc des "Time Stamp"

- KDC
  - ➤ Le tiers parti de confiance
  - ➤ Génère un "royaume" kerberos (kerberos realm)
  - ➤ Divisé en 2 services
    - — TGS - Ticket Granting Server qui fourni les tickets
    - — AS - Authentication Service auprès duquel les clients s'authentifient
- Une fois connecté à l'AS, on peut demander des tickets au TGS
  - ➤ Les clefs secrètes sont les hash des mdp $K_a$
  - ➤ Anti-rejeu
    - — Time Stamp
    - — Durée de vie des tickets
- Etant donné qu'on utilise des time stamp, toutes les machines doivent être configuré à la même geure
- Différence d'heure max sur un active diretory : max 5 min.

## 17.2 En résumé

- — REQ = Request
- — REP = Reply
- — AS = Authentication Service
- — TGS = Ticket Granting Server
- — Bob = Service
- — Alice = Utilisateur

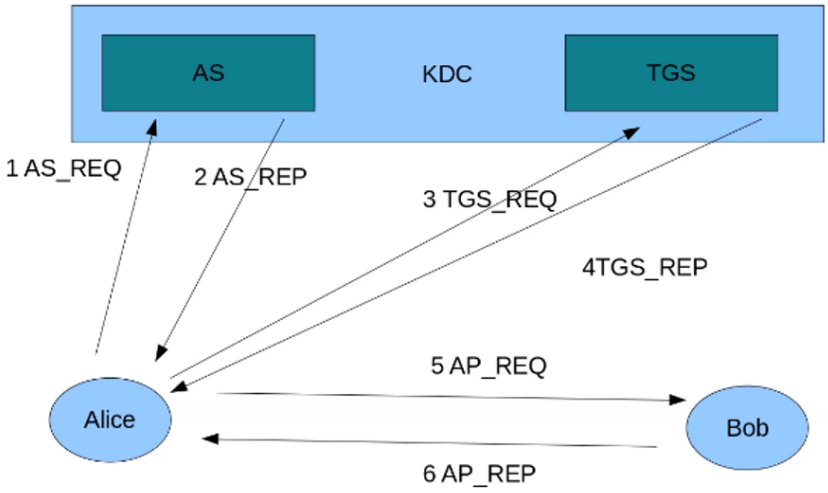


FIGURE 5 – Fonctionnement de Kerberos

1. Alice s'authentifie dans le royaume Kerberos (Demande d'un TGT)
   - A → AS : A, ticketlifetime,[$K_a${timestamp}]
   - Pré authentification optionnelle
     - Si il n'y a pas de pré authentification l'AS n'est pas sûr que c'est elle mais ce n'est pas important car si ce n'est pas Alice, elle ne comprendra pas la réponse de l'AS car l'usurpateur ne connaitra pas le secret partagé.
2. AS envoie la durée de vie du ticket et la session key (TGT - Ticket Granting Ticket) entre ALice et le TGS
   - AS → A : $K_a${Ticket lifetime, $K_{a,tgs}$}
   - AS → A : $K_{tgs}${A, timestamp, $K_{a,tgs}$,ticket lifetime}
3. Alice contacte TGS (Demande d'un TS)
   - A → TGS : $K_{a,tgs}${A, timestamp, ticket lifetime}
     - Le TGS saura grâce au TGT que ça vient d'Alice
4. TGS répond à la demande pour le service bob avec un ticket de service TS
   - TGS → A : $K_{a,tgs}${Bob, timestamp, ticket lifetime}
   - TGS → A : $K_b${Alice, Bob, $K_{ab}$, timestamp, ticket lifetime}
5. Alice s'authentifie auprès de Bob
   - A → B : $K_{ab}${A, timestamp, ticket lifetime}
   - B → A : $K_{ab}${timestamp - 1}
     - La réponse de Bob va permettre de vérifier que le timestamp n'a pas déjà été utilisé (rejeu)
     - Bob peut alors déchiffrer le message d'Alice grâce à TS qui contient la clef de session entre Alice et Bob
     - Une fois le message déchiffré, Bob est sensé voir apparaître l'ID d'Alice, le Ticket Lifetime, et le timestamp

## 17.3 Points Negatifs

- Dépendances au temps
- Si KDC tombe en panne, plus possible de s'authentifier nulle part (Single point of failure)
- Si mdp compromis, alors KDC est compris aussi (Solution : Passer par des certificats)
- Les services doivent supporter Kerberos
- Il faut utiliser des algorithme symétrique fort (AES par exemple)

# Table des matières

**1 A savoir sur Linux** **1**
1.1 Qu'est-ce que c'est ? . . . 1
1.2 Les distributions . . . 1
1.2.1 Qu'est-ce que c'est ? . . . 1
1.2.2 Quelques exemples . . . 1

**2 Principes de base** **2**
2.1 Comptes utilisateurs . . . 2
2.2 Login . . . 2
2.3 Arrêt/Démarrage . . . 2
2.4 Shell . . . 2
2.5 Commandes . . . 3
2.6 All -> Fichier . . . 3

**3 Interface graphique** **3**
3.1 Introduction à X . . . 3
3.2 Serveur X . . . 3
3.3 Les gestionnaires de fenêtres . . . 3
3.4 Gestionnaires de bureau . . . 3

**4 Installation** **3**
4.1 Mise en place du/des FS . . . 3
4.2 Partitions . . . 4
4.3 Disques & partitions sous Linux . . . 4

**5 Arborescence & File System** **4**
5.1 Introduction . . . 4
5.2 Règles des FS UNIX . . . 5
5.3 Arborescence . . . 5
5.4 Référence relative & Absolue . . . 6
5.5 Manipuler des fichiers . . . 6
5.6 Afficher le contenu d'un fichier . . . 6
5.7 Autres commandes utiles . . . 7

**6 Le Shell : BASH** **8**
6.1 Shell . . . 8
6.1.1 Introduction au Shell . . . 8
6.1.2 Lancer des commandes dans le Shell . . . 8
6.1.3 L'environnement des commandes . . . 8
6.1.4 Les caractères spéciaux . . . 9
6.1.5 Les wildcars ou Jokers . . . 9
6.1.6 Variables d'environnement . . . 9
6.2 Les commandes de BASH . . . 10
6.2.1 Historique . . . 10
6.2.2 Complétion des commandes . . . 10
6.2.3 Touches reconnues par le shell . . . 10
6.3 Toutes les infos sur Linux . . . 10
6.3.1 L'aide . . . 10
6.3.2 man . . . 10
6.3.3 info . . . 10
6.3.4 apropos . . . 10
6.3.5 La documentation . . . 11
6.4 Entrée & sortie standart . . . 11
6.4.1 Redirections . . . 11

6.4.2 Pipes . . . 11
6.4.3 Tee . . . 11
6.4.4 Editeur de texte . . . 11

**7 Gestion des paquets** **11**

**8 Users and groups** **12**
8.1 La base . . . 12
8.2 Fichiers concernés . . . 13
8.3 Changer le mot de passe . . . 13
8.4 Limiter l'usage d'un compte . . . 13
8.5 Modifier les propriétés d'un compte . . . 13
8.6 Informations sur les utilisateurs . . . 14

**9 Gestion avancées des fichiers** **14**
9.1 Droits spéciaux . . . 14
9.2 Inodes . . . 14
9.3 Types de fichiers . . . 14
9.4 Lien . . . 15
9.4.1 Liens durs . . . 15
9.4.2 Liens symboliques . . . 15
9.5 Liste des fichiers périphériques . . . 15
9.6 Numéro de périphériques . . . 15

**10 Gestion des systèmes de fichiers** **15**
10.1 Outils de partitionnement . . . 15
10.2 Systèmes de fichiers . . . 16
10.3 Montages . . . 16
10.4 Taille des partitions . . . 16
10.5 Gestion des quotas . . . 16
10.5.1 Différentes limites de quota . . . 16
10.5.2 Mise en place des quotas . . . 17
10.5.3 Modification des quotas . . . 17
10.6 RAID - Redundant Array of Independent Disks . . . 17
10.7 LVM . . . 18

**11 Gestion Processus** **19**
11.1 Afficher les processus . . . 19
11.2 Interaction Processus . . . 19
11.3 Priorité . . . 20
11.4 Contrôle des processus . . . 20
11.5 Planification . . . 20

**12 Etat du système (/proc)** **21**

**13 Démarrage Linux** **21**
13.1 UEFI - BIOS . . . 21
13.1.1 UEFI . . . 21
13.1.2 BIOS . . . 22
13.2 Boot Manager . . . 22
13.2.1 Configuration . . . 22
13.3 Kernel Linux . . . 22
13.4 Gestionnaire d'initialisation du système - Systemd . . . 22
13.5 Init . . . 23
13.5.1 RunLevels . . . 23
13.5.2 /etc/init.d . . . 23
13.5.3 Problème d'init . . . 24

**14 Gestion du kernel** **24**
14.1 Rôle du kernel . . . 24
14.2 Type de kernel . . . 24
14.3 Version du noyau . . . 24
14.4 Modules . . . 24
14.5 Compilation . . . 25

**15 PAM** **25**
15.1 Fonctionnement de PAM . . . 25
15.2 Config. . . . 26
15.3 Modules . . . 27

**16 Radius** **27**

**17 Kerberos** **28**
17.1 Protocole Needham et Schroeder . . . 28
17.2 En résumé . . . 29
17.3 Points Negatifs . . . 30