

# CHAPITRE 8 : Matériel et systèmes d'exploitation 2

---

1	Modèle d'architecture séquentielle (von Neumann).....	2
1.1	Histoire .....	2
1.2	Le marché .....	5
1.3	Les constituants d'une machine .....	6
1.4	Architecture de von Neumann .....	8
2	Systèmes d'exploitation .....	9
2.1	Les fonctions d'un système d'exploitation .....	9
2.2	Les familles de système d'exploitation.....	9
2.3	L'interface "ligne de commande" .....	10
2.3.1	Définition.....	10
2.3.2	Système de fichiers sous Unix (et Linux) .....	11
2.3.3	Interface en ligne de commande d'Unix / Linux.....	14
2.3.4	Les commandes de base pour Unix / Linux.....	14
2.3.5	Gérer les droits d'accès d'un fichier ou d'un répertoire.....	16

# CHAPITRE 8 : Matériel et systèmes d'exploitation 2

---

## 1 Modèle d'architecture séquentielle (von Neumann)

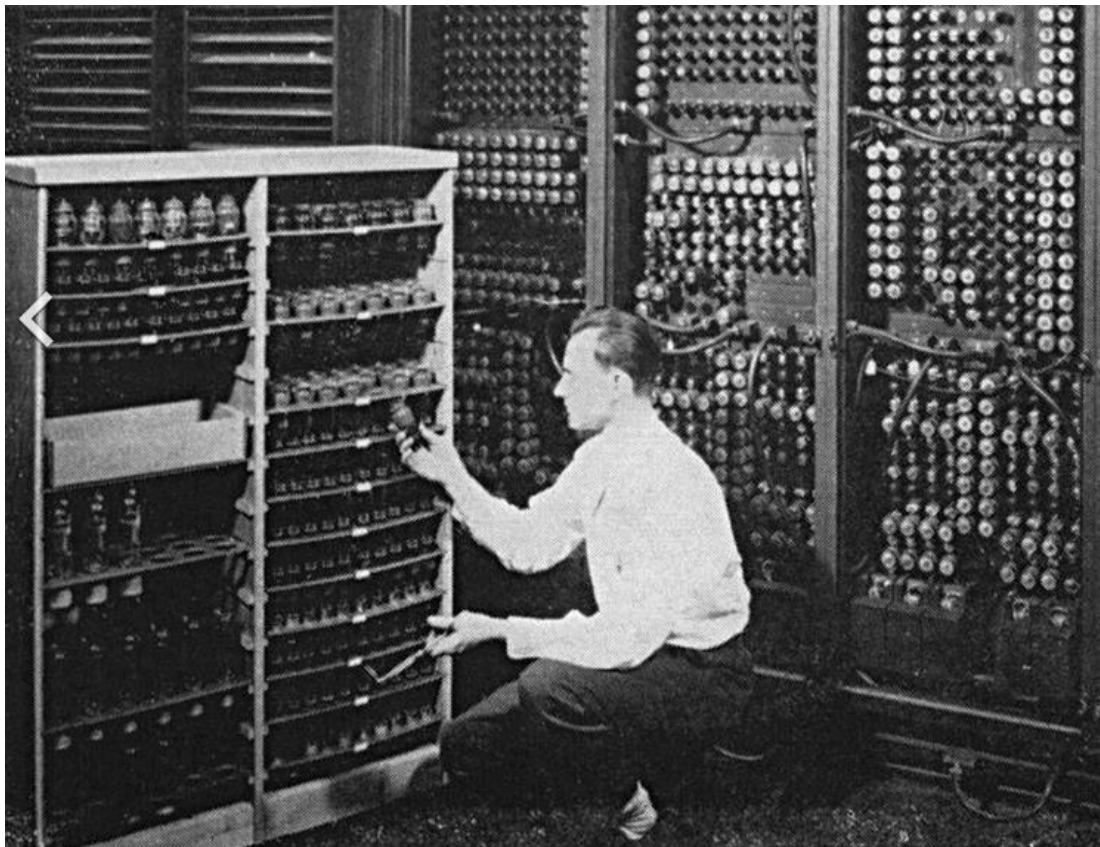
### 1.1 Histoire

- Le matériel

**1943** • Mise au point aux Etats-Unis, l'**ENIAC** (*Electronic Numerical Integrator And Computer*) est une machine à calculer programmable (à des fins militaires).

Les composants sont des tubes électroniques à vide.

Le programme est réalisé en câblant des fils. Les entrées sorties se font sur des cartons perforés.



*Un technicien en train de changer un tube de l'ENIAC*

L'ENIAC est une grosse machine. Il contient 17 468 tubes à vide.  
Son poids est de 27 t pour des dimensions de 2,4 × 0,9 × 30,5 m.



*Un tube à vide*

- 1945 • Le mathématicien, informaticien, physicien américain John von Neumann (1903 – 1957) conçoit une architecture d'ordinateur (architecture qui reste encore théorique en 1945).

L'architecture de von Neumann extrêmement innovante pour l'époque, est à la base de la conception de la plupart des ordinateurs aujourd'hui.

Les ordinateurs construits avec l'architecture de von Neumann sont constitués de quatre composants :

1. L'**unité arithmétique et logique** (UAL) ou unité de traitement, qui effectue les *opérations* de base.
2. L'**unité de contrôle**, qui est chargée du *séquençage* des opérations, c'est à dire l'enchaînement des opérations les unes après les autres.
3. La **mémoire**, qui contient *à la fois les données et le programme* qui indique à l'unité de contrôle quels calculs faire sur ces données. La mémoire se divise en mémoire vive (programmes et données en cours de fonctionnement) et mémoire de masse (programmes et données de base de la machine).
4. Les **dispositifs d'entrées-sorties**, qui permettent de communiquer avec le monde extérieur.

Le fait que l'unité arithmétique et logique soit *séparée* de l'unité de contrôle est novateur.  
Autre élément novateur : la mémoire contient *à la fois les données et le programme*.



John von Neumann

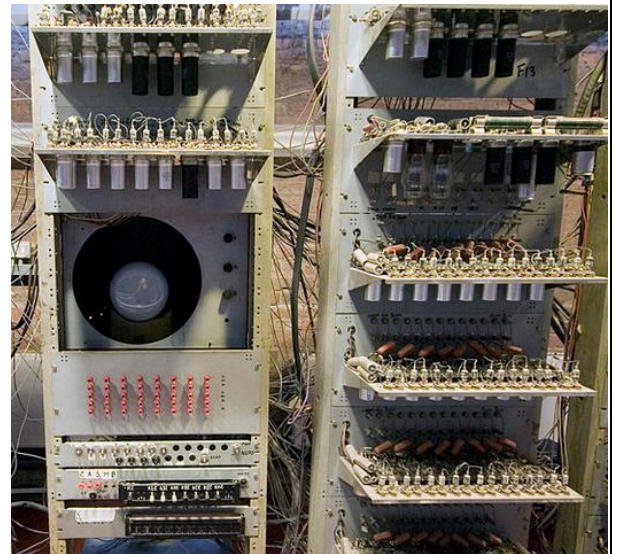
- 1948 • La **Small-Scale Experimental Machine** (SSEM) est la première machine réelle à architecture de von Neumann.

Son poids est de 1 t pour des dimensions de 2,2 × 5,2 m.

La SSEM utilisait une mémoire de 32 mots de 32 bits.

Le premier programme écrit pour cette machine trouva le plus grand diviseur propre<sup>1</sup> de  $2^{18}$  en testant chaque entier de  $2^{18}-1$  à 1 dans l'ordre décroissant.

Il lui fallut 52 minutes pour donner la réponse correcte, 131 072. La SSEM effectua 3,5 millions opérations.



Écran et périphérique d'entrée du SSEM

- Les systèmes d'exploitation

Un **système d'exploitation**, ou OS pour *Operating System*, définit un ensemble de programmes chargé d'établir une relation entre les différentes ressources matérielles, les applications et l'utilisateur.

<sup>1</sup> **Diviseur propre** : un diviseur propre d'un entier naturel n est un diviseur de n autre que lui-même et que 1.

Il reçoit des **demandes d'utiliser les ressources de l'ordinateur** de la part des logiciels applicatifs. Par exemple : demande d'accès à la mémoire vive, aux mémoires de masse, demande de ressources de calcul du processeur, demande de communication vers des périphériques ou vers le réseau.

Année d'apparition	Nom	Famille	Éditeur	Matériel pris en charge	Utilisation	Graphique
1969	UNIX		Bell Labs			
1973	Réécriture d'UNIX en langage C		Dennis Ritchie			
1973	SYSMIC		R2E	Micral	ordinateurs personnels, stations de travail	
1977	VMS		DEC	VAX, DEC Alpha, Hewlett-Packard	serveurs, ordinateurs centraux	
1978-1985	CP/M		Digital Research	Amstrad CPC, Commodore 128, TRS-80	ordinateurs personnels	
1981-1990	DOS		IBM & Microsoft	Compatible PC	ordinateurs personnels	
1982	QNX		<i>Quantum Software Systems</i>	compatibles PC, MIPS, PowerPC, ARM	systèmes embarqués, automates industriels	✓
1984	Mac OS		Apple	Apple Macintosh	ordinateurs personnels	✓
1985	TOS		Atari	Atari ST, Eagle, Medusa, Hades, Milan, FireBee, ColdFire	ordinateurs personnels	✓
1985	AmigaOS		Commodore	Commodore Amiga, PowerPC	ordinateurs personnels et consoles de jeu	✓
1987-2006	OS/2		IBM et Microsoft	PS/2 et Compatible PC	ordinateurs personnels	✓
1989	Symbian OS		Symbian ltd	Nokia, Siemens, Samsung, Panasonic	téléphones mobiles, smartphone, assistants personnel	✓

1990	Windows 3	Windows	Microsoft	Surcouche logicielle à DOS	ordinateurs personnels	✓
1991	Solaris	Unix	Sun	machines de Sun et x86/64	serveurs, stations de travail, superordinateurs	✓
1991	GNU/Linux	Unix	(communautaire)	nombreux	tous	✓
1991	Windows NT	Windows	Microsoft	Compatible PC	serveurs, stations de travail, ordinateurs personnels	✓
1994	FreeBSD	Unix	(communautaire)	nombreux	tous	✓
1996	Windows CE	Windows	Microsoft	x86, MIPS, ARM	smartphone, assistants personnels, automates industriels	✓
1999	Mac OS X	Unix	Apple	x86, PowerPC d'Apple	ordinateurs personnels, serveurs, station de travail	✓
1999	BlackBerry OS		Research In Motion	téléphones mobiles de BlackBerry	smartphone	✓
2007	Android	Unix	consortium Open Handset Alliance	produits des fabricants de l'Open Handset Alliance	smartphone, tablette électronique	✓
2007	iOS	Unix	Apple	appareils d'Apple (iPhone, iPod, iPad...)	smartphone, tablette électronique, baladeur numérique	✓

## 1.2 Le marché

- **Ordinateurs de bureau** (août 2021)

Linux : 2%    Chrome OS : 2%    Autres : 4%    **OS X : 16%**    **Windows : 76%**

Source : <https://gs.statcounter.com/os-market-share/desktop/worldwide>

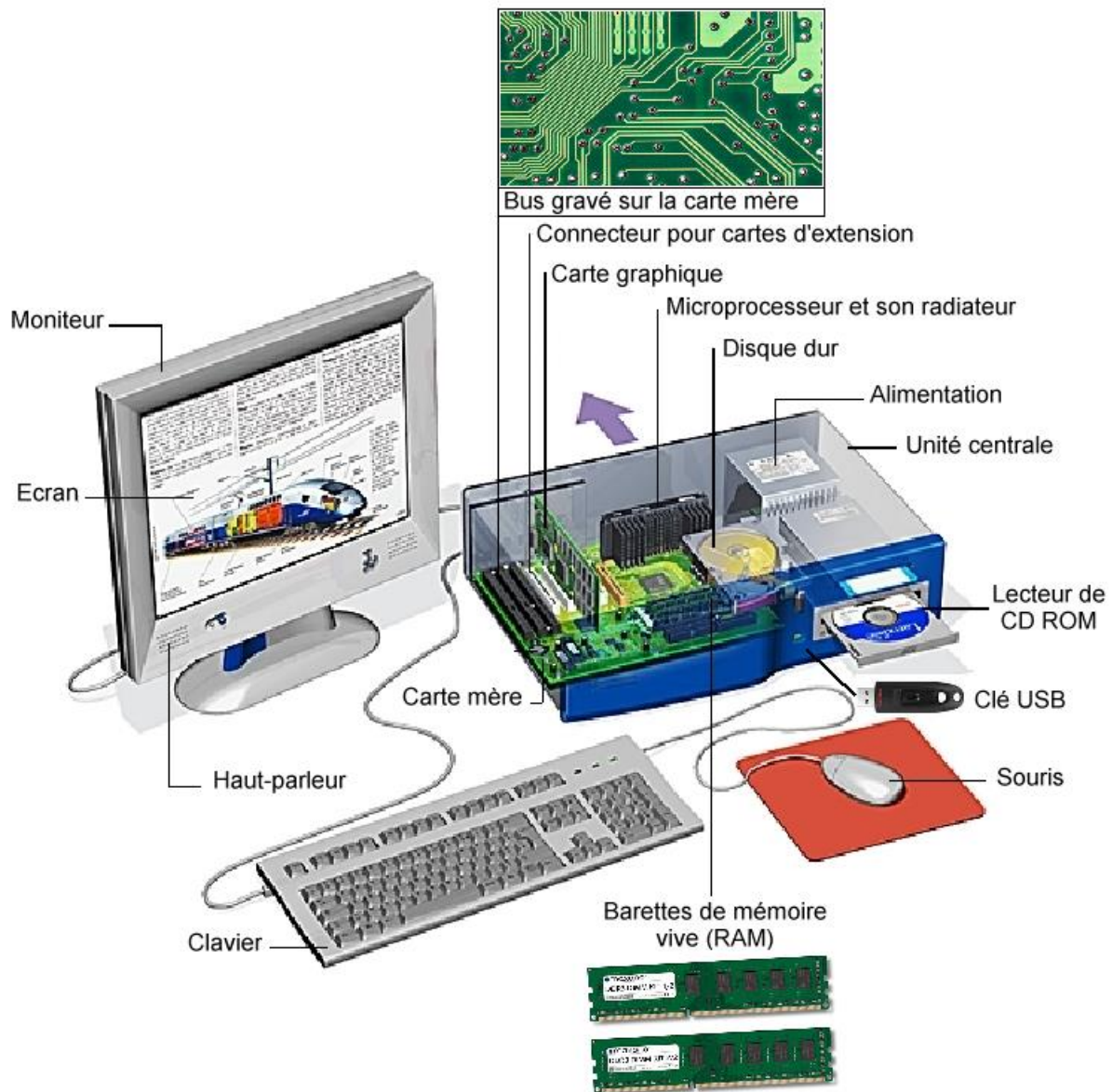
- **Smartphones** (août 2021)

Autres : 1%    **iOS : 26%**    **Android : 73%**

Source : <https://gs.statcounter.com/os-market-share/mobile/worldwide>



### 1.3 Les constituants d'une machine



- **Périphérique** : dispositif connecté à la machine et qui lui permet de **recevoir** ou d'**émettre** des données. Exemples : Hub USB, lecteur de CD ROM, moniteur, clavier, souris.
- **Carte mère** : C'est le circuit imprimé principal sur lequel se trouvent les bus de données, les supports des composants (microprocesseur, barrettes de RAM, mémoire morte ROM, mémoire cache) et différents connecteurs (alimentation, cartes d'extension etc.)
- **Microprocesseur** : c'est une puce de silicium sur laquelle sont gravés des **millions de transistors** organisés selon une architecture<sup>2</sup>.

---

<sup>2</sup> **Architecture externe d'un microprocesseur** : elle précise au programmeur les modes d'accès aux données de la mémoire et le jeu d'instructions machine directement exécutables. Exemples : architecture CISC (microprocesseur à jeu d'instructions étendu), architecture RISC (microprocesseur à jeu d'instructions réduit).

Si un ordinateur est **multiprocesseur** alors il est doté de plusieurs processeurs. C'est une forme d'architecture où plusieurs instructions sont exécutées en parallèle.

La technologie nommée **multicœurs** (*multicore* en anglais) permet d'assembler plusieurs cœurs de processeurs côte-à-côte sur le silicium : le support (connectique qui relie le processeur à la carte électronique) lui ne change pas. Certains éléments, comme l'antémémoire (la mémoire cache<sup>3</sup>) peuvent être mis en commun.

- **Mémoire RAM** : *Random Access Memory* est une mémoire "vive" ou "**volatile**", c'est à dire qu'on peut écrire des mots binaires dedans ou les lire. Mais si on coupe l'alimentation électrique alors les données sont perdues. C'est pourquoi on la qualifie de *volatile*. Elle sert donc uniquement à stocker des informations lorsque l'ordinateur fonctionne.

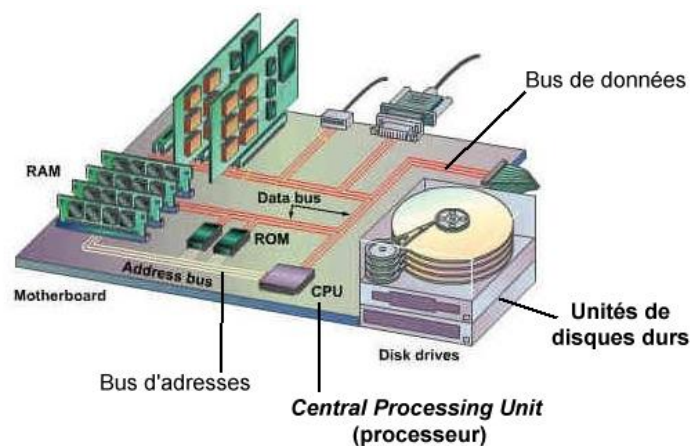
- **Mémoire ROM** : *Read Only Memory* est une mémoire "morte", c'est à dire qu'elle contient des informations une fois pour toutes **destinées seulement à la lecture**. Son avantage est que les informations persistent en cas de coupure d'alimentation. Elles sont donc disponibles dès la mise en route de l'ordinateur. La mémoire ROM est utilisée pour stocker les données nécessaires au démarrage de l'ordinateur.



Deux mémoires ROM sur une carte mère

- **Mémoire de masse** : de grande capacité, **non volatile** comme la ROM mais qui **peut être lue et écrite**. Les disques durs, les cartes SD, les clés USB font partie de cette catégorie.

- **Bus** : ce sont **plusieurs circuits en parallèle** gravés sur la carte mère ou en nappes de fils. Ils servent à transporter les données, envoyer les adresses aux mémoires, les instructions au microprocesseur.



<sup>3</sup> Une **mémoire cache** ou **antémémoire** est une mémoire qui enregistre **temporairement** des copies de données provenant d'une source, afin de **diminuer le temps d'un accès** ultérieur (en lecture) d'un matériel informatique (en général, un processeur) à ces données.

La mémoire cache, **plus rapide** et plus proche du matériel informatique qui demande la donnée, est plus petite — en raison de ses performances et donc de son coût — que la mémoire pour laquelle elle sert d'intermédiaire.

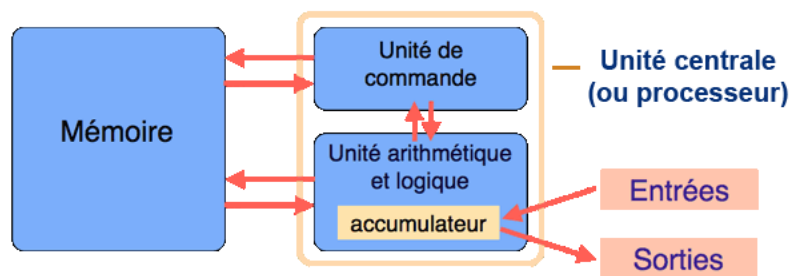
## 1.4 Architecture de von Neumann

Dans l'architecture de von Neumann (qui date de 1945) selon laquelle sont construits tous les ordinateurs d'aujourd'hui, la mémoire contient *à la fois les instructions et les données*. Elle est composée de quatre parties :

- **L'unité arithmétique et logique** est un ensemble de circuits électroniques capables de faire :
  - Les opérations arithmétiques (addition, changement de signe etc.).
  - Les opérations de base (complément à un, complément à deux, and, or, not, nand etc.)
  - Les comparaisons (test d'égalité, d'inférieur, de supérieur etc).
  - Les décalages de bits, les rotations de bits etc.

Elle reçoit des données lues de la mémoire et présente des "données résultats" à écrire en mémoire. Elle contient un registre <sup>4</sup> nommé "accumulateur" qui sert à stocker les résultats intermédiaires.

- **L'unité de contrôle ou de commande** est aussi un ensemble de circuits électroniques pilotés par une horloge (de fréquence souvent plusieurs millions ou milliards de fois par seconde, soit plusieurs mégahertz ou gigahertz) ce qui permet **le séquençage**<sup>5</sup> des opérations :
  - Elle lit les instructions qu'elle trouve en mémoire.
  - Elle envoie à la mémoire des commandes de lecture ou d'écriture.
  - Elle envoie à l'unité arithmétique et logique les codes hexadécimaux d'opérations à faire.
- La mémoire
- Les entrées et sorties



Architecture de von Neumann selon le site [interstices.info](http://interstices.info)

- En pratique, les deux unités (arithmétique et logique et de commande) sont sur la même puce "processeur" appelée **CPU** (Central Processing Unit).
- Les instructions lues dans les cases de la mémoire et présentées à l'unité arithmétique et logique sont en **langage machine** (des octets). Ce type de langage est dit de **bas niveau**.
- Pour l'humain, il existe le **langage assembleur** qui est de bas niveau aussi. Le programmeur peut écrire des instructions symboliques en assembleur (propre à chaque microprocesseur).

Puis, grâce à *un programme assembleur*, les instructions symboliques sont converties en langage machine et écrites en mémoire. Le processeur peut alors exécuter les instructions.

Instructions et données en langage machine (ici les octets A2 et 00)

Adresse	Hexdump	Désassemblage
\$0600	a2 00	LDX #\$00

Assemblage

Instructions symboliques en langage assembleur.

<sup>4</sup> **Registre** : C'est une mémoire qui ne contient qu'un seul mot binaire.

<sup>5</sup> **Séquençage** : Détermination de la séquence (suite ordonnée) des opérations à effectuer.



## 2 Systèmes d'exploitation

### 2.1 Les fonctions d'un système d'exploitation

Le système d'exploitation sert d'intermédiaire entre les logiciels d'application et le matériel.

Pour fonctionner, chaque périphérique a ses propres instructions. Le système d'exploitation traduit les instructions du logiciel d'application en simple demande d'écriture et de lecture sur le périphérique.

#### Exemples

- Grâce au système d'exploitation, un logiciel peut indifféremment sauvegarder des données sur un disque dur ou sur une clé USB.
- Le système d'exploitation prend en compte la capacité de la mémoire de masse encore disponible à l'écriture.
- Il surveille les permissions d'accès. Par exemple, dans une entreprise, l'administrateur informatique a plus de droits qu'un simple utilisateur.
- Le système d'exploitation gère les erreurs. Si une écriture ou une lecture à une certaine adresse mémoire a provoqué une erreur, il va recommencer l'opération pour tenter de la corriger. S'il échoue, il signalera à l'utilisateur qu'une erreur s'est produite.

### 2.2 Les familles de système d'exploitation

Deux grandes familles existent : Windows et Unix.

- Windows

Cette famille dans laquelle on trouve par exemple **Windows 10**, **Windows Server**, **Windows CE** est la propriété de Microsoft. Les systèmes d'exploitation de cette famille sont dits *propriétaires*.

- Unix

Les origines d'Unix remontent à la fin des années 1960 et à un laboratoire d'AT&T (à l'époque, la compagnie nationale de télécommunications américaine) appelé « Bell Labs ».

À l'heure actuelle, il y a **deux grandes branches d'Unix** :

- Systèmes *libres* issus de BSD. La **Berkeley Software Distribution** ou **BSD**, « collection de logiciels de Berkeley » en français, est un système d'exploitation dérivé d'Unix et originaire de l'université de Californie à Berkeley. Un Unix de cette branche est par exemple **FreeBSD**.
- Systèmes *propriétaires* issus de System 5 : par exemple **Solaris** (de Sun Microsystems).

Un **logiciel libre**, selon la définition de la Free Software Foundation, garantit aux utilisateurs quatre libertés fondamentales :

- La liberté d'exécuter le programme, pour tous les usages.
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins.
- La liberté de redistribuer des copies.
- La liberté d'améliorer le programme et de publier ses améliorations.

Un logiciel qui ne garantit pas ces quatre libertés est dit *propriétaire*.

**Remarque : libre ≠ gratuit.** Un logiciel libre peut être payant. Exemple : le système d'exploitation **RedHat** était commercialisé et les modifications et versions créées par les utilisateurs n'étaient pas commercialisables. Un logiciel propriétaire peut être gratuit. Exemple : un antivirus "gratuit".

En 1991, l'étudiant finlandais **Linus Torvalds**, indisposé par la faible disponibilité du serveur informatique Unix de l'université d'Helsinki, entreprend le développement d'un noyau de système d'exploitation, qui prendra le nom de « noyau **Linux** ». Au sens strict, Linux n'est pas un Unix puisqu'il ne comprend pas de code provenant de l'original. D'ailleurs, Linux à proprement parler n'est que le *noyau*, c'est à dire le cœur du système d'exploitation. Le système d'exploitation est **GNU/Linux**, le noyau plus les outils basiques fournis par le projet GNU (pour *Gnu's Not Unix* : « Gnu N'est pas Unix ») de la *Free Software Foundation*.

Néanmoins, GNU/Linux a en commun avec Unix une bonne part de son fonctionnement et de son comportement. Linux est un système d'exploitation **libre**.

Le caractère *libre* du noyau Linux et les outils GNU a eu pour conséquence qu'il existe 30 ans après la première version, des **dizaines de versions de Linux**. On peut citer Suse, Fedora, Chromium OS, Debian. La version Debian a elle-même plusieurs versions dérivées telles que Ubuntu, Linux Mint, Raspbian etc.

- Les systèmes d'exploitation embarqués

Ils sont plus simples que ceux des familles Windows et Unix car ils sont spécialisés pour certaines tâches restreintes au matériel auquel ils servent. On les trouve dans les objets électroniques du quotidien par exemple les serveurs de fichiers, les montres connectées, les box Internet etc.

## 2.3 L'interface "ligne de commande"

### 2.3.1 Définition

Une interface **en ligne de commande** permet la communication entre l'utilisateur et la machine **uniquement sous forme de texte**. Ces lignes de commande sont interprétées par un programme d'interface appelé **Shell**. Le Shell<sup>6</sup> permet à l'utilisateur d'interagir avec le système d'exploitation.

```
debian login: Angie
Password:
Last login: Sun Sep 12 18:26:55 CEST 2021 on hvc0
Linux debian 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Angie@debian:~$ ls
Danse  Encodage  Jeux  NSI  Types
Angie@debian:~$
```

*Interface en ligne de commande Linux Debian*

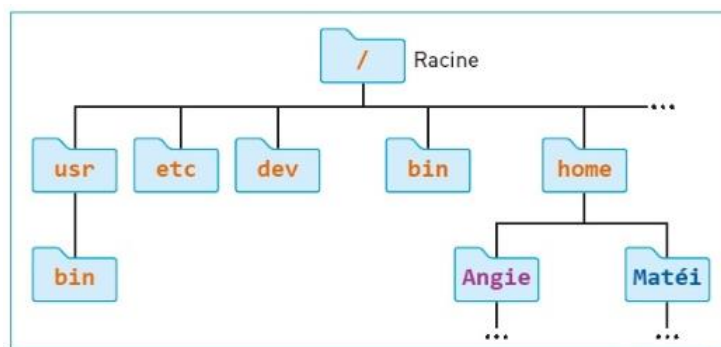
---

<sup>6</sup> Il existe plusieurs types de Shell. Par exemple le Shell du DOS de Microsoft ou le PowerShell de Windows. Le Shell qui est utilisé de façon standard par Unix et Linux s'appelle "**Bash**" (Bash = Bourne Again SHell).

- L'utilisateur peut entrer une à une des commandes telles que `mkdir`, `ls` ou bien écrire un fichier texte de plusieurs lignes composées de commandes à exécuter. Dans ce cas on dit que ce fichier est un **script Shell**.

### 2.3.2 Système de fichiers sous Unix (et Linux)

- Pour y stocker les données, la mémoire de masse est composée de **répertoires** (ou dossiers).
- Dans les répertoires on met les **fichiers**.
- Le système de fichiers est organisé en **arborescence**, c'est à dire qu'on part de la mémoire de masse elle-même qui peut être vue comme la **racine** d'un arbre. Sur cette racine il y a les premiers répertoires. Dans ces premiers répertoires il peut y avoir d'autres répertoires à l'image des branches principales d'un arbres sur lesquelles il y a d'autres branches etc.



*Arborescence d'un Linux standard*

Dans une arborescence Linux standard on trouve toujours les répertoires suivants **sur la racine** (notée " / " ) :

`/usr` (de "**U**ser **S**ystem **R**esources") : Le répertoire des **applications** des utilisateurs.

`/etc` (de "**E**t **C**etera") : Les fichiers de **configuration** de Linux et des applications.

`/dev` (de "**d**evice" = "périphérique") : Les fichiers liés aux **périphériques** connectés au système.

`/bin` (de "**b**inaries") : Où sont stockés les fichiers **exécutables et binaires** essentiels.

`/home` : Où sont placés les répertoires personnels des différents **utilisateurs**.

...

#### **Remarques**

- La barre de fraction ("slash") placée devant les noms des répertoires `usr` `etc` `dev` ... signifie que ces répertoires sont rattachés à la racine directement.
- Pour désigner le répertoire `Angie` qui est dans le répertoire `home` qui est sur la racine on écrit:

`/home/Angie`

On appelle cette écriture "**chemin absolu**" du répertoire personnel d'Angie, puisqu'il commence sur la racine.

- À tout moment, lorsqu'on utilise le Shell, on peut saisir la commande pwd (pour "print working directory") qui affiche le répertoire courant dans lequel on se trouve.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Angie@debian:~$ ls
Danse Encodage Jeux NSI Types
Angie@debian:~$ pwd
/home/Angie
Angie@debian:~$
```

*Le répertoire courant est le répertoire d'Angie*

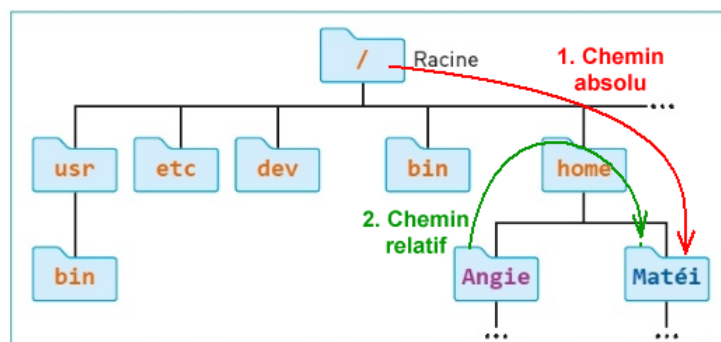
### Changer de répertoire

La commande cd (pour "change directory") permet de changer de répertoire. Il suffit d'indiquer soit le **chemin absolu** du répertoire dans lequel on veut aller (c'est à dire en partant de la racine), soit le chemin relatif du répertoire dans lequel on veut aller (c'est à dire en partant du répertoire courant).

### Exemple

Le répertoire courant est /home/Angie et on veut aller dans le répertoire personnel de Matéi.

On va utiliser la commande cd (change directory). L'arborescence est la suivante :

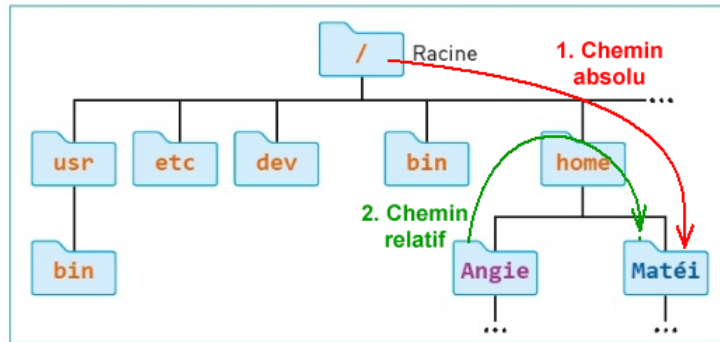


**1<sup>ère</sup> possibilité : changer de répertoire en indiquant le chemin absolu : cd /home/Matéi**

```
Angie@debian:~$ pwd
/home/Angie
Angie@debian:~$ cd /home/Matéi
Angie@debian:/home/Matéi$ pwd
/home/Matéi
```

Explications :

- **Ligne 1** : La commande pwd demande d'afficher le répertoire dans lequel on se trouve.
- **Ligne 2** : La réponse montre que le répertoire courant est /home/Angie
- **Ligne 3** : La commande cd est ici suivie du **chemin absolu** du répertoire de destination.
- **Ligne 4** : Le prompt Angie@debian:/home/Matéi\$ montre qu'on est bien dans le répertoire personnel de Matéi. La commande pwd permet de le confirmer puisque la réponse affichée à la cinquième ligne montre que le répertoire courant est /home/Matéi



2<sup>e</sup> possibilité : changer de répertoire en indiquant le chemin relatif : `cd ../Matéi`

```
Angie@debian:~$ pwd
/home/Angie
Angie@debian:~$ cd ../Matéi
Angie@debian:/home/Matéi$ pwd
/home/Matéi
```

Explications :

- **Ligne 1** : La commande `pwd` demande d'afficher le répertoire dans lequel on se trouve.
- **Ligne 2** : La réponse montre que le répertoire courant est `/home/Angie`
- **Ligne 3** : La commande `cd` est ici suivie du **chemin relatif au répertoire de départ** pour aller dans le répertoire de destination. Les deux points " `..` " équivalent à "répertoire parent du répertoire courant". Donc, puisqu'on est dans le répertoire `Angie`, " `..` " équivaut à `home`.
- **Ligne 4** : Le prompt<sup>7</sup> `Angie@debian:/home/Matéi$` montre qu'on est bien dans le répertoire personnel de `Matéi`. La commande `pwd` permet de le confirmer puisque la réponse affichée à la cinquième ligne montre que le répertoire courant est `/home/Matéi`

#### **Deux répertoires spéciaux**

" `..` " est le "répertoire parent du répertoire courant". Ainsi la commande `cd ..` dit d'aller dans le répertoire parent du répertoire courant.

" `.` " est le "répertoire courant". Ainsi la commande `cd .` dit d'aller dans le répertoire courant et donc ne fait rien !

#### **Dernier exemple**

Si le répertoire courant est `/home/Matéi` alors la commande qui permet de se rendre par le chemin relatif dans le répertoire `usr` est `cd ../../usr` Explication : étant dans `Matéi`, il faut aller dans le parent du parent (c'est à dire la racine) puis descendre dans `usr`.

**Remarque** : Un chemin absolu commence toujours par `/` et un chemin relatif par autre chose que `/`

<sup>7</sup> **Prompt** est le mot anglais pour "invite de commande". C'est la chaîne de caractères en début de ligne qui indique que le Shell est prêt à recevoir une commande. Dans Linux le prompt est de la forme `Utilisateur@NomDeLaMachine:~$` pour signifier que *le répertoire courant est /home/Utilisateur*. Le `$` indique que l'utilisateur est un utilisateur standard qui n'a pas les droits d'administrateur.



### 2.3.3 Interface en ligne de commande d'Unix / Linux

- L'interface entre l'humain et le système d'exploitation Unix ou Linux peut être du type graphique avec des fenêtres, des icônes, des boutons à cliquer. On parle de GUI (Graphical User Interface).

L'interface peut aussi être en ligne de commande. On a vu dans le paragraphe précédent des explications sur le prompt et sur quelques commandes. on va les compléter ici.

Après s'être connectée, l'utilisatrice Angie voit le prompt `Angie@debian:~$`

Le ~ "tildé" signifie qu'elle est dans son répertoire personnel, autrement dit dans /home/Angie

Le tildé est suivi d'un \$ qui suit signifie qu'elle est "simple utilisateur". Si elle était administrateur (on dit "super utilisateur") elle aurait à la place du \$ le symbole #

- Le super utilisateur se nomme root (racine en anglais). Si Angie se connecte en tant que "root" et qu'elle va dans son répertoire personnel, elle voit le prompt :

```
root@debian:/home/Angie#
```

Elle est maintenant super utilisateur. Elle a donc tous les droits sur le système. Elle peut même usurper l'identité d'un autre utilisateur.

Lorsqu'on passe en super utilisateur, on a le message d'avertissement suivant :

```
Nous espérons que vous avez reçu de votre administrateur système local les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :

#1) Respectez la vie privée des autres.
#2) Réfléchissez avant d'utiliser le clavier.
#3) De grands pouvoirs confèrent de grandes responsabilités.
```

### 2.3.4 Les commandes de base pour Unix / Linux

Commande	de l'anglais	Description
cd	change directory	Permet de <b>changer de répertoire</b> courant.
cp	copy files	Permet de <b>copier</b> des fichiers ou des répertoires.
ls	list	Permet d' <b>afficher le contenu</b> d'un répertoire.
man	manual	Affiche les pages du <b>manual</b> du système.
mkdir	make directory	<b>Crée un répertoire</b> vide.
mv	move	Permet de <b>déplacer ou de renommer</b> des fichiers ou des répertoires
pwd	print working directory	Affiche <b>le répertoire courant</b> .
rm	remove	Permet d' <b>effacer</b> des fichiers ou des répertoires.
cat	concatenate files	Permet la concaténation et l' <b>affichage du contenu</b> des fichiers.
clear	clear	Efface l'écran

- Les commandes sont généralement suivies d'**arguments** qui précisent le fichier ou répertoire.

Commande	Description
<pre>cd .. cd ~ ou simplement cd cd / cd -</pre>	<p>Fait aller dans le répertoire parent.</p> <p>Fait aller dans le répertoire personnel de l'utilisateur.</p> <p>Fait aller sur la racine.</p> <p>Fait aller dans le répertoire précédent</p>
<pre>cp photo.jpg /home/Angie cp photo.jpg /home/Angie/image.jpg</pre> <p>source                      cible</p> <p>source                      cible</p>	<p>Copie le fichier photo.jpg (qui doit être dans le répertoire courant) dans le répertoire personnel d'Angie.</p> <p>Même chose, et le fichier photo.jpg est renommé image.jpg</p>
<pre>cp -r /home/Angie/Jeux /home/Angie/NSI cp -r /home/Angie/Jeux /home/Angie/NSI/Z</pre>	<p>Copie le répertoire Jeux dans le répertoire /home/Angie/NSI</p> <p>Même chose et le répertoire Jeux est <b>renommé Z</b> (en supposant qu'il n'y a pas déjà de dossier Z dans le dossier personnel d'Angie).</p>
<pre>ls ls -l ls -lR ls -laR</pre>	<ul style="list-style-type: none"> <li>Affiche les noms des fichiers et des répertoires présents.</li> <li>L'option -l (format long d'affichage) affiche les permissions, la taille en octets, le propriétaire, le groupe du propriétaire, la date de création. Les fichiers sont en blanc et les répertoires en bleu.</li> </ul> <pre>Angie@debian:~/monRepertoire\$ ls monFichier rep1 rep2 rep3 Angie@debian:~/monRepertoire\$ ls -l total 4 -rw-r--r-- 1 Angie enfants 32 sept. 12 15:40 monFichier drwxr-xr-x 2 Angie enfants 1024 sept. 12 15:37 rep1 drwxr-xr-x 2 Angie enfants 1024 sept. 12 15:37 rep2 drwxr-xr-x 2 Angie enfants 1024 sept. 12 15:37 rep3</pre> <ul style="list-style-type: none"> <li>L'option -lR (format long <b>R</b>écursif) affiche tout ce qui se trouve dans l'arborescence descendante.</li> <li>-laR (format long <b>a</b>ll <b>R</b>écursif) affiche aussi <b>les fichiers cachés</b>.</li> </ul>
<pre>man ls</pre>	<p>Affiche toutes les options possibles avec la commande ls.</p>
<pre>mkdir monRepertoire mkdir ../monRepertoire</pre>	<p>Crée dans le répertoire courant le répertoire monRepertoire.</p> <p>Crée dans le répertoire parent le répertoire monRepertoire.</p>
<pre>mv /home/Angie/Jeux /home/Angie/NSI mv /home/Angie/Jeux /home/Angie/NSI/Z</pre>	<p>Déplace le répertoire Jeux dans le répertoire /home/Angie/NSI</p> <p>Même chose et le répertoire Jeux est <b>renommé Z</b> (en supposant qu'il n'y a pas déjà de dossier Z dans le dossier personnel d'Angie).</p>
<pre>pwd</pre>	<p>Affiche le <b>répertoire courant</b> dans lequel nous nous trouvons.</p>
<pre>rm monFichier rm -r monRepertoire</pre>	<p>rm doit être utilisé pour <b>effacer un fichier</b> (exemple : monFichier)</p> <p>rm -r doit être utilisé pour <b>effacer un répertoire</b> (exemple le répertoire monRepertoire situé dans le répertoire courant).</p>
<pre>cat monFichier</pre>	<p>Affiche le contenu texte d'un fichier. Par exemple si on a le fichier texte monFichier et qu'on veut voir ce qui est écrit dedans :</p> <pre>Angie@debian:~/monRepertoire\$ cat monFichier bla bla ligne 1 bla bla ligne 2 Angie@debian:~/monRepertoire\$</pre>

**Remarque :** Le caractère \* est le caractère "joker". Il permet de remplacer n'importe quelle chaîne de caractères. Par exemple : `ls TP*.py` n'affiche que les fichiers commençant par `TP` et terminant par `.py` comme par exemple `TP1.py` mais pas `Travail.py` ni `TP1.jpg`

### 2.3.5 Gérer les droits d'accès d'un fichier ou d'un répertoire

Les systèmes d'exploitation inspirés d'Unix possèdent la capacité de définir de façon poussée la gestion des droits d'accès aux divers fichiers du système d'exploitation.

Les droits d'accès définissent deux notions :

- **La possession** (ou non possession) d'un fichier ou d'un répertoire à un utilisateur et à un groupe d'utilisateurs.

La possession d'un fichier se définit sur trois catégories :

1. L'utilisateur propriétaire du fichier (**u**). Il s'agit généralement du créateur du fichier. (Prenez note qu'un fichier créé par une commande exécutée à l'aide de `sudo` appartiendra à l'utilisateur `root` ; vous serez potentiellement amené à devoir changer le propriétaire de ce fichier pour pouvoir vous en servir avec votre propre compte utilisateur.)
2. Le **groupe** propriétaire du fichier (**g**). Si un utilisateur est membre d'un certain groupe qui possède la propriété d'un fichier, l'utilisateur aura aussi certaines permissions particulières sur ce fichier.
3. Les autres, *other*, le reste du monde (**o**). Bref, tout un chacun n'étant ni propriétaire du fichier, ni membre du groupe propriétaire du fichier.

Faisons une analogie avec les voitures. Le *propriétaire* serait la personne au nom de laquelle la voiture est immatriculée. Le *groupe propriétaire* est l'ensemble des personnes qui sont inscrites en tant que conducteurs secondaires de la voiture chez l'assureur. Enfin, les *autres* correspondent à toutes les autres personnes n'étant ni détenteur de l'immatriculation ni inscrites en tant que conducteurs de la voiture chez l'assureur.

Les permissions désignent ce que les diverses catégories d'utilisateurs (propriétaire d'un fichier, membres du groupe propriétaire d'un fichier et le reste du monde) ont l'autorisation d'effectuer sur un fichier donné. Par exemple, une catégorie d'utilisateurs peut avoir accès en lecture et écriture à un fichier, alors qu'une autre catégorie a accès en lecture seulement à ce même fichier.

- **Les permissions** se définissent sur trois niveaux :

La présence des lettres `r`, `w`, `x` donne le droit correspondant. Le symbole `-` donne l'interdiction.

Lettre	Signification	Permissions accordées	
		Pour un fichier	Pour un répertoire
<code>r</code>	<code>read</code>	Lire avec un logiciel.	Afficher la liste des fichiers et des répertoires qu'il contient.
<code>w</code>	<code>write</code>	Modifier son contenu.	Créer, supprimer et changer le nom des fichiers qu'il contient, quels que soient leurs droits respectifs.
<code>x</code>	<code>execute</code>	Exécuter (s'il est exécutable).	Ouvrir le répertoire.

Lorsqu'on saisit en ligne de commande `ls` avec l'argument `-l` on voit le contenu détaillé du répertoire courant. Si le premier caractère est `d` ("`directory`") c'est un répertoire, si c'est `-` alors c'est un fichier.

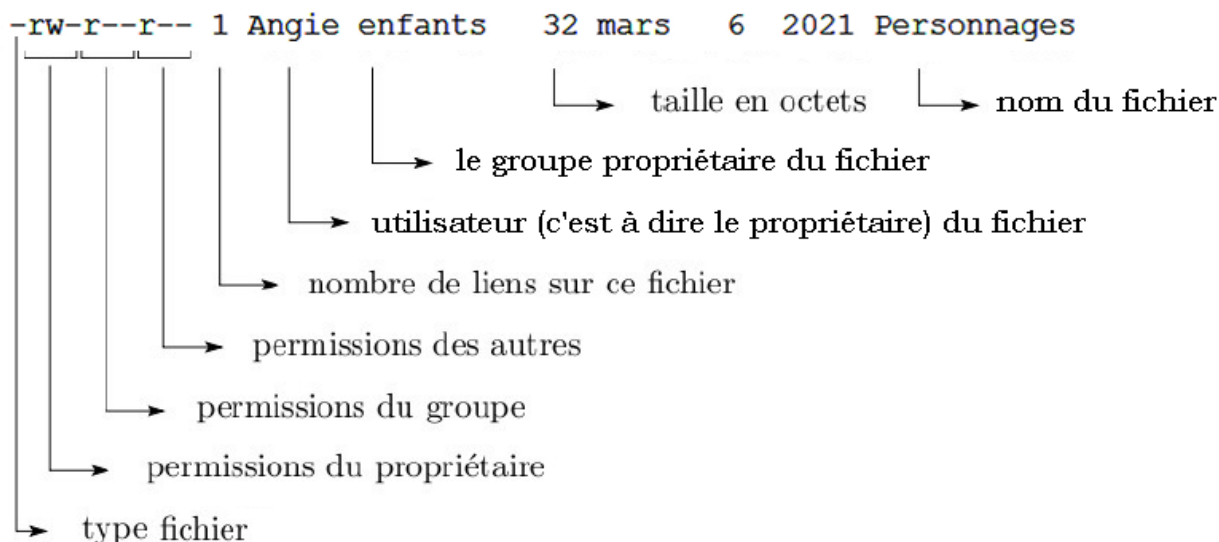


Figure 1

Il est possible de modifier les droits (aussi bien les permissions que les propriétés) :

#### Modification des permissions

- La commande `chmod` (**change mode**) permet de changer les permissions sur un fichier<sup>8</sup>.

#### Exemples avec le fichier *Personnages* de la figure 1

- Ajouter** le droit d'exécution au groupe du fichier (g) et à tous les autres utilisateurs de Linux (o) avec le signe plus :

```
chmod go+x Personnages
```

On obtient : `-rw-r-xr-x 1 Angie enfants 32 mars 6 2021 Personnages`

- Puis **retirer** le droit de lecture au groupe du fichier (g) avec le signe moins :

```
chmod g-r Personnages
```

On obtient : `-rw---xr-x 1 Angie enfants 32 mars 6 2021 Personnages`

- Enfin **ajouter** des droits **tout en supprimant** ceux qui n'ont pas été indiqués avec le signe égal :

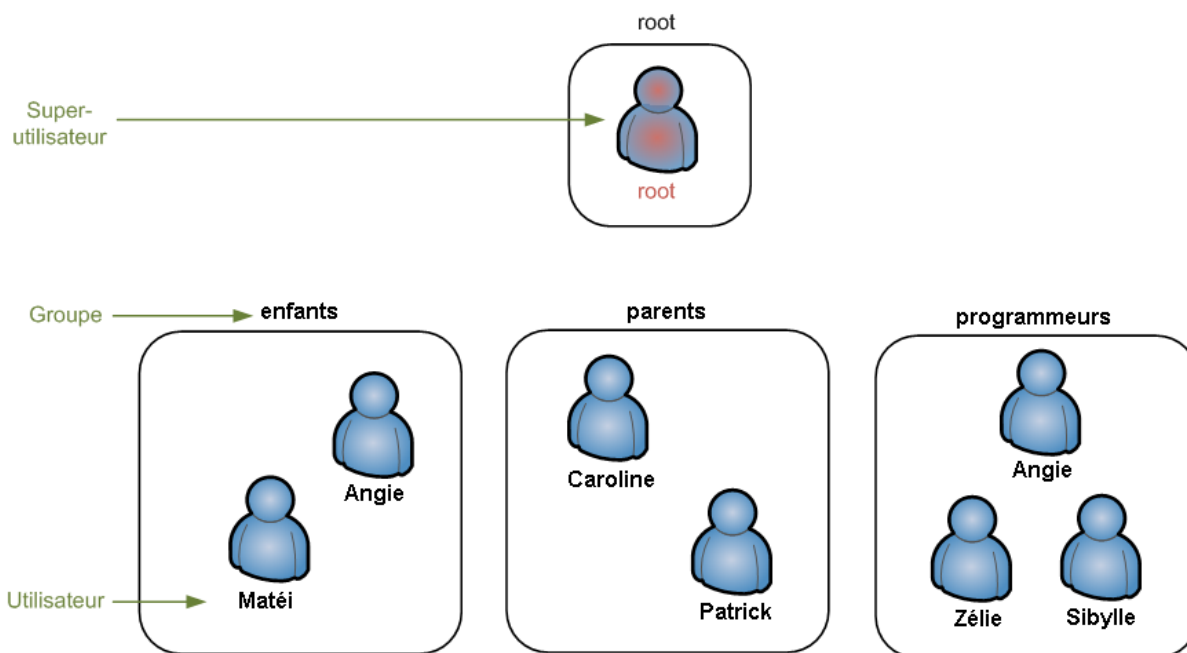
```
chmod u=rwx,g=r Personnages
```

Cette commande ajoute au propriétaire les droits `rwx` et ajoute au groupe le droit `r` et supprime tous les autres droits. On obtient : `-rwxr--r-x 1 Angie enfants 32 mars 6 2021 Personnages`

On voit que les utilisateurs autres (o) ont toujours le droit de lire et d'exécuter. Pour leur retirer tous leurs droits, il aurait fallu saisir `chmod u=rwx,g=r,o= Personnages`

<sup>8</sup> Pour effectuer le changement de permission sur un **répertoire complet** c'est-à-dire sur tous ses fichiers et tous ceux de sa descendance, il faut utiliser l'option `-R` (de façon **R**écursive) Exemple : `chmod -R g=rw Saisons`

**Remarque :** Si on veut **donner simultanément** les mêmes droits à tous les utilisateurs, on utilise **la lettre a** (all). Par exemple `chmod a+rw Personnages` ajoute simultanément les droits r et w à tous les utilisateurs sur le fichier Personnages.



*Utilisateurs et groupes d'utilisateurs dans Linux*

Changement du propriétaire d'un fichier ou d'un répertoire ; Changement du groupe d'un fichier ou d'un répertoire

- La commande<sup>9</sup> `chown` (**change owner**) permet de changer le propriétaire d'un fichier ou d'un répertoire.
- La commande `chgrp` (**change group**) permet de changer le groupe propriétaire d'un fichier ou d'un répertoire.

**Exemple :** Angie a créé le fichier Véhicules. Automatiquement le groupe propriétaire du fichier Véhicules était **enfants**. Puis elle a changé le groupe propriétaire pour qu'il soit **programmeurs**.

```
Angie@debian:~/Jeux$ ls -l
total 5
-rw-r--r-- 1 Angie enfants    34 mars  6 2021 Mondes
-rw-r--r-- 1 Angie enfants    32 mars  6 2021 Personnages
drwxr-xr-- 2 Angie enfants  1024 mars  6 2021 Saisons
-rw-r--r-- 1 Angie enfants     7 sept. 16 17:15 Scénarios
-rw-r--r-- 1 Angie programmeurs 20 sept. 16 17:02 Véhicules
```

<sup>9</sup> `chmod` est une commande qui nécessite d'avoir les droits de super utilisateur. C'est pourquoi on doit se donner ces droits temporairement, juste avant de saisir la commande, en saisissant `sudo` qui est l'abréviation de **substitute user do**. Donc `sudo chown` (ou `sudo chgrp`).  
Pour devenir utilisateur `root` (super utilisateur en permanence) saisir `sudo -i` et pour quitter `root` saisir `exit`