Introduction à GNU/Linux

Pascal Bessonneau

11/2017

Table des matières

1 Rapide historique de GNU/Linux

1.1 UNIX

Le premier système d'exploitation a été conçu par Ken Thompson et Dennis Ritchie (AT&T Bell Laboratories) en 1970. Il était écrit en C déjà.

BSD a tenté de faire un fork mais un procès a eu lieu car BSD embarquait des codes propriétés de AT&T.

(ce qui a changé depuis, BSD est open source complètement). Mais ça a freiné l'expansion de BSD.

1.2 Le noyau

Il assure l'interface entre les logiciels et le matériel.

MINIX, un système Unix-like pour usage académique a été écrit par Andrew S. Tanenbaum.

Puis Linus Torvals, le 25 August 1991, poste un message pour dire qu'il a écrit un MINIX like en C qui pouvait faire tourner un compilateur et bash (le mode console).

Appelée au début FreaX, il fut renommer par un administrateur d'un serveur FTP Linux. Ce qui fut accepté par Linus Torvalds.

Rapidement il fut placé sur GNU General Public License (GPL) : v1 puis v2 et enfin v3.

Il rassembla pas mal de développeurs et surtout des companies vont s'en emparer. En effet en 2015, 80% des développeurs sont rémunérés pour leurs contributions.

1.3 Le projet GNU

Tout cela (le succès de Linux) n'aura pas eu lieu sans le projet GNU lancé par Richard Stallman qui, en 1983, a démarré le projet GNU avec le projet de créer un clone d'UNIX opensource. La communauté est aussi à l'origine des licences qui encadrent la plupart des logiciels opensource.

1.4 Le débuts des distributions

En 1993 deux distributions ont été crées pour faciliter l'installation et la maintenance des systèmes GNU/Linux : Slackware et Debian.

Aujourd'hui faire Linux from Scratch (entièrement à la main) est reservé à l'apprentissage. Ce sont les distributions (la pléthore) de distributions qui permet d'installer et de gérer son ordinateur sous GNU/Linux.

2 Versions

2.1 Unstable

Dans une première étape, une version expérimentale est réalisée. Elle est dite *unstable*.

Elle n'est pas recommandé pour les novices et en production. A ce stade les versions des logiciels ne sont pas toujours fixées.

Souvent son numéro après la virgule est impaire (au moins pour les Debian et Fedora).

2.2 Release candidate

Dans cette version plus mature, la plupart des gros bugs sont résolus. A cette étape, les testeurs et développeurs chasse les derniers bugs mineures. La version des logiciels dans la distribution est fixée à ce stade.

Souvent on la voit retrouve appelée RC pour Release Candidate, plus le nombre augmente plus la version est fiable.

2.3 Stable

Dans cette version mature, les bogues sont résolus. A cette étape, la distribution est stable et peut être installé par tous.

Le temps durant lequel la communauté fournit les mises à jour varie selon la version et la distribution (support de la version). Par exemple pour la Debian c'est trois ans.

2.4 Long Term Support

Certaines distributions comme Ubuntu, distingue deux types de distributions :

les versions normales (pour Ubuntu x.10) supportées pendant une brêve période

les versions LTS qui sont supportées pendant plus longtemps (pour Ubuntu x.04 pendant 5 ans)

2.5 Trouver la version installée

En console il suffit de taper *lsb_release -a*.

Dans paramètres sous GNOME, on peut l'avoir dans le groupe « Détails ».

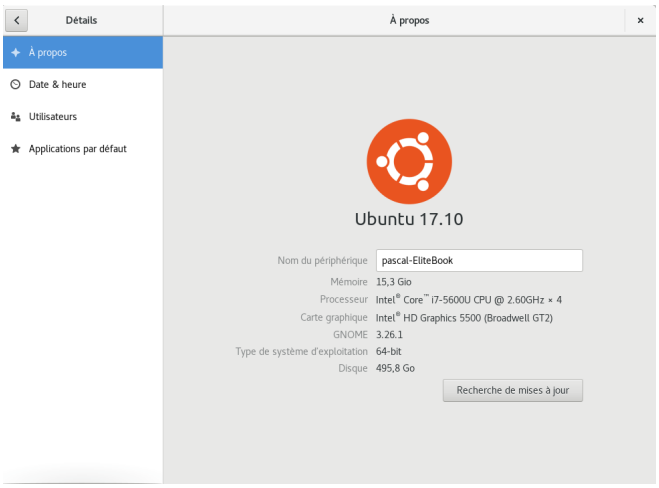

FIGURE 1 – Version sous GNOME

3 Recommandations

3.1 Versions préconisées

Ubuntu c'est la plus simple à utiliser pour un novice. Enfant de Debian elle a aussi le plus gros volume de documentation en ligne. Des variantes pour les ordinosaures existent comme Lubuntu

Mint Enfant d'Ubuntu, on a les mêmes avantages mais avec un bureau par défaut de type Cinnamon

Fedora c'est celle, qui dans la simplicité, vient juste après. Il y a quelques manipulations à faire pour avoir les contenus non libres comme les plug ins Flash... après l'installation.

Arch,Manjaro pour les aventureux qui veulent les logiciels « on the edge »

3.2 Préconisations à l'installation

– Lors de l'installation, faites la sur un disque entier puis avec ce que nous verrons, mettez le home sur un disque différent pour avoir plus de place et pour en cas de réinstallation avoir toujours votre HOME propre.

3.3 Préconisations à l'installation d'Ubuntu

Ubuntu installe le plugin flash, le MP3 si demandé à l'installation. Il ne reste plus grand chose à faire.

A la rigueur le dépot Google pour Chrome (Netflix, CanalPlay,...) :

```
wget -q -O - https://dl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
sudo echo "deb https://dl.google.com/linux/chrome/deb/ stable main" > /etc/apt/sources.list.
```

Et GNOME-tweak-tool si il n'est pas installé :

```
sudo apt-get install gnome-tweak-tool
```

3.4 Préconisations à l'installation pour Fedora

- Ajouter les dépots rpmfusion :
  ```
  sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(r
  sudo dnf install https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-relea
  ```
- Installer Yum *dnf install yum*
- Installer GNOME Tweak Tool *dnf install gnome-tweak-tool*
- Installer flash-player-ppapi et/ou flash-plugin
- pour installer les logiciels google comme Chrome pour Netflix, ...
  ```
  cat << EOF > /etc/yum.repos.d/google-chrome.repo
  [google-chrome]
  name=google-chrome - \$basearch
  baseurl=http://dl.google.com/linux/chrome/rpm/stable/\$basearch
  enabled=1
  gpgcheck=1
  gpgkey=https://dl-ssl.google.com/linux/linux_signing_key.pub
  EOF
  ```

4 Séquence de Boot

4.1 Recherche du software après le démarrage matériel

Le matériel va chercher le software qui est nécessaire pour booter :
- une partition bootable
- une partition *UEFI*

4.2 Old school

la partition doit être bootable. Pour la rendre bootable il suffit d'utiliser *gparted* ou *fdisk* pour la rendre bootable.

Sinon tous les installeurs le font d'eux-mêmes.

4.3 UEFI

L'UEFI installé sur les ordinateurs complexifie le processus en demandant que le soft de démarrage soit signé ce qui peut poser des problèmes car *GNU/-Linux*.

D'autre part il exige une partition en FAT32 pour démarrer bien particulière et il faut une signature particulière répondant au doux nom de *EF00*. Il faut également que la table des partitions soit au format *gpt*.

L'*UEFI* est connu pour une bête noire pour la configuration sous *Linux*. Les distributions mainstream gèrent maintenant l'*UEFI*. Tant qu'on a que *Linux* sur le poste. En cas de double boot, on commence à avoir des problèmes.

more...

4.4 GRUB

Ensuite le démarrage va charger en mémoire le petit noyau *Linux* ...

le noyau par défaut est celui qui s'appelle vmlinuz à la racine /. En fait c'est un lien symbolique vers le noyau qui se trouve vraiment dans */boot*.

GRUB selon ses réglages peut démarrer sur un autre noyau. La configuration de *GRUB* se fait en lançant *update-grub*.

La partie automatisée de *GRUB* est situé dans le répertoire */boot/grub*. C'est là qu'on retrouve des scripts *bash* qui vont construire le menu de démarrage et qui vont être appelé quand on fait *update-grub*.

Pour les modifications manuelles, il faut éditer le fichier *40_custom* dans */etc/grub.d* ou directement (plus simple) le fichier */etc/default/grub*.

Point sécurité : il est possible avec les bons arguments de booter avec le menu *démarage personnalisée* en mode single user, ie. en root (voir ici)

Pour ne pas le permettre il faut ajouter un mot de passe à taper avant d'accéder à cette personnalisation.

Vous trouverez la manipulation ici. Elle se ressemble sur *Feodra* et *Ubuntu*.

4.5 vmlinuz

Il s'appelle ainsi car il est compressé à une époque ou le programme ne devait pas dépasser une taille critique. D'ailleurs sur certains ordinateurs on étaient obligés de modifier (compiler soi même le noyau) pour qu'il rentre dans la taille spécifiée.

En effet il y a dans *vmlinuz* le système d'exploitation et et des "modules". Un module est un bout du noyau qui ajoute des fonctionnalités : une interface pour un matériel particulier, la possibilité de lire un format de fichier, ...

Au démarrage dans *GRUB*, on peut charger un module ou au contraire empêcher son chargement.

C'est utile par exemple à l'heure où j'écris ces lignes pour une installation avec la carte Nvidia pour *Feodra* (pareil sous *Ubuntu*) ici.

Le principe des modules est qu'ils sont chargés à la demande du noyau et évite ainsi d'avoir un noyau énorme et qui serait moins performant (?).

4.6 init/system.d

A ce stade, nous sommes en pleine transition en deux logiciels pour continuer le démarrage, le système *init* et le *systemd*.

Le système *init* est en phase d'être abandonné au profit de l'autre.

Dans la plupart des distributions vous pouvez constater leur travail en appuyant sur la touche *ESC* de votre ordinateur. Vous verrez une ligne par service chargé.

Par service, il faut entendre qu'il va charger le support réseau, le support du matériel, ...

4.7 init/systemd

Sous *Feodra* le systemd est presque complètement adopté. Sous Ubuntu c'est moins clair car ils ont essayé de lancer leur propre système de démarrage *Upstart* qui n'a pas pris dans la communauté. Donc Ubuntu c'est un peu le f...

Pour activer ou désactiver un service de systemd, il faut utiliser la commande *sysctl*.

Vous pouvez regarder là.

Les différents services sont théoriquement dans le répertoire */etc/systemd/-system*.

4.8 run.level

Pour gérer un service init, c'est les commandes :

```
update-rc.d <nom du service> stop|start|enable|disable
```

Il faut parfois rajouter l'option *-f* pour pour forcer la commande. Il vous le signale si c'est nécessaire.

Les *run.levels* sont fondamental dans *Linux*. Ils sont appelés aussi par l'appellation System V.

Une machine *Linux* est toujours dans un état défini par un *run.level*. Par exemple le mode multi-user (généralement *rc2.d*), il a un *run.level rc0* pour l'extinction, le single user le *rcS*...

Ces états sont décrits dans un fichier particulier */etc/inittab*.

4.9 Interface de connexion graphique

L'interface de connexion graphique est maintenant lancé par défaut pour la plupart des *GNU/Linux*.

Cette interface vous permet de vous authentifier et de lancer votre bureau préféré.

Une interface a été créée par la plupart des bureaux mais vous avez avoir une interface GNOME et lancer KDE.

Les plus connues sont évidemment celles des bureaux les plus connus : *kdm* (KDE), *gdm* (GNOME), *lightdm* (LXDE),...

La méthode la plus simple pour reconfigurer l'interface de démarrage (passer de l'une à l'autre) est de lancer la configuration de l'interface installée :

```
dpkg-reconfigure gdm
```

4.10 Xorg, Wayland

Xorg est une couche logicielle qui permet d'avoir des interfaces graphiques sous *GNU/Linux*.

Il provient d'une ancètre qui s'appellait X sous UNIX. Il est en passe d'être remplacer dans les prochains mois par Wayland. C'est déjà le cas sous Fedora.

C'est la couche qui va gérer votre écran, votre souris, ... Il est à l'interface entre le bureau et le matériel.

4.11 Bureau

Le bureau est schématiquement un gestionnaire de fenêtre avec des applications dédiées.

Les plus connus sont GNOME, KDE, Mate, Cinnamon, ...

4.12 En cas de problème lors du démarrage

Si votre ordinateur sous *GNU/Linux* ne démarre pas, il existe des solutions plus ou moins simples pour savoir ce qui cloche :

1. Si le menu GRUB ne s'affiche pas c'est que le boot ou l'UEFI ne marche pas
2. Si *Windows* se lance au lieu de *GNU/Linux* c'est que le boot ou l'UEFI ne marche pas

3. Si en appuyant sur *ESC* vous voyez que l'ordinateur bloque sur un service, c'est peut être ce service
4. Si l'ordinateur s'arrête et demande à passer en root pour des réparations, un service ou un périphérique (ex : disque dur manquant) ne fonctionne pas

1. Si l'ordinateur reste noir ou bloqué sur l'écran de démarrage, vous pouvez essayer de voir si la console est disponible sur les autres terminals (CTRL+ALT+F1, CTRL+ALT+F2,...). Si une console est disponible, alors c'est l'interface de démarrage graphique qui bloque ou Xorg.
2. Si l'ordinateur revient à l'interface graphique de connexion par exemple c'est le bureau qui est en cause ou votre *home* directory (le bureau ne démarre pas si votre home n'est pas accessible en lecture ou s'il est plein).

5 Accéder au terminal

5.1 Introduction

5.2 Terminal/Console

Le Terminal est souvent nécessaire sous *Linux*. Plus précisement il est souvent plus facile de passer par le Terminal pour faire des réglages qu'en mode graphique.

C'est à la fois un plus et le talon d'Achille de *Linux*.

Le Terminal est souvent dans le répertoire des outils systèmes (KDE, Cinnamon,...). Il s'appelle *Terminal* dans Gnome3.

On peut le lancer et ouvrir plusieurs sessions (les logiciels supportent quasiment tous plusieurs sessions). Chaque session étant indépendante.

La console est aussi accessible sur les bureaux CTRL+ALT+F1, CTRL+ALT+F2, ..., CTRL+ALT+F5. Initialement l'interface graphique était sur le bureau 6 ou 7 maintenant on peux le trouver sur le bureau 1 ou 2.

6 Commandes primaires

6.1 Raccourcis clavier

Vous pouvez utiliser la flêche haute pour rappeler des commandes déjà soumises. Si vous tapez le début d'une commande ou d'un fichier vous pouvez compléter avec Tab.

6.2 Terminal/Console

Les commandes primaires seront :

cd change de répertoire. Pour changer de répertoire il suffit de donner son nom, vide vous retournez à votre répertoire personnel.

ls liste les fichiers et répertoires. Pour avoir les détails sur les fichiers utiliser *ls -l*

cp copie les fichiers et répertoires.

mv copie les fichiers et répertoires.

nano éditeur de texte. il suffit d'ajouter le nom du fichier à ouvrir

6.3 cd

cd retourne au répertoire personnel

cd .. va au répertoire parent

cd items/sauvegarde va dans le répertoire items/sauvegarde

cd /var/items/sauvegarde va dans le répertoire items/sauvegarde

cd ~pascal va dans le répertoire personnel de pascal

6.4 ls

ls liste les fichiers

ls -l liste les fichiers avec les détails

ls -la liste les fichiers avec les détails y compris les fichiers cachés

6.5 cp

cp a b copie le fichier a en b

cp -f a b écrase le fichier b avec le contenu du fichier a

cp -R d e copie tout le répertoire d dans le répertoire e

6.6 mv

mv a b déplace le fichier a en b

mv -f a b déplace en écrasant le fichier b avec le contenu du fichier a

mv -R d e déplace tout le répertoire en répertoire e

7 Utilitaires pour le texte

7.1 nano

nano fichier ouvre le fichier fichier

- Pour écrire dans un fichier ou le sauvegarder, utilisez Ctrl-o
- Pour quitter Nano, Ctrl-x
- Pour rechercher dans le fichier, Ctrl-w

Pour le copier/coller, il faut au début du texte à copier faire CTRL-shift-6, puis flèche gauche/droite pour sélectionner le texte.

La sélection finie il faut appuyer sur Alt-Shift-6.

Pour le coller, il faut appuyer sur CTRL-u.

Pour l'affichage de fichiers texte comme les logs il y a 4 commandes à se rappeler :

head -¡n¿ affiche les n premières lignes d'un fichier

tail -¡n¿ affiche les n premières lignes d'un fichier

cat affiche tout le fichier dans la console du haut vers le bas

tac affiche tout le fichier dans la console du bas vers le haut

7.2 more

Pour l'affichage de grands fichiers texte : *more* ou *less* affiche le fichier page par page (espace pour changer de page et q pour quitter)

8 Introduction

8.1 Qu'est ce qu'un utilisateur ?

C'est un compte permettant généralement une connexion (graphique ou console).

Il est associé sous UNIX à un numéro qui est utilisé pour identifier les fichier et les répertoires

Il est généralement associé à la présence d'un répertoire utilisateur qui lui appartient.

Il existe des utilisateurs spéciaux qui n'ont pas de répertoire et/ou la connexion n'est pas possible.

C'est le cas pour de nombreux serveurs qui fonctionnent ainsi pour des raisons de sécurité.

9 Gestion des utilisateurs

9.1 Créer/supprimer un utilisateur

Pour créer un utilisateur, il suffit d'utiliser la commande *adduser ¡utilisateur¿* précédé de *sudo*.

Et après il faut répondre aux questions, relativement simple.

En effet la gestion des utilisateurs est réservé au super utilisateur.

Pour supprimer un utilisateur, il faut utiliser *deluser ¡utilisateur¿* et de répondre aux questions.

9.2 Les fichiers de référence

Le fichier des utilisateurs est le fichier /etc/passwd.

Le fichier des groupes est le fichier /etc/group.

Le fichier des groupes est plus intéressant car il permet de voir quels groupes contiennent quels utilisateurs. C'est utile par exemple pour copier les droits de l'utilisateur créé par défaut par votre distribution.

Mais il faut éviter de manipuler directement ces fichiers car s'ils sont corrompus le système d'authentification peut planter

9.3 Gestion des groupes

Pour ajouter un groupe à un utilisateur, il faut utiliser la commande : *usermod -a -G ¡groupe¿ ¡utilisateur¿*.

Pour ajouter un groupe à un utilisateur, il faut utiliser la commande : *gpasswd -d ¡utilisateur¿ ¡group¿*.

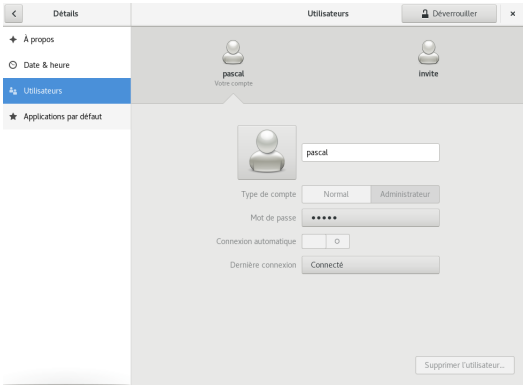

FIGURE 2 – Utilisateurs dans le menu Détails de l'application Paramètres

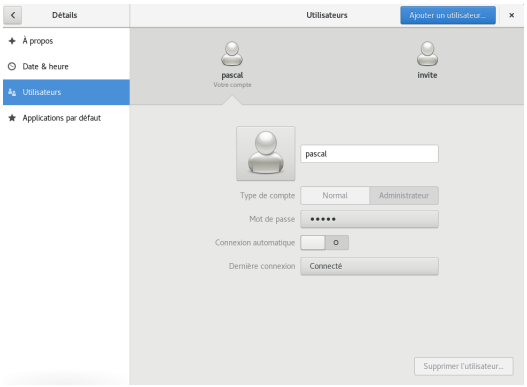

FIGURE 3 – Utilisateurs, déblocage avec le mot de passe

9.4 Nota bene

Les droits sont des numéros.

Si l'utilisateur pascal est 1001 sur un ordinateur et 1002 sur un autre ordinateur, alors si vous échangez les disques les fichiers ne seront pas reconnus comme pascal sur l'un ou l'autre des ordinateurs.

9.5 En mode graphique

Sous KDE, l'utilitaire KUser permet de le faire graphiquement comme system-config-users sous GNOME.

9.6 Sous GNOME 3

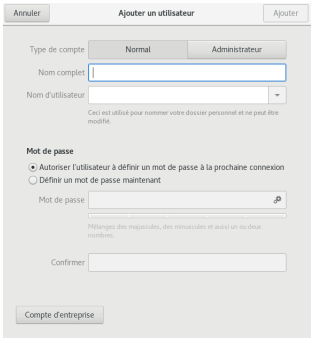

FIGURE 4 – Utilisateurs, nouvel utilisateur

10 La théorie

10.1 Les systèmes *GNU/Linux*

Le propos sera simplifié et pourra choquer des spécialistes. Les processus et leur gestion est un sujet complexe.

Un processus est par exemple quand vous lancez une application. Elle possède son petit bout de mémoire propre, des informations comme le répertoire où vous l'avez lancé, le droit d'avoir un certain temps d'utilisation du CPU, ...

Sous *GNU/Linux*, un processus (une application au sens large) est toujours le fils d'un autre processus. Ainsi à chaque boot, le premier processus va se multiplier à chaque que nécessaire avec des infos et des droits adaptés.

Cette hiérarchie peut être vu avec la commande *pstree*.

10.2 Au quotidien

Les élements importants à savoir au quotidien est principalement que chaque processus se voit affecter des droits (par exemple root, ou pascal ou gaston) et qu'il a un numéro unique qui l'identifie.

Identifier les droits avec lequel est lancé le processus est important car cela conditionne ce qu'on pourra faire au sein du processus.

C'est la raison pour laquelle on utilise *sudo*. On laisse en utilisant cette commande le processus en tant que root.

11 Les signaux

11.1 Les systèmes *GNU/Linux*

Un processus peut recevoir des signaux venant du noyau. Les signaux sont, en grande partie, la forme de communication pour gérer les processus.

Nous verrons ici les ordres envoyés aux processus par le noyau : ceux qui nous intéressent sont ceux qui vont gérer l'arrêt et le redémarrage des processus.

Ainsi les signaux à mémoriser sont :

TERM (15) pour dire à l'application de se fermer

KILL (9) pour tuer l'application

HUP (1) pour redémarrer l'application (par exemple un serveur pour qu'il relise ses fichiers de configurations)

11.2 Au quotidien

Le numéro de processus va permettre par exemple s'il se bloque de l'arrêter ou de le tuer en utilisant une commande et le numéro du processus.

Oui sous *GNU/Linux* on tue des processus, c'est cruel.

12 En mode graphique

12.1 Terminal/Console

GNOME *Moniteur Système.*

KDE *KDE System Guard* dans Système.

13 En mode console

13.1 Au quotidien

C'est la version simple. Si votre processus a un nom unique, la commande permet de tuer les processus. Par exemple pour stopper le navigateur Firefox vous pouvez taper :

```
pkill firefox
```

S'il ne répond pas du tout vous pouvez le tuer :

```
pkill -9 firefox
```

```
pascal   31945  105  5.6 9244360 903364 ?      Sl   19:32   0:23 /usr/lib/firefox/firefox
pascal   32073  0.0  0.0  14376  1088 pts/0    S+   19:33   0:00 grep --color=auto firefox
```

on voit que Firefox est le processus 31945 et qu'il est en sommeil.

Pour le stopper, *kill 31945* ou le tuer *kill -9 31945*.

Vous pouvez visualiser les tâches en arborescence avec ps à l'aide de cette commande : *ps aux -ejHu*

Il y a aussi l'utilitaire *top* il permet d'afficher en temps (presque) réel les processus actifs.

Vous pouvez classer différemment en utilisant les touches :

F trie selon une colonnes différentes

u n'affiche que cet utilisateur

k stoppe un processus

q quitter *top*

14 La gestion de paquets

14.1 Introduction

14.2 En avance sur tout le monde...

Les paquets sous *GNU/Linux* sont incontournables. Ce sont des fichiers qui contiennent des logiciels avec les instructions d'installation.

Toutes les distributions ont des dépôts qui regroupent de quelques paquets à quelques milliers. Ils sont généralement restreints à des logiciels open source.

Ils sont finalement en avance sur les kiosque de logiciels qu'ont mis en place Apple et Windows ces dernières années.

14.3 La diversité des formats

Les paquets sous *GNU/Linux* sont aussi une source de conflit car selon la distribution originale le format des paquets est différents.

Par exemple des rpm sous Fedora, des deb sous Debian, ... Ils changent car les instructions d'installations sont différentes et que le binaire est différent (par exemple les versions des librairies sont différentes).

D'autres distributions comme *Gentoo* installent des paquets sources, i.e. qu'il faut compiler les logiciels à l'installation comme sous BSD.

14.4 Gestion des dépots

14.5 Notion de dépots

Le dépots est un serveur en ligne qui contient des paquets pour votre distribution.

Ces paquets sont signés c'est-à-dire qu'un code unique permet de vérifier à la fois l'authenticité et la qualité du téléchargement.

Quand on ajoute un dépot, il faut ajouter la clef du dépot pour télécharger et installer les paquets sinon le système va soit vous avertir soit bloquer l'installation des paquets provenant de ce dépot.

Pour ajouter des dépots, il faut l'ajouter dans le répertoire */etc/apt/sources.d/* pour Debian/Ubuntu soit dans */etc/yum.repos.d/* pour Fedora.

Syntaxe de Fedora pour ajouter un paquet :

```
[nom-du-depôt]
name=Le nom du dépôt $releasever - $basearch
baseurl=http://adresse-du-dépôt.com/fedora/$releasever/$basearch/
mirrorlist=http://adresse-du-miroire.com/fedora/$releasever/
enabled=1
gpgcheck=1
gpgkey=http://adresse-de-la-clés-gpg/RPM-GPG-KEY-nomdudépôt
```

Sur cet exemple d'un dépot Fedora qu'il y a trois composants :

– l'url du serveur
– le numéro de version
– la clef d'authentification des paquets

Dans le cas de Debian, la gestion de la clef est déporté dans l'utilitaire *apt-key*.

14.6 Gestion des paquets en ligne de commande

14.7 Ajout d'une clef sous Debian/Ubuntu

Sous Ubuntu :

```
wget http://www.dotdeb.org/dotdeb.gpg
sudo apt-key add dotdeb.gpg
```

La clef est téléchargé puis *apt-key* est utilisé pour ajouter la clef dans le trousseau des clefs des dépots.

14.8 Ajout d'un dépot sous Debian/Ubuntu

Il faut ajouter, de préférence dans */etc/apt/sources.d* une ligne :

```
deb   http://www.serveur.tld   <branche>   <sections>
```

La branche est le nom de la version et sections sont les répertoires du serveur dans lequel les paquets seront recherchés : en effet sous Debian/Ubuntu,

Les sections définissent des catégories de paquets par exemple selon leur licence : *main* pour les paquets libres et *restricted* pour les paquets non libres.

14.9 Ajout d'un paquet

Le mieux est de connaître le nom du paquet à installer. Dans ce cas la syntaxe est pour Debian :

```
apt-get install nom_du_paquet
```

Pour Fedora :

```
dnf install nom_du_paquet
```

14.10 Recherche d'un paquet

Pour trouver le nom d'un paquet, il y a des fonctions particulières pour les chercher dans les dépots :

Sous Debian :

```
apt-search kernel
```

Sous Fedora :

```
dnf list "kernel*"
```

Pour trouver le nom d'un paquet, il y a des fonctions particulières pour les chercher dans les dépots :

Sous Debian :

```
apt-cache search kernel
```

Sous Fedora :

```
dnf list "kernel*"
```

Il y a aussi le site alternativeto qui est très utile.

Si vous connaissez le nom d'un logiciel proche, il affiche la liste des programmes référencés qui font la même chose.

14.11 Que faire avec un .deb ou un .rpm ?

Dans le cas précédent, *apt-get* ou *dnf* vont télécharger le paquet puis l'installer.

Parfois, par exemple quand le paquet n'est pas dans un dépot, il faut installer un paquet sans le télécharger.

Dans ce cas l'utilitaire est *dpkg* pour Debian et *rpm* pour Fedora.

L'utilisation est très simple, puisque qu'il suffit de faire :

```
dpkg -i fichier_du_paquet
```

ou sous Fedora

```
rpm -i fichier_du_paquet
```

14.12 Quand ça se passe mal sous Debian/Ubuntu...

dpkg est utile à connaître quand par exemple l'installation est interrompu.

La commande *dpkg –configure -a* permet de finir dee configurer tous les paquets en attente.

On peut même forcer la configuration même en cas d'erreur ce qui n'est toutefois pas conseillé.

14.13 L'entretien d'un système...

La gestion des paquets inclut le fait qu'au fil de l'eau certains paquets ne sont plus nécessaires.

Il faut alors nettoyer un peu avec :

```
apt-get autoremove
dnf autoremove
```

Dans le cas précédent on enlève les paquets obsolètes. C'est le cas du kernel sur les Ubuntu/Debian qui sont une plaie.

Souvent on se retrouve avec 20 kernel installés si on ne fait pas *autoremove*. C'est parce que le paquet *linux-image* pointe vers une version précise du kernel qui change au fil du temps. Ainsi le paquet installé deveint obsolète car il n'est plus pointé par linux-image. Debian/Ubuntu le signale dans ce genre de cas.

Le cas précédent diffère de la recherche des paquets orphelins. Un paquet orphelins est un paquet qui n'est relié à aucun autre.

C'est le cas par exemple d'une librairie qui n'est pas supprimée quand on supprime un logiciel qui utilisait cette librairie et qu'il était le seul à l'utiliser.

Quand on change de versions, les orphelins sont nettoyés (la dernière étape de l'installation, *Nettoyage*) mais vous pouvez demander à le faire vous même.

Debian/Ubuntu

```
deborphan --guess-all
```

Fedora

```
package-cleanup --quiet --leaves
```

14.14 Gestion des paquets au format graphique

14.15 Utilitaires graphiques

Il y a un utilitaire très pratique, *synaptic* qui permet la gestion avancée des paquets.

L'utilitaire le plus ressemblant est *yumex-dnf* pour Fedora.

14.16 Les versions

14.17 Différence entre les distributions usuelles et *ArchLinux*

14.18 Retrouver sa version

15 Les dossiers personnels

15.1 Le home directory *GNU/Linux*

Les arborescences de fichiers *GNU/Linux* sont souvent source de confusion. Pourtant ils ne sont on ne peux plus simple...

Votre *home directory* ou répertoire personnel est dans le répertoire */home/utilisateur* avec utilisateur votre login.

Vous y passerez la quasi totalité de votre temps.

Ce répertoire s'appelle aussi ~pour le répertoire de l'utilisateur actif. On peux écrire pour les utilisateurs ~utilisateur.

Tous les utilisateurs sont dans la même situation sauf le super utilisateur (ou root).

15.2 Les périphériques amovibles

Quand vous montez une clef usb par contre (ou un CD), le système ne va pas la monter dans votre répertoire personnel.

Il va la monter dans le répertoire */media/utilisateur/nom de la clef*. Les distributions plus vieilles montent dans le répertoire */mnt/*.

Par monter, on entends rendre possible l'accès du périphérique à l'utilisateur.

16 Montage et démontage

16.1 Montage et démontage

On a parlé de montage et démontage :

1. quand on branche un périphérique, il est réconnu par le système d'exploitation mais pas accessible par l'utilisateur
2. il faut lui affecter un répertoire (et son type de format) pour qu'il soit accessible
3. cette opération se fait par la commande *mount*. Elle nécessite parfois d'être le super-utilisateur

Par exemple pour monter une clef « à la main » :

```
mount -t vfat /dev/sdg /home/pascal/clef
```

Les deux élements les plus importants sont :

1. /dev/sdg : c'est la référence vers le matériel ici une partition
2. /home/pascal/clef : c'est le répertoire dans lequel le contenu de la clef est visible. Attention si il y a des choses dans ce répertoire, ces élements sont "'masqués"' tant que la clef est monté.

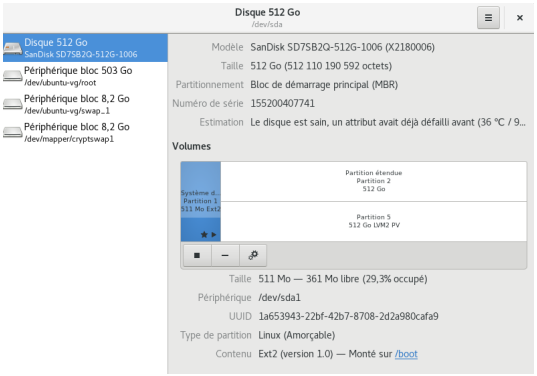

FIGURE 5 – Disques sous GNOME

16.2 Ajouter le montage d'un disque au démarrage

16.3 Avoir l'UUID

Il faut pour ça utiliser avec sudo l'utilitaire *blkid* :

```
/dev/sda1: UUID="8bf33340-e94c-..." TYPE="ext4"
/dev/sda2: UUID="ac56a704-260b-..." TYPE="swap"
/dev/sda3: LABEL="Home" UUID="8244710a-5cce-49ad-8b93-a92b5d2e53a0" TYPE="ext4"
/dev/sda4: UUID="DCF041AFF0419126" TYPE="ntfs"
```

Cette commande permet donc d'avoir l'UUID et d'identifier les disques et le type de format.

L'UUID est un identifiant unique qui, sauf formatage, restera identique « à vie ».

Si on utilise le nom du périphérique, par exemple /dev/sda1, pour identifier un disque plutôt que le numéro unique, en cas de changement de configuration, le nom de pérphérique risque de changer contrairement à l'UUID.

16.4 Gestion des disques en mode graphique

16.5 Sous GNOME 3

Il y a un utilitaire *Disques*.

16.6 Gestion des disques en mode graphique

16.7 Sous GNOME 3

Il y a un utilitaire *Disques*.

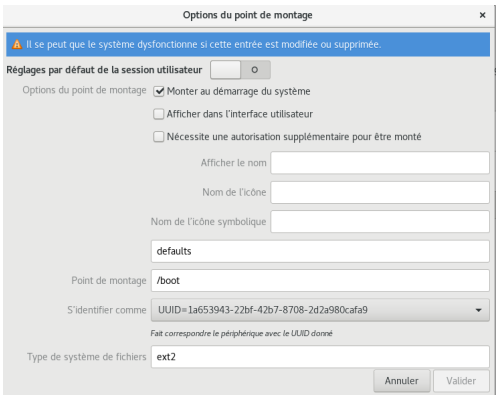

FIGURE 6 – Disques sous GNOME

16.8 Ajouter le montage d'un disque au démarrage

16.9 /etc/fstab

Comme on a vu précedemment, on utilise l'UUID dans ce fichier de configuration plutôt que le nom de périphérique.

```
UUID=c3cc32c0-b4bd-... /boot ext2 defaults 0 2
```

Attention, au démarrage, le système risque de se bloquer si le disque est marqué comme à monter par défaut et qu'il n'est pas présent. Avant d'enlever un disque penser à commenter la ligne (ou la supprimer) avant le redémarrage.

```
UUID=c3cc32c0-b4bd-... /boot ext2 defaults 0 2
```

Les arguments sont :

1. l'identifiant ou le nom du périphérique
2. le point de montage
3. le type de la partition
4. des mots clefs pour changer les propriétés du disque
5. le *dump*, utilisé pour les sauvegardes
6. le *pass*, pour les vérifications on démarrage

Parmi les mots-clefs à connaître :

user, no user définit si un utilisateur (et pas seulement le root) a le droit de monter la partition

auto,noauto définit si la partition est montée automatiquement au démarrage et en faisant *mount -a*

atime/noatime définit si le système marque la date de dernier accès. Il faut le mettre de préférence à *noatime* pour les SSD

rw/ro montage en lecture/écriture ou lecture seule (read only)

uid,gid,... permet de spécifier les droits des fichiers du disque

Pour le *dump*, il faut le laisser à 0.

Pour le *pass*, les valeurs sont à indiquer sont les suivantes :

1. pour la racine
2. pour les autres partitions Linux
3. pour le swap et les partitions windows

Plus d'infos là

16.10 /etc/mtab

17 Les autres répertoires

17.1 Les devices

17.2 Le répertoire /dev/

Il y a un dicton qui dit que tout dans *GNU/Linux* est fichier.

Ce qui traduit l'existence du répertoire */dev/*. A une entrée dans ce répertoire correspond un matériel ou une fonctionnalité du noyau.

Par exemple avec la commande *sudo blkid* vous pouvez voir les répertoires */dev/* correspondant aux disques.

Un autre exemple est */dev/urandom* qui contient des données aléatoires générées par le noyau.

```
dd bs=4M count=1 if=/dev/urandom of=random.txt
```

Les fichiers respectent des nomenclatures. Par exemple, *sdX* designe un disque SCSI (historiquement).

/dev/sdg désigne le disque tout entier, */dev/sdg1* la partition 1 du disque sdg.

Un utilitaire en ligne de commande pour explorer les disques est *fdisk* ou plus récent *parted*.

Graphiquement utiliser *gparted*.

17.3 Les autres répertoires

17.4 Les principaux répertoire...

- /bin, la plupart des programmes en ligne de commande
- /dev, pointe vers les périphériques
- /etc, Contient les fichiers de configuration du système
- /lib, les librairies
- /tmp, pour les répertoires temporaires
- /usr, là où s'installe la plupart des programmes
- /var, contient les informations partagée (par exemple site web, logs, ...)

La liste complète est là.

17.5 /bin

Dans ce répertoire on trouve les binaires c'est-à-dire les programmes exécutables.

Vous pouvez également trouver des exécutables dans les répertoires */usr/-share/bin* et */usr/local/bin*.

La disposition des exécutables dépend de la distribution et du type d'exécutables.

17.6 /etc

Si vous voulez modifier la configuration système, il est probable que vous ayez à intervenir dans ce répertoire.

Les fichiers sont presque tous des fichiers texte qu'il suffit d'éditer en tant que super utilisateur.

17.7 /lib

C'est le répertoire dans lequel on trouve les librairies c'est-à-dire des « morceaux » de programme qui sont mis dans le pot commun pour plusieurs programmes.

17.8 /tmp

Dans ce répertoire on trouve tous les fichiers qui sont destinés à avoir une durée de vie courte. D'ailleurs sur la plupart des distributions, le répertoire est vidé quand le système démarre.

17.9 /usr

Dans ce répertoire on trouve pas mal de choses. Il y a aussi des exécutables des librairies, l'emplacement de tout ça est régi par des règles dépendant de la distributions et de l'histoire de *GNU/Linux*.

Dans ce répertoire on trouve surtout les exécutables qui sont lancés en mode graphique.

17.10 /var

Ce répertoire est important notamment car vous trouverez par exemple deux répertoires très précieux :

- /var/www, le contenu de votre site web
- /var/logs, les fichiers journaux qui stockent les évènements qui se produisent sur le poste.

La page est assez exhaustive.

Elle contient pas d'informations intéressantes.

18 Les formats de partition

18.1 Les formats extX

Particularité des systèmes de fichier
sortie de fdisk -l

```
file -sL /dev/sd*
```

ext2 c'est le format standard UNIX qui n'est plus au goût du jour

ext3 il reprend le format ext2 avec une journalisation

ext4 il reprend ext3 avec en plus une augmentation de la taille des disques

A l'heure actuelle, il faut préférer le format le plus récent ext4.

18.2 Les formats Windows

Si vous voulez monter des disques Windows :

FAT,FAT32 c'est le vieux format qu'on retrouve sur les clefs USB notamment. Il est compatible *Linux*, Mac OS et Windows

NTFS c'est le « nouveau » format de Windows qui est propriétaire

Avant *Linux* ne pouvait écrire que sur des partitions FAT mais maintenant il peut aussi écrire sur des partitions NTFS dans la plupart des distros (mais pas toutes !).

Il y a d'autres types de partitions proposées : reiserfs, xfs, zfs, ...

Il ne se sont pas installés et/ou sont propriétaires donc ne sont pas lus par toutes les distributions.

18.3 gparted

19 Les permissions

19.1 permission

19.2 les sorties d'un ls

```
  -rw-r--r-- 1 root root 1426 nov.  26  2016 debug
-rw-r--r-- 1 root root 1735 nov.  26  2016 dhclient.conf
drwxr-xr-x 2 root root 4096 nov.   1 12:29 dhclient-enter-hooks.d
drwxr-xr-x 2 root root 4096 nov.   1 12:29 dhclient-exit-hooks.d
```

Les premiers caractères sont les droits pour le propriétaire et le groupe attribué au fichier ou au répertoire :

d directory ou répertoire.

r on peut lire le fichier

w on peut écrire le fichier

x le fichier est exécutable ou le répertoire peut être traversé

si au lieu d'un de ces caractères est remplacé par un - alors la propriété est inversé : si au lieu d'un w on a -, alors on ne peut pas écrire dans le fichier.

19.3 Les différents utilisateurs

On peut voir que le trio rwx est répété trois fois.

Car il y a trois types d'utilisateurs :

1. le propriétaire du fichier ou *u(ser)*
2. un membre du groupe auquel est attaché le fichier ou *g(roup)*
3. tous les autres utilisateurs ou empho(thers)

19.4 Dans l'explorateur de fichier

Avec un clic droit sur le fichier on peut changer avec nautilus ou dolphin, changer les droits : il suffit de choisir *Propriétés* puis *Permissions*.

En ligne de commande vous avez chmod. Il y a plusieurs utilisations de chmod.

19.5 Utilisation simple de chmod

par exemple pour changer les propriétés d'un fichier, il faut faire :

```
chmod ug+rwx fichier
```

cette commande permet de rendre lisible et en écriture *fichier*.

L'inconvénient est que la syntaxe ne permet de ne changer que par différence les permissions.

19.6 Masque

Au lieu de modifier les permissions de façon relative, elle définisse les droits mais sont moins intuitifs. A chaque catégorie rwx, est assigné un numéro et la somme définie les droits du fichiers.

1 exécution

2 écriture autorisée

4 lecture autorisée

En combinant les sommes de chacun de ces chiffres on obtient tous les combinaisons de droits.

Ex : *chmod 754 fichier*

7= $4+3+1$, tous les droits sont données

5= $4+1$, lecture et exécution

4= 4, lecture autorisée

et ainsi de suite.

19.7 Récursif

Pour changer les droits de tout un répertoire et ses sous répertoires, il suffit de faire *chmod 700 -R repertoire*

Attention, changer les droits des fichiers doit être fait de façon très prudente car il est diffcile après une erreur de retrouver les droits de tous les fichiers d'une arborescence.

Et parfois le système peut être bloqué pour des problèmes de droit.

Par exemple, vous bloquerez la mise à jour des fichiers de log si vous en changez les droits.

19.8 Utilisateur et groupe

il y a deux commandes :

chown pascal fichier change le propriétaire de fichier pour pascal

chown pascal :pascal fichier change le propriétaire de fichier pour pascal et le groupe pour pascal

chgrp pascal fichier change le propriétaire de fichier pour pascal et le groupe pour pascal

20 Utilitaires

20.1 Utilitaires

Il y a quelque utilitaires qui n'ont pas d'équivalent en mode graphique.

Le premier est *df* qui permet d'avoir la place libre sur les disques.

Il faut le lancer avec l'argument *df -h* pour avoir l'espace en unités intelligible.

Il y a aussi la commande *du -ch* qui permet d'avoir l'espace disque pris par des répertoires.

Sous GNOME, un utilitaire peu efficace qui tente de les remplacer : *Analyseur d'utilisation de disques*

21 Réseau

21.1 Déboguer un réseau

21.2 Je n'ai pas de connexion du tout

Si vous faites un *ifconfig* et que l'interface réseau ethernet (enX ou ethX) n'apparait pas...

Votre controleur réseau n'est pas reconnue par *GNU/Linux*. Dans ce cas il faut regarder sur Internet comment l'activer pour *GNU/Linux*. Ca peut être compliqué.

Généralement il faut télécharger le firmware ou charger le bon module du noyau.

Si vous faites un *ifconfig* une adresse internet n'est pas assignée...

Dans ce cas, il est probable que ce soit un cable ou que la box ne joue pas son rôle de DHCP. Le réseau est indisponible.

Essayer de jouer avec les cables, rebooter votre box, ...

Si vous faites un *ifconfig* une adresse internet n'est assignée...

La box joue son rôle de DHCP. Le réseau est disponible. Essayer de faire *ping 4.4.4.4*. *4.4.4.4* est un serveur de Google qui répond au ping sauf cataclysme.

Si vous obtenez une réponse alors c'est surement le serveur DNS qui est mal configuré si vous avez un serveur DNS privé ou vous avez un peu trop joué avec les réglages de resolv.conf ou un problème de pare-feu mal configuré sur la box ou votre routeur.

Si vous n'avez pas de réponse au *ping* alors vous n'êtes pas connecté à internet. Dans ce cas c'est souvent un problème de pare-feu mal configuré sur la box ou votre routeur.

Autre possibilité si tout parait au vert mais que vous n'avez pas accès au web, vérifier les réglages Proxy de votre navigateur.

22 Sécurisation

22.1 Super utilisateur et compte ROOT

22.2 Compte ROOT

Être *root* permet de tout faire sur un poste *GNU/Linux*. C'est le super utilisateur qui permet notamment d'éditer les fichiers systèmes, arrêter/démarrer des services, ...

Mais il ne faut pas utiliser le compte *root*. En effet on peux supprimer, altérer des fichiers systèmes ou d'autres utilisateurs.

Et si vous le pouvez, surfer sur le web ou lancer des programmes vous expose considérablement car un malware peux de profondes modifications sur votre ordinateur.

C'est pour cette raison que le compte root doit absolument être désactivé sur un poste convenablement sécurisé. C'est le cas par défaut sur les distributions modernes.

Cette désactivation est visible dans le fichier */etc/passwd* car comme interpréteur de fichier il doit y avoir */bin/false*.

false est un programme qui échoue lorsqu'on le lance.

22.3 sudo

Comme il n'y a pas de compte *root* et qu'il faut quand même pouvoir modifier des fichiers systèmes, lancer/démarrer des services, ... il existe la possibilité d'utiliser l'utilitaire *sudo*.

Les administrateurs (et non le commun des mortels) doit avoir un accès à *sudo*. Cela est possible car les administrateurs appartiennent au groupe *sudo* (ou *wheel*).

Lorsque vous tapez une commande précédée par *sudo* alors cette commande est exécutée en tant que *root* après vous être authentifié avec votre mot de passe. Le mot de passe n'est pas redemandé pendant quelques minutes après une première utilisation de sudo.

Attention, vous pouvez tout casser avec une commande sudo comme si vous êtiez *root*.

La différence avec un compte *root* est que le fait d'être *root* est transitoire. Ce qui permet de se protéger de beaucoup de malwares.

Le nombre de personnes pouvant être *sudoers* doit être limité au maximum. D'autant plus que la mise à jour des paquets non critiques est possible sur les dernières distributions par des administrateurs et non par le super utilisateur.

Enfin il est possible de limiter les commandes que peuvent lancer les *sudoers*. Ceci est possible en éditant le fichier de configuration de *sudo* avec la commande *visudo*.

Par exemple on peut ajouter paul qui a besoin de lancer/relancer apache car il est webmaster. Il suffit alors de rajouter :

```
paul ALL = /usr/bin/apachectl
```

Et il ne faut pas l'ajouter aux utilisateurs sudo.
Plus d'information sur ces modifications sont disponibles là.

23 Rappels réseaux

23.1 Protocoles réseaux

On s'intéressera principalement à trois protocoles : TCP, UDP et ICMP.

Le protocole TCP est le plus utilisé. Il est utilisé quand une connexion stable et qui nécessite l'envoi et la réception de paquets dans un ordre défini.

Le protcole UDP est moins utilisé. Il sert souvent à envoyer des messages où l'aspect transactionnelle est moins importante comme la vidéo sur le LAN. Il ne garantit ni que les données arrivent dans l'autre ni que les deux ordinateurs ont établi une connexion.

Le protocole ICMP est un protocole dont on peux se passer mais qui permet d'envoyer des messages sur le réseau : saturation du réseau, écho, ... Il n'est pas associé à un port particulier contrairement à TCP et UDP.

Quand vous voulez ouvrir un port pour SSH, votre site web ou une autre application il s'agit souvent de TCP. Quelques applications comme la communication sur internet (genre Skype) peuvent demander d'ouvrir un port UDP.

Le protocole ICMP doit être accessible par exemple sur des opérateurs de serveurs tels qu'OVH qui utilisent ce protocole pour vérifier que la machine est bien « Up and running ».

Par défaut quand une application n'écoute pas sur le port, le port est fermé. Si une application l'utilise et peut établir une connexion dans le sens internet vers votre ordinateur, le port est ouvert. Si c'est dans l'autre sens le port est fermé : c'est votre ordinateur qui initiera la connexion dans le sens votre ordinateur vers internet.

Un pare-feu permet de protéger votre ordinateur en masquant des ports ouverts qui ne devraient pas être visibles depuis l'extérieur (imprimante 631, ssh 22, ...). Les ports sont alors masqués : votre ordinateur ne renvoie pas un message port fermé ce qui ne trahi pas sa présence et empêche la connexion depuis l'extérieur.

24 Pare-feu

24.1 arno-iptables-firewall

C'est un petit pare-feu disponible sous Debian/Ubuntu.

A partir de quelques questions, il met en place un pare-feu tout à fait performant et sécuritaire.

Après installation, les réglages supplémentaires sont situés dans le fichier */etc/default/arno-iptables-firewall* et */etc/arno-iptables-firewall/firewall.conf*.

Vous trouverez un tutoriel là.

24.2 FirewallID

là là là là

25 Réparation

25.1 Utilitaire Live CDs

Pour la plupart des distributions, il y a des live CDs qui vont vous permettre en mode graphique ou en ligne de commande sauvez votre installation.

Il y a même des des distributions dédiées comme ici.

Attention la liste contient aussi des distributions de *forensics* pour étudier un disque après une intrusion (Kali, Backbox).

25.2 Montage

Le type de problème fréquent est d'altérer /etc/fstab et ça empêche le système de booter.

Il suffit d'appliquer ce que vous avez appris.

Il faut identifier votre disque avec la commande *sudo blkid* et/ou *fdisk -l*

```
sudo mkdir /media/crash
sudo mount /dev/sdX /media/crash
sudo nano /media/crash/etc/fstab
```

25.3 Corruption d'un disque

Sauf très gros problème, vous pouvez lors du démarage accéder à la console pour réparer une parititon défecteuse.

Mais si vous voulez corriger une partition défectueuse vous pouvez le faire avec la commande :

```
sudo e2fsck /dev/sdX
```

Attention !!! la partition ne doit pas être montée pour le faire

e2fsck détecte le type de partition et tente de réparer le disque.

Si *e2fsck* ne détecte pas le type de partition là, soit vous vous êtes trompé en spécifiant la partition soit elle est très abimée et sera éventuellement récupérable en forçant

e2fsck.ext4, *e2fsck.ext3*, ...

La récupération des disques Windows (fat, ntfs) est un problème sous Linux car elle ne marche pas très bien et vous demande souvent de faire la réparation sous... Windows.

25.4 Grub

Si l'ordinateur ne démarre pas du tout hors matériel c'est potentiellement grub qui pose problème. Dans ce il faut réparer grub.

La solution c'est le chroot. Cet utilitaire va faire croire au système que la racine est différente et vous pourrez faire fonctionner comme si vous aviez booter sur le sytème altéré.

C'est-à-dire que si vous voulez accéder à /mnt/t1/boot/grub.cfg, vous le ferez en accédant directement par /boot/grub.cfg une fois chrooté.

Il faut monter la partition par exemple dans le répertoire /mnt/t1 puis :

```
mount /dev/sda1 /mnt/t1
mount --bind /run  /mnt/t1/run
mount --bind /dev /mnt/t1/dev
mount --bind /dev/pts /mnt/t1/dev/pts
mount -t proc /proc /mnt/t1/proc
mount -t sysfs /sys /mnt/t1/system/sys
chroot /mnt/t1 /bin/bash
```

une fois chrooté il suffira d'un :

```
sudo update-grub
```