**Linux Foundation Certified SysAdmin (LFCS)**

Length: 24:06:49

# Essential commands(25% of the Exam)

## Log into Local and Remote Graphical and Text Mode Consoles – Debian Version

Ctrl + Alt + t afin d'ouvrir le terminal.

Cliquer sur icône show applications and tape terminal

For a remote access :

ssh user@ip

Exemple :  ssh [pi@192.168.0.12](mailto:pi@192.168.0.12)

On peut aussi prendre la main avec VNC viewer, il faut créer un compte sur Real VNC puis cela te permettra de prendre la main sur un ordinateur en indiquant son nom ou IP.

Search for files

Linux case is sensitive

find / -name "test.txt"

Everything in Linux is a file.

Pour ignorer la case:

find / -iname "test.txt"

Pour chercher tout ce qui est n'est pas "test.txt" par exemple :

find / -not -name "test.txt"

Pour chercher tous les dossiers :

find / -type d

Pour chercher les dossiers intitulés "log"

find / -type d -name "log"

Pour chercher tous les files qui se terminent par log :

find / -type f -name "\*.log"

Chercher un fichier de plus de 27 k :

find /usr/bin -size +27000c

find /sur/bin -size +27k

Chercher les fichiers créés il y a plus d'un jour:

find / -type f -mtime 1

Chercher les fichiers créés il y a moins d'un jour:

find / -type f -mtime -1

Chercher tous les fichiers appartenant à Qeyss:

find / -user qeyss

Chercher tous les fichiers du dossier etc appartenant à l'utilisateur root:

find /etc -user root

Chercher tous les fichiers par des permissions:

find /usr/bin -perm 755

Chercher tous les fichiers et modifier la permission:

find / -name "test.txt" -exec chmod 700 {} \ ; ( ici on donne full permission au propriétaire et rien aux autres.)

ls -la permet d’avoir la liste des fichiers, leurs permissions, tailles, etc.

Which command

Afin de savoir où se trouve l'exécutable d'une commande, ex :

which ifconfig

which python

 locate command

La commande locate permet de trouver très rapidement un fichier. Contrairement à ce que l’on pourrait penser **locate** ne vas pas chercher le fichier au sein de l’arborescence, mais au sein d’une base de données contenant la liste des fichiers existants. De ce fait, il se peut que lorsque vous venez de créer un fichier ou un répertoire que ce dernier ne soit pas dans cette base de données, il faut pour cela réactualiser la base de données à l’aide de la commande **updatedb**.

## Evaluate and compare the Basic file system features and options.

A file system is where computer system persists general data for users and/or applications

Les **systèmes de fichiers** (il en existe de multiples) gèrent l'organisation des informations mémorisées sur les périphériques de stockage de l'ordinateur

Dans le cœur d'un ordinateur, tout est constitué de 1 et de 0, mais l'organisation de ces données n'est pas aussi simple.

Un **bit** est un 1 ou un 0 ; un **octet** (byte en anglais) est composé de huit bits ; un kilo-octet binaire (**kibioctet**, ou kibibyte) est un groupe de 1024 octets ; un méga-octet binaire(**mébioctet**, ou mebibyte) se constitue de 1024 kilo-octets binaires ; et ainsi de suite.

Un disque dur (hard drive ou hard disk) stocke vos données : à chaque fois que vous enregistrez un fichier, vous écrivez des milliers de 1 et de 0 sur un disque métallique, le piquant littéralement et créant une indentation qui pourra plus tard être relue en tant que 1 ou 0 par votre ordinateur. En réalité, vous magnétisez des milliers de petits bâtons (que nous appellerons clusters) qui seront polarisés dans un sens ou dans un autre (N-S ou S-N) ; cette polarisation différente sera ré-interprétée plus tard par l'ordinateur en tant que 1 ou 0.

Il y a tellement de données sur un disque dur qu'il doit obligatoirement y avoir un moyen de les organiser.

Les systèmes de fichiers organisent les fichiers de votre ordinateur sur votre disque dur de façon à pouvoir les retrouver lorsque vous en aurez besoin. Les systèmes de fichiers les plus répandus à l'heure actuelle sont sûrement le FAT32 et le NTFS, qui sont les deux seuls systèmes de fichiers que Microsoft® Windows® peut nativement lire. Mais, tout comme il existe d'autres systèmes pour classer des livres dans une bibliothèque, il existe de nombreux autres systèmes de fichiers : ext2, ext3, ext4, ReiserFS, JFS, XFS, Btrfs, ZFS,….

Les qualifications d’un système de fichiers :

La taille maximale d’un fichier

La taille maximale d’une partition

La gestions des droits d’accès aux fichiers et répertoires

La journalisation :

Parlons maintenant de la journalisation. Un système de fichiers journalisé est **plus fiable** lorsqu'on entre dans le domaine du stockage des données. Il a été expliqué plus haut ce qui se produit réellement lorsqu'un fichier est enregistré sur un disque dur (une suite de 1 et de 0 est inscrite sur le disque) ; mais que se produit-il si l'écriture de la chaîne est interrompue avant son terme (ce qui se produit, par exemple, lors d'une coupure de courant) ? Votre fichier devient « corrompu », incomplet.

Un système de fichiers journalisé travaille de façon à prévenir une telle corruption : lors de la sauvegarde d'un fichier, au lieu d'écrire immédiatement sur le disque dur les données à l'endroit exact où elles devraient être enregistrées, le système de fichiers écrit les données dans une autre partie du disque dur et note les changements nécessaires dans un journal, et ensuite, en arrière-plan, il repasse chacune des entrées du journal et termine le travail commencé ; lorsque la tâche est accomplie, il raye la tâche de la liste.

EXT ( Extended File System)

 BtrFS, XFS, HPFS, SWAP, FAT(16, 32, exFAT)

## Compare and manipulate File Content and Use Input Output redirection

**Commande Cat**

Permet de voir le contenu d’un fichier sans l’ouvrir

cat ShoppingList.txt

**Commande More**

Permet de voir le contenu d’un fichier page par page sans l’ouvrir.

**Commande Less**

Permet de voir le contenu d’un fichier page par page ( s’il est long) sans l’ouvrir tout en pouvant revenir en arrière

less ShoppingList.txt

**Commande Sort**

Permet de trier l’ensemble des lignes du fichier par ordre croissant.

sort ShoppingList.txt

On peut combiner deux commandes :

sort ShoppingList.txt | more

**Commande Man**

Te donne des infos sur une commande, ex : man sort

Redirection

>

Permet de rediriger le résultat d’une commande dans un autre fichier ou une autre commande.

Exemple : ls -la > test.txt

>> : redirige à la fin d'un fichier et le crée s'il n'existe pas

**Commande cut**

Permet de couper un fichier selon le nombre de caractères ou un délimiteur

Exemple de fichier intitulé notes.txt :

Fabrice,18 / 20,Excellent travail

Mathieu,3 / 20,Nul comme d'hab

Sophie,14 / 20,En nette progression

Mélanie,9 / 20,Allez presque la moyenne !

Corentin,11 / 20,Pas mal mais peut mieux faire

Albert,20 / 20,Toujours parfait

Benoît,5 / 20,En grave chute

Coupure selon le nombre de caractères :

cut -c -3 notes.txt

Va afficher :

Fab

Mat

Sop

Mél

Cor

Alb

Ben

Coupure selon un délimiteur :

cut -d , -f 1 notes.txt

Là on demande à couper en prenant pour délimiteur la virgule et à n’afficher que les premiers termes avant la virgule soit :

Fabrice

Mathieu

Sophie

Mélanie

Corentin

Albert

Benoît

Si je veux afficher que les commentaires donc situés après la troisième virgule on tape :

cut -d , -f 3 notes.txt

Excellent travail

Nul comme d'hab

En nette progression

Allez presque la moyenne !

Pas mal mais peut mieux faire

Toujours parfait

En grave chute

Si je veux afficher les deux premiers éléments je tape :

cut -d , -f 1,2 notes.txt

Fabrice,18 / 20

Mathieu,3 / 20

Sophie,14 / 20

Mélanie,9 / 20

Corentin,11 / 20

Albert,20 / 20

Benoît,5 / 20

## Analyse text using basic regular expressions

Nous allons avoir besoin des deux symboles suivants, retenez-les :

* ^(accent circonflexe) : indique le début d'une chaîne ;
* $(dollar) : indique la fin d'une chaîne.

Ainsi, si vous voulez qu'une chaîne commence par « Bonjour », il faudra utiliser la regex ( expression régulière) :

'^Bonjour '

Si vous placez le symbole « ^ » devant le mot, alors ce mot devra obligatoirement se trouver au début de la chaîne, sinon on vous répondra FAUX.

De même, si on veut vérifier que la chaîne se termine par « zéro », on écrira cette regex :

'zéro$ '

Si je cherche toutes les lignes commençant par The dans le texte Alice.txt, je taperais :

grep '^The' Alice.txt

Les classes de caractères

Si je veux chercher les lignes commençant par un a, b ou c il faut mettre cela entre parenthèse [abc] ou [a-c], exemple :

grep '^[a-c]' Alice.txt

grep '^T[a-z][aeiouy]' Alice.txt

Là je cherche les lignes commençant par un T, suivi d’une lettre, suivie d’une voyelle.

#### Et pour dire que je n'en veux pas ?

Si vous ne voulez PAS des caractères que vous énumérez dans votre classe, vous devrez placer le symbole « ^ » au début de la classe.

Ainsi, la regex suivante :

'[^0-9]'

… signifie que vous voulez que votre chaîne comporte au moins un caractère qui ne soit pas un chiffre.

grep '^T[a-z][^e]' Alice.txt

Cela signifie toutes les lignes commençant par un T, suivi d’une lettre et dont le troisième caractère n’est pas un e.

Les quantificateurs

* + ?(point d'interrogation) : ce symbole indique que la lettre est facultative. **Elle peut y être 0 ou 1 fois**.

Ainsi,#a?#reconnaît 0 ou 1 « a » ;

* + +(signe plus) : la lettre est obligatoire. **Elle peut apparaître 1 ou plusieurs fois**.

Ainsi,#a+#reconnaît « a », « aa », « aaa », « aaaa », etc. ;

* + \*(étoile) : la lettre est facultative. **Elle peut apparaître 0, 1 ou plusieurs fois**.

Ainsi,#a\*#reconnaît « a », « aa », « aaa », « aaaa », etc. Mais s'il n'y a pas de « a », ça fonctionne aussi !

Les quantificateurs sont des symboles qui permettent de dire combien de fois peuvent se répéter un caractère ou une suite de caractères.

La regex#^Yaho+$#signifie que la chaîne doit commencer et finir par le mot « Yahoo ». Il peut y avoir un ou plusieurs « o ». Ainsi « Yaho », « Yahoo », « Yahooo », etc. marchent…

Les accolades

Parfois on aimerait indiquer que la lettre peut être répétée quatre fois, ou de quatre à six fois… bref, on aimerait être plus précis sur le nombre de répétitions.  
C'est là qu'entrent en jeu les accolades.

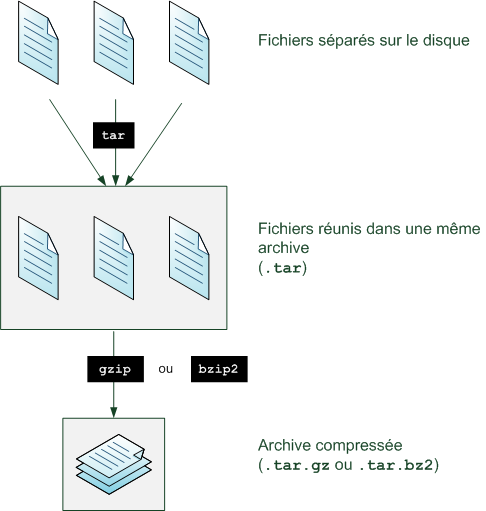
Il y a trois façons d'utiliser les accolades.

* {3} : si on met juste un nombre, cela veut dire que la lettre (ou le groupe de lettres s'il est entre parenthèses) doit être répétée **3 fois exactement**.  
  #a{3}#fonctionne donc pour la chaîne « aaa ».
* {3,5} : ici, on a plusieurs possibilités. On peut avoir la lettre de **3 à 5 fois**.  
  #a{3,5}#fonctionne pour « aaa », « aaaa », « aaaaa ».
* {3,} : si vous mettez une virgule, mais pas de 2e nombre, ça veut dire qu'il peut y en avoir jusqu'à l'infini. Ici, cela signifie « **3 fois ou plus** ».  
  #a{3,}#fonctionne pour « aaa », « aaaa », « aaaaa », « aaaaaa », etc. Je ne vais pas tous les écrire, ça serait un peu long.

|  |  |  |
| --- | --- | --- |
| eeeee | #e{2,}# | VRAI |
| Blablablabla | #^Bla(bla){4}$# | FAUX |
| 546781 | #^[0-9]{6}$# | VRAI |

## Archive,backup, Compress,Unpack, and Decompress Files

Tar permet d'assembler des fichiers avant de les compresser.



1° On rassemble tous les fichiers à compresser dans un même dossier.

2° On crée une archive avec tar, exemple sur le dossier intitulé tuto

tar -cvf tuto.tar tuto/

* -c : signifie créer une archive tar ;
* -v : signifie afficher le détail des opérations ;
* -f : signifie assembler l'archive dans un fichier.

3° On compresse avec gzip :

gzip tuto.tar

Tu auras donc un dossier tuto.tar.gz

Il y a aussi la commande bzip2 pour compresser :

bzip2 tuto.tar

4° Pour décompresser on utilise la commande gunzip :

gunzip tuto.tar.gz

5° Pour extraire les fichiers d’une archive on utilise la commande tar avec l’option -xvf

tar -xvf tuto.tar

Il est possible d’archiver et compresser en même temps avec la commande tar et l’option -zcvf :

tar -zcvf tuto.tar.gz tuto/

Pour décompresser et extraire directement on utilise la commande tar avec l’option -zxvf :

tar -zxvf tuto.tar.gz

On peut afficher le contenu d’une archive sans l’extraire :

On doit taper : tar -tf nomdelarchive.tar

Pour voir le contenu d’un fichier compressé on utilise la commande zcat

Ex : zcat tuto.tar.gz

## Create,Delete,Copy and move files and directories

Create with touch or vim and the name of the file

Copy with cp :

cp test test2 🡪 Va copier test dans un fichier test2

cp test text/ 🡪 va copier le fichier test dans le dossier text

Remove with mv

mv test2 text/ 🡪 Va déplacer le fichier test2 dans le dossier text

mv test2 text/test3 🡪 Va déplacer le fichier test2 dans le dossier text et le renommer en test3

Delete a file with rm

rm test2

Delete a directory with rm -r

rm -r text/

rm -rf text/ 🡪 va supprimer le dossier et tout ce qu’il contient.

## Create and Manage Hard and Soft Links

La commande ln permet de créer des liens entre fichiers.

Ces liens sont des raccourcis

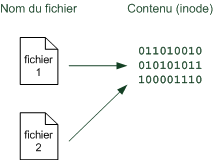
On peut créer des liens physiques ou symboliques

**Les liens physiques**

Un lien physique permet d'avoir deux noms de fichiers qui partagent exactement le même contenu, c'est-à-dire le même inode (figure suivante).

Tu commences par créer un fichier1

Pour le créer tu fais : ln fichier1 fichier2, ça va automatiquement créer le fichier2



Ainsi, que vous passiez par fichier1 ou par fichier2, vous modifiez exactement le même contenu. En quelque sorte, **le fichier est le même**. On peut juste y accéder via deux noms de fichiers différents.

Si tu modifies n’importe lequel de ces deux fichiers, ça modifiera aussi l’autre.

En faisant un ls -i tu verras que les fichiers partagent le même inode.

**Les liens symboliques**

Le principe du lien symbolique est que l'on crée un lien vers un autre nom de fichier. Cette fois, on pointe vers le nom de fichier et non vers l'inode directement (figure suivante).

Pour en créer un tu tapes : ln -s fichier1 fichier2

En faisant un ls -l tu verras que le fichier 2 au niveau des droits commence par l qui signifie link, ex :

-rw-r--r-- 1 mateo21 mateo21 0 2008-07-31 13:55 fichier1

lrwxrwxrwx 1 mateo21 mateo21 8 2008-07-31 14:15 fichier2 -> fichier1

Sinon ils sont similaires en tout aux liens physiques sauf que les liens symboliques fonctionnent aussi sur des répertoires, contrairement aux liens physiques.

## Lecture: List,set and Change Standard File Permissions

In Linux, everything is a file, so we can restrict based on some basic file permissions

On utilise la commande ls -la afin de voir les permissions des files and directories.

Sous Linux on ne se connecte en root que très rarement, **lorsque c'est nécessaire**. Certaines commandes de Linux que nous allons voir dans ce chapitre ne sont accessibles qu'à root.  
Le reste du temps, on utilise le compte « limité » que l'on a créé.

Cette simple protection permet de largement limiter les dégâts en cas de fausse manipulation, de virus sur votre PC, etc. En effet, un virus ne peut rien faire de plus que vous quand vous êtes connectés avec des droits limités. En revanche, si vous êtes en root il pourra tout faire, même détruire votre ordinateur.

Sous Windows, vous êtes toujours connectés en administrateur par défaut (équivalent de root), ce qui explique pourquoi les virus y sont si dangereux.

Avec Ubuntu on ne peut pas se connecter en root au démarrage.

On peut s’y connecter temporairement à l’aide de la commande **sudo** en tapent sudo puis la commande souhaitée ou bien définitivement en tapant sudo su

Pour quitter le « mode root », tapez exit (ou faites la combinaison Ctrl + D).

Sous les autres distributions qu'Ubuntu, écrire « su » suffit à passer root.

**Gestion des users**

**adduser** pour ajouter un user

**deluser** pour supprimer un user

**passwd** pour changer le password, ex : passwd qeyss

Gestion des groupes

addgroup pour ajouter un groupe

delgroup pour supprimer un groupe

usermod pour modifier un user

**Les droits d’accès**

Chaque fichier et chaque dossier possède une liste de droits. C'est une liste qui indique qui a le droit de voir le fichier, de le modifier et de l'exécuter.

Pour afficher les droits on tape **ls –l** et pour les modifier on utilise la commande **chmod**

mateo21@mateo21-desktop:~$ ls -l

total 40

drwxr-xr-x 2 mateo21 mateo21 4096 2007-11-13 21:53 Desktop

drwxr-xr-x 2 mateo21 mateo21 4096 2007-11-13 13:46 Documents

lrwxrwxrwx 1 mateo21 mateo21   26 2007-09-19 18:31 Examples -> /usr/share/example-content

drwxr-xr-x 2 mateo21 mateo21 4096 2007-09-25 20:28 images

Vous voyez tous ces d, r, w et x au début ? Ce sont ce qu'on appelle les droits d'accès du fichier ou dossier.

On peut voir cinq lettres différentes. Voici leur signification :

* **d** (Directory) : indique si l'élément est un dossier ;
* **l** (Link) : indique si l'élément est un lien (raccourci) ;
* **r** (Read) : indique si on peut lire l'élément ;
* **w** (Write) : indique si on peut modifier l'élément ;
* **x** (eXecute) : si c'est un fichier, « x » indique qu'on peut l'exécuter. Ce n'est utile que pour les fichiers exécutables (programmes et scripts).

Les droits sont découpés en fonction des utilisateurs (figure suivante).



* Le premier triplet rwx indique les droits que possède le **propriétaire** du fichier sur ce dernier ;
* Le second triplet rwx indique les droits que possèdent les autres membres du **groupe** sur ce fichier ;
* Enfin, le dernier triplet rwx indique les droits que possèdent tous les **autres** utilisateurs de la machine sur le fichier.

| **Droit** | **Chiffre** |
| --- | --- |
| r | 4 |
| w | 2 |
| x | 1 |

Voici la liste des droits possibles et la valeur correspondante :

| **Droits** | **Chiffre** | **Calcul** |
| --- | --- | --- |
| --- | 0 | 0 + 0 + 0 |
| r-- | 4 | 4 + 0 + 0 |
| -w- | 2 | 0 + 2 + 0 |
| --x | 1 | 0 + 0 + 1 |
| rw- | 6 | 4 + 2 + 0 |
| -wx | 3 | 0 + 2 + 1 |
| r-x | 5 | 4 + 0 + 1 |
| rwx | 7 | 4 + 2 + 1 |

Par exemple, « 640 » indique les droits du propriétaire, du groupe et des autres (dans l'ordre).

* 6 : droit de lecture et d'écriture pour le propriétaire.
* 4 : droit de lecture pour le groupe.
* 0 : aucun droit pour les autres.

Pour changer les droits sur le fichier rapport.txt, et être le seul autorisé à le lire et l'éditer, je dois exécuter cette commande :

chmod 600 rapport.txt

##### Attribuer des droits avec des lettres (chmod relatif)

Il existe un autre moyen de modifier les droits d'un fichier. Il revient un peu au même mais permet parfois de paramétrer plus finement, droit par droit.  
Dans ce mode, il faut savoir que :

* **u** = user (propriétaire) ;
* **g** = group (groupe) ;
* **o** = other (autres).
* **a = all**

… et que :

* **+** signifie : « Ajouter le droit » ;
* **-** signifie : « Supprimer le droit » ;
* **=** signifie : « Affecter le droit ».

Maintenant que vous savez cela, vous pouvez écrire :

chmod g+w rapport.txt

Signification : « Ajouter le droit d'écriture au groupe ».

chmod g+w,o-w rapport.txt

Signification : « Ajouter le droit d'écriture au groupe et l'enlever aux autres ».

chmod go-r rapport.txt

Signification : « Enlever le droit de lecture au groupe et aux autres ».

chmod +x rapport.txt

Signification : « Ajouter le droit d'exécution à tout le monde ».

chmod a+rwx rapport.txt

Signification : « Ajouter tous les droits à tout le monde ».

#### Récursivement

Pour chacune de ces commandes, on peut les lancer récursivement sur un répertoire. C'est à dire que l'action sera effectuée sur le répertoire désigné et sur tous les fichiers ou répertoires qu'il contient. Ceci se fait en ajoutant l'option **-R** .

Gestion des propriétés d’un fichier

**chown** : Pour changer le propriétaire d’un fichier

La commande chown, qui doit être utilisée **en tant que root**, attend deux paramètres au moins :

* le nom du nouveau propriétaire ;
* le nom du fichier à modifier.

Cela donne donc :

chown patrick rapport.txt

**chgrp** : changer le groupe propriétaire d'un fichier

chgrp s'utilise exactement de la même manière que chown à la différence près qu'il affecte cette fois le groupe propriétaire d'un fichier.

chgrp amis rapport.txt

Cette commande affectera le fichier rapport.txt au groupe amis.

## Read, and Use System Documentation

Linux is an amazing operating system and the best part? The documentation is built right in!

We use the command man, like: man ls or info ls

/usr/share/doc contient aussi de la documentation.

## Manage Access to the Root Account

Root access on Linux machines means all the power to do whatever you need and whatever you want, and it's really important to limit who has access.

Pour supprimer tous les dossiers et fichiers d’un dossier je tape : rm -rf \*

Si la commande sudo ne fonctionne pas avec un user car il n’a pas les droits pour l’utiliser, il faut taper sudo visudo et le mettre dans la section où il y a root avec les mêmes droits que root.

La commande sudo !! permet de retaper la précédente commande mais avec sudo.

# Operation of Running Systems (20% of the Exam)

## Lecture: Boot, reboot, and shut down a system safely

We type **shutdown**

sudo shutdown –r : Pour faire un reboot

## Boot of changes system into different operating modes

Les runlevel sont les différents niveaux de démarrage/arrêt sous Unix/Linux.  
De petites différences existent suivant les différentes distributions. Je m'intéresse ici aux runlevel sous Debian GNU/Linux.

Il existe donc 6 runlevel sous Debian GNU/Linux:  
Le runlevel 0 correspond à l'arrêt du système  
Le runlevel 1 correspond au démarrage single-user  
Le runlevel 2 correspond mode de démarrage normal  
Les runlevels 3-5 correspondent à des modes de démarrage que l'on peut se configurer  
Le runlevel 6 correspond au redémarrage.

Les runlevel sont des niveaux constitués d'un ensemble de liens situés dans les répertoires /etc/rcX.d/ qui pointent vers les scripts situés dans /etc/init.d/

Au démarrage de Linux, le noyau est chargé puis il passe la main au processus INIT.  
Ce processus démarre tout ce qui se trouve dans /etc/rcS.d/ puis il lance par défaut le runlevel 2 (on peut changer cela dans le fichier /etc/inittab). Par exemple, si l'on veut démarrer en mode single-user (runlevel 1) (souvent en cas de problème du système), on tape **Linux 1** au prompt de LILO.

En tapant runlevel tu auras le level sur lequel tu es actuellement

## Install, configure and troubleshoot bootloaders

There's a lot that needs to happen from the moment when the power is turned on until the system decides which operating system to boot. That's where bootloaders come in! In this lesson, we'll take a close look at the bootloader that most Linux distributions use today -- Grub 2. We'll look at the files that make it up and discuss how you can install, configure, and troubleshoot it. At the end of this lesson, you'll understand how to configure grub2 and boot one or more operating systems on a machine!

***GR****and* ***U****nified* ***B****ootloader, version 2* (ou **GRUB 2**) est la seconde version de GNU GRUB. C'est un logiciel permettant de [charger un système d'exploitation](https://doc.ubuntu-fr.org/amorcage). Il pourra amorcer lui-même des systèmes compatibles avec la norme POSIX (GNU/Linux, \*BSD, Mac OS, etc.) ou enchaîner vers un autre chargeur d’amorçage pour les autres systèmes.

Il est dans /etc/grub.d

Pour changer l’ordre de changement du boot faut aller dans /etc/default/grub

GRUB 2 est le chargeur d'amorçage installé par défaut, ce document concerne la version de GRUB 2 pour les ordinateurs Compatibles PC (la plupart des ordinateurs actuels)

Pour connaitre sa version d’installée on tape : grub-install –V

Utilisation du menu

**Si vous n'avez qu'un seul système d'installé, Grub ne s'affiche pas !** Pour le voir, il suffit de maintenir la **touche Majuscule (shift)** ( ou la touche[**ECHAP**](https://forum.ubuntu-fr.org/viewtopic.php?pid=21778237#p21778237)) enfoncée **au démarrage** de l'ordinateur.

**Fichier grub.cfg**

* /boot/grub/**grub.cfg** : est un fichier de configuration généré automatiquement par *update-grub* (il est inutile d’éditer ce fichier) et **ne doit** donc **pas être modifié manuellement**.

**Fichier /etc/default/grub**

Dans ce fichier, on peut **activer ou désactiver un paramètre** en le commentant / décommentant avec le caractère **dièse « # »**([croisillon](https://fr.wikipedia.org/wiki/wiki" \o "https://fr.wikipedia.org/wiki/wiki)).

**paramètres présents par défaut**

* **GRUB\_DEFAULT=0** correspond au menu qui sera sélectionné par défaut. (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#menu_par_defaut))
* **#GRUB\_HIDDEN\_TIMEOUT=0** : avec le « # », le menu de grub sera visible. Sans le « # », le menu de grub sera invisible. (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#affichermasquer_le_menu))
* **GRUB\_HIDDEN\_TIMEOUT\_QUIET=false** De pair avec ‘GRUB\_HIDDEN\_TIMEOUT’. (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#affichermasquer_le_menu))
* **GRUB\_TIMEOUT\_STYLE=hidden** Paramètre apparu avec la version 18.04.1 en remplacement des deux qui précèdent.
* **GRUB\_TIMEOUT=10** est la durée en secondes de l’affichage du menu avant de se lancer sur le menu sélectionné par défaut. (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#menu_par_defaut))
* **GRUB\_DISTRIBUTOR=``** est la ligne qui définit la syntaxe des titres du menu (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#intitule_dans_le_menu))
* **GRUB\_CMDLINE\_LINUX\_DEFAULT=""** et **GRUB\_CMDLINE\_LINUX=""** on y met les paramètres à ajouter lors de la détection automatique des systèmes (ex: radeon.modeset=1, logo.nologo,..).
* **#GRUB\_GFXMODE=640x480** (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#grub_gfxmode)).
* **#GRUB\_INIT\_TUNE="480 440 1"** : sans le « # », on aura un bip à l’affichage du menu Grub.

**Paramètres qu'on peut ajouter**

* **GRUB\_GFXPAYLOAD=1024x768** : définit la résolution d’affichage entre Grub et celle définie dans la distribution (pour le splash screen par exemple). Défini à 'keep' par défaut.
* **GRUB\_GFXPAYLOAD\_LINUX=auto** : définit la résolution d’affichage par défaut du linux lancé FIXME
* **GRUB\_BACKGROUND="/boot/grub/images/fjord.jpg"** : pour mettre une image en fond d'écran (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#fond_d_ecran))
* **GRUB\_DISABLE\_OS\_PROBER="true"** : si on souhaite désactiver la recherche d'autres systèmes à chaque update-grub (inutile si on les lance autrement)
* **GRUB\_OS\_PROBER\_SKIP\_LIST="…"** : liste de partitions à ne pas inclure dans la recherche automatique de systèmes. (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#os-prober_partiel))
* **GRUB\_DISABLE\_LINUX\_RECOVERY="true"** : on mettra ce paramètre si on ne veut plus avoir la possibilité de lancer les systèmes en mode 'maintenance'
* **GRUB\_DISABLE\_SUBMENU=y** : si on souhaite voir directement une entrée de menu par noyau linux disponible (On peut supprimer les [noyaux](https://doc.ubuntu-fr.org/kernel) les plus anciens → [Nettoyer Ubuntu](https://doc.ubuntu-fr.org/kernel#suppression_des_anciens_noyaux))
* **LANG=fr\_FR** : pour définir la langue utilisée pour l'édition du menu depuis Grub ou le mode 'ligne de commande'
* **GRUB\_TERMINAL\_INPUT=at\_keyboard** : nécessaire pour paramétrer le clavier français (pour ne plus avoir à chercher les ()[]= .. du clavier Qwerty). (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#clavier_francais))
* **GRUB\_THEME="/boot/grub/themes/ubuntu-mate/theme.txt"** : chemin vers le fichier de configuration du thème (mode graphique). (→ [**Détails**](https://doc.ubuntu-fr.org/tutoriel/grub2_parametrage_manuel#theme))

Top of Form

Bottom of Form

**Dossier /etc/grub.d/**

Ce dossier contient tous les scripts qui seront utilisés (en respectant l'ordre de numérotation) par update-grub pour créer le fichier grub.cfg.

* **00\_header** : script gérant les paramètres définis dans /etc/default/grub ;
* **05\_debian\_theme** : script pour gérer le thème en mode texte (fonds d'écran et couleurs) ;
* **10\_linux** : contient le script de lancement du système sur lequel on est ;
* **20\_memtest86+** : script permettant de générer les entrées memtest ; semble absent de certaines installations EFI
* **20\_linux\_xen** : script pour Xen Linux et Xen Hypervisor ;
* **30\_os-prober** : contient le script de recherche des autres systèmes installés ;
* **30\_uefi-firmware** FIXME
* **40\_custom** : configuration personnelle (systèmes à lancer en configuration manuelle, paramètres qui n'existent pas pour /etc/default/grub,..)
* **41\_custom** : identique à 40\_custom (si on le renomme par exemple 07\_custom, on pourra y mettre le lancement du mediacenter ou de Windows en premier sur la liste)

**Nous n'utiliserons** pour nos paramétrages **que les fichiers 'custom'**.

Paramétrage manuel

Avant de commencer, il est souhaitable :

* de lancer dans un [terminal](https://doc.ubuntu-fr.org/terminal) une [commande](https://doc.ubuntu-fr.org/commande_shell) de mise à jour : *sudo update-grub* ;
* de créer une [sauvegarde](https://doc.ubuntu-fr.org/tutoriel/comment_sauver_et_restaurer_un_fichier) (ex. : *sudo cp /boot/grub/grub.cfg /boot/grub/grub.cfg.autogénéré*) ;

De même, aucun script de configuration présent dans **/etc/grub.d/** ne doit être effacé.

**Menu par défaut**

On utilisera ici les paramètres **GRUB\_DEFAULT** et **GRUB\_TIMEOUT** du fichier **/etc/default/grub**

Pour changer le système d'exploitation sur lequel grub démarre par défaut on paramètrera **GRUB\_DEFAULT=** au choix avec :

* un **chiffre** (0 = le premier, 1 = le deuxième, etc.) ;
* **saved** (sans guillemet) pour ce qui a été sélectionné lors du précédent lancement (nécessite de rajouter GRUB\_SAVEDEFAULT=true) ;
* ou le **titre** de menu exact d’un menu mis **entre guillemets** (par exemple : "Microsoft Windows 10 (on /dev/sda1)"). Pour obtenir les titres exacts des menus : grep menuentry /boot/grub/grub.cfg

La **temporisation** avant un lancement automatique se définit avec le paramètre **GRUB\_TIMEOUT** :

* **GRUB\_TIMEOUT=10**, lancera le système par défaut au bout de **10 secondes**. Si on ne veut **pas de temporisation**, on mettra : **GRUB\_TIMEOUT=-1**

## Diagnose and manage processes

Knowing what is running on your Linux system, as well as what's very busy and what is not so much, is a critical aspect of system administration.

**w** – Show who is logged on and what they are doing

**top** – Display linux processes

Pour faire simple, dites-vous qu'un **processus** est un programme qui tourne en mémoire. La plupart des programmes ne font tourner qu'un processus en mémoire (une seule version d'eux-mêmes). C'est le cas d'OpenOffice par exemple. D'autres lancent des copies d'eux-mêmes, c'est le cas du navigateur Google Chrome qui crée autant de processus en mémoire que d'onglets ouverts

Pour lister les processus qui tournent sous Windows, on utilise Ctrl + Alt + Suppr et on va dans l'onglet « Processus ».  
Sous Linux, on peut utiliser deux commandes différentes : ps et top

**ps** vous permet d'obtenir la liste des processus qui tournent au moment où vous lancez la commande. Cette liste n'est pas actualisée en temps réel, contrairement à ce que fait top

On distingue quatre colonnes.

* PID : c'est le numéro d'identification du processus. Chaque processus a un numéro unique qui permet de l'identifier. Ce numéro nous sera utile plus tard lorsque nous voudrons arrêter le processus.
* TTY : c'est le nom de la console depuis laquelle a été lancé le processus.
* TIME : la durée d'exécution du processus. Plus exactement, cela correspond à la durée pendant laquelle le processus a occupé le processeur depuis son lancement.
* CMD : le programme qui a généré ce processus. Si vous voyez plusieurs fois le même programme, c'est que celui-ci s'est dupliqué en plusieurs processus (c'est le cas de MySQL, par exemple)

Quand on utilise ps sans argument comme on vient de le faire, il affiche seulement les processus lancés par le même utilisateur.

**ps -ef : lister tous les processus**

Avec ps -ef, vous pouvez avoir la liste de tous les processus lancés par tous les utilisateurs sur toutes les consoles

**ps -u UTILISATEUR : lister les processus lancés par un utilisateur**

Pour filtrer un peu cette longue liste, on peut utiliser -u afin d'obtenir par exemple uniquement les processus que l'on a lancés nous-mêmes.

**top : liste dynamique des processus**

La liste donnée par ps a un défaut : elle est **statique** (elle ne bouge pas). Or, votre ordinateur, lui, est en perpétuel mouvement. De nombreux processus apparaissent et disparaissent régulièrement.

**Ctrl +C & kill** : arrêter un processus

**Ctrl + C : arrêter un processus lancé en console**

Taper **Ctrl + C** ne coupe pas le programme brutalement, cela lui demande gentiment de s'arrêter, comme si vous aviez cliqué sur la croix pour fermer une fenêtre.

**kill :** tuer un processus

Ctrl + C ne fonctionne que sur un programme actuellement ouvert dans la console. De nombreux programmes tournent pourtant en arrière-plan, et Ctrl + C n'aura aucun effet sur eux.

C'est là que vous devez utiliser kill si vous voulez les arrêter

Pour vous en servir, il faudra auparavant récupérer le PID du ou des processus que vous voulez tuer. Pour cela, deux solutions :

* ps ;
* top.

Exemple si je veux tuer le process firefox lance par l’utilisateur qeyss, je recherché son PID :

root@ubuntu:~# ps -u qeyss | grep firefox

14692 tty2 00:00:21 firefox

root@ubuntu:~# kill 14692

Vous pouvez aussi tuer plusieurs processus d'un seul coup en indiquant plusieurs PID à la suite :

kill 32678 2768 33071

Vous voulez tuer un processus sans lui laisser le choix ?  
C'est tout à fait possible, mais à n'utiliser que dans le cas d'un programme complètement planté que vous voulez vraiment arrêter !

Avec kill -9 (comme le chiffre 9, oui, oui), vous demandez à Linux de tuer le processus sans lui laisser le temps de s'arrêter proprement. Cela peut faire le ménage quand rien ne va plus.

kill -9 32678

**killall :** tuer plusieurs processus

Souvenez-vous : je vous ai dit que certains programmes se dupliquaient en plusieurs processus. Si vous voulez arrêter l'ensemble de ces processus, comment faire ? Heureusement, vous avez des armes pour éradiquer cette vermine.

Vous pourriez, certes, tuer tous les processus en récupérant un à un leur PID. Mais il y a plus rapide : killall (« tuez-les tous ! »).

Contrairement à kill, killall attend le nom du processus à tuer et non son PID.

Supposons que nous ayons trois processus find en cours d'exécution que nous souhaitions arrêter.

$ ps -u mateo21 | grep find

  675 pts/1    00:00:01 find

  678 pts/2    00:00:00 find

  679 pts/3    00:00:01 find

Pour tous les tuer, il faudra donc taper :

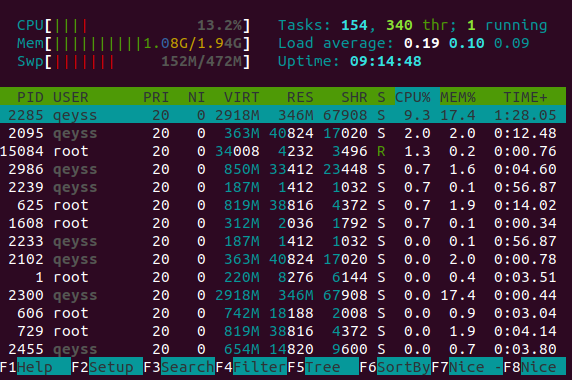
$ killall find

**htop** – Display linux processes

htop permet de visualiser, je dirais même de gérer les processus de manière interactive.

Grâce à lui vous pourrez rechercher, tuer les processus et mêmes les trier selon un critère (utilisation de mémoire, de CPU,...), et d'[autres](https://www.it-connect.fr/cours-tutoriels/administration-systemes/serveur-web/autres-serveur-web/) fonctions disponible toute via l'interface de htop.

Il faut l’installer avec : apt-get install htop



Dans la partie haute de l'interface on peut voir l'utilisation du **CPU** en pourcentage, l'utilisation de la mémoire en Mo par rapport à la totalité de mémoire disponible ainsi que la mémoire swap utilisée. Le temps depuis lequel le [système](https://www.it-connect.fr/cours-tutoriels/administration-systemes/windows-server/systeme/) est démarré est indiqué également. Ensuite, un "**tableau**" listant les processus est présenté où vous pouvez retrouver une multitude d'information comme l'utilisateur qu'il l'a exécuté, le pourcentage de mémoire et de **CPU** qu'il utilise ainsi que la commande qui sert à l'exécuter. Pour finir, dans la partie basse un menu explique les différentes actions qui sont disponibles.

**Trier les processus selon un critère**

Pour cela appuyez sur "**F6**" ce qui correspond à "**SortBy**" soit "**Trier par**". Un listing de critères apparaît sur la gauche, choisissez celui que vous souhaitez



**Rechercher un processus**

Pour recherche un processus appuyez sur **F3** puis saisissez le nom ou une partie du nom de celui-ci :

**Tuer un processus**

Positionnez-vous sur le processus que vous désirez "killer" puis appuyez sur F9, ensuite choisissez une méthode de **KILL** dans la liste puis faites Entrée.

**Voir processus d'un utilisateur en particulier**

Plutôt que de voir l 'ensemble des processus, si vous préférez voir les processus exécutés par un utilisateur en particulier, utilisez la commande :

htop -u nom\_utilisateur

## Locate and analyze system log files

When something goes wrong, you need to know where to look to fix it! In this lesson, we'll explore the main system log files on Centos/Redhat systems and Debian/Ubuntu systems and take a look at a few tips and tricks to making reviewing the log files a bit easier. At the end of this lesson, you'll know where the main system logs are and have a good idea for where to find application-specific logs.

cd /var/log

Any time you have any kind of problems in your Linux system, the log files are generally a good place to start, pour les trouver :

Sur Debian/Ubuntu tu tapes : sudo less syslog

Sur Rhel/Centos tu tapes : sudo less messages

Si je cherche une erreur spécifique je peux taper ceci : grep "Error" syslog

root@ubuntu:/var/log# grep "Error" syslog

Dans Debian/Ubuntu il y a plusieurs fichiers de logs.

## Schedule tasks to run at a set date and time

V

## Verify completion of scheduled jobs

v

To do :

Se connecter à distance en ssh à ton serveur Linux raspberry

dd