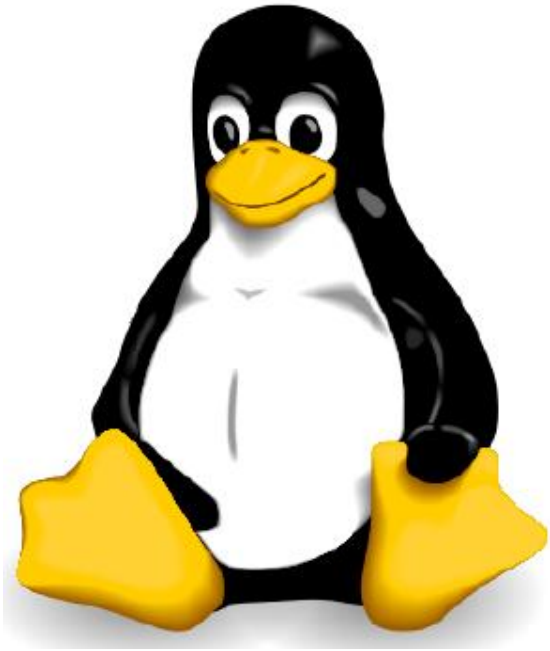
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté des Sciences - Département de Physique

Manuel
Initiation à Linux



Tux est un manchot, mascotte officielle du noyau Linux dessiné par Larry Ewing en 1996.

Cheballah Yamina

Table des matières

Avant-propos

Linux est une famille de systèmes d'exploitation open source de type Unix, fondé sur le noyau Linux, créé en août 1991 par Linus Torvalds à l'Université d'Helsinki, en Finlande. Linux a été développé avec l'aide de nombreux programmeurs et spécialistes Unix.

Ce manuel consiste principalement en un aperçu des commandes de base en console Linux. Il s'adresse à tous ceux qui désirent migrer vers Linux, plus particulièrement aux étudiants Master des filières techniques.

Je tiens à remercier Monsieur Zenia Hand, enseignant à l'Université de Tizi-Ouzou, pour m'avoir fait part de ses remarques lors de la lecture du manuscrit initial.

Cheballah Yamina
Tizi-Ouzou (Algérie), Mars 2023

Initiation à Linux

1. Introduction

Un système d'exploitation est un ensemble de programmes responsables de la liaison entre l'ordinateur et l'utilisateur.
Dans le monde de Microsoft, que tout le monde connaît, il existe un seul système d'exploitation pour les ordinateurs, c'est Windows, qui évolue simplement au cours du temps : Windows XP, Windows 7, Windows 8, Windows 10, Windows 11…, différentes versions apportant des modifications mineures, comme les éditions familiale ou professionnelle, qui ajoutent simplement quelques fonctionnalités et logiciels. Dans le monde de Linux, c'est un peu différent. Linux est un système d'exploitation libre et à code source[1] ouvert, c'est-à-dire que tout le mode peut l'utiliser pour créer son propre système d'exploitation, on dit que Linux est personnalisable alors que Windows, dont le code source est inaccessible, n'est pas personnalisable.

2. Quelle est la différence entre les systèmes d'exploitation Linux et Unix ?

Unix fait référence au système d'exploitation d'origine développé par AT&T (American Telephone and Telegraph) et Linux se réfère à la famille des distributions dérivées.

2.1. Système d'exploitation UNIX

- Le code source d'Unix n'est pas disponible pour le grand public.
- Unix utilise principalement les lignes de commandes.
- Unix a un environnement dépendant du matériel. Par conséquent, il ne peut être installé sur n'importe quelle machine.
- Unix est principalement utilisé dans les systèmes serveur, les ordinateurs centraux et les ordinateurs haut de gamme.
- Les différentes versions d'Unix sont IBM AIX (International Business Machines Corporation Advanced Interactive eXecutive), HP-UX (Hewlett Packard UniX), BSD (Berkeley Software Distribution), Sun Solaris (Sun Microsystems : un constructeur d'ordinateurs et un éditeur de logiciels Américain. Solaris est un système d'exploitation Unix développé à l'origine par Sun Microsystems), Iris (système d'exploitation Unix possédant des extensions BSD, de la société SGI (Silicon Graphics Inc : société Américaine qui construisait des stations de travail dédiées aux domaines de l'infographie de la 3D, du traitement de vidéo et du calcul haute performance))…etc.
- Unix est développé par l'entreprise de télécommunications AT&T.

2. 2. Système d'exploitation LINUX

- Le code source de Linux est disponible gratuitement pour les utilisateurs.
- Linux utilise principalement une interface graphique avec une ligne de commande optionnelle.
- Linux est très flexible et peut être installé sur la plupart des ordinateurs de bureau, les ordinateurs personnels, les téléphones mobiles,…etc.
- Les différentes distributions de Linux sont Debian, Ubuntu, Linux Mint, ReadHat, Manjaro Linux, Opensuse,…etc.
- L'installation de Linux est économique et ne nécessite pas de matériel spécifique.

[1] Le code source est le fichier qui a permis au développeur de programmer le logiciel, grâce à des lignes écrites dans un langage de programmation, qui sera compris et ensuite compilé en un programme. Certains codes sources sont mis gratuitement à la disposition de développeurs notamment dans le cadre des logiciels libres.

- Linux est développé par une communauté Linux active dans le monde entier.

Le système d'exploitation Linux est idéal pour les petites et moyennes entreprises. Il est également utilisé aujourd'hui dans les grandes entreprises où Unix était auparavant considéré comme la seule option.

3. Raisons de migration vers Linux

Partout dans le monde, des entreprises, des institutions scolaires ou universitaires et d'autres organisations abandonnent de plus en plus leur système d'exploitation Microsoft Windows et le remplacent par Linux et cela pour de nombreuses raisons, parmi lesquelles nous citons les suivantes :

- Etant enregistré sous une licence logicielle libre, Linux (ainsi que d'autres logiciels libres) peut être obtenu gratuitement.
- Linux est aussi un logiciel libre dans le sens où quiconque peut le modifier, y compris son code source.
- Un support de qualité pour Linux est disponible gratuitement sur internet, notamment par le biais de groupes de diffusion ou de forums.
- Il n'est pas à craindre que Linux devienne un jour complètement obsolète, pour la simple raison que son architecture est extrêmement performante, robuste et sûre.
- Les utilisateurs de Linux ne sont jamais poussés vers des mises à jours forcées, car les versions antérieures bénéficient toujours d'un support et parce que les nouvelles versions sont disponibles gratuitement et offrent par essence une compatibilité élevée avec les précédentes.
- Lorsqu'un utilisateur décide de passer à une version plus récente de Linux, il ne devra payer aucun frais de licence et n'aura rien à dépenser en logiciels s'il choisi une distribution gratuite.
- Linux offre une sécurité supérieure, à savoir un taux très faible d'infection par virus.
- Linux est très résistant aux crashes système et nécessite rarement un redémarrage. Linux a été intégralement bâti pour être un système d'exploitation extrêmement robuste et stable.

4. Pourquoi Linux ?

- Toutes les machines de calcul scientifique sont sous Linux.
- Le système Linux est un système d'exploitation multiutilisateur et multitâche.

4. 1. Multiutilisateurs

Plusieurs utilisateurs peuvent se connecter à une même machine, au même temps, sous Linux; le système partage alors toutes les ressources de l'ordinateur entre les différents usagers. Il existe deux types d'utilisateurs :

- Utilisateurs normaux

Sous Linux, il faut s'identifier pour avoir accès à un ordinateur.
Compte avec - login
- password (changement avec la commande passwd)
- Espace de travail protégé

Sécurité : pas d'utilisateur non autorisé.

Confidentialité : user1 ne lit un fichier appartenant à user2 et se trouvant dans son espace de travail que si user2 l'autorise (système de droits).

Afin d'ajouter de nouveaux utilisateurs on utilise la commande adduser, qui ne peut être exécutée que par le super utilisateur.

- Super utilisateur (root)

L'utilisateur root gère tout le système. C'est l'administrateur. Certaines opérations ne peuvent êtres faites que par lui. Il peut tout faire.

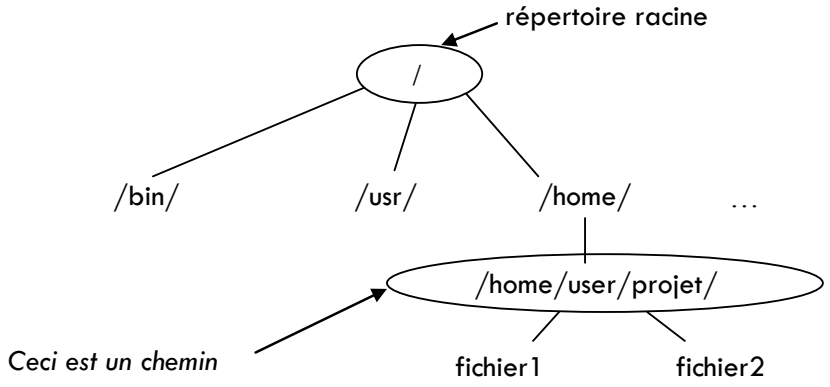
4. 2. Multitâche

Linux est multitâche, car plusieurs programmes peuvent être en cours d'exécution au même temps sur une même machine.

5. Système de fichiers

Linux utilise le standard FHS (Filesystem Hierarchy Standard) pour définir son arborescence. Ce standard propose une structure de répertoires dont chacun possède un rôle spécifique défini dans FHS. Parmi ces répertoires, plusieurs sont importants. On peut citer :

- /bin/ : contient toutes les commandes de base nécessaires au démarrage et à l'utilisation d'un système minimaliste.
- /sbin : contient les commandes système réservées aux administrateurs.
- /boot : contient les fichiers nécessaires au démarrage du système d'exploitation.
- /dev : contient des fichiers correspondant à un périphérique (disque,..).
- /etc : contient la plupart des fichiers de configuration du système.
- /home/ : utilisé pour stocker les répertoires utilisateurs (/home/user/).
- /opt : utilisé comme emplacement d'installation d'un logiciel.
- /tmp/ : utilisé pour stocker les fichiers temporaires tout comme /var/tmp et /run/tmp et généralement vidé à chaque démarrage.



Système de fichier de type arborescence.

6. Console (ou terminal)

Linux fonctionne en mode ligne de commande mais aussi en mode graphique, ce qui permet d'effectuer des opérations très complexes. Chaque utilisateur connecté au système d'exploitation est capable d'interagir avec le Shell en exécutant une commande sur une console.

Une console (anciennement le pupitre des commandes) est généralement un terminal dédié uniquement à l'envoi et au retour des commandes.
En lançant un terminal, une fenêtre s'ouvre. Il n'y a que l'invite de commande qui s'affiche :

```
user@user-Asus:~$_
```

user est l'utilisateur de la machine et Asus est le nom de la machine.

Le tilde ~ veut dire que nous sommes dans le répertoire personnel de user, dans son home ou /home/user (home directory ou répertoire d'accueil), que l'on peut aussi désigner par ~/ ou $HOME.

Le dollar $ indique que nous sommes en mode utilisateur. Par contre, en mode root ou super utilisateur, on a #, comme suit :

```
root@user-Asus:/home/user#
```

A la suite de l'invite de commande, un curseur clignote, sous forme d'un petit rectangle, souvent gris par défaut.

7. Interpréteur de commandes (ou Shell)

Après le login, dans une console, nous voyons par exemple :

```
user@user-Asus:~$_
```

L'interpréteur de commande (ou Shell) attend les instructions.

Le Shell est un programme qui vérifie, interprète les commandes Linux, exécute et renvoie les réponses. Il existe plusieurs Shells. Les plus communs sont bash et tcsh. L'environnement de Shell est définit par des variables, par exemple PATH[2]. Cette variable définit la liste des répertoires où le Shell cherche le binaire ou l'exécutable correspondant à la commande tapée.

8. Modifier le contenu d'une variable

Supposons qu'un nouveau compilateur est installé dans /opt/nanosoft/. Le Shell ne le trouvera pas. Il faut modifier la variable PATH.

Avec bash (catégorie des sh) :

```
user@user-Asus:~$ export PATH=$PATH :/opt/nanosoft
```

Avec tcsh (catégorie des csh) :

```
user@user-Asus:~$ setenv PATH $PATH :/opt/nanosoft
```

Ceci modifie PATH uniquement pour cette console.

[2] PATH : Spécifie l'ensemble des chemins d'accès des exécutables. Le système cherche successivement dans ces répertoires afin de trouver l'exécutable correspondant à la commande tapée ou exécutée dans le Shell.

Le Shell qui se lance automatiquement dès l'ouverture d'une console, lit, entre autres, les fichiers suivants : /etc/profile, /etc/bashrc, /home/user/.bashrc et /home/user/.profile ou /home/user/.bash_profile. En insérant les commandes ci-dessus dans .profile ou .bashrc, la variable PATH sera modifiée automatiquement à chaque ouverture d'une console.

NB : La commande env donne la liste et le contenu de toutes les variables définies.

9. Introduction à la ligne de commande

Une commande est un programme. Pour l'exécuter, saisir son nom éventuellement suivi d'options et d'arguments sur un terminal. La syntaxe standard d'une commande Linux est la suivante :

```
Commande -option arg1 arg2 ...
```

- Dans une commande, chaque mot est séparé des autres par un espace ou une tabulation.
- La commande est suivie d'une ou plusieurs options facultatives qui servent à modifier le comportement de la commande.
- Les arguments, facultatifs aussi, représentent les objets sur lesquels la commande agit.

Exemple 1

user@user-Asus:~$ ls -a /home

où ls est la commande qui permet de lister l'ensemble des fichiers d'un répertoire, en l'occurrence, le répertoire /home et l'option -a permet d'inclure les fichiers cachés dans le résultat.

Exemple 2

user@user-Asus:~$ cp -r rep1 rep2

La commande cp (copy) copie un répertoire (option r) vers un autre répertoire. On a ici deux arguments.

- Conventions courantes sur les options

- Une option commençant par un seul tiret est composée d'une seule lettre (exemple -h, -v). Les options plus longues commencent par deux tirets (exemples : --help, --version).
- Plusieurs options courtes peuvent être combinées après un seul tiret (exemple -la au lieu de -l -a).
- L'option --help sert à afficher une courte documentation de la commande.

Exemple 3

user@user-Asus:~$ cd --help : on aura une courte documentation de la commande cd.

cd: cd [-L|[-P [-e]] [-@]] [rép]
Change le répertoire de travail du shell.

Change le répertoire actuel vers DIR. Le répertoire DIR par défaut est donné par la variable « HOME » du shell. La variable CDPATH définit le chemin de recherche du répertoire contenant DIR. Les noms de répertoires alternatifs dans CDPATH sont séparés par un deux-point « : ». Un nom de répertoire vide est identique au répertoire actuel. Si DIR commence avec une barre oblique « / », alors CDPATH n'est pas utilisé.

Si le répertoire n'est pas trouvé et que l'option « cdable_vars » du shell est définie, alors le mot est supposé être un nom de variable. Si la variable possède une valeur, alors cette valeur est utilisée pour DIR…

9. 1. Consulter l'aide en ligne : man et info

- Commande man

La commande man x (pour manuel) permet d'afficher une documentation, souvent très complète sur la commande x. Elle est, en particulier, utile pour explorer les options possibles d'une commande.

NB : Flèches ↑ et ↓, barre d'espace pour faire défiler le manuel.

Lettre q pour fermer la page du manuel.

La documentation sera donnée en langues Française ou Anglaise, selon le choix de la langue du système.

Exemple

user@user-Asus:~$ man ls : on obtient un manuel de la commande ls.

```
LS(1)          User Commands
LS(1)
NAME
       ls - list directory contents
SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified…
```

- Commande info

La commande info est d'un fonctionnement très similaire à man. Elle peut être utilisée pour produire des pages d'information sur un sujet spécifique.

Exemple

user@user-Asus:~$ info ls : on obtient des informations sur la commande ls.

```
Next: dir invocation, Up: Directory listing
10.1 'ls': List directory contents
==================================
The 'ls' program lists information about files (of any type, including directories). Options and file arguments can be intermixed arbitrarily, as usual…
```

9. 2. Naviguer : ls, pwd et cd

- Commande ls (list)

Permet d'afficher la liste des fichiers et sous répertoires du répertoire courtant.

Exemple 1

user@user-Asus:~$ ls rep1/rep2 : affiche la liste des fichiers et sous répertoire du répertoire rep1/rep2.

Options

user@user-Asus:~$ ls -l : affiche une liste détaillée (droits (*), propriétaires, taille,…etc).
user@user-Asus:~$ ls -a : affiche également les fichiers cachés.
user@user-Asus:~$ ls -t : affiche par ordre de date de dernière modification.

(*) droits :
r : droit de lire.
w : droit d'écrire.
x : doit d'exécuter.

L'option -l de la commande ls permet d'afficher les informations suivantes :
- Le type de fichier :
  - - pour les fichiers ordinaires.
  - d pour les répertoires (directory).
  - …etc.
- Les permissions d'accès.
- Le nombre de liens physiques[3].
- Le nom du répertoire et du groupe.
- La taille en octets.
- L'horodatage (date de dernière modification).

Exemple 2

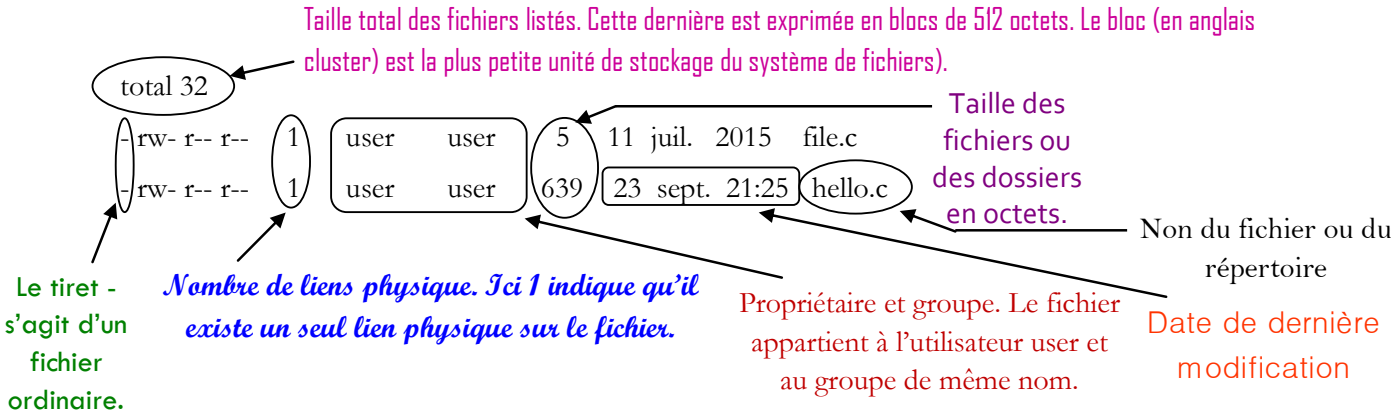
user@user-Asus:~$ ls -l

Le champ rw- r-- r-- indique les permissions d'accès. Ce dernier se décompose en trois parties :

- Première partie (rw-) :
  Concerne des droits d'accès du propriétaire. Ce dernier possède un accès en lecture (r pour read) et écriture (w pour write). Le fichier ne peut donc pas être exécuté, sauf si la lettre x avait été présente.

- Deuxième partie (r--) :
  Concerne les droits d'accès des utilisateurs du groupe auquel appartient user. Ici r indique que les utilisateurs n'ont que le droit de lecture (pas d'écriture ni d'exécution).
- Troisième partie (r--) :

  Concerne les droits d'accès des autres (n'appartenant pas au groupe). Ici r indique un accès possible qu'en lecture.



[3]Lien physique : permet de donner plusieurs noms/chemin d'accès à un même fichier en pointant sur un numéro de fichier (en interne, Linux enregistre les fichiers sur la base d'un numéro, appelé numéro d'indice ou inode et pas sur la base d'un nom). Un fichier peut donc avoir plusieurs noms et existera tant qu'il a au moins un nom.

Exemple 3
Par défaut, sous Linux, tous les fichiers ne sont pas affichés. Les fichiers dont le nom commence par un point « . » ne sont pas affichés. Afin de les afficher, on utilise l'option -a de la commande ls.

```
user@user-Asus:~$ ls  -a
. .. .file  file.c  hello.c  Makfile…
```

La commande ls avec l'option -a affiche trois résultats supplémentaires :
. : désigne le répertoire courant.
.. : désigne le répertoire parent du répertoire courant.
.file : correspond à un fichier caché.

Exemple 4
Afin d'afficher la taille des fichiers de manière plus lisible (en kilo, méga et giga octets) on utilise l'option -h de la commande ls -l.

```
user@user-Asus:~$ ls  -lh
total 22M
- rw- r-- r-- 1 user  user   1,1 M  14  mai  2015  fichier2.tar.gz
- rw- r-- r-- 1 user  user   21  M  16  sept. 2014 fichier.rar
```

Exemple 5
Afin de lister seulement les répertoires :
user@user-Asus:~$ ls -ld */

- Commande pwd (print working directory)

Affiche le chemin absolu (l'emplacement) du répertoire courant.

Exemple
user@user-Asus:~$ pwd
/home/user

- Commande cd (change directory)

Permet de changer de répertoire courant et de se situer sur un autre (c'est-à-dire se déplacer au sein de l'arborescence du système de fichier Linux).

Exemples
user@user-Asus:~$ cd rep1 : aller dans le sous répertoire rep1 du répertoire courant.
user@user-Asus:~$ cd /rep1 : aller dans le répertoire de chemin absolu /rep1.
user@user-Asus:~$ cd .. : pour se diriger vers le répertoire supérieur (parent).
user@user-Asus:~$ cd / : pour se diriger vers le répertoire racine.
user@user-Asus:~$ cd (sans paramètre) : retourner dans le répertoire de connexion de l'utilisateur.

NB : user@user-Asus:~$ cd ..
user@user-Asus:/home$ pwd
/home
user@user-Asus:/home$ cd ..
user@user-Asus:/$ pwd
/ → nous sommes dans le répertoire racine « / ».

Afin de remonter au dossier personnel (répertoire de connexion) :

```
user@user-Asus:~$ cd /home/
user@user-Asus:~$ cd user/
```

9. 3. Chemins d'accès sous Linux

La référence à une ressource (fichier ou répertoire) s'appelle un chemin d'accès (en Anglais path). Dans ce chemin, les noms des répertoires et fichiers sont séparés par un slash /. Il existe deux types de chemins : absolu et relatif.

- Chemin absolu

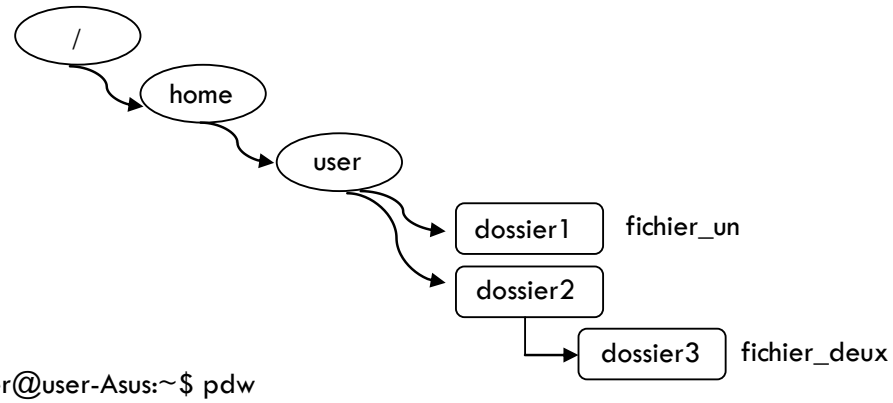
Un chemin absolu se trouve sur la racine de l'arborescence et commence par /.

- Chemin relatif

Un chemin relatif est à priori relatif au répertoire courant où se trouve l'utilisateur.

Exemples

Nous sommes dans /home/user




```
user@user-Asus:~$ pdw
/home/user
user@user-Asus:~$ ls
dossier1 dossier2
user@user-Asus:~$ more /home/user/dossier1/fichier_un   « chemin absolu »
Contenu du fichier_un
user@user-Asus:~$ more /home/user/dossier2/dossier3/fichier_deux
Contenu du fichier_deux
user@user-Asus:~$ pwd
/home/user
user@user-Asus:~$ more  dossier1/fichier_un   « chemin relatif »
Contenu du fichier_un
user@user-Asus:~$ ls
dossier1 dossier2
user@user-Asus:~$ more dossier2/dossier3/fichier_deux
Contenu du fichier_deux
user@user-Asus:~$ cd dossier2
user@user-Asus:~$ ls
dossier3
user@user-Asus:~$ cd dossier3
user@user-Asus:~$ ls
fichier_deux
user@user-Asus:~$ pwd
```

/home/user/dossier2/dossier3
user@user-Asus:~$ more fichier_deux
Contenu du fichier_deux
user@user-Asus:~$ more /home/user/dossier1/fichier_un « chemin absolu »
Contenu du fichier_un
user@user-Asus:~$ pwd
user@user-Asus:~$ /home/user/dossier2/dossier3
user@user-Asus:~$ more ../../dossier1/fichier_un « chemin relatif »
Contenu du fichier_un

9. 4. Commandes de sortie echo et cat

- Commande echo

Permet d'afficher une ligne.
echo permet aussi d'écrire le contenu dans un fichier moyennant le signe > pour écraser le contenu du fichier, ou >> pour suffixer le contenu du fichier.

Exemple 1
user@user-Asus:~$ echo 'Jolie phrase'
Jolie phrase

Exemple 2
user@user-Asus:~$ echo -e 'Jolie phrase\nBelle phrase'
Jolie phrase
Belle phrase

- Commande cat

La commande cat abréviation de concaténer (concatenate en Anglais) permet de créer un ou plusieurs fichiers, d'afficher le contenu des fichiers et de rediriger la sortie vers un terminal ou un fichier.

Exemples
user@user-Asus:~$ cat fichier1 fichier2 : affiche le contenu des fichiers fichier1 et fichier2 dans la fenêtre du terminal.

user@user-Asus:~$ cat -b fichier1 : affiche le contenu du fichier1 dans la fenêtre du terminal et numérote toutes les lignes sauf les lignes vides.

user@user-Asus:~$ cat > fichier3 : crée un fichier nommé fichier3 (redirige la sortie vers fichier3).

user@user-Asus:~$ cat fichier1 fichier2 > fichier4 : joint deux fichiers 1 et 2 et enregistre le résultat dans un nouveau fichier nommé fichier4.

9. 5. Visualiser : more et less

- Commande more

Permet de visualiser le contenu d'un fichier par page.

NB : La touche 'entr' ou 'espace' permet d'avancer l'affichage du contenu du fichier.
La touche q permet de reprendre la main.

Exemple
user@user-Asus:~$ more cp.txt

- Commande less

less permet de visualiser un fichier page par page. Sa fonction est similaire à la commande more, mais a des fonctionnalités plus avancées et permet de naviguer à la fois vers l'avant et vers l'arrière dans le fichier.

Exemples
user@user-Asus:~$ less fichier1 : affiche le contenu du fichier1.
user@user-Asus:~$ less fichier1 fichier2 : affiche plusieurs fichiers à la fois.

Less affiche le premier fichier.
Pour afficher le suivant, taper : puis n pour next.
Pour revenir au fichier précédent, taper : puis p pour previous.
La touche q permet de reprendre la main.

user@user-Asus:~$ less **-N** fichier1 : affiche le contenu du fichier1 et numérote toutes les lignes, même les lignes vides.

Contrairement à la commande cat, le contenu du fichier ne s'affiche pas directement dans le terminal. less bloque alors sur la première page puis on utilise des commandes pour naviguer dans le fichier afin d'afficher le contenu.

- Les commandes de navigation les plus courantes

Les touches ↑ et ↓ permettent de déplacer par ligne.
La touche 'espace' permet de descendre d'une page.
La touche b permet de monter d'une page.
Les touches g ou < permettent d'aller à la première ligne.
Les touches G ou > permettent d'aller à la dernière ligne.

/motif, permet de chercher le motif à partir de la position.
Le mot à rechercher apparaît en surbrillance. Pour rechercher les motifs suivant dans le fichier taper n. Pour rechercher en arrière taper N.

9. 6. Créer : touch et mkdir

- Commande touch

Permet de changer la date du dernier accès ou modification d'un fichier. Permet également de créer un fichier vide.

Exemple
user@user-Asus:~$ touch article : créer un fichier nommé article, si celui-ci n'existe pas déjà.

- Commande mkdir (make directory)

Permet de créer des répertoires comme sous répertoire du répertoire courant.

Exemples
user@user-Asus:~$ mkdir rep1 rep2 rep3 : créer les répertoires rep1, rep2 et rep3 comme sous répertoires du répertoire courant.

user@user-Asus:~$ mkdir -p rep1/A : créer le sous répertoire A dans le sous répertoire rep1 du répertoire courant.

user@user-Asus:~$ mkdir -p rep/a/b : créer le sous répertoire b dans le sous répertoire a, qui a son tour sera créer dans le sous répertoire rep du répertoire courant.

NB : l'option -p permet la création des répertoires parents.

9. 7. Copier, déplacer et renommer : cp et mv

- Commande cp (copy)

Permet de copier des fichiers ou des répertoires.
user@user-Asus:~$ cp fichier1 fichier2 : copie le fichier1 dans le fichier2 du répertoire courant.
user@user-Asus:~$ cp -r rep1 rep2 : copie toute l'arborescence de rep1 dans rep2.
user@user-Asus:~$ cp fichier1 rep1 : copie le fichier1 dans le répertoire rep1.
user@user-Asus:~$ cp -f fichier1 fichier2 : efface les fichiers cibles existants.
user@user-Asus:~$ cp -i fichier1 fichier2 : interroge l'utilisateur avant d'écraser les fichiers réguliers existants (répondre en tapant n pour non ou y pour oui).

- Commande mv (move)

Sert à renommer ou déplacer un fichier ou un répertoire.

Exemples
user@user-Asus:~$ mv fichier1 fichier2 : renomme le fichier1 en fichier2.
user@user-Asus:~$ mv -i fichier1 fichier2 : interroge l'utilisateur avant de renommer le fichier1 en fichier2.
user@user-Asus:~$ mv fichier1 ../fichier2 : déplace en le renommant le fichier1 vers le répertoire parent.
user@user-Asus:~$ mv fichier1 rep1/ : déplace fichier1 au répertoire rep1.
user@user-Asus:~$ mv * rep/ : déplace tous les fichiers vers le répertoire rep.
user@user-Asus:~$ mv *.ext rep/ : déplace que les fichiers ayants l'extension .ext vers le répertoire rep.
user@user-Asus:~$ mv rep1/ rep2/ : renomme le répertoire rep1 en répertoire rep2.
user@user-Asus:~$ mv rep1/ rep3/ : déplace le répertoire rep1 au répertoire rep3.

NB : * remplace des chaînes de caractères quelconque.
La commande mv écrase les fichiers et dossiers de façon irréversible.

9. 8. Supprimer : rm et rmdir

- Commande rm (remove)

Cette commande permet de supprimer un fichier ou un répertoire, elle est à utiliser avec précaution car avec l'option -f ou -rf, cette commande permet de supprimer le système de fichiers de manière irréversible.

Exemples
user@user-Asus:~$ rm -r rep : supprime un dossier vide nommé rep.
user@user-Asus:~$ rm -rf rep : supprime le contenu du dossier plain rep à la fois : (-r) détruit récursivement le dossier plain, (f) détruit sans demander de confirmation.
user@user-Asus:~$ \rm -r rep : supprime le tout à la fois.
user@user-Asus:~$ rm fichier(s) : détruit le fichier/les fichiers.

user@user-Asus:~$ rm rep1/* : détruit tous les fichiers dans rep1.
user@user-Asus:~$ rm rep1/*.dat : détruit tous les fichiers du répertoire rep1 dont les noms se terminent par .dat.
user@user-Asus:~$ rm -i : demande une validation avant destruction.

NB : La commande rm supprime les fichiers ou dossiers de façon irréversible.

- Commande rmdir (remove directory)

Permet de supprimer des répertoires.

Exemples
user@user-Asus:~$ rmdir rep1: supprime le répertoire rep1.
user@user-Asus:~$ rmdir rep1 rep2 : supprime les répertoires rep1et rep2.
user@user-Asus:~$ rmdir -p : supprime les répertoires parents s'ils deviennent vides après la suppression des répertoires mentionnés en arguments. Ainsi, rmdir -p a/b/c est équivalent à rmdir a/b/c, rmdir a/b, rmdir a.

9. 9. Les flux de redirection

Les signes > (prononcé chevron) et >> (prononcé double chevron) permettent de rediriger le résultat d'une commande (la sortie standard (sortie écran)) dans un fichier.

> : redirige dans un fichier et l'écrase s'il existe déjà.
>> : redirige à la fin d'un fichier et le crée s'il n'existe pas.

Exemples
user@user-Asus:~$ ls -l > list.txt : écrit la liste des fichiers du répertoire courant dans le fichier nommé list.txt.
user@user-Asus:~$ ls -t >> list.txt : ajoute dans le fichier list.txt, la sortie de la commande ls -t.

9. 10. Exécuter des programmes en arrière-plan

- & et nohup : lancer un processus en arrière-plan

Les utilisateurs de Linux peuvent lancer des applications en utilisant des lignes commandes (traitement de texte, script,…etc) tout en les détachant du terminal, afin de reprendre l'invite de commandes et de faire autre chose. Il faut donc connaître les techniques qui permettent de lancer une commande en tâche de fond.

- & : lancer un processus en arrière-plan

Rajouter le symbole & à la fin de la commande qu'on veut envoyer en arrière-plan. Le symbole & s'appelle « et commercial » ou encore « esperluette », est présent sur la touche 1 d'un clavier AZERTY.

Exemple
Il est plus pratique d'avoir la main sur un terminal lors de l'édition d'un programme, afin de pouvoir compiler et exécuter au même temps.

user@user-Asus:~$ gedit programme.f90 &
[1] 16504
Le système renvoie deux informations :
[1] c'est le numéro du processus en arrière-plan dans cette console. Comme c'est le premier processus envoyé en arrière-plan, il prend le numéro 1.

16504 c'est le numéro d'identification général du processus (PID). Cette information permet de tuer le processus à l'aide de la commande kill si nécessaire.

Le processus est entrain de tourner en tâche de fond. Il reste toutefois un problème : le processus est 'attaché' à la console. Si on ferme la console sur laquelle on travail, le processus sera tué.

Important
On lance une commande par erreur en arrière-plan et on veut récupérer l'invite de commandes. Il faudra donner l'ordre : ctrl+z pour mettre en pause le programme et récupérer l'invite de commandes.

- nohup : détacher le processus de la console

Si on veut que le processus continue, il faut lancer la commande via nohup.
nohup est l'abréviation de « pas de raccrochage », est une commande supplémentaire qui indique au système Linux de ne pas arrêter une commande jusqu'à ce que son exécution soit terminée.

Exemple
user@user-Asus:~$ nohup script.sh & : lancer un script en arrière-plan.

La sortie de la commande nohup est par défaut redirigée vers un fichier nohup.out. Aucun message ne risque d'apparaître dans la console. D'autre part, la commande est maintenant 'immunisée' contre la fermeture de la console.

- Jobs : connaître les processus qui tournent en arrière-plan

On peut envoyer autant de processus en arrière-plan qu'on veut, au sein d'une même console. Comment savoir les processus qui tournent en arrière-plan ?

```
user@user-Asus:~$ jobs
[1] - Stopped top
[2] + Stopped find/…
```

9. 11. Gérer les processus

Un processus est un programme en cours d'exécution. Il lui est attribué un PID (process ID), un nombre qui le caractérise de manière unique.

- Information sur le process en cours

top : liste en temps réel les process sur l'ordinateur, et ressources utilisées.

NB : ctrl+c : interrompt le processus en mode interactif.

- Modifier un process en cours

user@user-Asus:~$ kill 8564 : tue le process de PID 8564.

9. 12. Vérifier l'espace disque

- Occupation du disque dur

La commande df (disk free) permet d'afficher la taille de l'espace disque occupée et la taille de l'espace disque libre.

L'exécution de la commande df permet d'afficher certaines colonnes par défaut telles que :
Sys. de fichiers : fournit le nom du système de fichiers.
Taille : donne la taille totale du système de fichiers.
Utilisé : indique la quantité d'espace disque déjà utilisé dans le système de fichiers.
Dispo : indique la quantité d'espace disponible dans le système de fichiers.
Uti% : indique la quantité d'espace disque déjà utilisé en pourcentage.
Monté sur : indique le point de montage de ce système de fichiers.

Exemple

```
user@user-Asus:~$ df
```

Sys. de fichiers	blocs de 1K	Utilisé	Disponible	Uti%	Monté sur
Udev	3983680	0	3983680	0%	/dev
tmpfs	802760	1936	800824	1%	/run
/dev/sda6	44295108	17691520	24323756	43%	/
tmpfs	4013796	0	4013796	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	4013796	0	4013796	0%	/sys/fs/cgroup
/dev/loop0	128	128	0	100%	/snap/bare/5
/dev/loop1	56832	56832	0	100%	/snap/core18/2074
/dev/loop2	56832	56832	0	100%	/snap/core18/2128
/dev/loop3	224256	224256	0	100%	/snap/gnome-3-34-1804/66
/dev/loop5	224256	224256	0	100%	/snap/gnome-3-34-1804/72
/dev/loop4	66688	66688	0	100%	/snap/gtk-common-themes/1515
/dev/loop6	66816	66816	0	100%	/snap/gtk-common-themes/1519
/dev/loop7	52224	52224	0	100%	/snap/snap-store/547
/dev/loop8	52352	52352	0	100%	/snap/snap-store/518
/dev/loop10	33280	33280	0	100%	/snap/snapd/13640
/dev/loop9	33152	33152	0	100%	/snap/snapd/12704
/dev/sda1	262144	36844	225300	15%	/boot/efi
/dev/sda7	421892768	25846940	374545108	7%	/home
tmpfs	802756	36	802720	1%	/run/user/1000

L'option -h permet d'afficher les informations de toutes les statistiques du système de fichiers en G (Giga-octets). Si la taille est inférieure à 1 G, elle sera affichée en M (Méga-octets) ou même en o (octets).

Exemple

```
user@user-Asus:~$ df -h
```

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
Udev	3,8G	0	3,8G	0%	/dev
tmpfs	784M	1,9M	783M	1%	/run
/dev/sda6	43G	17G	24G	43%	/
tmpfs	3,9G	0	3,9G	0%	/dev/shm
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	3,9G	0	3,9G	0%	/sys/fs/cgroup
/dev/loop0	128K	128K	0	100%	/snap/bare/5
/dev/loop1	56M	56M	0	100%	/snap/core18/2074
/dev/loop2	56M	56M	0	100%	/snap/core18/2128
/dev/loop3	219M	219M	0	100%	/snap/gnome-3-34-1804/66
/dev/loop5	219M	219M	0	100%	/snap/gnome-3-34-1804/72
/dev/loop6	66M	66M	0	100%	/snap/gtk-common-themes/1519
/dev/loop4	66M	66M	0	100%	/snap/gtk-common-themes/1515

/dev/loop7	52M	52M	0	100%	/snap/snap-store/518
/dev/loop8	51M	51M	0	100%	/snap/snap-store/547
/dev/loop9	33M	33M	0	100%	/snap/snapd/12704
/dev/loop10	33M	33M	0	100%	/snap/snapd/13640
/dev/sda1	256M	36M	221M	15%	/boot/efi
/dev/sda7	403G	25G	358G	7%	/home
tmpfs	784M	36K	784M	1%	/run/user/1000

Autres options

- -m : permet d'afficher des informations sur l'utilisation de tous les systèmes de fichiers en M (Méga-octets).
- -k : permet d'afficher toutes les informations et l'utilisation du système de fichiers dans des blocs de 1024 octets.
- -T : permet d'afficher le type de système de fichiers avec d'autres informations.

- Taille des répertoires

La commande du (disk usage) permet d'afficher la taille d'un répertoire et de tous les sous répertoires récursifs qu'il contient.

Exemple

```
user@user-Asus:~$  du -sh /home/user/Bureau/Thèse/
195M /home/user/Bureau/Thèse/
```

L'option -h affiche les informations dans un format lisible. L'option -s donnera la taille totale d'un dossier spécifié (dans ce cas, il montrera la taille totale du répertoire Thèse).

9. 13. Gérer les archives compressées

- Dossiers

Archiver

```
user@user-Asus:~$ tar -cvf  dossier.tar  dossier
user@user-Asus:~$ ls
dossier  dossier.tar
```

Compresser l'archive

```
user@user-Asus:~$ gzip dossier.tar
user@user-Asus:~$ ls
dossier.tar.gz
```

On peut archiver et compresser un dossier au même temps, en utilisant la commande :

```
user@user-Asus:~$ tar  -zcvf dossier.tar.gz  dossier
user@user-Asus:~$ ls
dossier  dossier.tar.gz
```

Décompresser

```
user@user-Asus:~$ tar  -zxvf dossier.tar.gz
user@user-Asus:~$ ls
dossier  dossier.tar.gz
```

On utilise les options

- -c : signifie créer une archive tar.
- -v : signifie afficher le détail des opérations.

- -f : signifie assembler l'archive dans un fichier.
- -x : pour 'extract'.

- Fichiers

Compresser

```
user@user-Asus:~$ gzip fichier
user@user-Asus:~$ ls
fichier.gz
```

Décompresser

```
user@user-Asus:~$ gunzip fichier.gz
user@user-Asus:~$ ls
fichier
```

9. 14. Gérer les droits d'accès

- Commande chmod

Permet de définir et de changer les droits d'accès d'un fichier ou d'un ensemble de fichiers.
Il y a deux modes d'utilisation de la commande chmod : de façon littérale ou numérique. Nous ne verrons que la façon littérale. Il s'agit de l'utilisation des lettres r, w et x pour spécifier les droits voulus. On utilise aussi des lettres pour désigner les identités : le propriétaire, le groupe et les autres.

chmod utilise les notations suivantes :

- u pour le propriétaire (user).
- g pour le groupe (groupe).
- o pour les autres (other).
- a pour tous (all).

Une autre notation sera utilisée pour attribuer et/ou retirer des droits :

- + pour attribuer.
- - pour retirer.
- = pour fixer l'accès exact.

Exemple 1

```
user@user-Asus:~$ chmod  go-wx  fichier
```

Pour les membres du groupe (g) et les autres (o), on retire (-) les droits d'écriture (w) et d'exécution (x).

Exemple 2

```
user@user-Asus:~$ chmod  u+x,og-w  fichier
```

On attribue (+) le droit d'exécution (x) pour le propriétaire (u) et on retire (-) les droits d'écriture (w) pour les membres du groupe (g) et les autres (o).

Autres exemples

```
user@user-Asus:~$ man cp > cp.txt
user@user-Asus:~$ ls -l
-rw-rw-r-- 1 user  user  5594  fév. 2  17:18  cp.txt
```

user@user-Asus:~$ chmod u+x cp.txt : on attribue (+) le droit d'exécution (x) pour le propriétaire (u).

```
user@user-Asus:~$ ls -l
```

-rwx-rw-r-- 1 user user 5594 fév. 2 17:18 cp.txt

user@user-Asus:~$ chmod u-x cp.txt : on retire (-) le droit d'exécution (x) pour le propriétaire (u).
user@user-Asus:~$ ls -l
-rw-rw-r-- 1 user user 5594 fév. 2 17:18 cp.txt

user@user-Asus:~$ chmod g+x,go-r cp.txt : on attribue (+) le droit d'exécution (x) pour le groupe (g) et on retire le droit de lecture (r) pour les membres du groupe (g) et les autres (o).

user@user-Asus:~$ ls -l
--w--wx--- 1 user user 5594 fév. 2 17:18 cp.txt

user@user-Asus:~$ chmod g-x,go+r cp.txt : on retire (-) le droit d'exécution (x) pour le groupe (g) et on attribue le droit de lecture (r) pour les membres du groupe (g) et les autres (o).

user@user-Asus:~$ ls -l
-rw-rw-r-- 1 user user 5594 fév. 2 17:18 cp.txt

9. 15. Gérer les alias

- Commande alias

Un alias sous Linux peut être considéré comme un raccourci qui représente une ou plusieurs commandes spécifiques. Au lieu de taper à chaque fois une ligne de commande, on utilise un raccourci.

- Création d'un alias

Exemple
user@user-Asus:~$ alias mk='mkdir'
user@user-Asus:~$ mk work
user@user-Asus:~$ ls
work

- Lister les alias définis sur la machine

On peut lister les alias définis sur le système à l'aide de la commande alias sans argument.

Exemple
user@user-Asus:~$ alias
alias cp = 'cp -i'

- Alias permanent

Pour rendre un alias permanent, il suffit de le mettre dans un fichier Shell de la configuration du profil comme : .bashrc.

Exemples
alias mv = 'mv -i'
alias rm = 'rm -i'
alias su = 'sudo -s'

- Supprimer un alias Linux

On peut supprimer un alias sur Linux grâce à la commande unalias. Comme on peut supprimer tous les alias avec l'argument -a c'est-à-dire unalias -a.

Exemple
user@user-Asus:~$ unalias mk
user@user-Asus:~$ mk rapport
mk : commande introuvable.

9. 16. Compilation d'un programme fortran

- Méthode 1

user@user-Asus:~$ gfortran fichier.f90 -o fichier1 : compilation fortran de fichier.f90 dans fichier1.

user@user-Asus:~$ ./fichier1 : exécution du programme.

- Méthode 2

user@user-Asus:~$ gfortran fichier.f90 : compilation fortran de fichier.f90. L'exécutable a.out sera créé par défaut.

user@user-Asus:~$ ./a.out : exécution du programme.

NB : Il existe d'autres langages de programmation autre que fortran. Par exemples : C #, C ou C++, Python, Java…etc.

9. 17. Autres commandes

- *Commande grep*

Permet de rechercher une chaîne de caractères dans un flux de texte (fichier ou sortie d'une commande) et elle accepte les expressions régulières.

Les options de la commande grep sont les suivantes :

- -v : affiche les lignes ne contenant pas la chaîne.
- -c : compte le nombre de lignes contenant la chaîne.
- -n : chaque ligne contenant la chaîne est numérotée.
- -x : ligne correspondant exactement à la chaîne.
- -w : ligne où le mot apparaît tel quel.
- -l : affiche le nom des fichiers qui contiennent la chaîne.
- -i : permet d'ignorer la casse.
- -r : recherche récursive dans tous les fichiers et dossier du répertoire.

La syntaxe d'une commande grep :

grep options 'recherche' chemin

avec :
options : les options possibles.
recherche : le terme à rechercher entre guillemets.
chemin : le chemin du fichier ou dossier où faire la recherche.

NB : Les guillemets autour du terme à rechercher ne sont pas obligatoires. C'est conseillé lorsqu'il y a des caractères autres qu'alphanumériques dans la recherche.

Il existe plusieurs variantes de la commande grep, par exemple :
agrep : permet la recherche de chaîne approximative.
zgrep : permet la recherche dans un fichier compressé.
rgrep : permet la recherche dans tous les fichiers d'un répertoire.

Exemples
Le contenu d'un fichier nommé 'exemple.txt' est le suivant :

Le Shell est un programme.
Ubuntu est une version du système d'exploitation Linux.
Commandes Linux de base.
Programmation en Fortran.
Rédaction sous Latex.

user@user-Asus:~$ grep Linux exemple.txt : permet d'obtenir les lignes contenant la chaîne de caractère Linux.
Ubuntu est une version du système d'exploitation *Linux*.
Commandes *Linux* de base.

user@user-Asus:~$ grep 'Programmation en Fortran' exemple.txt
Programmation en Fortran.

- Commande find

Permet de chercher un fichier sous Linux.
find, contrairement à locate cherche le fichier au sein de l'arborescence. La syntaxe d'une commande find est la suivante :

find répertoire critères

dont répertoire est le chemin sur lequel la recherche est exécuté et critères sont une suite d'expressions permettant d'afficher la recherche.

NB : Si l'on veut rechercher un fichier dans toute l'arborescence, on peut remplacer répertoire par /. Cela signifie qu'on effectue la recherche à partir de la racine et donc dans toute l'arborescence, ce qui est très long.

Les options les plus utilisées, de la commande find, sont les suivantes :

- -name : filtrer par rapport au nom du fichier.
- -type : filtrer par rapport au type du fichier.
- -size : filtrer par rapport à la taille du fichier.

Exemple
user@user-Asus:~$ find . -name artcle.tex
./ Bureau /work/Article/article.tex

On peut effectuer une recherche dans un répertoire :
user@user-Asus:~$ find /home/user/ -name article.tex
/home/user/ Bureau /work/Article/artcle.tex

Ou bien faire une recherche dans toute l'arborescence :
user@user-Asus:~$ find / -name article.tex
/home/user/ Bureau /work/Article/artcle.tex

Lorsque l'on recherche un répertoire, on utilise la syntaxe suivante :

```
find répertoire -type d -name nom-du-répertoire
```

où répertoire désigne le répertoire dans lequel on recherche le répertoire.

Exemple

```
user@user-Asus:~$ find /home/ -type d -name wor*
/home/user/ Bureau /work
```

La recherche est effectuée dans le répertoire home. On cherche les répertoires commençant par wor. Il existe bien évidemment d'autres options de la commande find.

- Commande which

Permet de localiser une commande en effectuant une recherche dans différents répertoires.

Exemples

```
user@user-Asus:~$ which pwd
/bin/pwd

user@user-Asus:~$ which mkdir
/bin/mkdir
```

- Commande locate

Permet de trouver très rapidement un fichier.
locate ne va pas chercher le fichier au sein de l'arborescence, mais au sein d'une base de données contenant la liste des fichiers existants. De ce fait, il se peut que lorsque la création d'un fichier ou d'un dossier est très récente, ce dernier ne soit pas encore ajouté à la base de données. Il faut pour cela réactualiser la base de données à l'aide de la commande updatedb.

Exemple

```
user@user-Asus:~$ locate article.tex
/home/user/Bureau/work/Article/article.tex
```

- Commande head

Permet d'afficher le début d'un fichier (par défaut, les 10 premières lignes).

Exemple 1

user@user-Asus:~$ cat data : afficher le contenu du fichier data.

```
1
2
3
4
5
6
7
8
9
10
11
```

user@user-Asus:~$ head data : affiche les 10 premières lignes du fichier data.

```
1
2
3
4
5
6
7
8
9
10
```

option -n : permet d'afficher les n premières lignes d'un fichier (n est un nombre).

Exemple 2

Le contenu d'un fichier nommé exemple.txt est le suivant :

```
Le Shell est un programme.
Ubuntu est une version du système d'exploitation Linux.
Commandes Linux de base.
Programmation en Fortran.
Rédaction sous Latex.
```

```
user@user-Asus:~$ head  -2 exemple.txt
Le Shell est un programme.
Ubuntu est une version du système d'exploitation Linux.
```

- Commande tail

Permet d'afficher la fin d'un fichier (par défaut, les 10 dernières lignes).

Exemple 1

```
user@user-Asus:~$ tail data
2
3
4
5
6
7
8
9
10
11
```

Exemple 2

```
user@user-Asus:~$ tail  /var/log/auth.log
Oct 27 19:17:02 user CRON[2684]: pam_unix(cron:session): session closed for user root
Oct 27 19:30:01 user CRON[2778]: pam_unix(cron:session): session opened for user root by (uid=0)
Oct 27 19:30:01 user CRON[2778]: pam_unix(cron:session): session closed for user root
Oct 27 19:31:37 user gdm-password]: pam_unix(gdm-password:auth): Couldn't open/etc/securetty: Aucun fichier ou dossier de ce type
Oct 27 19:31:45 user gdm-password]: pam_unix(gdm-password:auth): Couldn't open/etc/securetty: Aucun fichier ou dossier de ce type
Oct 27 19:31:45 user gdm-password]: gkr-pam: unlocked login keyring
```

Oct 27 19:34:03 user pkexec: pam_unix(polkit-1:session): session opened for user root by(uid=1000)
Oct 27 19:34:03 user pkexec[2998]: user: Executing command [USER=root][TTY=unknown] [CWD=/home/ user] [COMMAND=/usr/lib/update-notifier/package-systemlocked]
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session opened for user root by(uid=0)
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session closed for user root

option -n : l'option -n <chiffre> détermine le nombre (<chiffre>) de ligne que l'on désir afficher.

Exemple 3
Pour lire les trois dernières lignes du même fichier que précédemment :

user@user-Asus:~$ tail -n 3 /var/log/auth.log
affiche les 3 dernières lignes :
Oct 27 19:34:03 user pkexec[2998]: user: Executing command [USER=root][TTY=unknown] [CWD=/home/user] [COMMAND=/usr/lib/update-notifier/package-systemlocked]
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session opened for user root by(uid=0)
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session closed for user root

On peut omettre le n en le remplaçant par le chiffre ainsi :

user@user-Asus:~$ tail -3 /var/log/auth.log
affiche les 3 dernières lignes :
Oct 27 19:34:03 user pkexec[2998]: user: Executing command [USER=root][TTY=unknown] [CWD=/home/user] [COMMAND=/usr/lib/update-notifier/package-systemlocked]
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session opened for user root by(uid=0)
Oct 27 20:17:01 user CRON[3302]: pam_unix(cron:session): session closed for user root

option -f : f pour 'follow', suivre en anglais. Ce paramètre permet de suivre la fin du fichier au fur et à mesure de son évolution.

C'est extrêmement utile pour suivre un fichier de log qui évolue souvent :

NB : syslog est un protocole permettant de collecter les messages des services qui tournent sur Linux. L'enregistrement se fait dans des fichiers appelés « fichiers log ». Ces fichiers sont placés dans /var/log.

Exemple avec syslog
user@user-Asus:~$ tail -f /var/log/syslog
Oct 27 20:30:09 user systemd[1517]: Starting Tracker metadata database store and lookup manager...
Oct 27 20:30:09 user dbus-daemon[1531]: [session uid=1000 pid=1531] Successfully activated service 'org.freedesktop.Tracker1'
Oct 27 20:30:09 user systemd[1517]: Started Tracker metadata database store and lookup manager.
Oct 27 20:30:21 user systemd[1]: systemd-hostnamed.service: Succeeded.
Oct 27 20:30:39 user tracker-store[3741]: OK
Oct 27 20:30:39 user systemd[1517]: tracker-store.service: Succeeded.
Oct 27 20:31:19 user systemd[1]: Started Run anacron jobs.
Oct 27 20:31:19 user anacron[3795]: Anacron 2.3 started on 2021-10-27
Oct 27 20:31:19 user anacron[3795]: Normal exit (0 jobs run)
Oct 27 20:31:19 user systemd[1]: anacron.service: Succeeded.

Ctrl+c pour quitter.

- | : Chaîner les commandes

Le pipe | (prononcé païpe ou tube). Son but est de chaîner les commandes.
Sur un clavier AZERTY Français, on peut l'écrire en combinant les touches AltGr et 6 au même temps.

Chaîner des commandes signifie connecter la sortie d'une commande à l'entrée d'une autre commande.

Exemple 1

```
user@user-Asus:~$ ls  -l
-rw-rw-r--  1  user  user  134  oct.  27  19:49  fichier1
-rw-rw-r--  1  user  user   50  oct.  27  19:32  fichier2
```

user@user-Asus:~$ ls -l | grep oct : affiche les lignes de résultat de la commande ls -l, contenant oct.

```
-rw-rw-r--  1  user  user  134  oct.  27  19:49  fichier1
-rw-rw-r--  1  user  user   50  oct.  27  19:32  fichier2
```

Exemple 2

```
user@user-Asus:~$ cat  /proc/cpuinfo  |  grep  processor
```

user@user-Asus:~$ cat /proc/cpuinfo : le fichier /proc/cpuinfo contient des informations à propos du microprocesseur dans un format texte simple, de son ou ses cœurs et de ce qu'il supporte.

On aura, par exemple

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      :  6
model           :  142
model name  :  Intel(R)  Core(TM) i7-8550U CPU @ 1.80GHz
stepping        : 10
microcode       : 0xea
cpu MHz         : 795.519
cache size      : 8192 KB
:
:
```

user@user-Asus:~$ grep processor écrite après cat /proc/cpuinfo | permet d'afficher les lignes possédant le mot 'processor' de la sortie de la commande cat /proc/cpuinfo

```
processor       : 0
processor       : 1
processor       : 2
processor       : 3
processor       : 4
processor       : 5
processor       : 6
processor       : 7
```

Exemple 3

```
user@user-Asus:~$ cat  /proc/cpuinfo  |  grep  'cpu family'
cpu family   : 6
cpu family   : 6
```

```
cpu family   : 6
cpu family   : 6
cpu family   : 6
cpu family   : 6
cpu family   : 6
cpu family   : 6
```

- Comparer des fichiers avec diff

Il nous arrive d'avoir à comparer des fichiers texte. Il est possible de le faire en mode texte avec diff.

Exemple

```
user@user-Asus:~$ cat  fichier1
1
2
6
4
5

user@user-Asus:~$ cat fichier2
1
2
3
4
5
```

On a donc une ligne de différence (la ligne 3).
La commande de comparaison est la suivante :

```
user@user-Asus:~$ diff  fichier1  fichier2

3c3
<6
---
>3
```

Ce qui veut dire qu'il y a une différence sur la ligne 3 du premier fichier avec la ligne 3 du second fichier (3c3). Sur le premier fichier on trouve 6 (<6) et 3 sur le second fichier (>3).

- Commande paste

La commande paste concatène les lignes de même niveau des fichiers en argument. Les éléments concaténés sont séparés par une tabulation.

Exemple

```
user@user-Asus:~$ cat Mars_2020
Electricité 123.50
Eau         25.00
Gaz         50.00

user@user-Asus:~$ cat Avril_2020
Electricité 156.00
Eau         34.50
```

Gaz 67.00

Mettre sur la même ligne les montants de chaque catégorie :

```
user@user-Asus:~$ paste  Mars_2020  Avril_2020
Electricité  123.50      Electricité 156.00
Eau          25.00       Eau         34.50
Gaz          50.00       Gaz         67.00

user@user-Asus:~$ paste  Avril_2020  Mars_2020
Electricité  156.00      Electricité 123.50
Eau          34.50       Eau         25.00
Gaz          67.00       Gaz         50.00
```

Concaténer toutes les lignes en une seule :

```
user@user-Asus:~$ paste  -s  Mars_2020  Avril_2020
Electricité 123.50  Eau  25.00  Gaz  50.00
Electricité 156.00  Eau  34.50  Gaz  67.00
```

- Commande clear

Cette commande trouve son utilité lorsqu'une succession de commandes surcharge le terminal, limitant ainsi la lisibilité. Elle permet de replacer la ligne de commande en haut de l'écran. Les commandes effectuées sont toujours visibles en remontant la page.

Exemple

```
user@user-Asus:~$  clear
```

- Commande reset

Permet de réinitialiser un terminal à son état initial.

Exemple

```
user@user-Asus:~$ reset
```

NB : Les touches du clavier ctrl+l jouent le même rôle que la commande reset.

- Astuces pour aller plus vite

➢ Complétion automatique

La touche Tab ⇄ permet de compléter automatiquement une commande ou un nom de fichier. S'il y a plusieurs possibilités la liste de choix apparaît.

➢ Rappel de commande

Les touches ↑ et ↓ permettent de faire défiler la liste des commandes précédemment tapées (liste stockée dans /home/user/ .bash_history).

9. 18. Editeurs de texte

- gedit

gedit est un éditeur de texte libre (logiciel destiné à la création et l'édition de fichiers textes).

Exemple

user@user-Asus:~$ gedit article : éditer un fichier nommé article.

- Editeur des fichiers texte : vi

vi est un éditeur de texte que l'on trouve par défaut sur l'ensemble des systèmes Linux.

➢ Lancer vi

```
user@user-Asus:~$ vi  fichier
```

➢ Modes vi

On trouve plusieurs modes :

- ***Mode commande (c'est le mode par défaut)***

Tout caractère est traité comme une commande. Permet :

- Mouvements (déplacements).
- Effacement (copier/couper/coller).
- Rechercher.

- ***Mode insertion***

Permet d'ajouter/insérer des caractères. Tout caractère est traité comme du texte. Afin de revenir en mode commande taper « ESC ».

- ***Mode lignes de commande***

- Quitter, enregistrer, fermer.
- Remplacer.
- Exécuter une commande externe.

➢ Guide vi

- Mode insertion

Ce mode est invoqué par une des commandes :

- **`i`** : insère des caractères après le curseur.
- **`A`** : ajoute des caractères à la fin d'une ligne où est positionné le curseur.
- **`o`** : insère une ligne après le curseur.
- **`O`** : insère une ligne avant le curseur.
- **`a`** : insère après le curseur.

- Mouvements

On appelle les déplacements du curseur dans le fichier des "mouvements".

- **`:0`** ou **`:1`** : le curseur se met au début de la ligne.
- **`:10`** : déplace le curseur à la ligne 10.
- **`:$`** : le curseur va à la fin de la ligne.
- **`w`** : le curseur va au début du mot suivant.
- **`e`** : le curseur va à la fin du mot courant.
- **`b`** : le curseur va au début du mot précédent.
- **`gg`** : aller au début du document.
- **`G`** : aller au début de la dernière ligne du document.
- **`G$`** : aller à la fin de la dernière ligne du document.

- Quantificateur

- **`2w`** : aller à 2 mots à partir du curseur.

- Effacer/Couper

- **x** : efface le caractère sous le curseur.

Avec mouvement :

- **dw** : efface le mot sous le curseur.
- **d$** : efface jusqu'à la fin de la ligne à partir du curseur.
- **de** : efface jusqu'à la fin du mot à partir du curseur.
- **dd** : efface la ligne du curseur.

Avec quantificateur :

- **d2w** : efface les deux mots à partir du curseur.
- **2dd** : efface les deux lignes à partir du curseur.

- Annuler

- **u** : annule la dernière commande.
- **U** : annule tous les changements sur une ligne.
- **CTRL-R** : annule l'annulation.

- Copier/Coller

- **yy** : copie la ligne.
- **y$** : copie jusqu'à la fin de ligne.
- **Y** : copie dans le tampon la ligne du curseur.
- **p** : colle à l'endroit du curseur.

- Numérotation des lignes

- **:se nu** : affiche les numéros de ligne.
- **:se nonu** : désactive l'affichage des numéros de ligne.

- Rechercher

- **/** : recherche une occurrence.

- Fichier

- **:q!** : quitte sans enregistrer.
- **:x** : quitte en enregistrant.
- **:w** : enregistre le fichier.
- **:w nomdefichier** : enregistre sous un nom.
- **:wq** : enregistre et quitte.

- Remplacer

- **:s/aa/bb** : remplace sur une ligne les lettres `aa` par des lettres `bb`.
- **r** : remplace le caractère sous le curseur.
- **R** : remplacement des caractères sous le curseur.

NB : Il existe d'autres éditeurs de texte sous Linux**.** Par exemples : Libre Office, Kile, Lyx, Emacs…ect.

Bibliographie

1. Kiki Novak, *Administration Linux par la pratique*. Tome 1 : *les fondamentaux de l'administration système*. EDITIONS EYROLLES, 2019.

2. Stéphane Gill. *Le système d'exploitation GNU-Linux*. Récupéré de la page : https://fr.wikibooks.org/w/index.php?title=Le_système_d%27exploitation_GNULinux/Version_imprimable&oldid=590902. La dernière modification été faite le 12 avril 2018.

3. Mathieu Nebra, *Reprenez le contrôle à l'aide de Linux !* www.openclassrooms.com. La dernière mise à jour été faite le 4 janvier 2013.

4. Vincent Lozano. *Unix, Pour aller plus loin avec la ligne de commande*. ILV. BIBLIOTHECA Editions, 2010.

5. Jean-Paul Armspach, Pierre Colin et Frédérique Ostré-Waerzeggers. *LINUX Initiation et utilisation*. Dunod, 2004.

6. Liens utiles :

- https://www.01net.com/astuces/surveiller-loccupation-du-disque-dur-540712.html. La dernière mise à jour été faite le 15 septembre 2011.
- https://blog.hubspot.fr/website/langage-de-programmation. La dernière mise à jour été faite le 20 janvier 2023.
- https://www.hostinger.fr/tutoriels/espace-disque-linux. La dernière mise à jour été faite le 18 janvier 2023.