

LA LIGNE DE COMMANDE (CLI) SOUS GNU/Linux

Table des matières

| | |
|---|----|
| Gameshell..... | 1 |
| Références à lire..... | 2 |
| Aide des commandes..... | 2 |
| La touche TAB..... | 2 |
| Gestion de l'historique..... | 3 |
| Éditeurs de fichiers en console..... | 3 |
| Rappel des dossiers à la racine /..... | 4 |
| Consoles..... | 4 |
| Types de terminaux..... | 4 |
| Les variables d'environnement..... | 5 |
| Les alias de commande..... | 5 |
| Connexion en root..... | 7 |
| Gestion des logiciels..... | 8 |
| Pour un paquet .deb..... | 8 |
| Avec les sources de logiciels apt..... | 9 |
| Gestion des fichiers et dossiers :..... | 10 |
| Processus de démarrage simplifié..... | 13 |
| Chargeur de démarrage..... | 13 |
| Noyau vmlinux..... | 14 |
| Initrd / Initramfs..... | 15 |
| Réinstallation de GRUB via chroot..... | 15 |
| Systemd / Services au démarrage..... | 16 |
| Display Manager / GUI..... | 17 |
| Processus..... | 18 |
| Journaux système..... | 19 |

| | |
|--|----|
| Matériel..... | 19 |
| Tâches répétitives..... | 20 |
| Aspirateur de site..... | 20 |
| Gestion des périphériques disques..... | 20 |
| Gestion des périphériques batterie..... | 21 |
| Gestion des usagers et groupes :..... | 21 |
| Compression de fichiers..... | 22 |
| Connexion et transfert sécurisés en SSH/SCP/RSYNC..... | 23 |
| Vérification d'empreintes SSH..... | 24 |
| Opérations réseau de base..... | 24 |
| Utilisation du network-manager..... | 26 |
| Serveur ou client windows..... | 27 |
| Extras..... | 28 |
| Exercice 1 : création d'un conteneur Xen..... | 29 |
| Exercice 2 : installer un serveur apache2..... | 31 |
| Exercice 3 : installer WordPress depuis ses sources..... | 32 |
| Autres ressources..... | 32 |

Gameshell

Gameshell est ce qu'on nomme un « serious game », comprenez un jeu sérieux à vocation professionnelle, pour apprendre aux débutants complets à maîtriser les commandes de base sous GNU/Linux.

C'est un petit jeu de rôle, intéressant à découvrir avec des dizaines de défis à relever. Prévoyez quand même quelques heures de temps libre pour cette découverte...

À noter que vous pouvez interrompre le jeu à tout moment, et le reprendre au point où vous étiez.

Comme l'explique <https://github.com/phyver/GameShell>, la procédure d'installation se fait en trois lignes de commandes :

```
# sur une ligne :
sudo apt install gettext man-db procps psmisc
nano tree bsdmainutils x11-apps wget

# sur une ligne :
wget
https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh

bash gameshell.sh
```

Références à lire

- http://doc.ubuntu-fr.org/tutoriel/console_ligne_de_commande
- http://doc.ubuntu-fr.org/tutoriel/console_commandes_de_base
- https://debian-facile.org/doc:systeme:commandes:gnu_linux

Aide des commandes

Astuce : les pages de manuel sont disponibles en français dans le paquet **manpages-fr**.

```
man [nom_commande]
```

La première chose à faire est donc :

```
man man
```

On trouve ainsi plusieurs sections de pages de manuel :

```
1 Programmes exécutables ou commandes de
l'interpréteur de commandes (shell)
2 Appels système (fonctions fournies par le
noyau)
3 Appels de bibliothèque (fonctions fournies
par les bibliothèques des programmes)
4 Fichiers spéciaux (situés généralement
dans /dev)
5 File formats and conventions, e.g.
/etc/passwd
6 Jeux
7 Divers (y compris les macropaquets et les
conventions), par exemple man(7), groff(7)
8 Commandes de gestion du système
(généralement réservées au superutilisateur)
9 Sous-programmes du noyau [hors standard]
```

Et suivant la commande recherchée, on peut tomber dans plusieurs sections :

```
man socket
```

La touche TAB

GNU/Linux fournit une autocomplétion des commandes via la touche **TAB**. Vous pouvez donc taper les premières lettres d'une commande, et appuyer **TAB** pour compléter cette dernière. Si rien ne s'affiche, c'est que plusieurs choix sont possibles : il faut alors

réappuyer sur **TAB** pour obtenir la liste des possibilités, et souvent rajouter un ou deux caractères en plus avant de retenter le coup.

Gestion de l'historique

Un autre mécanisme très sympathique à utiliser pour se faciliter la vie est l'historique, qui enregistre chaque commande tapée dans le terminal.

Les commandes précédentes peuvent ainsi être rappelées soit en utilisant un numéro de ligne, soit les premières lettres de la commande, soit une recherche texte en mode arrière.

- Afficher l'historique du terminal :

```
history
```

- Rejouer la commande numéro XX :

```
!XX
```

- Rejouer la dernière commande qui commençait par `ls` :

```
!ls
```

- Rejouer la dernière commande :

```
!!
```

- Rechercher une commande manuellement :

```
Utilisez les flèches de direction haut/bas
```

- Rechercher une commande par mot-clé, en remontant l'arborescence :

```
CTRL+R + [motif] (+CTRL-R pour remonter petit à petit dans l'historique)
```

- Reprise du dernier argument de la ligne de commande précédente (avec recherche arrière si plusieurs appuis) :

```
ALT+(MAJ)+.
```

- Refaire la commande précédente en administrateur :

```
sudo $(history -p \!\!)
```

- Astuces :

La variable d'environnement **HISTFILESIZE** définit la taille maximale (en nombre de lignes) de ce fichier d'historique, généralement stocké dans votre `~/.bash_history`.

La variable d'environnement **HISTSIZE** contrôle le nombre de lignes affichées par la commande `history` seule.

Éditeurs de fichiers en console

Les débutants choisissent généralement **nano**, un éditeur très simple mais très limité.

Les professionnels préfèrent **vim** ou **emacs**, qui demandent un apprentissage de départ plus important, mais qui offrent bien plus de possibilités dans le traitement des fichiers texte, et notamment une redoutable efficacité en mode rechercher/remplacer, via les expressions régulières (ou **regexp**).

Rappel des dossiers à la racine /

| | |
|---------|---|
| / | (root ou slash) la racine du système de fichier |
| —/bin | programmes de base pour tous les utilisateurs |
| —/boot | fichiers de démarrage |
| —/dev | fichiers d'accès aux périphériques |
| —/etc | fichiers de configuration |
| —/home | dossiers utilisateurs (équivalent du C:\Users) |
| —/lib | librairies partagées |
| —/media | utilisé par le système pour le montage/démontage de périphériques amovibles |
| —/mnt | dossier autrefois utilisé pour le montage/démontage de périphériques amovibles, mais que l'on peut désormais utiliser pour un montage manuel temporaire |
| —/opt | programmes additionnels |
| —/proc | dossier virtuel contenant la liste des processus actifs |
| —/root | dossier par défaut du super-utilisateur |
| —/sbin | programmes plus dédiés aux administrateurs |
| —/tmp | dossier temporaire pour tous les usagers |
| —/usr | dossier principal où s'installent les programmes choisis par l'utilisateur |
| —/var | données variables : journaux systèmes, dossiers emails POP3, bases de données, stockage des impressions, etc... |

Consoles

- Consoles texte :

CTRL+ALT+F1 à F6

- N.B. : quand l'interface graphique est bloquée, les consoles texte sont très intéressantes pour revenir en mode CLI, et analyser/débloquer la situation (tuer un processus bloquant par exemple).
- Console graphique (Ubuntu) :

CTRL+ALT+F2

- Remonter d'une page :

MAJ+Page haut

- Descendre d'une page :

MAJ+Page bas

- Effacer la console

clear

ou son raccourci clavier :

CTRL+L

Types de terminaux

- **TTY** (Teletypewriter): terminal natif, connecté directement à un système comportant clavier/souris, ou à un matériel via une connexion série (par exemple les consoles de base du système).
- **PTS** (Pseudo Terminal Slave): dispositif de terminal émulé par un

autre programme (`xterm`, `screen`, `ssh`, etc).

- En résumé : les **ttyX** sont des terminaux physiques et les **ptsX** sont des terminaux virtuels.

Les variables d'environnement

- Afficher les variables d'environnement :

```
env | less
```

- Exporter une variable d'environnement pour les processus fils :

```
export [NOM VARIABLE]=[VALEUR]
```

Exemple :

```
echo $PATH
export PATH=$PATH:/home/stagiaire/monprojet
echo $PATH
```

- Principales variables d'environnement :
 - **PATH** : décrit les chemins vers les programmes
 - **DISPLAY** `host:display.screen` : décrit l'hôte réseau qui reçoit l'affichage graphique (**localhost** par défaut), l'affichage (sous la forme d'un numéro – **0** par défaut), et l'écran (**0** sous la forme d'un numéro – 0 par défaut).
 - **SHELL** : le shell actuellement utilisé (bash, sh, ksh, ...)
 - **USER** : utilisateur actif
 - **LANG** : langage et collation + encodage
 - **HOME** : dossier home de l'utilisateur actif

- **XDG_SESSION_TYPE** : session graphique (x11 ou wayland)

Les alias de commande

La plupart des distributions proposent des alias de commande par défaut, c'est-à-dire des raccourcis permettant d'émuler des commandes plus complexes, comme `ll` à la place de `ls -l` par exemple.

La liste des **alias** s'obtient avec la commande :

```
alias
```

Chaque usager peut créer ses propres alias dans `~/.bashrc` ou `~/.bash_aliases`, mais on peut aussi créer des alias généraux pour tous les usagers du système. Dans ce cas, le mieux est de les placer dans `/etc/bash_aliases`, et de rajouter à la fin de `/etc/bash.bashrc` une ligne en :

```
. /etc/bash_aliases
```

Même si c'est possible techniquement, il est fortement déconseillé de placer ses alias généraux à la fin de `/etc/bash.bashrc`, ce fichier pouvant être mis à jour et donc réécrit à tout moment.

ATTENTION : les deux exemples suivants contiennent certaines commandes qui peuvent définitivement « casser » votre système si vous les utilisez au mauvais endroit ! À bon entendeur...

Exemples d'alias utilisateurs (`~user/.bash_aliases`) :

```
alias ..='cd ..'
alias .2='cd ../..'
```

```
alias .3='cd ../../..'
alias .4='cd ../../../../..'
alias d755='find ./ -type d -print0|xargs -0iX
chmod 755 X'
alias d770='find ./ -type d -print0|xargs -0iX
chmod 770 X'
alias d775='find ./ -type d -print0|xargs -0iX
chmod 775 X'
alias f600='find ./ -type f -print0|xargs -0iX
chmod 600 X'
alias f644='find ./ -type f -print0|xargs -0iX
chmod 644 X'
alias f660='find ./ -type f -print0|xargs -0iX
chmod 660 X'
alias f664='find ./ -type f -print0|xargs -0iX
chmod 664 X'
alias fgr='fgrep -r'
alias g='grep'
alias h='history'
alias ka='killall'
alias pk='pkill'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias md='mkdir'
alias psc='ps xawf -eo pid,user,cgroup,args'
alias psg='ps aux|grep -v grep|grep'
alias pst='pstree -paul'
```

```
alias ssh='ssh -XY'
# s'te plaît !
alias stp='sudo $(history -p \!\!\!)'
alias man='man -a'
alias tail='tail -n 1000'
alias dns='systemd-resolve --status'
alias ip='ip -c'
```

Exemples d'alias administrateurs (*/etc/bash_aliases*) :

```
PS1='${debian_chroot:+($debian_chroot)}\[\
033[01;31m\]\u@\h\[\033[00m\] \[\033[01;34m\]\w\
[\033[00m\] # '
```

```
alias ai='apt install'
alias as='apt search'
alias ar='apt remove'
alias au='apt update && apt dist-upgrade'
alias ac='apt clean && apt autoremove'
```

```
alias dlG='dpkg -l|grep'
alias dL='dpkg -L'
```

```
# auth logs
alias ta='tail -f /var/log/auth.log'
alias va='vi /var/log/auth.log'
```

```
# html logs
alias th='tail -f /var/log/apache2/*.log'
```

```
alias vh='vi /var/log/apache2/error.log'

# syslog logs
alias ts='tail -f /var/log/syslog'
alias vs='vi /var/log/syslog'

# messages logs (for old distros)
alias tm='tail -f /var/log/messages'
alias vm='vi /var/log/messages'

# unattended upgrades logs
alias tu='tail -f /var/log/apt/history.log'

# Systemd aliases and functions
.
/usr/share/bash-completion/completions/systemctl

# sc like systemctl
alias sc='systemctl'
complete -F _systemctl sc

alias lus='systemctl list-units -t service'
alias luf='systemctl list-unit-files'

alias reboot='systemctl reboot'
alias poweroff='systemctl poweroff'

alias start='systemctl start'
```

```
alias restart='systemctl restart'
alias stop='systemctl stop'
alias status='systemctl status'
alias reload='systemctl reload'
# we stop here : remember that
# 'enable' is a bash command !

alias dr='systemctl daemon-reload'

# logs
alias jc='journalctl -n 300 -f'
# kernel only
alias jk='journalctl -n 300 -k'
```

Appel d'une commande originale au lieu de son alias – il suffit de rajouter le caractère \ devant le nom :

```
\ls
```

Connexion en root

Les machines de type UNIX ou GNU/Linux intègrent par défaut un super-administrateur nommé **root**, qui a tous les droits sur le poste.

Par défaut - et pour des raisons évidentes de sécurité - la plupart des distributions ne permettent pas de se connecter directement sous ce compte privilégié, et proposent donc un autre mécanisme basé sur la commande **sudo**, permettant à un utilisateur

(généralement le premier définit lors de l'installation) d'obtenir temporairement les droits administrateurs.

Bien entendu, on peut passer outre cette mesure de sécurité, en attribuant à **root** un mot de passe (si possible le plus long et complexe possible...), ce qui permet alors d'utiliser la commande **su** au lieu de **sudo**.

Mais gardez bien à l'esprit qu'un pirate SAIT que le compte **root** existe sur une machine GNU/Linux, et qu'à la moindre porte ouverte trouvée, il tentera sûrement une attaque de type « force brute » pour trouver le mot de passe du super-administrateur...

- Exécuter temporairement une commande en **root** :

```
sudo [commande]
```

N.B. : il faut que l'utilisateur appartienne au groupe **admin** ou **sudo** (réglage définit dans **/etc/sudoers**). Notez l'existence du raccourci **visudo** pour les accros de VIM.

- Si **root** n'a pas de mot de passe attribué :

```
sudo bash
```

ou

```
sudo -s
```

- Si **root** a un mot de passe attribué :

```
su - (ou su -l)
```

N.B. : suivant les distributions, les variables d'environnement retournées par un **su** et un **su -**, et notamment la variable **PATH**,

ne sont pas toujours les mêmes... En outre, l'option **-l** vous ramène dans le dossier de l'utilisateur.

Gestion des logiciels

Pour un paquet .deb

- Lister les logiciels installés :

```
dpkg -l | less
```

Remarque : Les lignes qui débutent avec **ii** sont les logiciels installés et configurés, ceux avec **rc** sont des restes de configuration (généralement laissés dans **/etc**). La purge d'un paquet consiste à effacer le programme ET ses fichiers/dossiers de configuration.

- Rechercher si un paquet est installé :

```
dpkg -l | grep [motif recherché]
```

- Si le paquet est installé, en détailler son contenu :

```
dpkg -L [nom du paquet]
```

- Installer un paquet :

```
dpkg -i [nom du paquet]
```

- Désinstaller un paquet en laissant ses fichiers de configuration :

```
dpkg -r [nom du paquet]
```

- Purger un paquet (avec ses fichiers de configuration) :


```
dpkg -P [nom du paquet]
```

- Rechercher le paquet de provenance d'un programme :

```
dpkg -S [chemin complet du programme – cf. which]
```

- Reconfigurer un paquet :

```
dpkg-reconfigure [nom du paquet]
```

Avec les sources de logiciels apt

N.B. : toutes les distributions récentes basées sur **Debian** fonctionnent désormais avec **apt** au lieu de **apt-get**.

- Lister les dépôts officiels :

```
cat /etc/apt/sources.list
```

- Lister les dépôts tiers :

```
ls -l /etc/apt/sources.list.d/*.list
```

- Mettre à jour la base des paquets disponibles :

```
apt update
```

- Mettre à jour la distribution avec les dépendances :

```
apt dist-upgrade
```

- Installer un paquet :

```
apt install [nom_paquet]
```

- Installer un paquet dont il manque encore des dépendances (exemple : **anydesk**) :

```
apt -f install [nom_paquet]
```

- Retirer un paquet (en laissant ses fichiers de configuration) :

```
apt remove [nom_paquet]
```

- Purger un paquet (avec ses fichiers de configuration) :

```
apt purge [nom_paquet]
```

- Rechercher un paquet par mot-clé :

```
apt search [motif]
```

- Effacer le cache local des paquets (dans **/var/lib/apt**) :

```
apt clean
```

- Rechercher et effacer les paquets devenus inutiles :

```
apt autoremove
```

- Rechercher le chemin d'une commande **SHELL** connue :

```
which [commande shell]
```

- Les deux commandes suivantes utilisent le paquet **locate** :

- Rechercher un fichier quelconque sur le système en utilisant une partie de son nom :

```
locate [motif]
```

- Mettre à jour la base des fichiers (scanne l'ensemble du système) :

```
updatedb
```

Gestion des fichiers et dossiers :

- Afficher le dossier courant :

```
pwd
```

- Afficher le contenu d'un dossier :

```
ls [nom dossier ou rien si dossier actuel]
```

- Afficher le contenu du dossier actuel avec sortie colorée :

```
ls --color=auto
```

- Afficher le contenu du dossier actuel avec les fichiers/dossiers cachés :

```
ls -a (ou ls -A pour ignorer . et ..)
```

- Afficher le contenu du dossier actuel avec détails en ligne :

```
ls -l
```

- Afficher le contenu du dossier actuel avec les **inodes** :

```
ls -i
```

- Utiliser **less** pour contrôler le défilement :

```
ls -l|less
```

- Afficher l'arborescence des fichiers :

```
tree|less
```

- Se déplacer dans un dossier :

```
cd [nom dossier]
```

- Revenir au dossier parent :

```
cd ..
```

- Revenir au dossier par défaut (généralement **\$HOME**) :

```
cd ~ ou cd ~[nom utilisateur]
```

ou

```
cd
```

- Se déplacer dans un dossier en mémorisant le chemin actuel :

```
pushd [nom du dossier]
```

- Retourner au dossier précédent, mémorisé via **pushd** :

```
popd
```

- Afficher la taille des fichiers et dossiers :

```
du -sm [nom dossier ou *]
```

- Créer un fichier vide :

```
touch [nom fichier]
```

P.S. : la commande **touch** permet également de manipuler les dates d'accès et de modification d'un fichier.

- Envoyer un texte sur la console :

```
echo [texte libre]
```

- Concaténer un texte à un fichier de sortie :

```
echo [texte] >> [nom fichier de sortie]
```

- Afficher le contenu d'un fichier texte (avec numérotation des lignes) :

```
cat (-n) [nom_fichier]
```

- Concaténer des fichiers texte vers un fichier de sortie :

```
cat [fichiers] > [nom fichier de sortie]
```

N.B. : le **>** écrase le contenu du fichier, le **>>** ajoute à la fin du fichier existant (ou créé un nouveau fichier si le fichier n'existe pas).

- Compter le nombre de lignes d'un fichier :

```
cat [nom fichier]|wc -l
```

- Extraire une colonne d'un fichier :

```
cat [nom fichier]|cut -d[délimiteur] -f[numéro colonne]
```

- Créer (récursivement) un dossier :

```
mkdir (-p) [nom dossier]
```

- Afficher les différences entre deux fichiers :

```
diff fichier1 fichier2
```

- Afficher les différences entre deux dossiers :

```
diff -r dossier1 dossier2
```

- Copier un fichier :

```
cp [fichier ou dossier source] [fichier ou dossier destination]
```

- Copier un fichier avec conservation des attributs (mode archive) :

```
cp -a [fichier ou dossier source] [fichier ou dossier destination]
```

- Déplacer ou renommer un fichier :

```
mv [fichier ou dossier source] [fichier ou dossier destination]
```

- Changer rapidement l'extension d'un groupe de fichiers :

```
for i in *.JPG;do mv $i ${i%.JPG}.jpg;done
```

- Effacer un fichier :

```
rm [fichier]
```

- Effacer récursivement un dossier :

```
rm -r(f) [dossier]
```

- Créer un lien symbolique ou « doux » (soft link) :

```
ln -s [fichier ou dossier réel] [nom du lien]
```

- Créer un lien dur (hard link) :

```
ln [fichier ou dossier réel] [nom du lien]
```

- Changer (récursivement) le propriétaire :

```
chown (-R) [nom utilisateur ou uid]:[nom groupe ou gid] [fichier ou dossier]
```

- Changer (récursivement) les droits propriétaire/groupe/défaut :

```
chmod (-R) [lettres ou numéro octal] [fichier ou dossier]
```

- P.S. : sur une partition ext4 (format par défaut sous GNU/Linux), la gestion des droits étendus (qui permettant d'attribuer plusieurs propriétaires et groupes à un même fichier/dossier) se fait via `getfacl/setfacl`, et `getfattr/setfattr`. Ces commandes deviennent indispensables dans des cas plus complexes d'administration, notamment quand on doit gérer un serveur **Samba** / *Active Directory*, avec profils itinérants et dossiers partagés.

- Changer récursivement les droits sur des fichiers (exemples) :

```
find . -type f -print0|xargs -0iX chmod 600 X
find . -type f -print0|xargs -0iX chmod 644 X
find . -type f -print0|xargs -0iX chmod 660 X
find . -type f -print0|xargs -0iX chmod 664 X
```

- Changer récursivement les droits sur des dossiers (exemples) :

```
find . -type d -print0|xargs -0iX chmod 755 X
find . -type d -print0|xargs -0iX chmod 770 X
find . -type d -print0|xargs -0iX chmod 775 X
```

- Rechercher et afficher un motif texte contenu dans les fichiers **.html**, depuis le dossier courant :

```
find . -type f -name "*.html" -exec grep -H "[motif recherché]" {} \;
```

- Rechercher tous les fichiers **HTML** dans l'arborescence fille, et les concaténer dans **all.html** :

```
find . -type f -name "*.html" -print0|xargs -0iX
cat X >> all.html
```

- Rechercher les fichiers/dossiers fils dont le propriétaire est inconnu :

```
find . -nouser
```

- Rechercher les fichiers/dossiers fils dont le groupe est inconnu :

```
find . -nogroup
```

- Rechercher les fichiers modifiés dans les dernières 24h :

```
find . -mtime -1
```

- Enlever les commentaires (**#**) d'un fichier de configuration :

```
grep -o '^[^#]*' [nom du fichier]
```

ou (avec les lignes vides en plus) :

```
egrep -v '^(^#|^$)' [nom du fichier]
```

- Rechercher récursivement un motif dans un dossier :

```
fgrep -r "[motif]" [dossier]
```

- Identifier un fichier :

```
file [nom du fichier]
```

- Trier un fichier texte :

```
sort -u [fichier entrée] -o [fichier sortie]
```

- Afficher les premières lignes d'un fichier :

```
head -n [nombre de lignes] [fichier]
```

- Afficher les dernières lignes d'un fichier :

```
tail -n [nombre de lignes] [fichier]
```

- Afficher le contenu d'une archive compressée :

```
zcat [nom archive.zip]
```

- Idem mais avec contrôle du défilement :

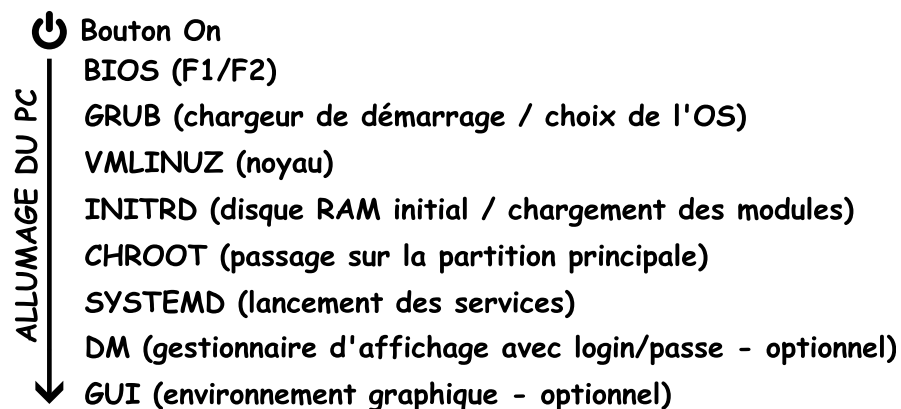
```
zless [nom archive.zip]
```

Processus de démarrage simplifié

GRUB (chargeur de démarrage) > **VMLINUZ** (noyau) > **INITRD** (mini arborescence contenant notamment des modules que le noyau charge à la volée pour supporter les périphériques

découverts) > **CHROOT** (passage sur la partition principale) > **SYSTEMD** (lancement des services : Web, SSH, Messagerie, Partage de fichiers, ...) > **LIGHTDM** (gestionnaire d'affichage permettant de saisir son identifiant et son mot de passe : GDM, KDM, XDM, ...) > **GUI** (environnement graphique au choix : Mate, Gnome, KDE, XFCE, Cinnamon, LXDE, ICEWM, etc)

Plus schématiquement :



Chargeur de démarrage

- Installation du chargeur de démarrage :
 - si BIOS anciens, limité à 4 partitions primaires en MBR : c'est le paquet **grub-pc** qu'il faut installer.
 - si machine récente en BIOS UEFI et partitions GPT : c'est le paquet **grub-efi-amd64** qu'il faut installer.
- À savoir pour les BIOS UEFI :

- la partition EFI/ESP (100 Mo maxi en FAT32 + flag EFI/ESP) doit être montée dans `/bios/efi` avant d'appeler `update-grub`. Lors d'une nouvelle installation, cette partition est automatiquement ajoutée dans `/etc/fstab`.
- avant d'installer une distribution GNU/Linux, pensez à toujours bien désactiver le *Secure Boot* dans le BIOS UEFI, sinon l'installation échouera !
- Partitionnement du disque dur :
 - pour les anciens BIOS, on utilisera `fdisk`, restreint aux partitions en MBR (Master Boot Record)
 - pour les nouveaux BIOS UEFI, on utilisera `gdisk`, restreint aux partitions GPT (GUID partition table)

Les deux utilitaires utilisent les mêmes commandes, la plupart sous formes de simples caractères, dont la touche `m` une fois lancé pour obtenir l'aide en ligne...

- Conseil : toujours faire un `lsblk -f` préalable pour bien vérifier les périphériques de blocs présents sur la machine.
- Installer **GRUB** sur un disque SATA :

```
grub-install /dev/sdX avec X=a,b,c,d,...
```

- Installer **GRUB** sur un disque NVMe :

```
grub-install /dev/nvmeXYZ
```

- Mettre à jour **GRUB** (maj de `/boot/grub/grub.cfg`) :

```
update-grub
```

- Informations sur le chargeur de démarrage :

```
bootctl status
```

- Liste des chargeurs en BIOS UEFI :

```
efibootmgr
```

- Changer l'ordre de démarrage en BIOS UEFI :

```
efibootmgr -o XXXX,YYYY,ZZZZ
```

- Forcer le prochain démarrage en BIOS UEFI :

```
efibootmgr -n XXXX
```

Noyau vmlinuz

- Afficher le noyau actuel :

```
uname -a
```

- Afficher les détails de la distribution GNU/Linux :

```
lsb_release -a
```

- Récapituler les principales informations de l'hôte :

```
hostnamectl status
```

- Changer le nom d'hôte :

```
hostnamectl set-hostname $NAME
```

- ou

```
modifier /etc/hostname à la main
```

- Afficher les options du noyau utilisées lors du démarrage :

```
cat /proc/cmdline
```

- Analyser le temps de démarrage total :

```
systemd-analyze time
```

- Analyser les temps de démarrage par service :

```
systemd-analyze blame
```

- Créer un graphique SVG pour visualiser les temps de démarrage puis affichage :

```
systemd-analyze plot > plot.svg  
eog plot.svg
```

- Afficher le temps écoulé depuis le démarrage :

```
uptime
```

- Affichage/réglage de la date/de l'heure :

```
date
```

ou

```
timedatectl
```

- Affichage du calendrier :

```
cal -y
```

Initrd / Initramfs

- Visualiser le système de fichier RAM au démarrage :

```
lsinitramfs /boot/initrd.img-$(uname -r)
```

- Mettre à jour les fichiers **initrd** pour les noyaux installés :

```
update-initramfs -u -k all
```

- Afficher la liste des modules chargés :

```
lsmod
```

- Afficher les informations sur un module :

```
modinfo [nom module]
```

- Retirer un module noyau en cours d'utilisation (attention!) :

```
rmmod [nom_module]
```

- Ajouter un module en dynamique :

```
modprobe [nom_module]
```

Réinstallation de GRUB via chroot

L'astuce suivante sert à reprendre le contrôle d'un PC dans le chargeur **GRUB** serait défaillant. Il faut d'abord démarrer depuis une clé USB GNU/Linux, et ouvrir un terminal sous **root**.

On admettra ici que la partition / du système GNU/Linux à récupérer est **/dev/sda1** et qu'on est sur une machine en BIOS UEFI :

```
mount /dev/sda1 /mnt
mount --bind /dev /mnt/dev
mount --bind /proc /mnt/proc
mount --bind /sys /mnt/sys
chroot /mnt
sed -i "lnameserver 8.8.8.8" /etc/resolv.conf
apt update
apt install grub-efi-amd64
update-grub
grub-install /dev/sda
CTRL-D ou exit
umount /mnt
reboot
```

Systemd / Services au démarrage

Historiquement, *InitV* (*init*) était le premier processus lancé après le noyau pour démarrer et contrôler les services. Composé de scripts BASH dans */etc/init.d*, *InitV* était relativement facile à manipuler et configurer, mais souffrait de plusieurs tares, dont notamment un démarrage séquentiel lent.

Systemd est aujourd'hui présenté comme le successeur d'*InitV*, et permet un démarrage des processus en parallèle, accélérant donc le chargement des services.

Cela étant, contrairement à son prédécesseur, il ne se contente plus de contrôler uniquement le démarrage et l'arrêt des services : il offre désormais un contrôle complet sur quasiment tous les composants principaux du système : réseau (*networkctl*), nom

de domaine (*resolvctl*), nom d'hôte (*hostnamectl*), date et temps (*timedatectl*), langue et collation (*localectl*), communication inter-processus via DBUS (*busctl*), journaux système (*journalctl*), authentification (*loginctl*), etc.

Bref, *Systemd* entre en conflit avec de nombreux logiciels et manières de faire historiques, et continue de faire couler beaucoup d'encre entre pros et cons, en particulier parce qu'il « casse » la philosophie **KISS** (Keep it Simple, Stupid) historique d'UNIX...

Ainsi en embarqué, on évite généralement d'installer *Systemd*, qui n'apporte pas d'avantages particuliers, et peut donc parfaitement être remplacé par de simples scripts BASH souvent plus légers et plus simples à maintenir. Et le projet *devuan* de continuer de proposer une Debian complète sans *Systemd*, pour ceux qui veulent vraiment éviter ce logiciel !

- Démarrer un service :

```
systemctl start [service]
```

- Arrêter un service :

```
systemctl stop [service]
```

- Afficher le statut un service :

```
systemctl status [service]
```

- Redémarrer un service :

```
systemctl restart [service]
```

- Activer un service au démarrage :


```
systemctl enable [service]
```

- Désactiver un service au démarrage :

```
systemctl disable [service]
```

- Masquer un service :

```
systemctl mask [service]
```

- Démasquer un service :

```
systemctl unmask [service]
```

- Lister les unités actives :

```
systemctl list-units
```

- Lister les unités actives ou non :

```
systemctl list-units --all
```

- Lister les fichiers des unités :

```
systemctl list-unit-files
```

- Lister le contenu d'un fichier d'unité :

```
systemctl cat [unité systemd]
```

- Modifier un fichier d'unité (mode **override**) :

```
systemctl edit [unité systemd]
```

- Revenir à l'unité d'origine (efface les modifications antérieures) :

```
systemctl revert [unité systemd]
```

- Recharger les unités (après une modification) :

```
systemctl daemon-reload
```

- Redémarrer la machine :

```
systemctl reboot
```

- Arrêter la machine et l'éteindre :

```
systemctl shutdown
```

Display Manager / GUI

- Le gestionnaire d'affichage/de sessions affiche les champs identifiant/mot de passe après le démarrage de la machine, sauf si vous avez réglé votre machine pour démarrer automatiquement sous un utilisateur donné.
- Il permet, en cours de session, de changer d'utilisateur quand le mode multi-utilisateurs est activé.
- Sous GNU/Linux, sa présentation varie d'une distribution à l'autre, mais est généralement lié à la « saveur » installée.
- Nous prendrons ici exemple de *LightDM*, gestionnaire de connexion (*display manager*) par défaut sous Ubuntu MATE, mais il existe d'autres gestionnaires comme *GDM* (Gnome), *KDM* (KDE), *LXDM* (LXDE), *XDM* (X11 par défaut), *SDDM* (Simple Desktop Display Manager), ...

- Tous proposent un fichier de configuration texte pour affiner la présentation et les modalités d'identification de l'utilisateur, et certains permettent même de se connecter à une autre machine GNU/Linux, via le protocole XDMCP (UDP 177), non sécurisé.
- Afficher la configuration du gestionnaire de session **lightdm** (N.B : le fichier de configuration par défaut est **/etc/lightdm/lightdm.conf**) :

```
lightdm --show-config
```

- Donner les droits à un usager d'afficher des programmes dans votre session graphique :

```
xhost +si:localuser:[nom usager]
```

- Donner les droits à tous les usagers d'afficher des programmes dans votre session graphique (non sécurisé !) :

```
xhost +
```

- Tuer un processus visuellement :

```
xkill
```

- Désactiver la mise en veille de l'écran (par exemple quand on veut regarder une vidéo) :

```
xset -dpms
```

- Désactiver la « cloche » d'alerte :

```
xset -b
```

Processus

- Liste des processus actifs

```
top
```

- Afficher les processus :

```
ps aux
```

- Afficher l'arbre des processus :

```
pstree
```

- Afficher des informations sur les processus :

```
ps xawf -eo pid,user,cgroup,args
```

- Rechercher des processus par mot-clé :

```
ps aux|grep -v grep|grep [motif]
```

- Rechercher le processus d'un programme par son nom :

```
pidof [nom programme]
```

- Retrouver le **PID** du processus dans **/var/run** :

```
cat /var/run/[nom programme.pid]
```

- Tuer « gentiment » un processus :

```
kill -TERM [processus id]
```

ou

```
kill -15 [processus id]
```

- Tuer et redémarrer un processus :

```
kill -HUP [processus id]
```

ou

```
kill -1 [processus id]
```

- Tuer « violemment » un processus :

```
kill -KILL [processus id]
```

ou

```
kill -9 [processus id]
```

- Tuer tous les processus d'un utilisateur :

```
killall -u [nom utilisateur]
```

- Tuer « gentiment » tous les processus ayant le même nom de programme :

```
killall [nom programme]
```

- Afficher la ligne de commande liée à un processus en utilisant son **PID** :

```
cat /proc/[numéro processus]/cmdline
```

Journaux système

- Afficher les messages noyaux :

```
dmesg
```

- Afficher les journaux système en continu :

```
tail -f /var/log/syslog
```

ou

```
journalctl -n500 -f
```

Matériel

- Rechercher la liste du matériel :

```
lshw
```

ou

```
inxi -F (si Ubuntu)
```

- Identifier les processeurs :

```
lscpu
```

ou

```
cat /proc/cpuinfo
```

- Afficher la mémoire :

```
free
```

- Afficher l'espace disque occupé/restant :

```
df
```

- Rejouer une commande toutes les 2s :

```
watch [commande]
```

- Visualiser les périphériques PCI :

```
lspci -v
```

Cette commande donne le **nom du module** noyau associé au périphérique USB, que l'on peut ensuite manipuler via `rmmmod`, `modinfo` et `modprobe`.

- Visualiser les périphériques USB (en mode « arbre » verbeux) :

```
lsusb -tv
```

Tâches répétitives

- Éditer les tâches répétitives avec CROND :

```
crontab -e
```

- Afficher les timers **Systemd**:

```
systemctl list-timers
```

Aspirateur de site

- Télécharger rapidement un fichier :

```
wget [options] [url]
```

ou

```
curl [options] [url]
```

- Téléchargement sélectif des pages HTML d'un site internet :

```
wget --reject
'*.js,*.css,*.ico,*.txt,*.gif,*.jpg,*.jpeg,*.png
,*.mp3,*.pdf,*.tgz,*.flv,*.avi,*.mpeg,*.iso,*.zi
p' --ignore-tags=img,link,script --
header="Accept: text/html" --limit-rate=200k --
no-clobber --convert-links --random-wait -U
"Mozilla/5.0 (X11; Linux x86_64; rv:10.0)
Gecko/20100101 Firefox/10.0" -E html -r [url]
```

Gestion des périphériques disques

- Afficher les périphériques disques + types partitions :

```
lsblk -f
```

- Afficher les UUID des partitions :

```
blkid
```

- Afficher les points de montage au démarrage :

```
cat /etc/fstab
```

- Monter un système de fichier :

```
mount (-t [type]) [périphérique] [point de
montage]
```

- Lister tous les points de montages actifs :

```
mount -l
```

- Trier les points de montage actifs par type :

```
mount -l -t ext4
```

- Remonter une partition en lecture écriture :

```
mount -o remount,rw [partition]
```

- Contrôler manuellement une partition **ext4** démontée :

```
fsck.ext4 [périphérique]
```

Gestion des périphériques batterie

- Surveiller les interfaces :

```
upower --monitor-detail
```

- Afficher les chemins disponibles :

```
upower -e
```

- Afficher les informations détaillées sur le périphérique :

```
upower -i chemin
```

Gestion des usagers et groupes :

- Afficher les usagers connectés :

```
who
```

- Afficher les usagers connectés et occupations respectives :

```
w
```

- Afficher les dernières connexions :

```
last
```

- S'auto-identifier :

```
whoami
```

- Renvoyer les informations sur un usager :

```
id [nom usager]
```

- Renvoyer les groupes d'un usager :

```
groups [nom usager]
```

- Lister des usagers sur le système :

```
cat /etc/passwd
```

ou

```
getent passwd
```

- Lister des groupes sur le système :

```
cat /etc/group
```

ou

```
getent group
```

- Ajouter un groupe :

```
groupadd [nom du groupe]
```

ou

```
addgroup --gid [GID] [nom du groupe]
```

- Effacer un groupe

```
groupdel [nom du groupe]
```

- Ajouter des groupes à un utilisateur SANS CONSERVER ses anciens groupes :

```
usermod -G [groupe1,groupe2,groupeN]  
[utilisateur]
```

- Ajouter des groupes à un utilisateur EN CONSERVANT ses anciens groupes :

```
usermod -aG [groupe1,groupe2,groupeN]  
[utilisateur]
```

- Enlever un utilisateur d'un groupe :

```
gpasswd -d [utilisateur] [groupe]
```

- Ajouter un utilisateur :

```
adduser (--force-badname) (--home /home/[nom  
utilisateur ASCII]) (--uid [UID]) (--gid [GID])  
(--gecos \"[prénom et nom réels avec espaces]\")  
[nom utilisateur ASCII]
```

- Effacer un utilisateur EN CONSERVANT ses fichiers personnels dans /home :

```
userdel [nom utilisateur]
```

- Effacer un utilisateur EN SUPPRIMANT ses fichiers personnels dans /home :

```
userdel -r [nom utilisateur]
```

- Changer un mot de passe :

```
passwd [nom utilisateur]
```

Compression de fichiers

- Créer d'une archive compressée :

```
tar czvf [nom archive.tgz] [fichiers ou  
dossiers]
```

- Vérifier une archive **TGZ** :

```
tar tzvf [nom archive.tgz]
```

- Extraire une archive **TGZ** dans le dossier courant :

```
tar xzvf [nom archive.tgz]
```

- Extraire une archive **TGZ** dans un autre dossier :

```
tar xzvf [nom archive.tgz] -C [dossier de  
destination]
```

- Compresser des fichiers/dossiers en fichiers **ZIP** distincts :

```
gzip [fichiers ou dossiers]
```

- Compresser des fichiers/dossiers en une archive **ZIP** :

```
gzip -r [archive.gz] [fichiers ou dossiers]
```

- Décompresser depuis un fichier **ZIP** :

```
gunzip [fichier.gz]
```

- Exemple de compression + envoi sur un serveur distant via SSH après transmission de sa clé publique :

```
tar zcvf - /etc|ssh user@machine "tar zxvf - -  
C /tmp"
```

Connexion et transfert sécurisés en SSH/SCP/RSYNC

- Se connecter à une autre machine en **SSH** :

```
ssh -XY [nom utilisateur]@[nom machine ou ip  
machine]
```

- Créer des clés locales **SSH** pour authentification automatique :

```
cd $HOME && ssh-keygen -t ed25519 -b 4096
```

- Transmettre sa clé publique ED25519 vers un hôte distant auquel on veut pouvoir se connecter automatiquement en **SSH** :

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub  
[utilisateur]@[machine]
```

- Copier les fichiers vers une autre machine :

```
scp -r [fichier ou dossier source]  
[utilisateur]@[machine]:[dossier de destination]
```

- Copier les fichiers depuis une machine vers un dossier local :

```
scp -r [utilisateur]@[machine]:[fichier ou  
dossier source] [dossier local de destination]
```

- Synchroniser des fichiers vers une autre machine :

```
rsync -avz [fichier ou dossier source]  
[utilisateur]@[machine]:[dossier de destination]
```

- Synchroniser des fichiers vers une autre machine en mode incrémental avec sauvegarde :

```
DATE=$(date '+%Y%m%d%H%M%S') && rsync -avz --  
delete --backup-dir=backup/${DATE} [fichier ou  
dossier source] [utilisateur]@[machine]:[dossier  
de destination]
```

Exemple en local :

```
cd /tmp  
mkdir src dst  
echo "coucou" > src/foret.txt  
cat src/foret.txt  
# sur une ligne en respectant bien le src au  
lieu de src/ !  
DATE=$(date '+%Y%m%d%H%M%S') && rsync -avz --  
delete --backup-dir=backup/${DATE} src dst/  
ls -R dst
```

```

cat dst/src/foret.txt
echo "hibou" > src/foret.txt
cat src/foret.txt
DATE=$(date '+%Y%m%d%H%M%S') && rsync -avz --
delete --backup-dir=backup/${DATE} src dst/
ls -R dst
# l'ancien fichier a bien été archivé
cat dst/backup/*/src/foret.txt
# le nouveau contenu a été mis à jour
cat dst/src/foret.txt
rm src/foret.txt
echo "poisson" > src/lac.txt
DATE=$(date '+%Y%m%d%H%M%S') && rsync -avz --
delete --backup-dir=backup/${DATE} src dst/
ls -R dst
# le fichier a été effacé
cat dst/src/foret.txt
# mais les anciennes sauvegardes sont là !
cat dst/backup/*/src/foret.txt
# et notre poisson est bien dans son lac...
cat dst/src/lac.txt

```

Attention : **rsync** est particulièrement sensible aux **/** ! Si vous indiquez **src/** en tant que source, il prendra les fichiers/dossiers dans **src/**, alors que si vous indiquez **src** sans le **/**, il prendra le dossier **src/** lui-même ! L'option **-n** (dry run) est particulièrement utile lors de la mise au point de scripts, et permet de tester les transferts sans les exécuter réellement. L'option **-S** (**--sparse**) est indispensable quand vous transférez des fichiers au format qcow2.

L'option **-H** (**--hard-links**) permet de préserver les liens durs sur la destination. Enfin, si vous voulez préserver les ACL simples ou étendues, il faut respectivement utiliser les options **-A** et **-X**. C'est notamment utile si vous sauvegardez des fichiers Windows.

Vérification d'empreintes SSH

La vérification de l'empreinte est à faire quand vous ouvrez un hébergement, la première fois que vous vous connectez en SSH. Il s'agit de mémoriser l'empreinte ECDSA ou ED25519 (utilisée dans la Debian 12 par défaut), sensée être statique et définitive.

ECDSA = Elliptic Curve Digital Signature Algorithm.

- Affichage de la signature ED25519 sur le serveur distant :

```
ssh-keygen -l -E sha256 -f
/etc/ssh/ssh_host_ed25519_key.pub
```

- Affichage de la signature ED25519 lors de la connexion à un serveur en SSH :

```
ssh -o FingerprintHash=sha256 -o
"UserKnownHostsFile=/dev/null"
[utilisateur]@[machine]
```

Attention : si les deux signatures diffèrent, alors une attaque de l'homme du milieu est en cours !

Opérations réseau de base

- Afficher le nom d'hôte :

```
hostname
```


- Afficher le domaine hôte :

```
nslookup
```

- Retrouver son IP publique :

```
curl http://ifconfig.me/all
```

- Afficher la table **arp** (*Address Resolution Protocol*) :

```
arp -a
```

- Afficher l'état de la couche de liaison (niveau 2) :

```
ip l
```

- Activer/Désactiver une interface :

```
ip l set [interface] up/down
```

- Afficher les interfaces et leurs adresses IP (niveau 3) :

```
ip a (-c)
```

- Ajouter une IP manuelle sur une interface (niveau 3) :

```
ip a add [192.168.1.X/24] dev [interface]
```

- Enlever une IP manuelle sur une interface (niveau 3) :

```
ip a del [192.168.1.X/24] dev [interface]
```

- Tester la présence d'un hôte (s'il veut bien répondre) :

```
ping [nom réseau canonique ou adresse ip] -c 3
```

- Lister les connexions internet actives (serveurs et établies) :

```
netstat -ltaupe
```

- Rechercher des hôtes à proximité :

```
ip n
```

ou

```
netdiscover -i [interface réseau]
```

- Se connecter manuellement en DHCP – ATTENTION : toujours imposer un nom d'interface, sinon la recherche se fera sur toutes les interfaces par défaut :

```
dhclient [nom interface]
```

- Libérer le bail DHCP :

```
dhclient -r
```

- Afficher les routes (et notamment la passerelle par défaut)

```
ip r
```

- Ajouter une route par défaut :

```
ip r add [10.0.0.0/8] via [192.168.1.254] (dev [nom interface])
```

- Afficher des serveurs DNS :

```
cat /etc/resolv.conf
```

ou

```
resolvectl
```

- Afficher la configuration réseau fixe au démarrage :

```
cat /etc/network/interfaces
```

- Liste des connexions actives TCP/UDP :

```
ss -tu state connected
```

- Lister toutes les sockets TCP (en écoute ou non) :

```
ss -ta
```

- Lister toutes les sockets TCP (en écoute ou non) :

```
ss -ta state connected
```

- Lister toutes les sockets TCP en écoute :

```
ss -tl
```

- Lister toutes les sockets UDP (en écoute ou non) :

```
ss -ua
```

- Lister toutes les sockets UDP connectées :

```
ss -ua state connected
```

- Lister toutes les sockets UDP en écoute :

```
ss -ul
```

- Gérer un dongle bluetooth jusqu'à l'appairage :

```
bluetoothctl
```

puis :

```
list
select xx:xx:xx:xx:xx:xx
show
power on
discoverable on
pairable on
scan on
devices
paired-devices
agent on
pair yy:yy:yy:yy:yy:yy
trust yy:yy:yy:yy:yy:yy
paired-devices
info yy:yy:yy:yy:yy:yy
connect yy:yy:yy:yy:yy:yy
```

- Scanner les ports d'une machine :

```
nmap -v [ip ou nom machine]
```

Utilisation du network-manager

- Accéder à l'outil de configuration général :

```
nmtui
```

- Afficher les périphériques réseau :

```
nmcli (device)
```

- Afficher les connexions sauvegardées :

```
nmcli connection
```

- Scanner les points d'accès WiFi :

```
iwlist scan
```

- Afficher les points d'accès WiFi :

```
nmcli dev wifi list
```

- Activer/désactiver le WiFi :

```
nmcli radio wifi (on | off)
```

- Se connecter à un réseau WiFi :

```
nmcli device wifi connect [SSID] password [mot  
de passe] ifname [interface]
```

- Afficher l'état du switch radio :

```
rfkill list
```

- Bloquer/débloquer le switch WiFi logiciel :

```
rfkill unblock [numéro ou « all »]
```

Serveur ou client windows

- Rechercher les partages disponibles sur un serveur :

```
smbclient -L [ip ou nom machine]
```

- Se connecter à un partage :

```
smbclient '\\[nom ou ip serveur]\\[nom partage] -  
U [nom utilisateur] (-W [domaine])
```

- Ajouter un utilisateur Samba :

```
smbpasswd -a $USERNAME
```

- Ajouter une machine (le nom de l'hôte doit finir par \$) :

```
smbpasswd -a $HOST$
```

- Voir les utilisateurs Samba :

```
pdbedit -Lv
```

- Afficher le contenu :

```
dir
```

- Créer un dossier :

```
mkdir [dossier]
```

- Effacer un dossier :

```
rmdir [dossier]
```

- Naviguer dans un dossier :

```
cd [dossier]
```

- Téléverser des fichiers ou dossiers :

```
(m)put [fichier ou dossier]
```

- Télécharger des fichiers ou dossiers :

```
(m)get [fichier ou dossier]
```

- Création d'un partage windows :

```
apt install samba

# ajoutez à la fin du fichier
/etc/samba/smb.conf :
[linuxsharename]
    path=/path/to/folder
    force user = $USER
    public = yes
    guest ok = yes
    writable = yes

service samba restart
```

- Montage d'un partage windows :

```
apt install cifs-utils

mount.cifs //SERVER/SHARE /mnt -o
username=$USER,password=$PASS,workgroup=$WORKGRO
UP,vers=3.0
```

- Même chose mais en cachant les identifiants dans **/root/.cifs** :

```
cat /root/.cifs
```

```
username=$USER
password=$PASS
workgroup=$WORKGROUP
```

```
mount.cifs //SERVER/SHARE /mnt -o
credentials=/root/.cifs,vers=3.0
```

- Monter le partage au démarrage via **/etc/fstab** :

```
//SERVER/SHARE /mnt cifs
ro,user,credentials=/root/.cifs 0 0
```

Extras

- Trouver des doublons à partir du dossier courant :

```
fdupes -r . > doublons.txt
```

- Les afficher (attention à la mémoire) :

```
xdg-open doublons.txt
```

- Renommer des images JPEG en les numérotant :

```
ls *.jpg|awk 'BEGIN{n=1} {print "mv "$1"
"n".jpg";n=n+1}'
```

pour contrôler la sortie à exécuter, puis si ok :

```
ls *jpg|awk 'BEGIN{n=1} {system("mv "$1"
"n".jpg");n=n+1}'
```

Exercice 1 : création d'un conteneur Xen

- Création d'un fichier conteneur de 10 Go :

```
mkdir /opt/xen
cd /opt/xen
dd if=/dev/zero of=template.img bs=1G count=10
```

- Liste des périphériques de bouclage actifs :

```
losetup -a
```

- Prochain périphérique de bouclage libre :

```
losetup -f
```

On admettra dans la suite que la commande a retourné **/dev/loop0**.

- Mise en place d'un périphérique de bouclage sur le conteneur :

```
losetup /dev/loop0 /opt/xen/template.img
```

- Vérification :

```
losetup -a
```

- Partitionnement du conteneur bouclé :

```
fdisk /dev/loop0
```

- Création d'un table de partition DOS dans **fdisk** :

```
o
```

- Création d'une nouvelle partition DOS dans **fdisk** :

```
n
```

- Indication du type de nouvelle partition (primaire ou étendue) :

```
p
```

- Indication d'une taille de 2 Go dans **fdisk** pour l'espace d'échange (**SWAP**) :

```
+2G
```

- Liste des partitions supportées dans **fdisk** :

```
l
```

Changement du type de partition dans **fdisk** :

```
t (puis le numéro de partition 82 = swap, 83 = Linux)
```

- Écriture des changements dans **fdisk**

```
w
```

- Montage des partitions du conteneur dans **/dev/mapper** :

```
kpartx -av /dev/loop0
```

- Contrôle :

```
ls -l /dev/mapper
```

- Formatage de la partition 1 (**swap**) du périphérique de bouclage :

```
mkswap /dev/mapper/loop0p1
```

- Formatage de la partition 2 (**ext4**) du périphérique de bouclage :

```
mkfs.ext4 /dev/mapper/loop0p2
```

- Montage de la partition du conteneur dans le dossier **/mnt** :

```
mount /dev/mapper/loop0p2 /mnt
```

- Contrôle :

```
df
```

- Installation des fichiers de base d'une Debian **stable** dans la partition montée :

```
debootstrap --includes=less,tree,locales-  
all,vim,gpm,sudo,openssh-server,grub-  
pc,manpages-fr stable /mnt
```

- Modification de la table de montage du nouveau système :

```
nano /mnt/etc/fstab
```

- Contenu à ajouter :

```
/dev/xvda1 none swap sw 0 0  
/dev/xvda2 / ext4 defaults,errors=remount-ro 0 1
```

- Modification du nom d'hôte du nouveau système :

```
echo "debian" > /mnt/etc/hostname
```

- Préparation de la configuration réseau :

```
sed -i  
's/^GRUB_CMDLINE_LINUX_DEFAULT="quiet"/GRUB_CMD  
LINE_LINUX_DEFAULT="net.ifnames=0  
ipv6.disable=1"/' /mnt/etc/default/grub  
  
echo "  
auto lo  
iface lo inet loopback  
  
auto eth0  
allow-hotplug eth0  
iface eth0 inet dhcp  
" >> /mnt/etc/network/interfaces
```

- Contenu à ajouter dans **/mnt/etc/fstab** :

```
proc /proc proc defaults 0 0  
/dev/xvda1 none swap sw 0 0  
/dev/xvda2 / ext4 defaults,errors=remount-ro 0 1
```

- Copie des modules du noyau courant :

```
cp -a /lib/modules /mnt/lib/modules
```

- Déplacement sur la partition racine :

```
mount --bind /dev /mnt/dev  
mount --bind /proc /mnt/proc  
mount --bind /sys /mnt/sys  
chroot /mnt
```

- Mise en place du mot de passe **root** :

```
passwd
```

- Configuration de la localisation :

```
apt install locales console-data
dpkg-reconfigure locales
debconf-show locales
```

- Configuration de la disposition du clavier :

```
apt install keyboard-configuration
debconf-show keyboard-configuration
```

- Configuration de la langue en console :

```
apt install console-setup
debconf-show console-setup
```

- Configuration du fuseau horaire :

```
dpkg-reconfigure tzdata
debconf-show tzdata
```

- Ajouts de terminaux virtuels (utiliser ALT+flèche gauche/droite pour changer de VT)

```
sed -i 's/^#NAutoVTs=6/NAutoVTs=6/'
/etc/systemd/logind.conf
```

- Sortie du **chroot** :

```
CTRL+D ou exit
```

- Démontage de la partition **/mnt** :

```
umount /mnt
```

- Démontage du conteneur dans **/dev/mapper** :

```
kpartx -d /dev/loop0
```

- Démontage du périphérique de bouclage :

```
losetup -d /dev/loop0
```

- Exemple de fichier de configuration **/etc/xen/template** pour exploiter notre conteneur :

```
kernel="/boot/vmlinuz-...(numéro noyau)"
ramdisk="/boot/initrd.img-...(numéro noyau)"
memory=512
name="stable"
vif=['mac=00:11:22:33:44:55']
disk=['file:/opt/xen/template.img,xvda,w']
root="/dev/xvda2"
extra="fastboot xencons=tty console=tty0"
```

Exercice 2 : installer un serveur apache2

```
apt install apache2
chown -R www-data:www-data /var/www/html
cd /var/www/html
find ./ -type d -print0|xargs -0iX chmod 770 X
find ./ -type f -print0|xargs -0iX chmod 660 X
find ./ -type f -name .ht* -print0|xargs -0iX
chmod 600 X
adduser $USER www-data
```

Exercice 3 : installer WordPress depuis ses sources

```
apt install apache2 mysql-server mysql-client
php php-mbstring php-mysql
mysql -u root-p -e"CREATE DATABASE dbname;SHOW
DATABASES"
cd /var/www/html
mkdir monblog
cd monblog
wget https://fr.wordpress.org/latest-
fr\_FR.tar.gz
tar zxvf latest-fr_FR.tar.gz
chown -R www-data:www-data /var/www/html
find ./ -type d -print0|xargs -0iX chmod 770 X
find ./ -type f -print0|xargs -0iX chmod 660 X
find ./ -type f -name .ht* -print0|xargs -0iX
xdg-open http://mondomain.fr/monblog
```

Autres ressources

Vous trouverez sur mon site web <https://alsatux.com/UHA> tout un tas d'autres documents destinés aux étudiants de l'UHA, dont bon nombre d'alias de commandes, les nouvelles commandes réseau **ip** et leurs ancêtres **ifconfig**, la gestion des droits sur les fichiers en ext4, la création d'une machine virtuelle Debian sous KVM depuis la ligne de commande, l'installation d'un serveur Samba en Active Directory, etc