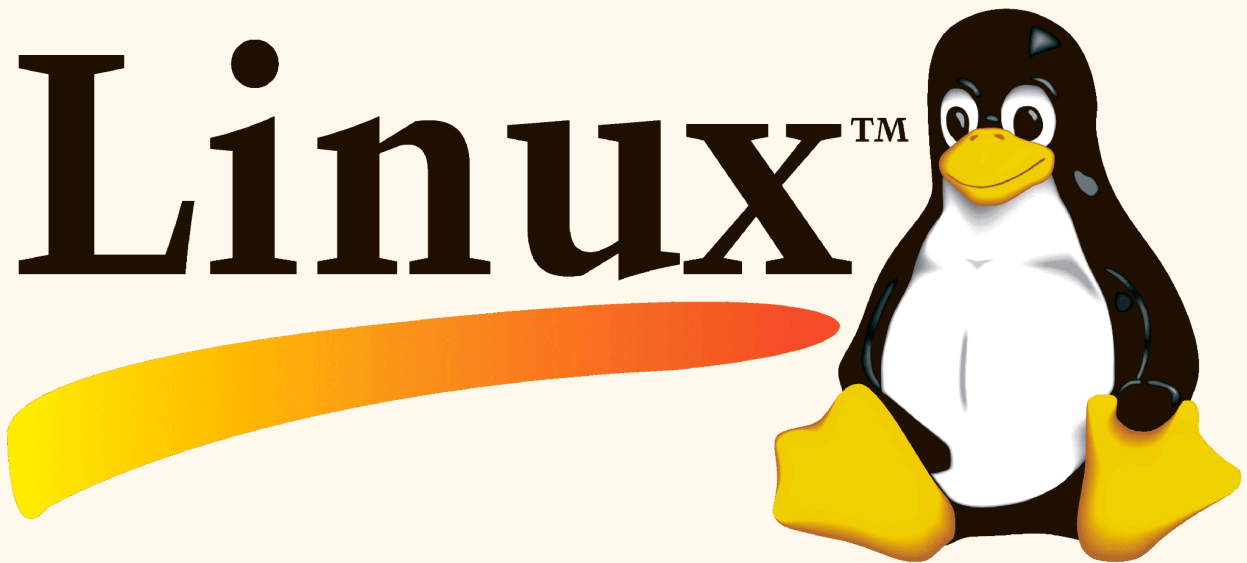


# Fiche de révision

## Systeme 1 : UNIX et Programmation Shell

---

Par SINADINOVIC Marko



## REMERCIEMENTS

Avant de débiter la fiche de révision, je tiens tout d'abord à remercier Monsieur GUIGNE Franck, responsable du cours de Système 1 : UNIX et Programmation Shell à l'Université Côte d'Azur, son cours m'a permis de constituer cette fiche de révision et à y voir plus clair à mes débuts dans l'apprentissage de la programmation Shell sous Unix. Certains éléments de ce document sont inspirés de son cours, si vous souhaitez approfondir vos connaissances, veuillez vous y référer.

Je suis aussi reconnaissant envers GIAUFER SAUBERT Corentin, étudiant et développeur freelance, il m'a permis de récupérer plusieurs ressources utiles afin de constituer cette fiche de révision. Son aide m'a été précieuse dans l'explication de certaines commandes et notions.

## Table des matières

<b>Introduction.....</b>	<b>4</b>
<b>Connaissance générale.....</b>	<b>4</b>
Champ lexical important.....	4
Architecture.....	7
UNIX et GNU/Linux.....	8
Les distributions.....	9
Virtualisation.....	10
Les serveurs.....	11
Réseau et protocoles.....	12
Questions de cours possibles à l'examen.....	14
<b>Commandes.....</b>	<b>17</b>
Expressions régulières.....	18
Naviguer .....	20
Manipuler les fichiers et répertoires.....	24
Permissions.....	32
Chercher .....	37
Questions possibles à l'examen.....	40
<b>Examen d'entraînement.....</b>	<b>46</b>
<b>Conclusion.....</b>	<b>49</b>

# INTRODUCTION

Cette fiche de révision est destinée aux étudiants et aux personnes qui souhaitent franchir le pas vers un système d'exploitation sous UNIX. Dans ce résumé vous trouverez des ressources utiles pour débutants, utilisateurs du quotidien et surtout pour préparer votre premier examen de Système 1. Ce document fait également suite à la séance de révision en distanciel organisée par le Google Student Club de l'Université Côte d'Azur.

À noter que ce document ne remplace pas vos leçons, vos TP et TD. Vous devez être assidue dans votre apprentissage et en cours afin de réussir. De plus, il est destiné à un usage purement personnel, il n'est pas destiné à une activité lucrative. Ce document contient des connaissances trouvables sur Internet et qui font partie du domaine public.

## CONNAISSANCES GÉNÉRALES

Dans ce chapitre vous trouverez du vocabulaire utile pour ce cours, il vous permettra de comprendre certaines commandes, définitions et surtout certains concepts qui peuvent être abstraits au premier regard. Les schémas sont là pour mieux visualiser certaines notions.

### 1. Champ lexical

#### Profil / utilisateur

Un profil utilisateur sur une machine est tout simplement un compte dédié à une personne afin qu'il puisse utiliser la machine sous une identité. Root est par exemple un profil, c'est le compte administrateur par défaut.

#### Répertoire et fichiers

Un répertoire (ou dossier) est comme une boîte qui entrepose des fichiers. Cependant, les ordinateurs considèrent les répertoires comme un fichier qui liste d'autres fichiers. Globalement, on les appelle les noeuds.

Un fichier est un document qui peut contenir plusieurs types de données comme du texte, une image, un programme, du son, une vidéo, un programme pilote et j'en passe. Il faut aussi savoir qu'il existe des fichiers qui permettent de rediriger l'utilisateur vers un autre fichier, ce sont les liens symboliques (comme les icônes de jeux vidéos quand on les installe).

## Périphérique

Un périphérique de manière générale est un objet, dispositif externe à une machine qui est connecté à cette dernière afin de lui ajouter une caractéristique en plus. On distingue 3 types de périphérique.

Périphérique d'entrée : Ils servent à envoyer des informations au système d'exploitation de la machine, ce sont tous les objets comme les claviers, souris, microphones et j'en passe.

Périphérique de sortie : Ils servent à recevoir des informations du système et de la machine, ce sont les écrans, les haut-parleurs etc.

Périphériques de stockage : Ils servent à la lecture et écriture de données, ce sont les disques dur, CD-ROM, clé USB...

## Mémoire virtuelle et vive

La mémoire vive (ou mémoire réelle) est la mémoire principale de votre ordinateur, elle correspond à la RAM. Sa tâche est de gérer et stocker les données en cours d'utilisation en temps réel, puis lorsque le processus ou programme est fermé, la RAM oublie et passe à autre chose.

La mémoire virtuelle est une mémoire qui n'est pas physique comme la RAM, elle sert à épauler la mémoire vive lorsque cette dernière est surchargée et pleine. Cela permet à la machine de continuer de fonctionner malgré tout. La mémoire virtuelle se sert du disque dur ou du SSD, elle lui emprunte de l'espace temporairement.

## Logiciel

Un logiciel est un ensemble de programmes qui indiquent à la machine ce qu'elle doit faire, exécuter, afficher et autre. On peut citer EMACS, VIM, VSCode comme logiciel de traitement

## Processus

Un processus est un objet (programme) avec un comportement et un état qui est en cours d'exécution par la machine. Les processus peuvent être nombreux et fonctionner en même temps. Chaque processus est identifié par un numéro entier, on peut aussi retrouver d'autres informations comme les données traitées par ce code, le répertoire courant du processus.

## Script

C'est un programme qui a pour but d'exécuter une action quand les conditions sont remplies.

## Partie logiciel et partie matérielle

La partie matérielle, ou le "hardware", correspond à tous les composants physiques d'une machine, ce sont les processeurs, cartes graphiques, RAM, carte mère etc.

La partie logicielle, "software", est l'ensemble des programmes, processus, instructions qui communiquent avec le hardware afin de faire une action. L'exemple parfait de software est le système d'exploitation.

## Shell

C'est un programme indépendant du noyau qui permet aux humains de communiquer directement avec le système d'exploitation. Il peut se présenter sous plusieurs formes, graphiques ou pas. Le shell est très utilisé en informatique, ce dernier permet d'exécuter des commandes très puissantes et utiles au quotidien.

## Système d'exploitation

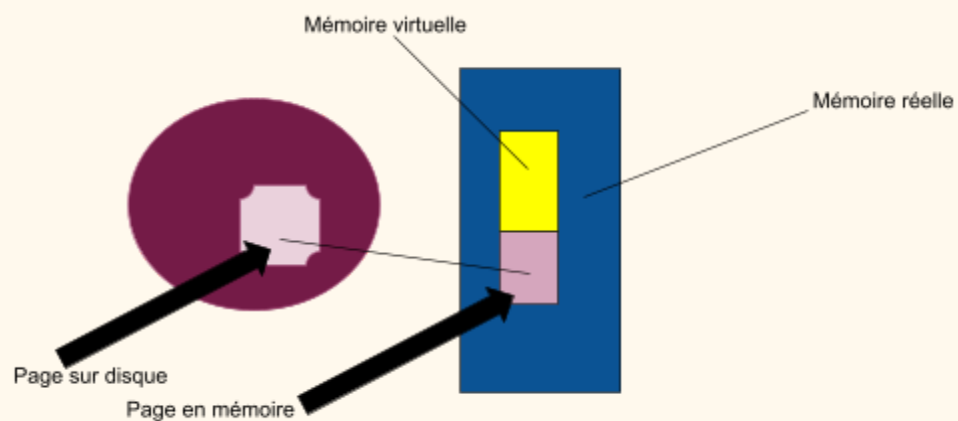
Un système d'exploitation, OS en anglais pour "operating system", est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur. Il est le logiciel principal d'un ordinateur car il permet la communication entre les différents logiciels, périphériques mais aussi entre l'humain et la machine par l'intermédiaire d'un shell.

Les systèmes d'exploitation sont aujourd'hui nombreux, on peut citer les plus célèbres comme Windows, Mac OS, UNIX, GNU/Linux, Android, iOS, VMS et j'en passe. Ce qui peut différencier ces derniers sont les plateformes/environnements sur lesquels ils sont installés, les utilisations, mais aussi sur leurs gestions des tâches et utilisateurs.

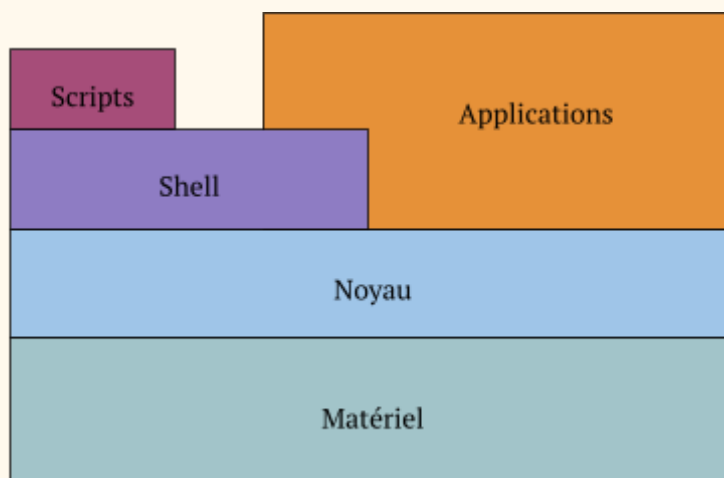
On parle de systèmes mono-tâche, mono-utilisateur ou multi-tâches, mais quelle est la différence ? La différence réside dans l'intitulé. Un système mono-tâche peut gérer juste une tâche (dis aussi programme) à la fois, contrairement à un multi-tâches qui peut gérer plusieurs programmes à la suite comme Windows, Mac OS, UNIX etc. Il en va de soit qu'un système mono-utilisateur peut avoir qu'un profil utilisateur pour une machine sur un système d'exploitation contrairement aux multi-utilisateur qui permettent à plusieurs personnes d'utiliser la même machine tout en ayant des comptes différents donc séparer les fichiers, répertoires des autres profils.

## 2. Architecture

### Gestion de la mémoire



### Machine



Pour plus d'explications concernant l'architecture d'un système informatique, référez vous au cours de Monsieur Guigne. Celui-ci est plus complet que ce document qui sert de révision.

### 3. UNIX et GNU/LINUX

#### UNIX

C'est une famille de systèmes d'exploitations multi-tâches et multi-utilisateurs qui n'est pas liée à un constructeur ou un fabricant de logiciel, et qui fonctionnent sur tous les ordinateurs.

#### Projet GNU

C'est un projet informatique qui avait pour but de développer un système d'exploitation GNU. Le projet a été scindé en plusieurs sous-groupes, chacun de ces derniers avaient un objectif précis, créer des logiciels libres afin de créer un système d'exploitation. A noter que ces logiciels peuvent être utilisés indépendamment en dehors de ce projet.

#### Noyau Linux

Dans un premier temps, un noyau est une des parties fondamentales des systèmes d'exploitation, c'est le cœur d'un système, il s'occupe des ressources de l'ordinateur, permettant aux différents composants de communiquer entre eux. Donc le noyau Linux est un noyau de système d'exploitation de type UNIX.

Plusieurs machines utilisent des systèmes d'exploitation avec un noyau Linux, on peut citer les téléphones portables Android.

#### GNU/Linux

C'est une famille de systèmes d'exploitation open source de type UNIX avec un noyau Linux.

#### Architecture arborescence répertoire dans un système d'exploitation GNU/Linux



Par exemple, /bin c'est le répertoire des binaires du système.



## 4. Les distributions

### Qu'est-ce qu'une distribution en système informatique ?

On peut voir une distribution comme un système d'exploitation contenant plusieurs logiciels d'un certain type compatibles entre eux qui créent une harmonie faisant fonctionner le système d'exploitation.

### Distribution GNU/Linux

Une distribution est appelée ainsi lorsqu'elle contient des logiciels du projet GNU et que le système d'exploitation possède un noyau Linux

### Exemples de distributions

- Debian
- Ubuntu
- Gentoo
- Red Hat
- Fedora
- Mandriva
- SuSE
- Mint
- Alpine
- Fedora
- Slackware

## 5. La virtualisation

### Définition

La virtualisation est un ensemble de procédés qui permettent de faire fonctionner sur une machine plusieurs systèmes d'exploitation, applications, comme s'ils fonctionnaient sur des machines physiques distinctes.

Pour la virtualisation il est nécessaire d'avoir un logiciel comme VMWare ou VirtualBox et un fichier .iso afin d'installer une distribution.

## Emulation

Le logiciel de virtualisation crée un ordinateur virtuel complet. Ce type de virtualisation permet d'utiliser des applications destinées à d'autres types de machine comme des consoles, téléphones sur votre ordinateur. Cependant, l'émulation utilise les instructions de votre propre système afin de les appliquer au système émuler.

## Virtualisation complète / Machine virtuelle

Le logiciel de virtualisation émule un ordinateur virtuel mais utilise certaines instructions afin de sécuriser le système de la machine. VMWare ou VirtualBox permettent cette virtualisation

## Environnement Virtuel

Le logiciel de virtualisation attribue à chaque environnement un espace mémoire, qui permet l'isolation des environnements. Il n'y a pas interception des instructions et l'environnement voulu doit être compatible avec celui du système sur la machine.

## Extension des fichiers

### **Pour VMware :**

.vmx : machine virtuelle

.vmdk : disques virtuelles et fichiers binaires représentant les disques

.log : événements relatifs à l'exécution de la machine virtuelle

### **Pour VirtualBox :**

.xml : machine virtuelle dans un répertoire

.vdi : disques virtuelles et fichiers binaires représentant les disques

.log : événements relatifs à l'exécution de la machine virtuelle

## 6. Les serveurs

### Définition

C'est un mélange de dispositif matériel et logiciel qui permet d'échanger des informations, communiquer entre plusieurs machines, stocker des fichiers etc. Cela peut être un réseau d'ordinateurs relié entre eux ou un programme qui effectue des actions en réseau avec d'autres programmes.

Généralement, les serveurs répondent de manière automatique, les requêtes et réponses sont envoyées/traitées selon des règles régies par ce qu'on appelle les "protocoles réseaux".

### Usage le plus fréquent

Les serveurs sont utilisés pour le courrier électronique, le stockage de base de données, connecter plusieurs personnes, accéder à des pages internet, partager des informations.

### Architecture requête réponse

Client



### Serveur X

C'est un système informatique qui permet d'afficher sur un terminal graphique, le vôtre par exemple, une fenêtre ou autre en exécutant l'action à partir d'une autre machine, l'action passe par le réseau et s'effectue sur votre terminal graphique.

## 7. Réseau et protocoles

### Définition réseau

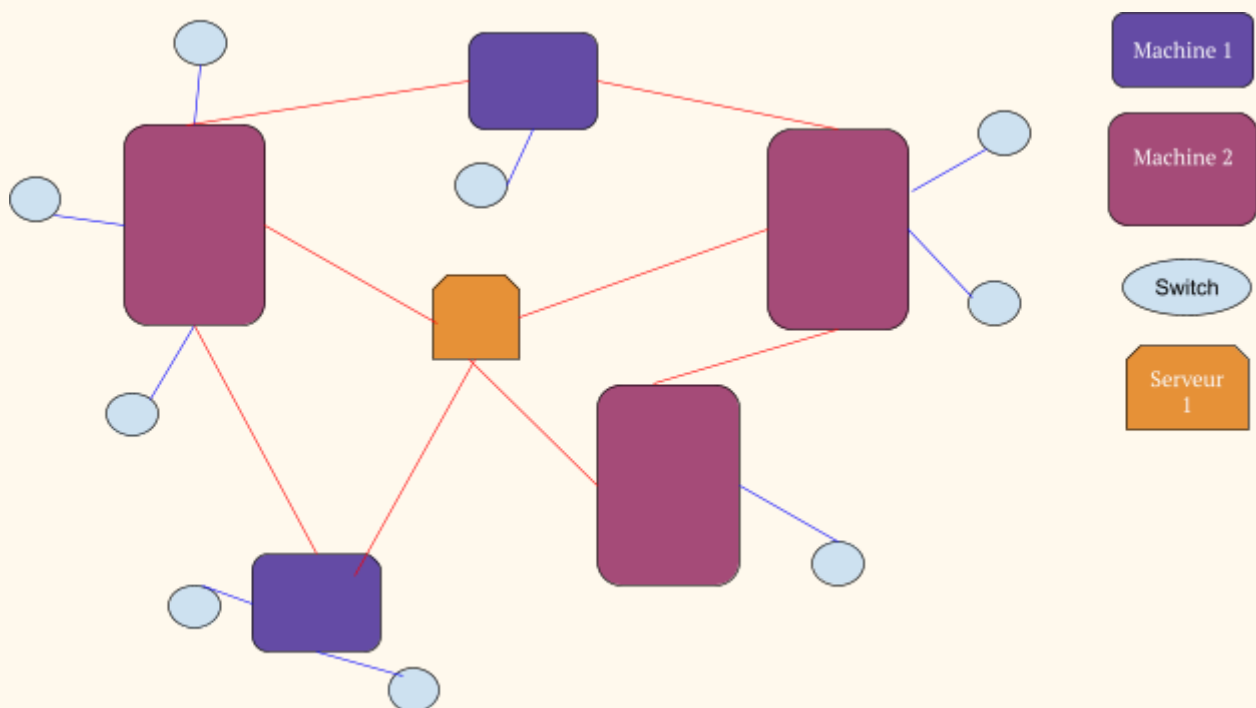
Un réseau est un ensemble de nœuds et de liens encore plusieurs machines (ordinateur, imprimantes, serveurs etc.)

Il existe des réseaux locaux, ils relient les machines d'un particulier, d'une entreprise par ligne téléphoniques, fibre optique, exemple : Campus d'Université

Quant au réseau étendu, il relie des machines qui sont loins l'une de l'autre (de l'ordre de milliers de kilomètres), et ils sont reliés par des satellites, réseaux publics, exemple : Université

Prenons le cas de "Internet", c'est un réseau des réseaux, il relie tous les réseaux nationaux, régionaux ou autre entre eux. Ces réseaux communiquent entre eux par l'intermédiaire de routeurs.

### Architecture d'un réseau simple



On remarque qu'il y a des liens entre certaines machines et que d'autres ont des liens "locaux" comme les ronds (switch) avec les machines 1 et 2.

### Transmettre des informations

**Liaison directe** : Connexion uniquement entre 2 machines dans un réseau simple.

**Diffusion** : Réception de l'information grâce à un émetteur qui la diffuse dans toutes les directions.

**Réseau Ethernet** : Les machines sont connectées entre elles et traitent l'information qui les concernent.

**Transmission switch** : Les machines reçoivent l'information grâce à des switch qui envoient l'information pendant quelque temps puis changent de "cible".

### Définition paquet

Un paquet est une information (message à faire passer) qui est découpé en plusieurs petits bouts afin de ne pas surcharger les canaux de communications. A l'arrivée, il est assemblé pour que la machine puisse le lire. L'échange de ces paquets est régie par les protocoles

### Définition protocole

Un protocole est une norme, une règle qu'on applique lors des échanges d'informations. Les protocoles prennent en compte la forme des données, la codification utilisée, appliquent certaines conventions et bien d'autres.

### IP & MAC

Une adresse IP (Internet Protocol) est un numéro unique (qui sert d'identifiant) attribué aux machines qui sont connectées à un réseau utilisant Internet. Il existe l'IPv4 codé sur 32 bits et l'IPv6 codé sur 128 bits.

Une Adresse MAC (Media Access Control), est un identifiant ancré dans les cartes réseaux des machines, chaque adresse MAC est unique au monde et permet d'identifier les machines sur les réseaux.

On peut aussi rajouter le NAT (Network adresse translation) qui permet de "cacher l'adresse IP des machines". Aussi, DNS (Domaine Name service), qui permet d'attribuer un nom/acheminement à une machine, ce nom remplace l'adresse IP de la machine. Cela simplifie l'accès et la communication entre machine

## Les protocoles

**TCP : Transmission Control Protocol.** Il découpe le message en paquet, il assure la sécurité et l'intégrité de la transmission, s'il y a une erreur le protocole dialogue avec l'expéditeur pour demander les paquets manquant, chaque application utilise un numéro de port spécialisé qui l'identifie, il rassemble les paquets à l'arrivée

**IRC : Internet Relay Channel.** Ce dernier permet la communication entre plusieurs personnes sous forme immédiate. Pour ce faire, l'utilisateur doit se connecter à un serveur IRC, choisir un canal et profiter des discussions

**FTP : File transmission protocol.** Ce protocole sert à partager des fichiers sur un réseau TCP/IP. En outre, il permet de copier, supprimer, manipuler des fichiers d'un ordinateur connecté au réseau vers un autre ordinateur

**NNTP : Network News Transfer Protocol.** Il permet la communication entre plusieurs personnes sous forme différée. L'utilisateur se connecte à un serveur et s'abonne à un canal de discussion qui l'intéresse.

**P2P : Peer to Peer.** C'est un protocole qui permet d'échanger des informations, messages directement entre deux ordinateurs connectés au système.

**SMTP : Simple Mail Transfer Protocol.** C'est un protocole de communication utilisé pour transférer les emails vers les serveurs de messagerie électronique.

**IMAP : Internet Message Access Protocol.** Ce protocole est le complémentaire de SMTP, il permet d'accéder à ses courriers électroniques directement sur les serveurs de messagerie, ce que SMTP ne permet pas.

**SSH : Secure Socket Shell.** Il permet aux administrateurs d'accéder à distance à une machine de manière sécurisée.

## 8. Questions de cours possible à l'examen

Quelle est la différence entre un répertoire et un fichier ?

Citez des systèmes de fichiers utilisés sous Linux et Windows

Qu'est-ce qu'un périphérique ?

A quoi sert la RAM ? Quelle est la différence entre mémoire virtuelle et vive ?

Qu'est-ce qu'un shell ? Que fait une commande intrinsèque ?

Que fait la touche TAB (tabulation) lorsqu'on est dans un shell ? La touche SUPR (suppression) ? Les flèches directionnelles ?

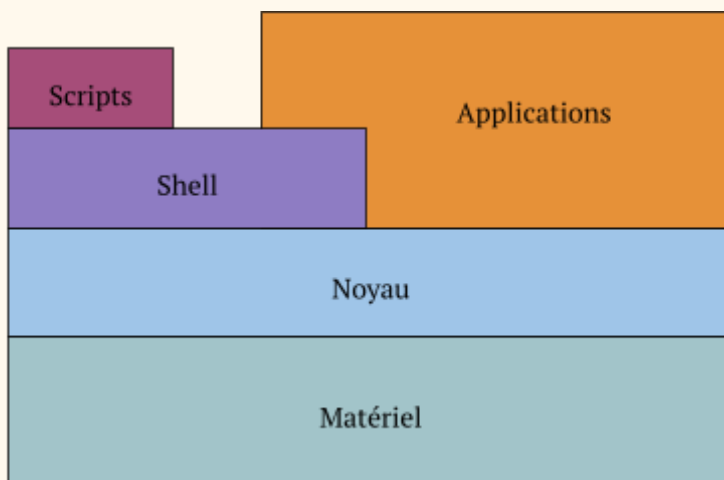
Qu'est-ce qu'un processus ? Quelle commande est associée à l'affichage des processus ? Citez moi au moins 2 processus

Citez deux types de construction d'image avec leur propriétés ainsi qu'un logiciel pour chaque type qui traite ces images (photos, dessins etc.).

Qu'est-ce qu'un système d'exploitation ? Donnez-moi son rôle ainsi que des exemples de systèmes d'exploitation.

Qu'est-ce qu'un OS multi-utilisateur mono-tâche ?

Expliquez moi le schéma ci-dessous. Que fait le noyau ? Pourquoi n'est-il pas au-dessus de shell ? Quelle est la différence entre applications et script ?



Qu'est-ce qu'un noyau Linux ?

A partir de la définition d'un noyau Linux et du projet GNU, expliquez ce qu'est GNU/Linux

Qu'est-ce qu'une distribution GNU/Linux ? Citez en 3.

Qu'est-ce que Fedora ?

Quelle est la sortie "poubelle" sous linux ?

Qu'est-ce que PWD ? HOME ? CC ? EDITOR ? BROWSER ? PATH ? DISPLAY ?

Citer trois répertoires obligatoires de l'arborescence du système de fichier d'une installation linux et expliquer les.

Qu'est-ce qu'un fichier système ? Citez moi en 3.

Quel est le principe de la virtualisation ? Citez moi deux types de "virtualisation" et donnez une brève explication pour chacun d'eux.

Que fait un fichier en .vmx ? .log ? .xml ?

Qu'est-ce qu'un "iso" ?

Qu'est-ce qu'une partition swap ?

Qu'est-ce qu'un serveur ? Quel est l'usage le plus fréquent ?

Qu'est-ce qu'un serveur X ?

Qu'est-ce qu'un réseau ? Quelle analogie peut-on faire pour mieux le comprendre ?

Qu'est-ce qu'un modèle "client/serveur" ?

Qu'est-ce qu'un réseau Ethernet ?

Qu'est-ce qu'un paquet ?

Qu'est-ce qu'un protocole ?

Quelle est la différence entre IP et MAC ? Expliquez en donnant les définitions.

Quelle est la différence entre IPv4 et IPv6.

Combien peut-on avoir d'adresse IP avec le IPv4 ? IPv6 ?

Citez moi 3 protocoles (hors TCP), et donnez leurs utilités.

Qu'est-ce que TCP?



# COMMANDES

Dans ce chapitre vous trouverez des commandes utiles à la vie quotidienne. Ces dernières vous permettront de naviguer dans l'arborescence de votre machine, de manipuler des répertoires, fichiers, rechercher des caractères dans un fichier texte, modifier les permissions des différents fichiers afin de garantir une sécurité et enfin, quelques commandes pour root pour le côté administrateur système.

## 9. Expressions régulières

Les expressions régulières ou regex sont des caractères qui permettent de représenter un ensemble de chaînes de caractères possible. Ces expressions sont très utiles pour rechercher des mots dans un fichier de plusieurs milliers voir millions de caractères. Elles sont souvent utilisées dans des commandes dites de "recherche" que nous verrons par la suite.

Voici une brève liste d'expression régulières que nous utiliserons :

### Classes de caractères

- `\s` Signifie white space
- `\d` Signifie chiffre
- `\w` Signifie mot

Vous pouvez mettre les lettres en majuscule pour avoir la négation de ces regex, par exemple `\D` signifie "pas chiffre"

Exemple d'utilisation : `ab\s` prend en compte "ab c"

### Quantificateur

- `*` Signifie que la chaîne de caractère ou le caractère est présent 0 ou plus de fois dans la chaîne recherchée.

`A*` Signifie que le A est présent 0 ou plus de fois, on peut trouver les mots suivant avec ce regex : AAAAA ou rien

- **+** Signifie que la chaîne de caractère ou le caractère est présent 1 ou plus de fois dans la chaîne recherchée.

A+ Signifie que le A est présent 1 ou plus de fois, on peut trouver les mots suivant avec ce regex : AAAAA ou A

- **?** Signifie que la chaîne de caractère ou le caractère est présent 0 ou 1 fois.

A? Signifie que le A est présent 0 ou 1 fois

- **{n}** Signifie que la chaîne de caractère ou le caractère est présent exactement n fois

A{3} Signifie que le A est présent 3 fois, on peut trouver les mots suivant avec ce regex : AAA

- **{n,}** Signifie que la chaîne de caractère ou le caractère est présent n fois ou plus dans la chaîne recherchée.

A{3,} Signifie que le A est présent 3 fois ou plus, on peut trouver les mots suivant avec ce regex : AAA ou AAAAAA

- **{3,n}** Signifie que la chaîne de caractère ou le caractère est présent 3 fois, ou 4 ou 5 jusqu'à n dans la chaîne recherchée.

A{3,5} Signifie que le A est présent 3 fois ou 4 fois ou 5 fois, on peut trouver les mots suivant avec ce regex : AAA ou AAAA ou AAAAA

### Ancrage

- **^** Signifie le commencement de la chaîne / ligne ou autre
- **\A** Signifie le commencement de la chaîne
- **\Z** Signifie le commencement de la chaîne

- \< Signifie le commencement du mot
- \> Signifie la fin du mot

### Métacaractères

- **^abc** Prend en compte le début d'une chaîne qui commence par "abc".  
^voiture Les caractères restant peuvent être par exemple : voiturebelle ou voitureX
- **abc\$** Prend en compte la fin d'une chaîne qui se finit par "abc".  
voiture\$ Les caractères restant peuvent être par exemple : magasinvoiture ou Zvoiture
- **a.c** Prend en compte le début d'une chaîne qui commence par "a" et se termine par "c".  
a.n Les caractères restant peuvent être par exemple : avion ou avignon
- **abc|xyz** Prend en compte la chaîne "abc" ou "xyz".
- **ab[cC]** Prend en compte la chaîne "abc" ou la chaîne "abC"

### Les groupes

- **[.....]** Prend en compte n'importe quel des caractères inclus entre les crochets  
On peut par exemple écrire [d-t], ce qui signifie n'importe quel caractère entre "d" et "t"  
[abcde] signifie n'importe quel caractères entre "a", ou "b" ou "c" ou "d" ou "e".  
[3-6] signifie n'importe quel chiffre entre 3 et 6

- **[^.....]** Prend en compte n'importe quel des caractères sauf ceux inclus entre les crochets

On peut par exemple écrire `[d-t]`, ce qui signifie n'importe quel caractère sauf ceux entre "d" et "t"

`[^xyz]` signifie n'importe quel caractères sauf "x" ou "y" ou "z".

- **[0-9a-z]** Prend en compte tous les lettres et chiffres

- **(a|b)** Prend en compte soit le "a" ou "b"

`(X|y)` signifie soit "X" (x majuscule) ou "y" (y minuscule)

### Posix

- `[:upper:]`
- `[:lower:]`
- `[:alpha:]`
- `[:digit:]`
- `[:alnum:]`
- `[:punct:]`
- `[:blank:]`

## 10. Naviguer

Pour pouvoir naviguer dans l'arborescence de votre machine sous GNU/Linux, il existe des commandes qu'on écrit dans le shell.

### LiSt

```
ls [option] [file]
```

“ls” permet d’afficher les informations à propos des répertoires et fichiers dans votre répertoire courant. Pour mieux visualiser les fonctionnalités de la commande, réalisons quelques exemples.

Imaginons qu’on se trouve dans le répertoire racine /, on souhaite voir les fichiers et répertoires présent :

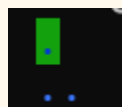
```
ls
```

“ls” tout seul liste que les répertoires présent. Cependant, nous avons vu en cours qu’il y a des “fichiers et répertoires cachés”, comment peut-on les voir ? Les options interviennent à ce moment-là.

Si on exécute :

```
ls -a
```

On remarque qu’on voit 2 répertoires en plus “.” et “..”



Il existe plusieurs options :

```
ls [option] [file]
ls -a : Affiche tous les fichiers et répertoires
ls -A : Affiche les fichiers cachés sauf . et ..
ls -d : Affiche les dossiers uniquement
ls -l : Affiche toutes les informations sur les dossiers et fichiers
ls -S : Affiche par taille (ordre croissant)
```

De plus, on peut voir le contenu d’un dossier avec la commande ls et en précisant le chemin absolu de ce dernier, ou chemin relatif si on a comme répertoire courant Desktop ou autre.

Par exemple, on souhaite voir si Toto a des dossiers sur son bureau mais à partir de la racine.

```
ls -a /mnt/c/Users/toto/Desktop
```

Pour un affichage plus complet, on peut utiliser :

```
ls -l
```

Qui affichera, les permissions, la taille, la date de création et le nom des fichiers et répertoires.

De plus on peut utiliser les jokers avec la commande ls, par exemple ls \*e, liste les fichiers et répertoires se terminant par “e”

On peut aussi citer la commande “pwd” (Print Working Directory) qui affiche le chemin absolu de votre répertoire courant.

```
pwd
```

```
/mnt/c/Users/marko
```

### Change Directory

```
cd [Chemin]
```

“cd”, change directory, est la commande phare pour naviguer, elle permet de se déplacer de répertoire en répertoire.

Avant de commencer à l'utiliser, il faut connaître la différence entre chemin absolu, chemin relatif.

**Chemin absolu** : C’est le chemin depuis la racine de l’arborescence.

exemple : /mnt/c/Users/toto/Desktop/Dossiers/Cours/L1

**Chemin relatif** : C’est le chemin depuis votre localisation

exemple : Dossiers/Cours/L1

Comment utiliser la commande cd ? Admettons qu’on se trouve dans le répertoire racine de la machine, /.

Nous voulons aller dans le répertoire Users. On sait qu’il se trouve dans le répertoire mnt, ensuite dans le répertoire c et enfin on trouve Users. Si on ne sait pas où se trouve ce dossier on peut utiliser la commande “ls” pour avoir une idée des répertoires présents dans votre répertoire courant

On commence par écrire la commande : `cd`

Ensuite on commence par écrire le premier répertoire qui nous mène à Users, puis le deuxième ainsi de suite, et on espace par `"/` les répertoires. Ce qui nous donne la commande suivante :

```
cd mnt/c/Users
```

Admettons qu'on souhaite aller dans le profil de Toto, puis dans son bureau et dans son dossier secret "Mot de passe" à partir du répertoire Users, nous exécutons la commande suivante :

```
cd Toto/Desktop/Mot de passe
```

Cependant, le shell renvoie un message d'erreur :

```
-bash: cd: too many arguments
```

Quelle est l'erreur ? Ce sont les espaces entre "mot" "de" "passe". Toto a nommé son fichier de cette manière (peu commode lorsqu'on utilise un shell), d'habitude on ne met pas d'espace lorsqu'on nomme des fichiers, répertoire et autre.

Pour palier à ce problème nous pouvons dire au shell que c'est un nom avec espace et de ne pas se préoccuper des espaces.

```
cd Toto/Desktop/"Mot de passe"
```

Un dernier exemple pour la route, maintenant qu'on se trouve dans le dossier "Mot de passe", on veut aller dans le répertoire qui contient le mot de passe de son compte en banque, imaginons une arborescence et trouvons le répertoire.

```
cd Perso/Argent/"Mes banques"/TotoBank|
```

Vous remarquerez qu'il a utilisé encore une fois des espaces, donc on met le nom du répertoire entre guillemets.

Voici quelques options avec cd et leurs significations :

```
cd .. : Remonte d'un niveau dans l'arborescence  
cd /  : Place dans la racine  
cd    : Place répertoire personnel
```

## 11. Manipuler les fichiers et répertoires

Il existe plusieurs commandes pour manipuler les répertoires et fichiers. Par manipuler on peut sous entendre : créer, supprimer, afficher, copier, déplacer, créer des liens etc.

### Make directory

Cette commande permet de créer un répertoire

```
mkdir [option] [file]
```

Exemple : Créons un répertoire se nommant Test sur le bureau tout en ayant comme répertoire Desktop

```
mkdir Testo
```

### Remove directory

Cette commande permet de supprimer un répertoire

```
rmdir [option] [directory]
```

Cependant, pour pouvoir exécuter cette commande il faut être le propriétaire et avoir les permissions.

Exemple : Supprimons un répertoire se nommant Testo sur le bureau tout en ayant comme répertoire courant /

```
rmdir /mnt/c/Users/Toto/Desktop/Testo
```



### Touch Create file

Cette commande permet de créer un fichier comme un .txt

```
touch [option] [file]
```

Exemple : Créons un fichier txt se nommant OuiOui à partir du bureau sur le bureau (pas besoin de spécifier l'emplacement)

```
touch OuiOui
```

Imaginons qu'on veut créer plusieurs fichiers, on peut procéder ainsi

```
touch Toto.txt Montagne.txt PC.txt Neige.txt
```

### MoVe

Cette commande permet de déplacer des fichiers, répertoires, mais aussi les renommer.

```
mv [option] [source] [destination]
```

Pour renommer un répertoire de "Toto1" à "BroBro1", il ne faut pas que BroBro1 existe.

```
mv Toto1 BroBro1
```

Pour déplacer le répertoire Toto vers MesAmis avec comme répertoire courant le bureau :

```
mv Toto MesAmis
```

Maintenant, on souhaite déplacer toutes les photos de Toto et moi vers le dossier MesAmis. Imaginons on a plus de 1000 photos en .png

On peut utiliser \*, on sait que ça signifie pas de caractères à une chaîne de caractères à n caractères. Donc si on écrit : \*.png, cela comprend tous les fichiers en .png quelque soit leurs appellations.

```
mv *.png MesAmis
```

Imaginons qu'il y a aussi des vidéos et des pdf ! Comment peut-on faire ?

Utilisons les regex, on sait que les fichiers peuvent comporter n'importe quel nom, et qu'il y a des mp4 et pdf. Donc : `*.{pdf,mp4}`

```
mv *.{pdf,mp4} MesAmis
```

### Copie

Cette commande permet de copier et de coller un fichier ou répertoire.

```
cp [option] [source] [destination]
```

Imaginons qu'on souhaite copier depuis mon bureau une image de Toto et Moi vers mon dossier MesAmis.

```
cp TotoMoi.png MesAmis
```

La différence entre move et copie, réside dans le fait qu'avec move on déplace le fichier de son répertoire initial vers un autre, alors qu'avec mv copie on fait juste un copier/coller du fichier sans lui attribuer un nouveau répertoire.

Imaginons qu'on veut copier plusieurs fichiers, on peut procéder ainsi :

```
cp Feu.pdf PC.png New
```

Maintenant je veux copier depuis mon bureau une image de Toto et Moi vers mon dossier MesAmis mais en lui changeant de nom juste dans le dossier MesAmis.

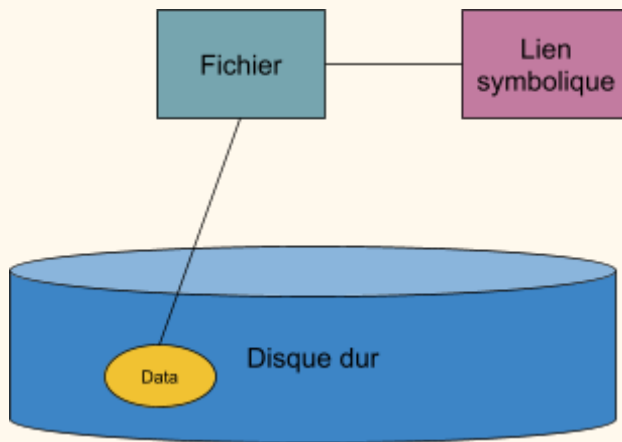
```
cp TotoMoi.png MesAmis/TotoMoiNeige.png
```

### Lien symbolique

Un lien symbolique est un fichier qui pointe vers un autre X, ce fichier crée peut être effacer et manipuler sans avoir d'incidence sur le fichier d'origine X. C'est un raccourci, nous pouvons prendre exemple avec les icônes des jeux sur votre bureau, ce sont que des liens qui pointent vers d'autres fichiers. Nous pouvons le voir grâce à la commande `ls *.lnk`

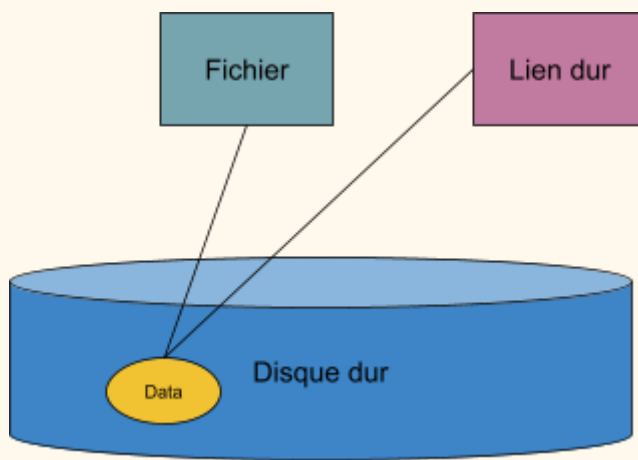
```
ls *.lnk  
Roblox Studio.lnk VALORANT.lnk
```

Lorsqu'on clique sur "VALORANT.lnk" ce fichier pointe vers un autre répertoire contenant un .EXE ou autre afin de lancer le jeu.



### Lien dur

Un lien dur permet de donner plusieurs chemins différents vers un même fichier. La différence avec un lien symbolique réside dans le fait qu'un lien dur pointe directement sur les données du disque dur, contrairement à un lien symbolique qui pointe vers un fichier qui lui pointe vers les données du disque dur.



### Crée un lien vers un répertoire ou fichier

La commande qui permet de créer un lien est la suivante.

```
ln [option] [fichier source] [nom_lien]
```

Voici comment créer un lien symbolique et dur.

```
ln -s [fichier source] [nom lien] : Créer lien symbolique  
ln [fichier source] [nom lien] : Créer lien dur
```

### Archiver

Cette commande permet de regrouper plusieurs fichiers ou répertoires en un seul fichier dit “archive”.

```
tar [option] [file]  
tar -cf = créer archive  
tar -df = supprimer archive  
tar -xf = extraire archive  
tar -tf = lister archive  
tar -rf = ajouter un fichier dans archive
```

### Compresser

Cette commande permet d’appliquer un algorithme sur un répertoire ou plusieurs fichiers afin de les archiver en un unique fichier et compresser (diminuer leurs tailles).

```
gzip [fichier]  
bzip2 [fichier]  
  
Pour les décompresser :  
  
ungzip [fichier]  
unbzip2 [fichier]
```

### More

Cette command permet d'afficher le contenu d'un fichier (limité)

```
more [option] [file]
```

### Less

Cette commande permet d'afficher le contenu d'un fichier (permet de naviguer dans ce dernier)

```
less [option] [file]
```

### Cat

Cette commande permet de lire et d'afficher le contenu d'un fichier texte

```
cat [option] [file]
```

On peut lire plusieurs fichiers en même temps

```
cat test.txt test2.txt
```

Cette command permet aussi de concaténer deux fichier ensemble

```
cat tot.txt >> final.txt
```

```
cat tsts.txt titi.txt toto.txt > final.txt
```

Pourquoi utilise-t-on ">>" et ">" ?

On utilise > pour rediriger le contenu d'un fichier vers un autre, de ce fait le contenu du fichier "receveur" est effacé.

On utilise >> pour rediriger le contenu d'un fichier vers la fin d'un autre, de ce fait le contenu du fichier "receveur" reste intact.

On peut aussi écrire à la fin d'un fichier avec la commande cat.

```
cat >> t.txt
```

On écrit le texte puis on sort avec CTRL+D

### WC

Cette commande permet de compter le nombre de lignes, mots, caractères, bytes etc.

```
wc [option] [file]
```

```
wc -l [file] : Compte ligne  
wc -w [file] : Compte mot  
wc -m [file] : Compte caractère
```

### Head

Cette commande permet d'afficher le contenu des 10 premières lignes d'un fichier

```
head [option] [file]
```

On utilise le plus souvent l'option -n qui permet de spécifier le nombre de lignes.  
Dans cet exemple, on affiche les 5 premières lignes de quizz.txt

```
head -n 5 quizz.txt
```

### Tail

Cette commande permet d'afficher le contenu des 10 dernières lignes d'un fichier

```
tail [option] [file]
```

On utilise le plus souvent l'option -n qui permet de spécifier le nombre de lignes.  
Dans cet exemple, on affiche les 20 dernières lignes de quizz.txt

```
tail -n 20 quizz.txt|
```

### Combinaison head & tail

La combinaison des deux commandes permet d'afficher une ligne précise ou un lot de ligne souhaité. Pour ce faire, on joue en premier avec les premières lignes grâce à "head" ensuite on essaye de jouer avec "tail". Pour transmettre l'output de head, on utilise le "pipe" |, il sert à dire qu'on va utiliser l'output de la commande précédente pour appliquer la nouvelle commande.

Exemple : On souhaite afficher la 11ème ligne du fichier

On sait que head affiche les 10 premières lignes, mais on souhaite la 11ème ligne uniquement. Donc on va étendre head à 11, et utiliser tail pour avoir que la 11ème ligne.

Contenu du fichier

```
Première
Dexuième
Troisi
Quatrio
Cinqui
Sixiem
Septi
Huitieme
Neuvieme
Dixième
Onzienme
Deuxi
13
14
```

On applique head

```
head -n 11 quizz.txt
```

```
Première
Dexuième
Troisi
Quatrio
Cinqui
Sixiem
Septi
Huitieme
Neuvieme
Dixième
Onzienme
```

Maintenant on souhaite garder “Onzienme”. On peut utiliser tail, on sait que la commande affiche que les 10 dernières lignes, donc de “Onzienme” à “Première”. Et on remarque que la 11ème ligne est la première en partant du bas, donc on peut juste afficher la première des dernières lignes avec tail.

```
head -n 11 quizz.txt | tail -n 1
```

On obtient bien “Onzienme”

```
Onzienme
```

Le rôle du | est de dire qu’on veut appliquer “tail -n 1” au output de “head -n 11”

## 12. Les permissions

Sur GNU/Linux, les permissions sont au nombre de 3, la permission de lecture, écriture et d'exécution.

r : La permission de lecture permet de lire le contenu d’un fichier et répertoire.

w : La permission d’écriture permet de modifier le contenu d’un fichier et répertoire.

x : La permission d’exécution permet d’exécuter le fichier.



Chaque permissions est associé à une valeur :

r = 4

w = 2

x = 1

Il y a 3 types de profil :

u = user

g = groupe

o = others

Les permissions sur un fichier sont représenté par groupe de 3, chaque profil a les 3 permissions

-rwxrwxrwx

Les permissions sur un dossier sont représenté par groupe de 3, comme pour le fichier

drwxrwxrwx

Que représentent réellement ces chaînes ? Ces chaînes montrent quelles permissions sont “activés” pour les différents profils. On peut enlever ou octroyer certaines permissions à d’autres avec des commandes.

-rw-rw-rw- : tous les profils n’ont pas la permission d’exécuter le fichier.

-rwxrwxrwx- : tous les profils sauf les others ont la permission d’exécuter le fichier.

-rwx-w-w- : le profil u a tous les droits, les profils u et o ont le droit d’écriture.

drwx----- : u a tous les droits, o et u n’ont pas de droits. sur le répertoire.

On remarque que “-” signifie que le droit n’est pas accordé. De plus on voit un paterne :

rwx | rwx | rwx

user | group | other

Il faut noter que les droits d’exécution des fichiers sont automatiquement omis sur les distributions GNU/Linux pour des raisons de sécurité.

Pour manipuler les permissions sur les répertoires et fichiers, il existe deux commandes : “chmod” et “umask”.

## Chmod

Cette commande permet de changer les permissions d'un fichier ou répertoire.

```
chmod [option] [profil][opérateur] [file]
```

L'utilisation de cette commande est simple voici des exemples :

```
chmod o-w,g+r| mdp
```

Cette commande enlève le droit d'écriture aux others, donne le droit de lecture au groupe sur le dossier mdp

```
chmod u+rw,g+rx-w,o+r-wx notes|
```

Cette commande :

Ajoute au user tous les droits

Ajoute au groupe le droit de lecture et exécution mais enlève le droit d'écriture

Ajoute aux others le droit de lecture et enlève les droits d'écritures et exécution

Dans ces exemples on a utilisé les commandes sans les valeurs. Pour plus de visibilité, on peut utiliser ces mêmes commandes avec les différentes valeurs des droits. Il nous faut d'abord la chaîne qui représente les permissions du fichier ou répertoire.

Imaginons qu'on part d'une chaîne où tous les profils ont tous les droits sur le dossier :

```
drwxrwxrwx
```

Nous avons vu les différentes valeurs pour chaque droit. Ici on peut traduire cette chaîne par:

```
7 7 7
```

Pourquoi ?  $(r = 4) + (r = 2) + (x = 1) = 7$

Lorsque toutes les permissions sont accordées à un profil on note la valeur 7.

```
dr-x-wxrw-
```

$$7-(2) \ 7-(4) \ 7-(1) \rightarrow 5 \ 3 \ 6$$

ou

$$(4+1) \ (2+1) \ (4+2) \rightarrow 5 \ 3 \ 6$$

Soit on part de la valeur 7 puis on soustrait les “-” donc les valeurs qui manquent.

Soit on additionne les valeurs visibles sur la chaînes

Exemple d'utilisation :

On a : d~~r~~w~~x~~r~~w~~x~~r~~w~~x~~

On veut : d~~r~~-~~x~~-~~w~~~~x~~r~~w~~-

On remarque que :

u n'a pas de w

g n'a pas de r

o n'a pas de x

Donc :

$$u = 7-(2) = 5$$

$$r = 7-(4) = 3$$

$$o = 7-(1) = 6$$

Cela signifie qu'on veut que 7 7 7 devient 5 3 6

On peut utiliser le chmod :

```
chmod 536 Toto.txt
```

On remarque que l'écriture est plus légère et compréhensible si on connaît ses valeurs !

Traduisons : `chmod u+rwx,g+rx-w,o+r-wx notes`

Admettons que la chaîne de permission est : d— — — (aucune permission pour tous  $\Leftrightarrow$  a-rwx)

u : on rajoute read,write,execute, donc : 4+2+1 = 7

g : on ajoute read,execute, et on enlève write, mais il est déjà enlever donc on fait rien : 4+1 = 5

o : on ajoute read, on enlève write et execute, donc : 4

```
chmod 754 Toto.txt
```

Tout dépend encore une fois de la chaîne de permissions d'origine mais le principe est le même.

### Umask

Cette commande permet de définir les permissions qu'on ne veut pas donner aux prochains fichiers et répertoires.

```
umask [option] [profil][opérateur] [file]
```

Le principe est similaire à chmod, cependant on définit les permissions d'une autre manière.

Pour définir les droits on applique l'algorithme suivant :

Pour les fichiers :  $666 - \text{chmod} = \text{umask}$

Pour les dossiers :  $777 - \text{chmod} = \text{umask}$

Donc si la chaîne de droits est : `dr-x-wxrw-`

$\text{chmod} = 536$

$\text{umask} = 777 - 536 = 241$

En résumé pour trouver le umask il suffit de trouver le chmod à partir de la chaîne de permissions et appliquer l'algorithme.

```
umask 241 Toto.txt
```

## 13. Chercher

### Find

Cette commande permet de parcourir une arborescence pour trouver un répertoire ou fichier

```
find [où] [option] [option]
```

A noter qu'on peut aussi utiliser les expressions régulières pour trouver le fichier.

Exemple : Imaginons qu'on souhaite trouver un fichier pdf dans le dossier MesAmis, tout en ayant comme répertoire courant le bureau

```
MesAmis/Feu.pdf
MesAmis/Fiche de révision Syst1.pdf
MesAmis/Impri.pdf
MesAmis/Kosovo.mp4
MesAmis/Montagne.png
MesAmis/Neige.png
MesAmis/New/Feu.pdf
MesAmis/New/PC.png
MesAmis/PC.png
MesAmis/Ski.mp4
MesAmis/Toto.png
MesAmis/Toto.txt
MesAmis/TotoMoi.png
MesAmis/TotoMoiNeige.png
```

Maintenant, si on souhaite peaufiner notre recherche, on peut spécifier la chaîne de caractères qui fait office de nom du fichier qu'on souhaite trouver. Pour cela on utilise les expressions régulières.

Exemple : On souhaite trouver que les fichiers commençant par T ou K dans le dossier MesAmis

```
find -type f -name "[KT]*"
```

```
./Kosovo.mp4
./Toto.png
./Toto.txt
./TotoMoi.png
./TotoMoiNeige.png
```

Exemple : On souhaite trouver que les fichiers commençant par tout sauf par T dans le dossier MesAmis

```
find -type f -name "[^T]*"
```

```
./Entraînement ALG01.txt
./Feu.pdf
./Fiche de révision Syst1.pdf
./final.txt
./Impri.pdf
./Kosovo.mp4
./Montagne.png
./Neige.png
./New/Feu.pdf
./New/PC.png
./PC.png
./question_cours
./quizz.txt
./Ski.mp4
./t.txt
```

Exemple : On souhaite trouver que les fichiers commençant par tout sauf par C et se terminant par un chiffre dans le dossier MesAmis

```
find -type f -name "[C]*[0-9]"
```

```
./Coucou9.mp3
```

Quelques options type très utile :

```
-type l : Le type est un lien
-type d : Le type est un répertoire
-type f : Le type est un fichier
```

Quelques options utiles :

```
-name [nom] : le nom
-path : l'arborescence
-mindepth [chiffre] : profondeur min
-maxdepth [chiffre] : profondeur max
-empty : fichier vide
-perm : les permissions
```

## Grep

Cette commande permet de rechercher une chaîne de caractère spécifique avec des expressions régulières dans un fichier.

```
grep [option] [expression régulière] [file]
```

```
grep -w [expression régulière] [file] : un mot
grep -n [expression régulière] [file] : numéro de la ligne
grep -c [expression régulière] [file] : nombre d'apparition
grep -r [expression régulière] [file] : récursif
grep -E [expression régulière] [file] : interpréter l'expression recherchée comme une
expression régulière étendue
```

Comment utilise-t-on cette commande ?

Dans un premier temps, on doit afficher le contenu du fichier, donc on utilise la commande cat, ensuite on cherche parmi le output de cat ce qu'on souhaite.

Reprenons le fichier quizz.txt, on souhaite rechercher les mots se finissant par "eme".

```
Première
Deuxième
Troisi
Quatrio
Cinqui
Sixiem
Septi
Huitieme
Neuvieme
Dixième
Onzième
Deuxi
13
14
```

```
cat quizz.txt | grep "eme$"
```

```
Huitieme
Neuvieme
```

Si on souhaite chercher les mots commençant par “S”.

```
cat quizz.txt | grep "^S"
```

```
Sixiem
Septi
```

Et si on ne veut pas voir les mots mais les compter ? On peut utiliser une troisième commande qui wc, elle permet de compter les mots, lignes etc.

```
cat quizz.txt | grep "^S" | wc -l
```

On a comme output **2**

Explication : Dans un premier temps on affiche le contenu de quizz.txt, puis on cherche les chaînes de caractères commençant par S, puis on compte le nombre de lignes du output de grep.

A noter qu'on peut directement passer par grep sans faire “cat” mais pour des raisons d'apprentissage et de compréhension des commandes, logique, nous privilégions l'écriture en plusieurs temps.

## 14. Questions possibles à l'examen

Comment lister tous les fichiers d'un répertoire avec ls et qui se termine par Z.

Je veux créer un dossier qui se nomme "mdp" qui est composé des fichiers amazon facebook unica, quelle commande permet de le faire ?



Comment ajouter la date et l'heure courante à la fin d'un fichier "mdp" créé à la question précédente

Qu'affiche la commande HOME ? PWD ? DISPLAY ?

On suppose que vous êtes dans votre répertoire personnel qui contient vos scripts shell dont le nom se termine par .sh. Vous voulez ajouter à votre groupe la possibilité d'exécuter vos scripts. Quelle commande devez-vous taper dans votre terminal ?

On suppose que vous êtes dans votre répertoire personnel qui contient vos mots de passe dont le nom se termine par "mdp". Vous voulez enlever à votre groupe et aux autres la possibilité de lire, ouvrir et exécuter. Quelle commande devez-vous taper dans votre terminal ?

Déterminez la valeur du umask avant la création des répertoire suivants :

drwxr-xr-- tot1  
 dr--r----- tot2  
 d----- tot3  
 d-wxr--rwx tot4  
 drw-rw-rw- tot5  
 drwx---rwx tot6  
 d-w--w-rw- tot7

Quels sont les droits sur les répertoires pour les valeurs umask suivantes (donner la chaîne de droits) :

345  
 136  
 777

517

130

Un fichier TOTO a les droits suivants : -rwxr-xr-x

Écrivez la commande qui permet de les passer à -r--rw-rw-

Écrivez la commande qui permet de les passer à ----r--r-

Écrivez la commande qui permet de les passer à -rw----rwx

Écrivez la commande qui permet de les passer à --wx--xrw sans connaître les droit sur le fichier

Écrivez la commande qui permet de les passer à -rwx---rw- sans connaître les droit sur le fichier

Écrivez la commande qui permet de les passer à --w----r-x sans connaître les droit sur le fichier

Écrivez la commande qui permet d'afficher l'ensemble des professeurs dont le nom commence par la lettre M (le nom est le premier mot de chaque ligne du fichier listeProfs.txt).

Comment trier les fichiers/dossiers par taille avec ls et qui se termine par O ou o ?

A quelle sortie correspond > ? A quelle sortie correspond >>>> ? A quelle sortie correspond 2> ?

Écrivez la commande qui permet de trouver uniquement les fichiers dans / ayant comme extension .mp4, avec une profondeur de 6, et une taille supérieure à 9 Go.

Citer 2 variables d'environnement génériques sous linux et les expliquer.

A quoi correspond la commande suivante :

```
find -type d -maxdepth 3 -name "[0-9][0-9][A-Z][A-Z][A-Z](0|1)$"
```

Créer les fichiers suivants en une seule ligne de commande : France PACA, COMORE, VoyaGE

Donner les valeurs numériques des permissions et les umask correspondant aux trois dossiers suivants :

```
drwx---r--
```

```
d--x--x-w-
```

```
d-wx-wx-wx
```

Compresser le dossier suivant : TotoPhoto et décompressez le sur une même ligne de commande

Archiver le dossier suivant : Toto Video

Afficher avec ls et les jokers :

Les fichiers commençant par "LoLo"

Les fichiers terminant par un "m"

Le fichier France PACA

On veut compter le nombre de lignes contenant la chaîne mdp dans le fichier perso.txt. Écrivez la commande qui produira le résultat.

Le fichier commande.txt contient la liste de toutes les commandes de Toto. Chaque ligne commence par un site.

Écrivez la commande qui permet d'afficher les 10 premiers sites.

Celle qui permet d'afficher tous les sites entre les lignes 28 et 38.

Celle qui permet d'afficher la ligne 17.

Expliquez ce que produit la commande (on précise que > est l'invite de la commande):

~> grep "M\$" commande.txt > M.txt

Écrivez la commande qui permet d'afficher le nombre de sites dont le nom commence par la lettre Z.

Écrivez la commande qui affiche tous les noms des fichiers qui commencent par une voyelle majuscule, puis contient 2 chiffres et qui contiennent exactement 3 consonnes suivie d'un b ou d'un q.

Comment lister tous les fichiers d'un répertoire avec ls ?

Que fait la commande suivante ? En quoi cette commande est-elle dangereuse ?

sudo rm -rf /

Comment copier le contenu du dossier ~/.config vers ~/backup ? Quels fichiers sont généralement dans le .config ?

Quelle commande permet d'éditer manuellement (avec un éditeur de texte) le fichier ~/.zshrc ?

Que fait ce fichier ?

Écrivez la commande qui permet d'afficher les lignes 12 à 19 du fichier listeProfs.txt

Comment (en une seule commande) relancer la même commande que l'exercice précédent et mettre le résultat dans le fichier d.txt ? (il ne faut pas réécrire la commande de la question 16 !)

Écrivez la commande qui permet d'afficher l'ensemble des professeurs dont le nom commence par la lettre C et se termine par e (le nom est le premier mot de chaque ligne du fichier listeProfs.txt)

Écrivez au moins 2 commandes qui permettent de lire le fichier d.txt

Comment trier les fichiers/dossiers par taille avec ls ?

Écrivez la commande qui permet de trouver uniquement les fichiers dans / ayant comme extension .iso, avec une profondeur de 2, et une taille supérieur à 1Go

Écrivez la commande qui donne les droits d'exécutions à tout le monde sur le fichier script.sh

Comment compresser dans une archive Backup.tar.gz le dossier ~/Backup ?

Comment compresser avec bzip2 le dossier ~/.config ? Et comment le décompresser ensuite ?

Comment afficher le contenu de l'archive Backup.tar.gz ?

Créer les fichiers : France, ALLEMAGNE, Italie, Bhoutan, Islande, Espagne

Afficher avec ls et les jokers :

Les fichiers contenant "an"

Les fichiers terminant par un "e"

Le fichier France

Le fichier contenant un chiffre

Que font les commandes suivantes : man, grep, cd, cut, ls, tar ?

On veut compter le nombre de lignes contenant la chaîne examen dans le fichier réussir.txt.  
Écrivez la commande qui produira le résultat.

Le fichier listeL1.txt contient la liste de tous les étudiants du cours de Système 1. Chaque ligne commence par un nom de famille.

Écrivez la commande qui permet d'afficher les 5 premiers noms.

Celle qui permet d'afficher tous les noms entre les lignes 25 et 35.

Celle qui permet d'afficher la ligne 11.

Expliquez ce que produit la commande (on précise que > est l'invite de la commande):

```
~> grep "^G" listeL1.txt | wc -l > G.txt
```

Écrivez la commande qui permet d'afficher le nombre d'étudiants dont le nom commence par la lettre L

## Examen d'entraînement

Voici un petit test fait par moi même qui regroupe un peu de tout. Il vous permettra de vous entraîner. N'hésitez pas à en faire vous même, c'est en pratiquant qu'on apprend l'informatique!

### Partie 1 : Questions de cours

- 1) Qu'est ce qu'un noyau ?
- 2) Définissez GNU/Linux.

- 3) Expliquez l'expression "distribution Linux", donnez moi 3 exemples.
- 4) Qu'est-ce qu'un shell ? Que permet-il de faire ? Où se trouve-t-il dans l'architecture d'une machine ?
- 5) Que permet de faire une commande intrinsèque ? Donnez moi 2 exemples.
- 6) Qu'est-ce qu'un script ?
- 7) En quoi les permissions sur une machine jouent un rôle majeur dans la sécurité ?
- 8) Est-ce que le droit d'exécution est omis sur les distributions Linux ? Si oui, pourquoi ?
- 9) Quelle est l'utilité du fichier /bin ? /tmp ? Qu'est-ce que / ?
- 10) Définissez la virtualisation, citez moi 2 "types de virtualisations" et l'utilité de ces dernières
- 11) Qu'est-ce que : new.vmx ?
- 12) Qu'est-ce qu'un serveur X ?
- 13) Qu'est-ce qu'un protocole ? Donnez moi 2 protocoles et leurs usages/utilités
- 14) Qu'est-ce que SSH ?

## Partie 2 : Exercice

- 1) On veut compter le nombre de mot dans un fichier se nommant dictionnaire", écrivez une commande qui produira le résultat
- 2) On veut compter le nombre de fois que la chaîne "heureux" apparaît dans un fichier se nommant League Of Legends, écrivez une commande qui produira le résultat
- 3) Le fichier commande.txt contient la liste de toutes les commandes de Toto. Chaque ligne commence par un site.  
Écrivez la commande qui permet d'afficher les 10 premiers sites.  
Celle qui permet d'afficher tous les sites entre les lignes 28 et 38.  
Celle qui permet d'afficher la ligne 17.
- 4) Expliquez ce que produisent les commandes suivante :  
cat Jeu | grep "l\$" | wc -l > Oui.txt  
cat Jeu | grep "{2}W\*" | wc -l >> Jeu.txt
- 5) Écrivez la commande qui permet d'afficher le nombre de sites dont le nom commence par la lettre Z dans le fichier Site.txt
- 6) Écrivez la commande qui permet d'afficher l'ensemble des chaînes de caractère qui commence avec un code postal de 4 chiffres, contient ensuite le nom de la ville, et que

cette ville se termine par z. ( Le code postale est la première chaîne de caractère du fichier ville.txt)

- 7) Je veux crée un fichier “gâteau” qui contient le contenu de mes fichiers suivants :  
framboise fraise myrtille
- 8) Qu’affiche la commande “echo DISPLAY” lorsqu’elle est invoquée dans le terminal ?  
Qu’affiche la commande “echo \$HOME” lorsqu’elle est invoquée dans le terminal ?
- 9) On suppose que vous êtes dans votre répertoire personnel qui contient vos mots de passe dont le nom se termine par "mdp". Vous voulez enlever à votre groupe et aux autres la possibilité de lire, ouvrir et exécuter. Quelle commande devez-vous taper dans votre terminal ?
- 10) Déterminer les valeurs umask avant la création des répertoires suivant :  
dr--rw-rw- tot5  
d-wx---rwx tot6  
d----w-rw- tot7  
d-----rw- tot8
- 11) Quels sont les droits sur les répertoires pour les valeurs umask suivantes (donner la chaîne de droits) :  
  
777  
130
- 12) Un fichier TOTO a les droits suivants : -rwxr-xr-x.  
Écrivez la commande qui permet de les passer à -r--rw-rw-  
Écrivez la commande qui permet de les passer à ----r--r-
- 13) Écrivez la commande qui permet de les passer à --w----r-x sans connaître les droits sur le fichier
- 14) Créez un répertoire se nommant Voyage, et déplacez-le dans votre dossier Photo sur votre bureau. Votre répertoire courant est Desktop
- 15) Supprimer maintenant le dossier Photo
- 16) Créez deux fichiers texte qui se nomment Fort et EAU ensuite créez un dossier Feu sur la même ligne de commande
- 17) Copiez le fichier Fort dans Feu en le renommant Lave
- 18) Archiver maintenant Feu



- 19) Créez un lien dur du fichier EAU
- 20) Enfin renommer le dossier Feu en Herbe

## Conclusion

J'espère que cette fiche de révision vous a permis de mieux comprendre les différentes notions présentées en cours de Système 1 : UNIX et Programmation Shell. Des modifications sont à venir sur cette fiche notamment des rajouts de commandes en plus, d'autres définitions, et d'autres exercices avec des exemples concrets.

Je rappelle que ce document ne remplace pas vos leçons, vos TP et TD. Vous devez être assidue dans votre apprentissage et en cours afin de réussir. De plus, il est destiné à un usage purement personnel, il n'est pas destiné à une activité lucrative. Ce document contient des connaissances trouvables sur Internet et qui font partie du domaine public.