

## L'ARBORESCENCE DES FICHIERS EXT ORGANISÉE DE LA FAÇON SUIVANTE :

- `/` : La racine du disque dur
- `/bin` : les commandes de base (mv, rm, ls...)
- `/dev` : les 'devices', accès aux périphériques du système
  - `/dev/stdin` : désigne l'entrée standard
  - `/dev/stdout` : désigne la sortie standard
  - `/dev/stderr` : désigne la sortie d'erreur standard(L'entrée (stdin), la sortie (stdout) et l'erreur standard (stderr) ont respectivement les descripteurs de fichiers 0, 1 et 2.)
- `/dev/tty` désigne le terminal auquel vous êtes connecté
- `/dev/null` : Ce fichier se comporte comme un « trou noir » en absorbant toute redirection sans être modifié.
- `/dev/zero` : Ce fichier renvoie des octets nuls à l'infini.
- `/dev/urandom` : Ce fichier renvoie des octets aléatoires à l'infini.
- `/etc` : les fichiers de configuration
  - `/etc/crontab` : configuration de crontab
  - `/etc/fstab` : configuration des partitions et des points de montage
  - `/etc/group` : la liste des groupes
  - `/etc/hosts` : correspondance adresse IP - Nom de machine codé en dur
  - `/etc/passwd` : La liste des utilisateurs
- `/home` : les répertoires des utilisateurs
  - `/home/user` : Répertoire d'un utilisateur (Vous êtes ici (^-^))
  - `/home/user/bin` : Répertoire des exécutable de l'utilisateur (ce qui les rends accessible depuis le terminal)
  - `/home/user/Bureau` : Bureau de utilisateur
- `/lib` : les librairies
- `/root` : le répertoire de l'utilisateur root
- `/sbin` : les commandes système pour root (fdisk, fsck...)
- `/tmp` : le répertoire des fichiers temporaires (effacés régulièrement)
- `/usr` : les programmes pour les utilisateurs
  - `/usr/bin` : les programmes de base des utilisateurs
  - `/usr/include` : les fichiers d'en-tête pour la programmation
  - `/usr/b` : les fichiers des librairies pour l'utilisation des programmes
  - `/usr/local` : les programmes compilés ou écrits en local
  - `/usr/sbin` : les programmes d'accès au système pour les utilisateurs
  - `/usr/src` : les sources du kernel (si installées)
  - `/usr/X11R6` : les programmes concernant l'interface graphique
- `/var` : les données variables (les logs, les spools, les mails...)

## DESCRIPTION DE LA HIÉRARCHIE DU SYSTÈME DE FICHIERS

man hier

## RACOURCIS CLAVIERS SUR UN TERMINAL

### GÉNÉRAL

- Tab** : Autocomplétion de mot/programme et de dossier/fichier
- Tab Tab** : Voir les proposition d'autocomplétion de mot/programme et de dossier/fichier
- Ctrl + b** : Permet de reculer le curseur d'un caractère à la fois. (on obtient le même résultat avec la flèche de gauche)
- Ctrl + f** : Permet d'avancer le curseur d'un caractère à la fois. (on obtient le même résultat avec la flèche de droite)
- Ctrl + d** : Efface le caractère sur lequel est positionné le curseur
- Ctrl + a** : Aller au début de la ligne de commande.
- Ctrl + e** : Aller à la fin de la ligne
- Ctrl + l** : Nettoyer l'écran (équivalent de cls ou clear)

### COPIER/COLLER

- Ctrl + k** : Pour couper à partir de là où est positionné votre curseur jusqu'à la fin de la ligne.
- Meta + d** : Pour couper à partir de là où est positionné votre curseur jusqu'à la fin du mot
- Meta + Retour arrière** (backspace): Pour couper à partir de là où est positionné votre curseur jusqu'au début du mot
- Ctrl + w** : Pour couper à partir de là où est positionné le curseur jusqu'à l'espace précédent
- Ctrl + y** : Pour coller ce que vous avez coupé
- Meta + .** : Permet de parcourir les derniers arguments des précédentes commandes pour les replacer dans la commande en cours.
- Ctrl + r** : Permet de rechercher à partir d'une lettre ou d'un mot une commande précédente. Et si vous appuyez plusieurs fois de suite, vous pouvez parcourir ces résultats.
- Ctrl + g** : Pour annuler la recherche et revenir à votre ligne en cours.

# MÉMO

## TERMINAL CMD + APP GNU/LINUX



RANDOM(lab) | ESADSE | version 1.0 | Janvier 2020 | <http://randomlab.io>

Mémo (guide de résistance) des usages du terminal à l'usage des Graphistes/Artistes/Designers/Ingénieurs

### BASES DE COMPREHENSION DU MÉMO

`rep`, `source` et `cible` (en vert) designe un répertoire/dossier du type :

`Bureau/`  
ou  
`Bureau/monDossier`  
ou  
`/home/bob/Bureau/monDossier`

`$file`, `$source` et `$cible` (en violet) designe un fichier du type :

`MonTexte.txt`  
ou  
`Bureau/monImage.png`  
ou  
`/home/bob/Bureau/maVideo.mp4`

Pour toutes les commandes, on peut afficher l'aide avec :

`NomDeMaCommande -h` ou  
`NomDeMaCommande --help` ou  
`man NomDeMaCommande`  
(touche **Q** pour quitter `man`)

**Cl** : Command Line Interface  
Interface en ligne de commande

**Tui** : Text User Interface  
Interface en mode text dans une ligne de commande

**Gui** : Graphic User Interface  
Interface graphique utilisateur

### NAVIGUER DANS LES RÉPERTOIRES

Afficher le répertoire courant  
`pwd`

Se placer dans le répertoire `rep`  
`cd rep`

Se placer dans le répertoire de l'utilisateur `~/`  
`cd ~`

Se placer dans le répertoire parent  
`cd ..`

Lister les fichiers du répertoire `rep`  
`ls rep`

ls avec les fichiers cachés  
`ls -a`

ls avec les droits d'accès et la taille  
`ls -l`

### ACTIONS SUR LES FICHIERS/DOSSIERS

Déplace le fichier `source` vers `cible`  
`mv source cible`

Copier le fichier `source` vers `cible`  
`cp source cible`

Copier le répertoire `source` vers `cible`  
`cp -R source cible`

Créer un lien fort de `source` vers `lien`  
`ln source lien`

Créer un lien symbolique de `source` vers `lien`  
`ln -s source lien`

Créer le fichier `file` ou met à jour sa date de modification  
`touch file`

Créer un répertoire `rep`  
`mkdir rep`

`mkdir` avec création du `rep` parent si nécessaire  
`mkdir -p rep/rep2`

Supprimer le fichier `file`  
`rm file`

Supprimer le fichier `file` protégé en écriture  
`rm -f file`

Supprimer un répertoire vide  
`rmdir rep`

Supprimer un répertoire (vide ou plein)  
`rm -R rep`

Afficher la taille de `file` ou du répertoire `rep`  
`du -h file ou rep`

Afficher la taille d'un répertoire en se limitant aux répertoires et trier la sortie par taille décroissante  
`du -hd 1 rep | sort -hr`

Afficher le contenu de l'arborescence des fichiers dans un répertoire  
`tree rep`  
(si il n'est pas encore installé :)  
`sudo apt install tree`

Afficher le type d'un fichier  
`file fichier`

Obtenir les informations d'un fichier  
`stat fichier`

Créer un lien symbolique  
`ln -s target_file linked_file`

Supprimer un lien symbolique  
`unlink linked_file`

Voir le contenu (la cible) d'un lien symbolique  
`readlink linked_file`

### MC (Tui)

GNU Midnight Commander (mc) est un gestionnaire de fichiers multiplate-forme en mode texte. L'interface principale se compose de deux « panneaux » qui affichent les fichiers présents par rapport à leur emplacement sur le disque.

`mc`  
si il n'est pas encore installé :  
`sudo apt install mc`

### LFM(Tui)

Last File Manager est un puissant gestionnaire de fichiers pour la console. Il possède une interface Cli.

`lfm`  
si il n'est pas encore installé :  
`sudo apt install lfm`

## AFFICHER/COMPARER LES FICHIERS

Compte le nombre de lignes, de mots, d'octets de fichier  
`wc fichier`

Concatène/affiche les fichiers  
`cat fichiers`

Affiche fichier page après page 'Espace'=page suivante, 'Entrée'=ligne suivante, 'u'=remonter  
`more fichier`

Affiche fichier avec une navigation au clavier  
`less fichier`

Affiche les x premières lignes de fichier  
`head -n x fichier`

Affiche les x dernières lignes de fichier  
`tail -n x fichier`

Affiche la dernière ligne de fichier en temps réel  
`tail -f fichier`

Affiche les différences entre deux fichiers texte  
`diff fichier1 fichier2`

Affiche les différences au format patch  
`diff -u fichier1 fichier2`

Compare deux fichiers binaires  
`cmp fichier1 fichier2`

Compare deux fichiers, file1 à partir du nième octet, et file2 à partir du Nième  
`cmp fichier1 fichier2 n N`

## UTILISATEURS

Affiche le login de l'utilisateur  
`whoami`

Affiche les utilisateurs connectés  
`who`

Afficher les uid, gid et groupes de l'utilisateur  
`id`

Afficher les uid, gid et groupes de user (root only)  
`id $user`

Affiche les informations de user  
`finger $user`  
ex : `finger bob`  
si il n'est pas encore installé :  
`sudo apt install finger`

Afficher un message sur le terminal de user  
`write $user`

Afficher le nom de son terminal  
`tty`

Passer en mode administrateur, super-utilisateur  
`su` (ou) `sudo`

Changer le mot de passe de l'utilisateur courant  
`passwd`

Ajouter un utilisateur  
`adduser`

Supprimer un utilisateur  
`deluser`

Ajouter un groupe  
`addgroup`

Supprimer un groupe  
`delgroup`

## PROCESSUS

Afficher les processus de l'utilisateur  
`ps`

Afficher tous les processus  
`ps ax`

Afficher tous les processus et leur utilisateur  
`ps aux`

Afficher les processus dans une arborescence  
`pstree`

Afficher un tableau des processus gourmands  
`top`

Tuer un processus en utilisant son pid  
`kill signal pid`

Tuer un processus en utilisant le nom du programme  
`pkill signal nom`

### SIGNAUX UTILISÉS PAR KILL/PKILL

- 1 (HUP) recharger le fichier de configuration du processus
- 2 (INT) interrompre le processus
- 3 (QUIT) quitter le processus
- 9 (KILL) tuer le processus (à éviter, tenter -15 avant)
- 15 (TERM) terminer le processus proprement
- 18 (STOP) geler le processus
- 20 (CONT) reprendre l'exécution d'un processus gelé

Trouver la liste des processus qui utilise un répertoire (en root)  
`lsdf -x +D /repertoire`

## SHELL

### MODIFIER L'INTERPRÉTEUR :

Interpréteur shell actuellement utilisé : (en général bash)

`ps -p $$`

Lister les interpréteur sur la machine :

`cat /etc/shells`

Changer d'interpréteur par default : (et relancer le terminal)

`chsh -s /bin/bash`

## NOTES :

.....

.....

.....

.....

.....

.....

.....

.....

## QUELQUES "HELLO WORLD"

### PROCESSING-JAVA | HELLO WORLD

Processing-java est un utilitaire permettant de compiler un dossier processing en programme exécutable. Il se trouve dans le dossier processing.

Dans le dossier `HelloWorld`, pour le fichier `HelloWorld.pde`:

```
void setup(){
  size(800,800);
  println("Hello World");
}
void draw(){
}
```

Compilation depuis la console:

```
processing-java --sketch="dossier" --run
```

Dans le dossier en cours on peut écrire :

```
processing-java --sketch="$PWD" --run
```

### JAVA | HELLO WORLD

Dans un fichier `HelloWorld.java` :

```
public class HelloWorld {
  public static void main(String [] args){
    System.out.println("Hello World");
  }
}
```

Compilation du code :

```
javac *.java
```

Exécution du code :

```
java HelloWorld
```

### PYTHON | HELLO WORLD

Dans un fichier `HelloWorld.py` :

```
print('Hello world!')
```

Exécution du code :

```
python3 HelloWorld.py
```

### C | HELLO WORLD

Dans un fichier `hello.c`

```
#include <stdio.h>
int main(int arc,char* arg)
{
  printf("hello, world\n");
  return 0;
}
```

Compilation depuis la console :

```
gcc hello.c
qui produit un a.out, et pour executer :
./a.out
```

### BASH | HELLO WORLD

Dans un fichier `hello.sh`

```
#!/bin/bash
echo 'Hello, world!'
```

Et pour executer :

```
./hello.sh
```

### NODE | HELLO WORLD

Dans un fichier `helloNode.js`

```
const { createServer } = require('http');
//Creation du serveur
const server = createServer(
  (request, response) => {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end('Hello World\n');
  }
);
server.listen(3000, () => console.log("Adresse du serveur : http://localhost:3000"));
```

Executer le code avec :

```
node helloNode.js
```

et ouvrir un page web à <http://localhost:8080/>

Trouver la liste des fichiers ouverts par un processus (en root)

```
ls -p -p PID
```

### HTOP (TUI)

VISUALISER ET GÉRER L'ENSEMBLE DES PROCESSUS

```
htop
si il n'est pas encore installé :
sudo apt install htop
```

### MATÉRIEL

Liste les périphériques de type USB connectés

```
lsusb
```

Liste les périphériques de type PCI connectés

```
lspci
```

Affiche les informations processeur

```
cat /proc/cpuinfo
```

Affiche les partitions montées

```
cat /proc/partitions
```

Afficher le modèle de sa carte graphique :

```
lspci | egrep "3D|Display|VGA"
```

Afficher le modèle de sa carte Wi-fi :

```
lspci | grep -i "net" | cut -d: -f3
```

Afficher le modèle de sa carte son :

```
lspci | grep -i audio | cut -d: -f3
```

Affiche les message du noyau linux. Très pratique pour voir les périphériques connecté ou déconnecté.

```
dmesg
```

dmesg avec option de suivie continue

```
dmesg -w
```

### RÉSEAU

Affiche le nom d'hôte de la machine

```
hostname
```

Envoie un ping à une machine

```
ping machine
ex : ping 198.168.0.10
```

Fait un trace route vers machine

```
traceroute machine
```

Liste les processus utilisant le réseau

```
netstat
```

Netstat + affichage des processus serveurs

```
netstat -a
```

Liste détaillée de l'usage des fichiers et du réseau

```
ls -l
```

Affiche la config des interfaces réseaux

```
ifconfig
Configure une interface réseau
ifconfig interface IP masque
```

Affiche la table de routage

```
route
```

IP publique

```
curl ifconfig.me
```

ou

```
curl -s ipecho.net/plain ;echo
```

### NMCLI (CLI)

SE CONNECTER EN MODE CLI

Affiche le status des connections

```
nmcli devices status
```

Affiche les réseaux wifi dans le terminal

```
nmcli device wifi list
```

Se connecte à un réseau wifi

```
nmcli device wifi connect $SSID password $pass
```

### NMTUI (TUI)

SE CONNECTER EN MODE TUI

Gestion du réseau en mode TUI (Text User Interface)

```
nmtui
```

### WAVEMON (TUI)

VISUALISER L'ENSEMBLE DES RESEAUX WIFI

Analyse du réseau wifi

```
sudo wavemon
si il n'est pas encore installé :
sudo apt install wavemon
```

### TELNET(CLI)

Telnet est un protocole utilisé sur tout réseau TCP/IP, permettant de communiquer avec un serveur distant en échangeant des lignes de texte et en recevant des réponses également sous forme de texte. [!] Il n'est pas sécurisé.... (voir ssh)

```
sudo apt install telnet
```

### SOCAT(CLI)

Socat est un utilitaire en ligne de commande qui établit deux flux d'octets bidirectionnels et transfère des données entre eux. Par exemple : rediriger un fichier texte vers un port local.

```
sudo apt install socat
```

### NETCAT(CLI)

En informatique, netcat, également abrégé nc, est un utilitaire permettant d'ouvrir des connexions réseau, que ce soit UDP ou TCP. Il est conçu pour être incorporé aisément dans un large éventail d'applications.

```
sudo apt install netcat
```

### NMAP(CLI)

Pour scanner les ports d'un ordinateur distant, Nmap utilise diverses techniques d'analyse qui s'appuient sur des protocoles tels que TCP, IP, UDP ou ICMP. De même, il se fonde sur les réponses qu'il obtient à des requêtes particulières pour obtenir une empreinte de la pile IP, souvent propre au système qui l'utilise.

```
sudo apt install nmap
```

### ZENMAP

Version GUI de nMap

```
sudo apt install zenmap
```

### WIRESHARK(GUI)

OUTIL DE DIAGNOSTIQUE RESEAU

Version de la distribution :

```
sudo apt install wireshark
```

Dernière version disponible :

```
sudo add-apt-repository ppa:wireshark-dev/stable
sudo apt-get update
```

## CALCULATRICE

lancer bc en ligne de commande (calcule entier simple)  
**bc**

lancer bc en ligne de commande (avec le module scientifique)  
**bc -l**

s(x)	sinus
c(x)	cosinus
a(x)	arctangente
l(x)	logarithme naturel
e(x)	exponentielle
sqrt(x)	racine carrée
x^y	puissance
x%y	reste de la division euclidienne

Calculer une opération avec bc  
**bc -l <<< "1+sqrt(5)"**

Calculer le nombre Pi a 10 décimal  
**pi=\$(bc -l <<< "scale=10; 4\*a(1)")  
&& echo "Pi est égal a \$pi"**

## MONITORING

Afficher la liste des processus en arrière plan  
**jobs**

Place au premier plan le processus **pid**  
**fg pid**

Place en arrière plan le processus **pid**  
**bg pid**

Redéfinir le niveau de priorité du processus **pid**  
**renice +1 pid**

Tuer le job [**n**]  
**kill %n**

## RECHERCHE

Recherche sur un nom correspond au **motif**  
**locate motif**

Mettre à jour la base de données de **locate**  
**updatedb**

Recherche les fichiers dans **chemin** avec **option**  
**find chemin options**

Recherche sur le nom du fichier  
**find -name motif**

Recherche par type où f=fichier,d=répertoire,l=lien  
**find -type f/d/l**

Exécute la commande **cmd** à tous les fichiers trouvés  
**find -exec cmd**

Rechercher un fichier d'après son nom dans le répertoire **dir**  
**find dir -type f -name filename**

## EXEMPLE :

Trouver toutes les images avec l'extension png dans le dossier 'Images' de l'utilisateur et les copier dans le dossier tmp ( '{} ' représente les fichiers trouvés).

```
find $HOME/Images -name "*.png"  
-exec cp {} $HOME/tmp/ \;
```

## ARCHIVES

**.TAR.BZ2, .TBZ2**

Compresser :  
**tar -cvjf archive.tar.bz2 repertoire**  
Décompresser :  
**tar xvjf**

**.TAR.GZ, .TGZ**

Compresser :  
**tar -cvzf archive.tar.gz repertoire**  
Décompresser :  
**tar xvzf**

**.bz2**

Compresser :  
**bzip2 fichiers**  
Décompresser :  
**bunzip2**

**.RAR**

Compresser :  
**rar a archive.rar repertoire**  
Décompresser :  
**unrar x**

**.GZ**

Compresser :  
**gzip fichiers**  
Décompresser :  
**gunzip**

**.TAR**

Compresser :  
**tar -cvf archive.tar fichiers**  
Décompresser :  
**tar xvf**

**.ZIP**

Compresser :  
**zip -r archive.zip fichiers**  
Décompresser :  
**unzip**

**.Z**

Compresser :  
**compress fichiers**  
Décompresser :  
**uncompress**

**.7z**

Compresser :  
**7z a fichiers**  
Décompresser :  
**7z x**

**.XZ**

Compresser :  
**xz -z repertoire**  
Décompresser :  
**unxz**

## PARLATYPE [LECTEUR AUDIO]

Parlatype est un lecteur audio minimal pour la transcription manuelle de la parole. Il lit les sources audio pour les transcrire dans votre application texte préférée.

<https://gkarsay.github.io/parlatype/>

Version de la distribution :  
**sudo add-apt-repository ppa:gabor-karsay/  
parlatype  
sudo apt-get update  
sudo apt install parlatype**

**SAYONARA-PLAYER**

<https://sayonara-player.com/>

**SOUND CONVERTER [CONVERTISSEUR AUDIO]**

Logiciel pour convertir des fichiers sons simplement.  
**apt install soundconverter**

**SOUND KONVERTER [CONVERTISSEUR AUDIO]**

Logiciel pour convertir des fichiers sons simplement.  
**apt install soundkonverter**

**SOUND JUICER [UTILITAIRE AUDIO]**

Logiciel pour ripper un CD audio  
**apt install sound-juicer**

**VcV RACK [SYNTHÉTISEUR MODULAIRE]**

Synthétiseur modulaire virtuel  
<https://vcvrack.com/>

**JITSI [COMMUNICATION]**

Skype en libre pour linux  
<https://jitsi.org/downloads/>

**SIMPLE SCREEN CAPTURE [CAPTURE D'ÉCRAN EN VIDÉO]**

SimpleScreenRecorder peut capturer un enregistrement audio-vidéo de tout ou partie de l'écran de l'ordinateur ou enregistrer directement des applications OpenGL telles que des jeux vidéo.  
<https://www.maartenbaert.be/simplescreenrecorder/>

Version de la distribution :  
**sudo apt-get install simplescreenrecorder**

Dernière version :  
**sudo add-apt-repository ppa:maarten-baert/  
simplescreenrecorder  
sudo apt-get update  
sudo apt-get install simplescreenrecorder**

**MAPMAP [MAPPING]**

MapMap est un logiciel de mapping vidéo gratuit.  
<https://mapmapteam.github.io/>

**GLMIXER [MIXAGE VIDÉO, VJING]**

GLMixer réalise le mixage graphique en temps réel de plusieurs clips vidéo et de graphiques générés par ordinateur. Déposez les fichiers vidéo dans l'espace de travail de mixage et placez-les dans une zone circulaire pour changer leur opacité ; si vous sélectionnez deux vidéos, les déplacer ensemble effectue une transition de fondu. Ce principe se généralise à un grand nombre de vidéos. L'interaction directe avec la vidéo permet d'être rapide et réactif, et de les déplacer et déformer à l'écran.

<https://sourceforge.net/projects/glmixer/files/Linux/>

## ALTERNATIVE [UTILITAIRE SYSTÈME]

Outil graphique pour configurer les alternatives, pour aider l'administrateur système à choisir quel programme devrait offrir un service donné.

**sudo apt install galternatives**



## THONNY [IDE PYTHON]

Ide de programmation simple dédié à Python  
<https://thonny.org/>

Version de la distribution :

**sudo apt install thonny**

## BLUEJ [IDE JAVA]

Ide de programmation simple dédié à Java

Allez sur le site et choisissez l'option .deb

<https://www.bluej.org/>

## CODE VISUAL STUDIO [IDE MULTILANGAGE]

Ide de programmation polyvalent et supportant le java, javascript, C, C++, C#, python, html,css,sass,... bref presque tout.

Allez sur le site et choisissez l'option .deb

<https://code.visualstudio.com/>

## GEDIT [TEXT]

Editeur de texte simple et efficace, supporte la colorisation syntaxique de code

**sudo apt install gedit**

## GHOSTWRITER [TEXT]

Editeur markdown

<http://wereturtle.github.io/ghostwriter/>

## REMARKABLE [TEXT]

Editeur markdown

<https://remarkableapp.github.io/linux.html>

## RETEXT [TEXT]

Editeur markdown

<https://github.com/retext-project/retext>

## VLC [LECTEUR VIDÉO]

Célèbre lecteur vidéo de l'école Central Paris

<https://www.videolan.org/>

Version de la distribution :

**sudo apt install vlc**

## GIMAGEREADER [LECTEUR OCR]

gImageReader est un lecteur OCR, il permet de reconnaître du texte sur des documents. Très pratique pour les documents scannés.

<https://github.com/manisandro/gImageReader>

Version de la distribution :

**sudo apt install gimagereader**

## LIBREOFFICE [BUREAUTIQUE]

Suite bureautique complète et open source

<https://fr.libreoffice.org/>

Version de la distribution :

**sudo apt install libreoffice**

## PSSENSOR [UTILITAIRE]

Utilitaire permettant de surveiller son ordinateur, température, activité.

**apt install psensor**

## BALENETCHER [UTILITAIRE]

Utilitaire permettant de faire des clefs de démarrage simplement, parfait pour faire des images RaspberryPi.

Télécharger l'AppImage sur :

<https://www.balena.io/etcher/>

## MUSIC FOR PROGRAMMING [MUSIQUES]

Pour rester zen quand on est concentré devant son écran

<http://musicforprogramming.net/>

## KMPLOT [MATHÉMATIQUE]

Visualisateur de courbe mathématique

**sudo apt install kmpplot**

## ARDOUR [COMPOSITEUR MUSICALE]

Ardour est une station audio-numérique libre souvent présentée comme une alternative libre au logiciel Pro tools.

Version de la distribution :

**apt install ardour**

Dernière version :

<https://community.ardour.org/download>

## PUREDATA [SYNTHÈSE MUSICALE]

Pure Data (en abrégé pd) est un logiciel de programmation graphique pour la création musicale et multimédia en temps réel.

Version de la distribution :

**sudo apt install pd**

## AUDACITY [TRAITEMENT DE SIGNAL AUDIO]

Audacity est un logiciel d'enregistrement de son numérique et d'édition de sources audio numériques sous différents formats (mp3, Wave, AIFF, Flac, Ogg...).

Version de la distribution :

**sudo apt install audacity**

## LMMS [COMPOSITEUR MUSICALE]

Linux Multi Media Studio, ou LMMS, est une station audio numérique, séquenceur, synthétiseur et compositeur pour GNU/Linux, Windows et MacOS.

Télécharger l'AppImage sur :

<https://lms.io/download#linux>

## REAPER

Alternative à Pro-Tools de première classe. Fonctionne parfaitement avec Wine ET existe en version native depuis peu. Ce n'est pas un logiciel libre, mais il fonctionne parfaitement et complètement même si on n'achète pas la licence.

<https://www.reaper.fm/download.php>

## KERNEL

Version du noyau Linux utilisé, son nom, la version du compilateur utilisé :

**cat /proc/version**

Version du kernel :

**uname -r**

Liste les noyaux installés sur votre machine

**dpkg -l | egrep "linux-(header|image)"**

## SON, MUSIQUE ET ENREGISTREMENT

Donne des informations sur les cartes sons

**amixer**

Permet de gérer les niveaux des différentes sorties de la carte son

**alsamixer**

Permet de gérer/sélectionner/dispacher les niveaux et sorties des différentes sources/sorties audios (Tui)

**pulsemixer**

**(apt install pulsemixer** , s'il n'est pas présent)

Permet de gérer les niveaux des différentes sorties audios (Gui)

**pavucontrol**

**(apt install pavucontrol** , s'il n'est pas présent)

Lire un fichier audio simple (.wav,...)

**aplay file**

Lire un fichier audio **file.mp3**

**mpg321 file.mp3**

**(apt install mpg321** , s'il n'est pas présent)

Lire un fichier audio **file.mp3** avec plus d'information

**mpg321 -v file.mp3**

## MOC (TUI)

PUISSANT LECTEUR MUSICAL DE DOSSIER/FICHER

Lecteur et exploreur de bibliothèque musical

**mocp**

**(apt install moc** , s'il n'est pas présent)

## CORRECTIONS ORTHOGRAPHIQUE ET GRAMMATICALE

Vérifier l'orthographe d'un fichier écrit en français

**aspell check --lang=fr infile**

## EXPORTER LE CONTENU, CHANGER DE FORMAT

Exporter un fichier texte en PDF (HTML, DOCX...)

**pandoc infile -o outfile.pdf**

Voir un fichier Markdown en rich text dans le terminal

**pandoc file.md | w3m -T text/html**

## MANIPULATION DE PDF

Afficher les informations d'un fichier PDF

**pdftk file.pdf**

Convertir un PDF en fichier texte

**pdftotext infile.pdf outfile.txt**

Renommer des fichiers (Debian)

**rename 's/exp1/exp2/' xxxexp1xx**

Renommer un fichier en \*.old

**mv filename{.,.old}**

Composer un PDF à partir de plusieurs

**pdftk A=file1.pdf B=file2.pdf cat A1-3 B2 A4 output outfile.pdf**

Supprimer des pages d'un PDF (ici 7,9,14)

**pdftk infile.pdf cat '~7~9~14' output outfile.pdf**

Inverser les pages d'un PDF

**pdftk infile.pdf cat end-1 output outfile.pdf**

Composer un PDF livret en 2 pages par feuilles

**pdftobook --short-edge --paper a4paper**

**--outfile outfile.pdf infile.pdf**

Composer un PDF avec 4 pages par feuilles

**pdfnup --nup 2x2 --paper a4paper**

**--landscape --outfile outfile.pdf infile.pdf**

## CRON

Crontab est un outil qui permet de lancer des applications de façon régulière, pratique pour un serveur pour y lancer des scripts de sauvegardes, etc.

Pour afficher le contenu du fichier crontab :

**crontab -l**

Pour supprimer toutes les actions du fichier crontab :

**crontab -r**

Pour éditer les actions du fichier crontab :

**crontab -e**

DANS LA COMMANDE **crontab -e**, FORME CANONIQUE :

**minute heure jour DansLeMois Mois JourDansLaSemaine user Command**

Exécution tous les jours à 22h00 d'une commande action.sh :

**00 22 \* \* \* /home/bob/bin/action.sh**

Exécution d'une commande action.sh toutes les 6 heures :

**00 \*/6 \* \* \* /home/bob/bin/action.sh**

Exécution d'une commande toute les heures :

**00 \*/1 \* \* \* /home/bob/bin/action.sh**

Exécution d'une commande toutes les minutes que les lundis :

**\* \* \* \* 1 /home/bob/bin/action.sh**

Exécuter chaque jour, de chaque mois à 2:15 :

**15 02 \* \* \* /home/bob/bin/action.sh**

Exécution d'une commande toutes les minutes :

**\* \* \* \* \* /home/bob/bin/action.sh**

Exécution d'une commande au démarrage :

**@reboot /home/bob/bin/action.sh**

## ÉTAT DE CRON

Voir l'état de cron :

**sudo service cron status**

Démarrer le service cron :

**sudo service cron start**

Arrêter le service cron :

**sudo service cron stop**

## HISTORIQUE DES COMMANDES

Afficher l'historique des commandes  
`history`

Executer la dernière commande  
`!!`

Executer la commande numérotée n  
`!  
n`

## EXÉCUTIONS SÉQUENTIELLES OU CONDITIONNELLES

Exécuter de façon séquentielle deux commandes  
`command1 ; command2`

Exécuter de façon parallèle deux commandes  
`command1 & command2`

Exécuter la command2 que si la command1 a réussie  
`command1 && command2`

Exécuter la command2 que si la command1 a échoué  
`command1 || command2`

## REDIRECTION DE RESULTAT

Rediriger la sortie standard de command vers le fichier outfile  
`command > outfile`

Rediriger la sortie standard de la command vers le fichier outfile en fin de fichier  
`command >> outfile`

Rediriger l'entrée standard de la command vers le fichier infile  
`command < infile`

Rediriger l'entrée standard de command2 vers la sortie standard de la command1  
`command2 < (command1)`

Rediriger la sortie d'erreur standard de command vers le fichier outfile  
`command 2> outfile`

Rediriger l'erreur standard de command au même endroit que la sortie standard  
`command 2>&1`

Rediriger la sortie standard de command au même endroit que la l'erreur standard  
`command 1>&2`

## TUBES (PIPELINE)

Exécuter la command2 avec le résultat de la command1  
`command1 | command2`

Scinder les éléments provenant du tube en différentes lignes  
`... | xargs -n 1 echo`

Concaténer les lignes provenant du tube en une ligne  
`... | xarg echo`

xtraire du tube les lignes contenant expression  
`... | grep expresion`

xtraire du tube les lignes ne contenant pas l'expression  
`... | grep -v expresion`  
Lancer une commande de manière répétitive avec chaque élément provenant du pipe comme paramètre  
`... | xargs -n 1 command`

Donner une chaîne de caractères en argument d'une commande  
`command <<< string`  
ou (équivalent mais moins bien)  
`echo string | command`

Écrire dans un fichier et en sortie standard stdout, par exemple pour continuer un long pipe  
`command | tee outfile | ...`

Éditer le résultat d'une commande avec Vim  
`command | vim -`

## MANIPULATION DE FICHIER

### RECHERCHER ET REMPLACER

Trouver une expression dans un fichier  
`grep 'expression' infile`

Trouver une expression dans un fichier avec les numéros de lignes (-n)  
`grep -n 'expression' infile`

### DÉCOUPER, ASSEMBLER

Découper un fichier texte par blocs de 100 lignes  
`split -l 100 -d infile`

Découper un fichier texte sur un motif  
`csplit infile '/\section/' '{*}' -f 'prefix_'`

Assembler une collection de fichiers texte de noms x00, x01  
`cat x* > outfile`

Concaténer des fichiers (verticalement)  
`cat infile1 infile2 ...`

Concaténer les colonnes de fichiers (horizontalement)  
`paste infile1 infile2`

### REMETTRE EN FORME

Limiter la longueur des lignes à 72 caractères  
`fold -sw 72 infile > outfile`

Remettre en forme un fichier texte en joignant les lignes successives  
`fmt infile > outfile`

### TRIER DES LIGNES : sort

`sort -n` Tri au format numérique  
`sort -h` Tri au format "human readable"  
`sort -M` Tri par mois  
`sort -k2` Tri effectué sur le second champ de la ligne

Exemple :  
`sort monFichier.txt`

## FONTMANAGER [TYPO]

Gestionnaire de typographie, permet de visualiser, trier, classer les typographies de son système.

Version de la distribution :  
`apt install font-manager`

Dernière version :  
`sudo add-apt-repository ppa:font-manager/staging`  
`sudo apt-get update`  
`sudo apt-get install font-manager`

## FONTFORGE [TYPO]

Editeur typographique historique et puissant sous linux  
`apt install fontforge`

## BIRDFONT [TYPO]

Editeur typographique simple

Version de la distribution :  
`apt install birdfont`

Dernière version :  
<https://birdfont.org/>

## KRITA [IMAGE]

Logiciels libres pour les Concept Art, la peinture numérique et les illustrateurs.

Version de la distribution :  
`sudo apt install krita`

## QSTOPMOTION [ANIMATION]

Logiciel d'animation stop motion  
<https://www.qstopmotion.org>

Version de la distribution :  
`sudo apt install qstopmotion`

## SYNFIG [ANIMATION]

Logiciel d'animation vectoriel  
<https://www.synfig.org/>

Version de la distribution :  
`sudo apt install synfig`

## PENCIL2D [ANIMATION]

Logiciel pour dessin animé  
<https://www.pencil2d.org/>

Version de la distribution :  
`sudo apt install pencil2d`

## OPENTOONZ [ANIMATION]

Le logiciel d'animation utilisé par le studio Ghibli.  
<https://opentoonz.github.io/>

Version de la flatpack :  
`flatpak install flathub io.github.OpenToonz`

## INSTALLER UNE IMPRIMANTE PDF [GRAPHIQUE]

`sudo apt-get install printer-driver-cups-pdf`

## HANDBRAKE [VIDÉO]

Logiciel de conversion vidéo simple et puissant  
<https://handbrake.fr/>

Version de la distribution :  
`sudo apt install handbrake`

Dernière version disponible :  
`sudo add-apt-repository ppa:stebbins/handbrake-releases`  
`sudo apt-get update`  
`sudo apt install handbrake-gtk handbrake-cli`

## CONVERSEEN [VIDÉO]

Logiciel de traitement d'image part lot.  
`apt install converseen`

## MYSTIQ VIDEO CONVERTER [CONVERSION VIDÉO]

Convertisseur vidéo bien pratique  
<https://mystiqapp.com/>  
Section download et choisir Applmage (le plus simple)

## PROCESSING [GRAPHIC API]

Ide de programmation simple dédié aux applications graphiques  
<https://www.processing.org/>

## ARDUINO [ELECTRONIQUE API]

Ide de programmation simple pour carte électronique  
<https://www.arduino.org/>

Version de la distribution :  
`sudo apt install arduino`

## LOVE2D [MOTEUR DE JEU]

Moteur de jeu vidéo 2D avec compilé avec amour, scriptable en LUA.  
<https://love2d.org/>

## FRITZING [DESSIN DE CARTE ÉLECTRONIQUE]

Logiciel simple de conception de carte électronique  
<https://fritzing.org/home/>

Version de la distribution :  
`sudo apt install fritzing`

## IANNIX [CONTROLEUR OSC]

Editeur / Contrôleur OSC basé sur la philosophie de Iannis Xenakis, fonctionne avec processing, etc... Créée et maintenue par l'IRCAM  
<https://www.iannix.org/fr/download-iannix/>

## GODOT [MOTEUR DE JEU]

Moteur de jeu vidéo complet et très prometteur  
<https://godotengine.org>

Dernière version disponible :  
<https://godotengine.org/download/linux>

## UNITY [MOTEUR DE JEU]

Moteur de jeu le plus utilisé actuellement, se code en C#. (C sharp)  
<https://unity3d.com/>

Choisir la version Unity Hub ici :  
<https://unity3d.com/fr/get-unity/download>

## LOGICIELS/PACKAGES UTILES SOUS : DEBIAN/UBUNTU/MINT/...

### INKSCAPE [GRAPHIQUE]

Logiciel de traitement vectoriel  
<https://inkscape.org/>

Version de la distribution :  
**apt install inkscape**

Version de la developement :  
**sudo add-apt-repository ppa:inkscape.dev/stable**  
**sudo apt-get update**  
**apt install inkscape**

### GIMP [GRAPHIQUE]

Logiciel de traitement d'image  
<https://www.gimp.org/>

Version de la distribution :  
**apt install gimp**

Version de la developement :  
**sudo add-apt-repository ppa:otto-kesselgulasch/gimp**  
**sudo apt-get update**  
**sudo apt-get install gimp**

### RAWTHERAPEE [PHOTO]

Logiciel de traitement de photo raw  
<https://rawtherapee.com/>

Version de la distribution :  
**apt install rawtherapee**

### DARKTABLE [PHOTO]

Logiciel de traitement de photo raw  
<https://darktable.fr/>  
<https://darktable.org/>

Version de la distribution :  
**apt install darktable**

### SCRIBUS [MISE EN PAGE]

Logiciel de mise en page  
<https://www.scribus.net/>

Version de la distribution :  
**apt install scribus**

Version en cours (dernière) :  
**sudo add-apt-repository ppa:scribus/ppa**  
**sudo apt-get update**  
**sudo apt-get install scribus-ng**

### OLIVE [VIDÉO]

Logiciel de motage vidéo  
<https://www.olivevideoeditor.org/>

Version en cours (dernière) :  
**sudo add-apt-repository ppa:olive-editor/olive-editor**  
**sudo apt-get update**  
**sudo apt-get install olive-editor**

### NATRON [EFFET SPÉCIAUX]

Logiciel de post production et effet spéciaux développer par l'ircam  
<https://natrongithub.github.io/>  
voir dans la section download

### KDENLIVE [VIDÉO]

Logiciel de motage vidéo  
<https://kdenlive.org/fr/>  
Section download et choisir ApplImage (le plus simple)

### OBS, OPEN BROADCASTER [VIDÉO BROADCAST]

OBS Studio, abrégé en OBS, est un logiciel libre et open source de capture d'écran et de streaming pour Microsoft Windows, MacOS et Linux.

Version de la distribution :  
**apt install obs-studio**

Dernière version sur :  
<https://obsproject.com/>

### BLENDER [3D]

Logiciel d'image et de rendu 3D professionnel  
<https://www.blender.org/>

Version de la distribution :  
**sudo apt install blender**

### Version en cours (dernière) :

<https://www.blender.org/download/>

### MESHLAB [3D]

Laboratoire de traitement d'imagerie 3D

Version de la distribution :  
**sudo apt install meshlab**

Version en cours (dernière) :  
<https://www.blender.org/download/>

### MESHROOM [3D]

Logiciel de photogrammetrie, tranformer une série de photo en image 3D (Scanner 3D)  
<https://alicevision.org/>

Allez sur le site et choisir l'option .deb  
<https://alicevision.org/#meshroom>

### OPENSCEAD [3D]

Modeleur 3D en passant par le code

Version de la distribution :  
**sudo apt install openscad**

Dernère version (Choisir l'ApplImage) :  
<https://www.openscad.org/downloads.html>

### GLYPHRS [TYPO]

Editeur typographique  
<http://glyphrstudio.com/>

Section download et choisir ApplImage  
<https://github.com/glyphr-studio/Glyphr-Studio-Desktop>

### RECHERCHER DES DOUBLONS : **uniq**

**Uniq** s'opère que sur les lignes qui se suivent. Compare la ligne suivante à la ligne précédente.

Supprimer les lignes en doublons de **fic.txt**  
**uniq fic.txt**

Idem  
**uniq -u fic.txt**

N'afficher que les lignes en doublons  
**uniq -d fic.txt**

Indiquer le nombre d'occurrences de chaque ligne  
**uniq -c fic.txt**

N Ne pas comparer les **N** premiers caractères  
**uniq -s N fic.txt**

Ne pas comparer les **N** premiers champs  
**uniq -f N fic.txt**

### MODIFICATION DE CARACTÈRES : **tr**

Remplacer les caractères **abc** par **zyz**  
**tr abc zyz**

Supprimer les doublons de caractère  
**tr -s [caractère]**

Supprimer les espaces doublons  
**tr -s " "**

Supprimer 123  
**tr -d 123**

### DÉCOUPAGE DE FICHIER PAR COLONNES : **cut**

Affichage de la première colonnes du fichier.txt  
**cut -f fic.txt**

Choix d'un champ à afficher  
**cut -f m-n fic.txt**

Choix d'une plage à afficher  
**cut -f n,x,y fic.txt**

Choix des champs à afficher  
**cut -d "char" fic.txt**

Affichage du champs **n** puis de tous les champs  
**cut -f n- fic.txt**

Choix du caractère délimiteur de champs  
**cut -d "char" fic.txt**

Choix de plage de caractères au lieu de champs  
**cut -c n-m fic.txt**

Afficher la 3ème colonne d'un **fic.txt**, celles-ci étant délimitées par ;  
**cut -d ; -f3 fic.txt**

Exemple :  
Affichage des logins, des commentaires et des informations qui suivent depuis le fichier **/etc/passwd**  
**cut -d ":" -f 1,5 /etc/passwd**

Affichages des points de montage actifs  
**cut -d " " -f 2 /etc/mtab**

Affichage de syslog à partir du caractère 27  
**cut -c 27- /var/log/syslog**

### MODIFICATION DE TEXTE LIGNE PAR LIGNE : **sed**

Modifie sur la sortie standard le contenu de **fic.txt** selon l'action indiquée. On peut invoquer plusieurs fois l'option **-e** et passer plusieurs actions à la commande **sed**.  
**sed -e 'action' fic.txt**

Écrit directement les modifications dans **fic.txt**  
**sed -i 'action' fic.txt**

Récupère les actions dans le fichier **act.sed**  
**sed -f act.sed fic.txt**

Utilisation de **sed** en mode silencieux  
**sed -n 'action' fic.txt**

Exemple de syntaxe d'action :  
(pour **'action'** précédemment utilisé) :

Substitution de **rech** par **modif** (première occurrence)  
**s/rech/modif**

Substitution sur toutes les occurrences  
**s/rech/modif/g**

Suppression des lignes contenant le motif **recherche**  
**/recherche/d**

Suppression des lignes 3 à 7  
**3,7d**

Ajout de nouveau avant la ligne contenant **rech**  
**/rech/i"nouveau"**

Ajout de nouveau après la ligne contenant **rech**  
**/rech/a"nouveau"**

Remplace la ligne contenant **rech** par nouveau  
**/rech/c"nouveau"**

### Exemples complets :

Remplacer une expression par une autre dans un fichier  
**sed -i 's/expression1/expression2/g' infile**

Supprimer les lignes vides  
**sed -i '/^\$/d' infile**

Modifie le nom de **srvtmp** par **srv001** dans le fichier **/etc/hostname**  
**sed -i 's/srvtmp/srv001' /etc/hostname**

Supprime les lignes vides du fichiers **fic.txt**  
**sed -i '/^\$/d' fic.txt**

Affiche le contenu de **/etc/passwd** en utilisant un espace comme séparateur des divers champs  
**sed -e 's/:/:x/g' -e '/\\_/g' -e 's:/\//g' /etc/passwd**



## MANIPULATION D'IMAGES

La manipulation d'image pass en général par imageMagick, en voici quelques fonctions :

Pour l'installer :

```
sudo apt install imagemagick
```

### animate

Animer une séquence d'images sur n'importe quel serveur X.

### compare

Annoter mathématiquement et visuellement la différence entre une image et sa reconstruction.

### composite

Superposer une image sur une autre.

### conjure

Interpréter et exécuter des scripts écrits dans le Magick Scripting Language (MSL).

### convert

Convertir entre les formats d'image ainsi que redimensionner une image, flouter, recadrer, découper, tergiverser, dessiner, retourner, joindre, ré-échantillonner, et bien plus encore.

### display

Afficher une image ou une séquence d'images sur n'importe quel serveur X.

### identify

Décrire le format et les caractéristiques d'un ou plusieurs fichiers d'images.

### import

Enregistrer toute fenêtre visible sur un serveur X et la sortir sous forme de fichier image. Vous pouvez capturer une seule fenêtre, l'écran entier ou toute partie rectangulaire de l'écran.

### mogrify

Redimensionner une image, flouter, recadrer, découper, tergiverser, dessiner, retourner, joindre, ré-échantillonner, et bien plus encore. Mogrify écrase le fichier image original, alors que, convert écrit dans un fichier image différent.

### montage

Créer une image composite en combinant plusieurs images séparées. Les images sont superposées sur l'image composite, éventuellement agrémentées d'une bordure, d'un cadre, d'un nom d'image, etc.

### stream

Un outil léger pour diffuser en continu un ou plusieurs composants de pixels de l'image ou d'une partie de l'image vers les formats de stockage de votre choix. Il écrit les composants de pixels tels qu'ils sont lus à partir de l'image d'entrée, une rangée à la fois, ce qui rend le flux souhaitable lorsque vous travaillez avec des images de grande taille ou lorsque vous avez besoin de composants de pixels bruts.

Exemples :

### CHANGER DE FORMAT

Changer de format une image

```
convert image.jpg image.png
```

Convertir une image bitmap (PNG, JPEG) en format vectoriel (SVG)

```
convert infile.png bmp:- | potrace -s - -o outfile.svg
```

### REDIMENSIONNER

Redimensionner une image

```
convert image -resize "largeur"x"hauteur"  
newimagedname
```

## GIF & Co.

Faire un gif avec une collection d'images

```
convert -delay 20 -loop 0 img_*.gif output.gif
```

Extraire les images d'un GIF

```
convert -coalesce infile.gif img_%d.png
```

Faire un GIF avec les pages d'un PDF

```
convert -verbose -delay 50 -alpha off -background  
#FFFFFF -loop 0 -density 300 infile.pdf outfile.gif
```

## VIGNETTES

Convertir une page de PDF en PNG

```
convert -trim infile.pdf[0] -quality 100 -flatten  
-resize x300 outfile.png
```

## AUTRES MÉDIAS : VIDÉO/SO

### AUDIO

Convertir un mp3 en wav

```
mpg123 -w outfile.wav infile.mp3
```

### VIDÉOS

Convertir un avi en mov

```
ffmpeg -i input.avi -acodec libmp3lame -ab 192  
output.mov
```

Découper une vidéo à 24'53

```
ffmpeg -i video.mp4 -acodec copy -vcodec copy -ss  
0 -t 00:24:53 video1.mp4
```

(ou)

```
ffmpeg -i video.mp4 -acodec copy -vcodec copy -ss  
00:24:53 video2.mp4
```

## MEDIAINFO

Afficher toute les infos d'un fichier

```
mediainfo file.png
```

( Si il n'est pas installé : )

```
sudo apt install mediainfo
```

## SSH

Se connecter en SSH

```
ssh user@host
```

Envoyer un fichier par SSH

```
scp infile user@host:directory/.
```

ou en sens inverse

```
scp user@host:directory/infile .
```

Envoyer un dossier par ssh

```
scp -r dir user@host:directory/.
```

## RSYNC

Synchroniser 2 répertoires

```
rsync -a --stats --progress --delete $SRC $DST/
```

Synchroniser 2 répertoires par SSH

```
rsync -azv --delete -e ssh $SRC user@host:$DST/
```

Sauvegarde incrémentale

```
rsync -abz --stats --progress --delete --  
suffix="old" --backup-dir=$(basename $SRC)"_$  
(date +%Y-%m-%d)" $SRC $DST/
```

## INCRÉMENTER/DÉCRÉMENTER UNE VARIABLE

```
var=$((var+n)) Incrémenter la variable var de n  
((var=var+n)) Incrémenter la variable var de n  
((var+=n)) Incrémenter la variable var de n  
((var-=n)) Décrémenter la variable var de n  
((var++)) Incrémenter la variable var de 1  
((var--)) Décrémenter la variable var de 1
```

## FONCTIONS

Les fonctions permettent de placer des éléments de code en mémoire puis de les rappeler dans nos script.

Exemple :

```
#!/bin/bash  
#Déclarer une fonction  
nomFonction()  
{  
    command1 -opt arg  
    command2 .  
}  
#Appel de la fonction  
echo "Etape1"  
nomFonction #Appel de la fonction  
echo "Etape2"
```

Exemple avec argument et retour:

```
#!/bin/bash  
#Déclarer une fonction  
nomFonction(){  
    res=$((($1 + $2))  
    # $1 = argument 1, $2 argument 2  
    return $res  
}  
#Appel de la fonction  
echo "step 1"  
r=`nomFonction 1 2`  
#[!] au caractère d'exécution  
echo $r  
echo "step2"
```

## INCLUSION DE FICHIER

La commande source permet d'inclure un fichier bash à l'intérieure d'un autre afin d'utiliser les variables et fonctions de celui-ci.

```
source functions.sh
```

## RÉCUPÉRATION DU RESULTAT D'UNE COMMANDE

Permet de récupérer le résultat d'une commande.

```
$(command -opt)  
ou  
`command -o arg`
```

Permet de récupérer le résultat d'une commande dans une variable.

```
result=$(command -opt)
```

## DPKG/APT : INSTALLATION DE PACKAGE

Tout d'abord, **apt-get** est une sur-couche de dpkg. dpkg se charge de toute la gestion de l'installation/désinstallation des packages .deb (les packages). **apt-get** permet lui les téléchargements automatiques, la gestion des dépendances, etc...

Les installations/désinstallations doivent être réalisées en tant que root, les recherches peuvent être faites par n'importe quel utilisateur.

### EN TANT QUE ROOT :

Mise à jour de la base de donnée des logiciels/packages

```
sudo apt update
```

Installe le programme **programme** passé en argument.

```
sudo apt install programme
```

Supprime les fichiers du package passé en argument. Ne touche pas les fichiers de configuration

```
sudo apt remove package
```

Supprime les fichiers du package passé en argument ET les fichiers de configuration

```
sudo apt remove --purge package
```

Met à jour tout le système

```
sudo apt upgrade
```

Reconfigure le package

```
dpkg-reconfigure package
```

Donne le nom du package ayant installé le fichier /usr/bin/aplay

```
dpkg -S /usr/bin/aplay
```

Recompile le cache de apt-show-versions

```
apt-show-versions -i
```

### N'IMPORTE QUEL UTILISATEUR :

Affiche la liste des packages qui sont installés sur l'ordinateur [ii] ou en cours de désinstallation (il reste les fichiers de configuration [rc])

```
dpkg -l
```

Affiche la liste des fichiers installés par le package **package** installé sur l'ordinateur

```
dpkg -L package
```

Affiche la liste des packages comportant le **mot-clé** dans leur description

```
apt search mot-clé
```

Affiche les caractéristiques et la description du package

```
apt show package
```

Affiche les dépendances et les packages utilisant le package passé en paramètre

```
apt showpkg package
```

Packages disponibles pour mise à jour

```
apt-show-versions -u -b
```



## CARACTÈRES SPÉCIAUX

Les caractères spéciaux modifient le comportement du shell

- `\` Banalise le caractère précédent (Ex : `\$2` va afficher `$2` et non la valeur dans `$2`)
- `'...'` Empêche le shell d'interpréter les caractères entre apostrophes
- `"..."` Empêche le shell d'interpréter les caractères entre guillemets autre que `$` (dollars), `\` (antislash) et ``` (backquote)

## METACARACTÈRES

- `*` 0 à n caractères quelconques
- `?` 1 caractère quelconques
- `[...]` 1 caractère compris dans une liste
- `[...-...]` 1 caractère compris dans un intervalle
- `[^...]` 1 caractère autre que ceux dans la liste ...
- `[!...]` 1 caractère autre que ceux dans la liste ...

### Exemple :

- `*.txt` Tous les coms avec l'extension `.txt`
- `abcd|cCd` `abcd` ou `abCd`
- `ab?d` Tous les noms de 4 caractères commençant par `ab` et se terminant par `d`
- `*[aeiouy]*.???` Un nom qui contient une voyelle et se termine par une extension de 3 caractères
- `[SK][0-9]*` Un nom qui commence par `S` ou `K`, qui se termine par une lettre minuscule ou majuscule.
- `*.*[!at01]*` Un nom dont l'extension ne contient ni `a`, ni `t`, ni `0`, ni `1`

## EXPRESSIONS RÉGULIÈRES (REGEX)

Une expression régulière est une manière compacte de représenter une chaîne de caractères.

- `.` Tout caractère (sauf newline)
- `^` Début de ligne
- `$` Fin de ligne
- `.*` Capture tout les caractères
- `^$` Une ligne vide
- `+` 1 à n fois le caractère précédent
- `?` 0 ou 1 fois le caractère précédent
- `\s` Un espace
- `\<` Début de mot
- `\>` Fin de mot
- `\n` Fin de ligne
- `\r` Retour chariot
- `\t` Tabulation
- `\c` Control
- `\s, \S` Espace, ou pas
- `\d, \D` Nombre, ou pas
- `\w, \W` Mot, ou pas
- `\x, \O` Nombre hexadécimal, octal

## CLASSES DE CARACTÈRES

- `[::upper:]` Une lettre en majuscule
- `[::lower:]` Une lettre en minuscule
- `[::alpha:]` Une lettre
- `[::alnum:]` Une lettre ou un chiffre
- `[::digit:]` Un chiffre
- `[::xdigit:]` Un chiffre hexadécimal
- `[::punct:]` Un caractère de ponctuation
- `[::blank:]` Un espace ou tabulation
- `[::cntrl:]` Un caractère de contrôle
- `[::word:]` Une lettre, un chiffre ou underscore

## GROUPES, ENSEMBLES

- `(...)` Groupe
- `\n` n-ième groupe
- `[abc]` Un des caractères spécifiés
- `[a-dh]` Une lettre de `a` à `d` ou `h`
- `[0-57]` 0, 1, 2, 3, 4, 5 ou 7
- `[a-d5-8-]` `a`, `b`, `c`, `d`, 5, 6, 7, 8 ou -
- `[012^]` 0, 1, 2 ou `^`
- `[^abc]` Un caractère ne figurant pas dans la liste
- `[^0-9]` Tout sauf un chiffre
- `[^a-zA-Z]` Tout sauf une lettre

## QUANTIFICATEURS

- `r*` Aucune ou plusieurs instances de l'expression `r`
- `r+` Au moins une instance de l'expression `r`
- `r?` Aucune ou une instance de l'expression `r`
- `{n}` Exactement `n` fois l'élément précédent
- `{n,m}` Entre `n` et `m` fois l'élément précédent
- `{n,}` `n` fois ou plus l'élément précédent
- `{,m}` `m` fois au plus l'élément précédent
- `r1|r2` Instance de l'expression `r1` ou `r2`

## RÉFÉRENCES

- `&, \0` tout le motif
- `\n` le n-ième groupe (`n>=1`)
- `\r` couper la ligne en 2 à ce point
- `\u` le caractère suivant en majuscule
- `\l` le caractère suivant en minuscule

## AFFICHER LA TABLE DES CARACTÈRES ASCII

`man ascii`

## ARITHMÉTIQUE ENTIÈRE

L'arithmétique entière peut être invoquée avec la commande `let` ou en utilisant les doubles parenthèses préfixées de `$`. Le fonctionnement est le même. Les caractères spéciaux du shell n'ont pas besoin d'être protégés.

Effectue l'opération `nb + 8`  
`let "$nb + 8"`  
ou (idem)  
`(( $nb + 8 ))`

Affiche le résultat de `2 + 3`  
`echo $((2 + 3))`

## CALCULS ARRITHMÉTIQUES

- `+` Addition
- `-` Soustraction
- `*` Multiplication
- `/` Division
- `%` Modulo (reste d'une division)

## COMPARATEURS

- `==` Égalité
- `!=` Différent de
- `<` Inférieure
- `>` Supérieure
- `<=` Inférieure et égal
- `>=` Supérieure et égal

## TÉLÉCHARGEMENT

Télécharger un fichier  
`wget http://path.to.the/file`  
ou  
`curl http://path.to.the/file`

Télécharger un fichier et le renommer  
`wget http://path.to.the/file -O newname`

Reprendre un téléchargement avorté  
`wget -c http://path.to.the/file`

Télécharger un fichier après authentification HTTP  
`curl -u username:password URL`

Télécharger tous les PDF d'un site  
`wget -e robots=off -r -l1 --no-parent -A.pdf URL`

Lister les formats disponibles sur un site de streaming  
`youtube-dl -F URL`

Télécharger une vidéo sur un site de streaming en limitant la bande passante à 150 ko  
`youtube-dl -r 150K -f <format code> URL`

## PARTAGE DE FICHIERS (LOCAL ET TEMPORAIRE)

`python3 -m http.server 8000`  
Un serveur web qui sert le dossier courant sur le port 8000

## FTP

Lister le contenu d'un répertoire FTP  
`curl ftp://username:password@example.com`

Envoyer un fichier par FTP  
`lftp -u $FTPuser,$FTPPwd ftp://@$FTPserver -e "put outfile ; quit"`

Créer un dossier miroir par FTP  
`lftp -u FTPuser,FTPPwd ftp://@FTPserver -e "mirror -e -R localdir FTPdir ; quit"`

## NAVIGATEURS

Afficher la météo à Paris dans son terminal  
`curl wttr.in/paris,france`

ou à Lyon en français  
`curl fr.wttr.in/Lyon,france`

## W3M (CLI)

Faire une recherche sur le web en tapant directement dans le terminal

Recherche sur duckduckgo : "google"  
`w3m https://www.google.com/search?q=\" $1 \"`

Recherche sur duckduckgo : "ma recherche"  
`w3m https://duckduckgo.com/?q=\" $1 \"`

## IMPRESSION

Lister les imprimantes  
`lpstat -p`

ou en se limitant aux noms  
`lpstat -a | cut -f1 -d ' '`

Soumettre un travail d'impression à l'imprimante `printer`  
`lpr -P printer filename`

Limiter l'impression à certaines pages  
`lpr -o page-ranges=1-4,7,9-12 filename`

Imprimer un document en format paysage ajusté à un A4  
`lpr -o landscape -o fit-to-page -o media=A4 filename`

Imprimer 4 pages par feuille (2 pages/page et recto-verso)  
`lpr -o number-up=2 -o sides=two-sided-long-edge filename`

Préciser le nombre de copies `n` à imprimer  
`lpr -#n filename`

Voir la pile d'impression de l'imprimante `printer`  
`lpq -P printer`

Supprimer le travail d'impression `job-id`  
`lprm job-id`

Changer l'imprimante destinataire d'un travail `job-id`  
`/usr/sbin/lpmove job-id new-printer`

## ADMINISTRATION SYSTÈME

Lancer une commande en administrateur  
`sudo command`  
ou  
`su -c command`

Autoriser un utilisateur à utiliser le système de fenêtres X  
`sudo xhost local:user`

Arrêter le système (en root)  
`shutdown -h now`  
ou  
`su -c poweroff`

Afficher depuis combien de temps vous êtes connecté et la charge de la machine  
`uptime`

Afficher l'historique des redémarrages du système  
`last reboot`

## ÉTAT DU SYSTÈME

Nom de machine  
`hostname`

Distribution  
`lsb_release -idrc`

Afficher les informations de la distribution  
`lsb_release -a`

Informations du système  
`uname -a`

Noyau  
`uname -r`

## DÉMARRER UN SCRIPT

Commencer un script (On déclare l'interpréteur de script)  
`#!/bin/bash`

Commentaire dans un script  
`#`

Sort du shell  
`exit -n`

Lance un script sans création de sous processus  
`exec script`

Afficher des information dans le terminal  
`echo "Hello World"`  
ou pour une variable  
`echo $var`

Faire une pause dans le script  
`sleep n`

## VARIABLE

### CRÉER/SUPPRIMER DES VARIABLES

Crée la variable var avec comme valeur "Hello World"  
`var = "Hello World"`

Détruit la variable var  
`unset var`

Récupère le resultat utilisateur et le stoke dans var1 et var2  
`read var1 var2`

Affiche le message entre guillemet avant de récupérer la saisie de l'utilisateur dans la variable choix  
`read -p "Votre choix" choix`

### EXPORTER DES VARIABLES

Exporter une variable dans l'environnement, dans les sous-shell.  
`export var`  
ou en effectant une valeur  
`export var = "Hello"`

### VISUALISER LES VARIABLES

Voirle contenu dela variable  
`echo $var`  
ou  
`echo ${var}`

Visualiser toute les variables d'environnement  
`env`

Visualiser toute les variables d'environnement + celles du shell  
`set`

Visualier les variables et les fonctions  
`declare`

### QUELQUES VARIABLES D'ENVIRONEMENT COMMUNES

`$HOME` Chemin du répertoire personnel de l'utilisateur  
`$LOGNAME` Nom de l'utilisateur dans le shell actif  
`$HOSTNAME` Nom de l'hôte de la machine  
`$PWD` Emplacement actuel du shell  
`$PS1` Définition de l'affichage du prompt principal  
`$IFS` Séparateur de champs (espace, tabulation, saut de ligne)  
`$LANG` Format de langue et encode du terminal  
`$SHELL` Chemin du shell par défaut de l'utilisateur

## VARIABLES RÉSERVÉES

`$?`  Code retour de la fonction précédente  
 `$0`  Nom la commande, du script lancé  
 `$n`  Appel de l'argument numéroté n passé avec la commande, la fonction ou le script. ( `$1`  argument 1,  `$5`  argument 5, etc)  
 `$#`  Affiche le nombre d'arguments passés avec la commande, la fonction ou le script.  
 `@$`  Appel de l'ensemble des arguments sans séparateurs  
 `*$`  Appel de l'ensemble des arguments sans séparateurs  
 `$$`  Numéro du processus en cours  
 `$!`  Numéro du dernier processus lancé en arrière-plan

### GESTION AVANCÉE DES VARIABLES

Mettre une variable en lecture seul  
`declare -r var`

Redonner l'attribut écriture à la variable  
`declare +r var`

Créer var avec son contenu en minuscules  
`declare -l var = "Hello"`

Créer var avec son contenu en majuscules  
`declare -u var = "Hello"`

Créer var avec l'attribut entier 123  
`declare -i var = 123`

### VARIABLES ET TABLEAUX

Affecte le contenu "World" à l'index n de la variable  
`var[n] = "Hello"`

Déclare un tableau avec les valeurs 1, 2, 3 et 4  
`declare -a var=(val1 val2 val3 val4)`

Affiche le contenu "World" à l'index n de la variable  
`echo ${var[n]}`

### TABLEAUX ASSOCIATIFS

Déclaration d'un tableau où les variables sont désignés par un nom  
`declare -A colours`

`colours[black]=noir`  
`colours[red]=rouge`  
`colours[green]=vert`

### SUBSTITUTION DE VARIABLE

Si var n'est pas défini, afficher la valeur  
`${var:-valeur}`

Si l'argument correspondant à la variable réservée \$n n'est pas définie, utiliser valeur  
`${n:-valeur}`

Si var n'est pas défini, affecter puis afficher valeur  
`${var=valeur}`

## STRUCTURES DE TEST

Test standard, (attention au espace entre les crochets)  
`[ condition ]`

Test standard inversé  
`[ !condition ]`

## COMPARAISON

Entier :  
`[ int1 -eq int2 ]` Entier1 égal à entier2  
`[ int1 -ne int2 ]` Entier1 différent de entier2  
`[ int1 -gt int2 ]` Entier1 supérieure de entier2  
`[ int1 -ge int2 ]` Entier1 supérieure ou égal de entier2  
`[ int1 -lt int2 ]` Entier1 inférieure a entier2  
`[ int1 -le int2 ]` Entier1 inférieure ou égal de entier2

Chaîne de caractère :  
`[ int1 = int2 ]` Chaîne est égal de chaîne2  
`[ int1 != int2 ]` Chaîne est différente de chaîne2  
`[ int1 -ef int2 ]` Fichier1 est identique à fichier2

### CONDITION

`[ -a fic ]` Vrai si le fichier existe  
`[ -f fic ]` Vrai si fic est un fichier standard  
`[ -d fic ]` Vrai si fic est un repertoire  
`[ -r fic ]` Vrai si fic est accessible en lecture  
`[ -w fic ]` Vrai si fic est accessible en écriture  
`[ -x fic ]` Vrai si fic est executable  
`[ -n $var ]` Vrai si la variable existe  
`[ -z $var ]` Vrai si la variable n'existe ou a une valeur null

### LA STRUCTURE IF

Version 1  
`if [ condition ] ; then`  
    commande  
    commande  
`fi`

Version 2  
`if [ condition ] ;`  
    then  
        commande  
        commande  
`fi`

Imbrication avec elif (si, sinon si)  
`if [[ condition ]]; then`  
    commande  
`elif`  
    commande  
`fi`

### LA STRUCTURE CASE

`case $var in`  
    expression1)  
        commande  
    ;;  
    expression2)  
        commande  
    ;;  
    \*)  
        commande  
    ;;  
`esac`

### Exemple:

```
case $reponse in
  [0-9])
    echo '$reponse est un chiffre'
    ;;
  O|o|Y|y)
    echo 'O,o,Y ou y'
    ;;
  *)
    echo 'Autre réponse'
    exit 1
  ;;
esac
```

### BOUCLES : WHILE/UNTIL

La structure while répète une action enboucle  
`while [ condition ] ;`  
    do  
        commande  
    done

### Exemple avec while:

```
i=0
while [ $i -lt 10 ] ;
do
  echo "$i"
  ((i = i + 1))
done
```

### Exemple avec until:

```
i=0
until [ $i -eq 10 ] ;
do
  echo "$i"
  ((i = i + 1))
done
```

### BOUCLES : WHILE/READ

Lecture d'un fichier  
`while read ligne1 ligne2 ligne3`  
do  
    echo "Bonjour \$ligne1 \$ligne2"  
    echo "ligne3 : \$ligne3"  
done < fichierDeLignes.txt

### BOUCLE FOR

Boucler sur une liste de valeur  
`for var in valeur1 valeur2 valeur3`  
do  
    commande  
    echo \$var  
done

### Boucler sur le contenu d'un fichier

```
for var in $(cat fichier.txt)
do
  echo $var
done
```

### Boucler sur un nombre

```
for ((i=0 ; i<10 ; i=i+1))
do
  echo "Numero -> $i"
done
```