Installation et maintenance des ordinateurs (v1.03)

1. Anatomie d'un ordinateur

1.1. Boitier

en: (computer) case

Les boitiers sont faits en métal pour éviter l'accumulation rapide de poussière et les perturbations extérieures (venues de l'électronique aux alentours).

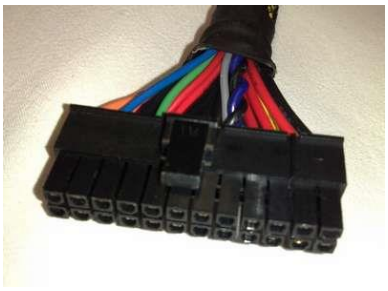
Les boitiers sont parfois déjà accompagnés :

- d'**alimentations** (souvent de piètre qualité) ;
- de **ventilateurs** (c'est bien, on n'en a jamais trop).

Assurez-vous d'acheter un boîter suffisamment grand pour contenir les composants que vous achetez ! Les meilleures processeurs et cartes graphiques du marché demandent de nombreux ventilateurs, souvent assez gros, qui prennent donc beaucoup de place.

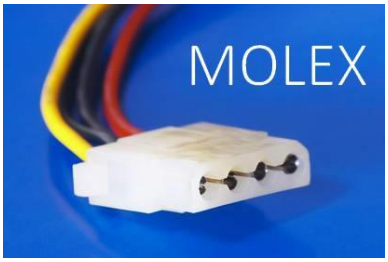
1.2. Alimentation

en: power supply unit (PSU)



Il en existe de différentes **puissances** (en watts), basées sur les **normes ATX**. L'alimentation est connectée à tous les composants qui ont besoin d'énergie : carte-mère, disques durs, lecteurs, ventilateurs (même si on verra que les ventilateurs peuvent être alimentés via la carte-mère), etc. Chaque composant vient avec des connecteurs d'alimentation différents (molex, SATA, Berg…).

L'alimentation est **en veille en permanence** et elle démarre la machine lors de l'appui sur un bouton défini.



L'alimentation dispose de deux rangées de douze broches (*en: pins*) pour différents signaux (3,3 V, 5 V, 12 V…) pour un total de **24 broches** (MOLEX ATX 24 broches). Ce sont les broches n°15 et 16 qui contrôlent l'allumage. Les différents connecteurs de l'alimentation possèdent des **détrompeurs** qui empêchent de les brancher dans le mauvais sens. Il y a parfois aussi des connecteurs supplémentaires pour alimenter des composants très gourmands (certains GPU…).

Le **courant** délivré par l'alimentation varie en fonction de la tâche demandée (la **puissance** augmente donc aussi), mais la **tension** sur les composants reste la même.

Une bonne alimentation a :

- un bon ventilateur (silencieux, idéalement) ;
- un rendement supérieur à 80 % ;
    - Un échauffement sur un composant le fait fonctionner moins bien. Il y a toujours des pertes, mais une meilleure alimentation (+ chère) a moins de perte.
- au minimum 350 W (souvent assez pour de la bureautique).
    - Pour des builds avec des composants plus gourmands, il faut faire la somme des puissances mentionnées par les fabricants (surtout GPU et CPU) pour savoir de combien de watts on a besoin.
- un certain prix, mais rien d'abusif ; éviter les alims à 30 € de marques inconnues sur Wish...

1.2.1. Test de fonctionnement

Les alimentations sont des composants qui posent souvent des soucis. Pour savoir si une alimentation ne fonctionne plus bien, il existe des **testeurs** (*en: power supply tester*) qui permettent de vérifier chaque broche.

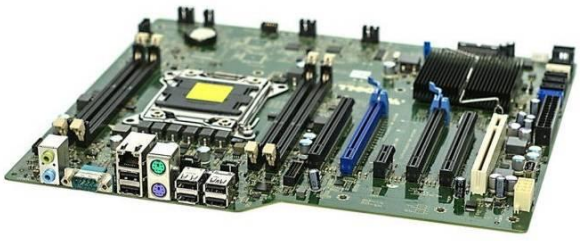
1.2.2. UE, Asie, Afrique, Amérique du Sud (50 Hz) vs Amérique du Nord (60 Hz)

L'électricité n'est pas délivrée avec les mêmes normes aux États-Unis (+ Canada) et à peu près partout ailleurs. De ce fait, les alimentations sont adaptées à ces différences ; on ne peut donc pas acheter des alims n'importe où pour les brancher chez nous !

1.3. Carte-mère

`en: motherboard (aka "mobo")`

C'est la plaque tournante de l'ordinateur. **Elle relie tous les composants entre eux**. En revanche, ce n'est pas elle qui se charge d'alimenter les composants (sauf des ventilateurs).



Y sont connectés : - le processeur ; - une ou des barrette(s) de RAM ; - les périphériques de stockage (HDD, SSD, USB…) ; - les lecteurs ; - des cartes d'extensions (pour étendre les capacités de la carte-mère) ; - des ventilateurs monitorables.

Le processeur est « socketé » directement sur la carte-mère. Il existe de nombreux **sockets** (supports) différents, donc il faut s'assurer que le processeur et la carte-mère qu'on achète sont compatibles. Dans la pratique, on choisira **d'abord le processeur** qui répond à nos besoins (ainsi que des éléments comme la carte graphique ou de la RAM puissante), puis on prendra une bonne carte-mère compatible.

Les composants communiquent entre eux grâces aux **pistes de cuivre** (bus) qui sont imprimés sur la carte-mère.

Sur les cartes-mères, on trouve aussi **une pile**, qui permet de garder l'horloge à jour quand elle n'est pas alimentée (voir 3.1.1. Mémoire CMOS).

1.3.1. Chipset

Auparavant, une carte-mère était composée d'une myriade de puces (*en: chips*) qui géraient toutes sortes de périphériques ; au fil des années, ces différentes puces ont été rassemblées. En effet, sur les cartes-mères modernes, il n'y a plus qu'un **chipset** (*fr : « groupe de puces »*) constitué de deux **ponts** (ici, ponts = puces, donc !) : un **nord** et un **sud**.

Le **pont nord** (*en: northbridge*) gère :

- les communications entre le processeur (CPU) et les périphériques « **rapides** » (RAM, bus AGP pour les GPU, bus PCI Express pour les périphériques externes & GPU modernes) ;
    - Le bus qui relie le pont nord au CPU s'appelle le « **front-side bus** ».
    - Le bus qui relie le pont nord à la RAM s'appelle le « **memory bus** ».
- un processeur graphique limité (dépend des modèles) ;
- (avant) le contrôle mémoire, mais c'est aujourd'hui le CPU qui se charge de ça.

Le **pont sud** (*en: southbridge*) gère :

- le fonctionnement des bus et dispositifs « **lents** » (clavier, PCI, port série…) ;
- la communication entre des contrôleurs dédiés et des composants ;
- l'interface Ethernet ;
- l'USB ;
- le FireWire (ancien).

La **qualité d'un chipset fait en grande partie la qualité d'une carte-mère** (et son prix). Il est soudé sur la carte-mère, donc impossible de le remplacer.

1.3.2. Ports de communication

Les cartes-mères possèdent différents ports, plus ou moins rapides, qui travaillent à des fréquences différentes.

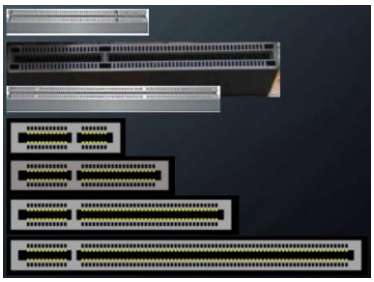
Le bus **PCI** existe toujours, car beaucoup de composants ont été fabriqués pour ce bus, mais il se fait doucement remplacé par le **PCI Express 1x**.

L'**AGP** est l'ancien standard pour les GPU ; on utilise aujourd'hui le bus **PCI Express 16x**.

Le **PCI-X** est rarement sur des ordinateurs domestiques ; il est adapté aux serveurs. Les bus **PCI Express 4x** et **8x** sont aussi plutôt utilisés par les serveurs.

La famille des **bus PCI-Express** fonctionnent tous en série. Un « PCI Nx » possède « N » lignes séries. Les informations sont envoyées les unes à la suite des autres ; **la série permet d'augmenter la fréquence de travail**.

Bus	Largeur	Fréquence	Débit maximal
PCI	32bits	33MHz	133 Mo/s
AGP 8x	32bits	66MHz	2,13Go/s
PCI-X	64bits	133MHz	4,26Go/s
PCI Express 1x	1bit	100MHz	500Mo/s
PCI Express 4x	4bits	100MHz	2Go/s
PCI Express 8x	8bits	100MHz	4Go/s
PCI Express 16x	16bits	100MHz	8Go/s



1.3.3. Connecteurs

De nos jours, les cartes-mères modernes fournissent de base toutes sortes de périphériques et connecteurs. Avant, il fallait des cartes additionnelles pour tout (son, image, clavier/souris…). On trouve ces connecteurs sur le panneau arrière de la carte-mère.

Les connecteurs modernes incluent :

- HDMI (sortie vidéo+son+Ethernet) et/ou DisplayPort (sortie vidéo+son+réseau+données) et/ou DVI (sortie vidéo) ;
- USB2 & USB3 (i/o pour plein de trucs) ;
- port Ethernet (i/o réseau) ;
- sortie audio optique.

Parmi les vieux trucs, on trouve :

- des ports PS2 (pour clavier/souris) ;
- des ports VGA (vidéo ; 15 pins) ;
- un COM PORT DB9 ;
- un port DB25 (imprimantes, scanners).

1.3.4. Fan headers

Les cartes-mères modernes sont pourvues d'alimentations pour ventilateurs. Ces connecteurs permettent souvent aussi de transmettre des informations sur la vitesse des ventilos et d'ajuster leur vitesse. En branchant les ventilos directement sur l'alimentation, la gestion de la vitesse n'est pas possible.

Si une carte-mère manque de fan headers, il est possible de les multiplier avec des « fan splitters » pour y brancher davantage de ventilateurs.

1.3.5. Tailles de cartes-mères

Il existe différentes tailles de cartes-mères. Habituellement, on retrouve les cinq mentionnées ci-contre :

Nom	Taille de la carte
ATX	12" x 9,6"
Mini-ATX	11,2" x 8,2"
Micro-ATX	9,6" x 9,6"
Flex-ATX	9" x 7,5"
Mini-ITX	6,7" x 6,7"

1.3.6. Marques à privilégier

Pour les cartes-mères, on se tournera vers des marques ayant fait leurs preuves, comme ASUS et MSI.

1.4. Processeur

en: processor

Permet **d'exécuter les programmes** qui sont en mémoire vive. C'est le « cerveau » du système, l'unité de calcul ; alors que la carte-mère est le « système nerveux ». On pourrait aussi faire une analogie du type : le processeur est le « moteur » et la carte-mère le « châssis ».

Un processeur fonctionne avec des milliards de **transistors** en silicium qui prennent deux états distincts : 0 ou 1. Depuis les années 2000, on augmente les performances des processeurs en multipliant le nombre de transistors, en les miniaturisant et en augmentant la taille de leurs **mémoires caches**. On arrive aujourd'hui aux limites physiques de la miniaturisation des transistors (7 nanomètres actuellement ; la taille d'un atome de silicium est de ~0,2 nm).

1.4.1. Cœurs et mémoire cache

Les milliards de transistors sont rangés en cœurs (*en: cores*), des sous-unités, elles-mêmes séparées en threads. Chaque core a sa **mémoire de travail interne**, de la RAM statique (SRAM) qui est *extrêmement* rapide : les **caches L1 et L2**. Depuis un moment, il existe aussi le **cache L3** qui est partagé entre tous les cœurs (même parfois un L4). Le cache L1 est le plus rapide, le L2 l'est moins et soutient la L1, et la L3 est encore plus lente.

Quand un processeur doit traiter des données, s'il ne les trouve pas dans sa mémoire interne, il va les chercher dans la RAM (voir 1.5.) ; il stocke alors les données les plus essentielles dans sa mémoire pour y avoir accès plus rapidement. Si un processeur n'avait pas de mémoire cache, il ne pourrait jamais fonctionner plus vite que la vitesse de la RAM connectée à la carte-mère.

1.4.2. Performance

Les performances d'un processeur dépendent de plusieurs choses :

- sa vitesse (fréquence) de travail (*en: clock speed*) = nombre d'opérations à la seconde ;
    - 3 GHz signifie donc 3 milliards d'opérations basiques à la seconde.
    - Pour le gaming, c'est avant tout la clock speed qui importe.
- la taille de sa mémoire cache ;
- le nombre de cœurs, mais uniquement pour certaines tâches.
    - La plupart des tâches demandée à un processeur ne peuvent pas être répartie sur des dizaines de cœurs en parallèle (un jeu tourne rarement sur plus de 8 cœurs). Avoir de nombreux cœurs est surtout utile lorsqu'on a de **multiples tâches simultanées**, par exemple lors de gaming + streaming + enregistrement vidéo, ou pour faire tourner de multiples machines virtuelles, ou bien pour le fonctionnement d'un serveur.
- la quantité de données traitées par unité de temps (FLOPS) ;
    - Cette quantité est limitée par la largeur (exprimée en bits) et de la fréquence (exprimée en Hz) du **bus frontal** de la carte-mère (*en: FSB ; Front-Side Bus*), parfois encore appelé « bus système » (ancien). Les FSB modernes ont une largeur égale à 64 bits.

- L'architecture originelle du FSB est aujourd'hui remplacée par des technologies comme l'**HyperTransport**, ou HT/HTT (chez AMD et Nvidia), et le **QuickPath Interconnect**, ou QPI (chez Intel).

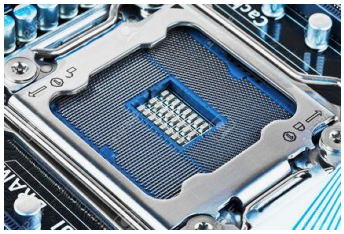
1.4.3. Ventirad

En faisant commuter des millions de transistors par seconde, un processeur chauffe **beaucoup**. S'il n'est pas ventilé, un processeur moderne est fichu en l'air en quelques minutes. On va donc le refroidir à l'aide d'un **ventirad** (*en: heatsink, cooler*). Le **venti**lateur apporte l'air frais, le **rad**iateur dissipe la chaleur. Un ventirad peut être ajusté en fonction de la puissance demandée, il ne tourne pas à fond tout le temps.

La plupart des processeurs sont fournis avec un ventirad qui suffira largement à le refroidir, mais si vous voulez overclocker votre processeur, il faudra sans doute penser à prendre un ventirad plus efficace. Ce faisant, assurez-vous que le ventirad que vous choisissez **rentre dans la tour** que vous avez choisie et que cela **ne bloque pas l'accès à des composants sur la carte-mère** (il est fréquent de bloquer des slots de RAM, par exemple).

On applique également une **pâte thermique** sur la surface de contact du processeur pour uniformiser la conduction de la chaleur entre le processeur et le ventirad. Si on retire le ventirad, il faudra remettre une nouvelle pâte thermique.

1.4.4. Socket



Chaque famille de processeur est liée à un socket (support) adapté sur les cartes-mères. De nos jours, on utilise des architecteurs Land Grid Array (vs Pin Grid Array) : les broches sont sur le socket de la carte-mère, et non plus sur le processeur. Il ne faut **jamais** toucher ces pastilles ou ces broches directement avec les doigts !

1.4.5. Jeux d'instructions

Chaque famille de processeur possède son propre **jeu d'instructions** (*en: instruction sets*). Ce sont les opérations élémentaires qu'il peut exécuter, c'est-à-dire son **code assembleur**. Ces instructions machines sont utilisées par les langages de haut niveau et sont transparente pour l'utilisateur. Les fabricants publient les jeux d'instructions quand ils sortent de nouveaux processeurs.

Il existe deux types de jeux d'instructions :

- RISC (Reduced Instruction Set Computer) : instructions peu nombreuses, mais codées simplement donc très rapides ;
- CISC (Complex Instruction Set Computer) : instructions complexes prenant plus de cycles d'horloge.

Tous les processeurs sont en CISC de nos jours.

1.4.6. Processeurs modernes

De nos jours, les processeurs gèrent le **contrôle de la mémoire vive** (ce n'est plus le pont nord qui fait ça) et ils intègrent une **puce graphique** (ce qui permet de se passer de GPU sur des ordis de bureautique pure).

1.4.7. Marques à privilégier

Pour les processeurs de PC et de serveurs, on se tournera exclusivement vers Intel et AMD. Pour checker les meilleurs CPU disponibles sur le marché, différents « benchmarks » régulièrement mis à jour existent : https://www.cpubenchmark.net/high_end_cpus.html

Pour des comparatifs rapide, le site Nanoreview fait du bon boulot : https://nanoreview.net/en/cpu-compare

Pour des chiffres en jeu, voir aussi la chaîne PC Benchmark : https://www.youtube.com/channel/UCpc_uUd25ObT3pQGbr5MpHg

1.5. Mémoire vive (RAM)

en: random access memory

La mémoire vive est placée directement sur la carte-mère, qui comporte en général 2 ou 4 emplacements pour des barrettes. La mémoire vive est assez rapide, mais volatile ; càd qu'à l'extinction du système, les données sont perdues, comme pour la mémoire cache des processeurs (elle est toutefois moins rapide). Son rôle est de charger temporairement les programmes pour qu'ils soient ensuite exécutés par le processeur.

Pour être exécuté, un programme est stocké sur un disque dur, puis il est chargé dans la RAM, et enfin, le CPU peut accéder à ses données et l'exécuter. Si un ordinateur manque de RAM, il va être plus lent car il y aura davantage d'échange entre la RAM et le disque dur. Avec suffisamment de RAM, toutes les données peuvent être mises en mémoire vive et le processeur peut s'en servir sans attendre.

1.5.1. DRAM vs SDRAM

Depuis un bon moment, la RAM est synchronisée avec l'horloge du système (Synchronous Dynamic RAM), ce qui permet de meilleures performances ; mais cela n'a pas toujours été le cas. La DRAM n'a pas totalement disparu de nos ordinateurs pour autant, on en trouve par exemple encore dans les SSD.

1.5.2. Générations et fréquences de RAM

La mémoire vive a aussi une fréquence de travail, qui n'a cessé d'augmenter au fil du temps. La fréquence de travail de la mémoire vive dépend du processeur. La fréquence la plus récente est la DDR5, mais elle n'est encore utilisée nulle part ; tous les PC tournent donc avec de la RAM **DDR4**, en gros. DDR signifie « Double Data Rate » et est une technologie qui permet le transfert de 2x plus de données qu'avec par signal d'horloge. Le nombre derrière DDR dans les appellations est une question de vitesse sur les bus. Cette vitesse n'a cessé de doubler au fil des années.

Voici un tableau reprenant les mémoires RAM depuis 1997 :

Année	Nom	Fréq Bus	Spécificités
1997	SDRAM	66-150 MHz	
2000	DDR-SDRAM	100-300 MHz	Vitesse doublée par rapport à SDRAM
2004	DDR2	200-600 MHz	Vitesse doublée par rapport à DDR-SDRAM
2007	DDR3	400-1066 MHz	Vitesse doublée par rapport à DDR2
2013	DDR4	666-1600 MHz	Vitesse doublée par rapport à DDR3

Ces mémoires ne sont pas compatibles entre elles ! Grâce à un système d'**encoches** différentes, il est impossible de se tromper de génération et de slotter une barrette incompatible dans une carte-mère. L'encoche évite aussi de **se tromper de sens** en insérant la barrette.

Attention aux dénominations trompeuses (tableau non exhaustif) :

Nom	Type et fréq. Annoncée	Bande passante	Fréquence réelle
PC-1600	DDR-200	1,6 Go/s	100 MHz
PC-3200	DDR-400	3,2 Go/s	200 MHz
PC2-3200	DDR2-400	3,2 Go/s	100 MHz
PC2-5300	DDR2-667	5,33 Go/s	166 MHz
PC3-10666	DDR3-1333	10,66 Go/s	166 MHz
PC3-19200	DDR3-2400	19,2 Go/s	300 MHz
PC4-17000	DDR4-2133	17 Go/s	133 MHz
PC4-22400	DDR4-2800	22,4 Go/s	175 MHz

Pour mieux comprendre ces appellations, voir : https://youtu.be/PVad0c2cljo?t=634

La fréquence maximum de RAM utilisable dépend du processeur. En 2020, il est rare de voir des processeurs accepter plus que de la RAM DDR4-3200 ; et sans overclocking, cela signifie qu'il est totalement inutile d'acheter de la RAM plus rapide que ce que le fabricant du CPU prévoit. Il faut également s'assurer que cette vitesse de RAM est garantie par la carte-mère.

1.5.3. Latence CAS

Le temps d'accès aux données dans la RAM n'est pas nul. Dans les spécifications du produit, cette **latence** peut être mentionnée par une suite de nombres (ex : 14-15-15-35 ; le premier nombre est le plus important) ou par une notation **CLx** (Column Address Strobe Latency ; où **x** est un nombre).

La **CAS** représente le nombre de cycles d'horloge entre l'envoi de la commande de lecture et l'arrivée des données. Ainsi, en théorie, plus le nombre qui suit « CL » est petit, mieux c'est (ce nombre indique des nanosecondes). En pratique, les RAM les plus récentes ont des CL plus élevées, mais des performances toujours accrues, car leurs fréquences sont de plus en plus hautes (voir https://en.wikipedia.org/wiki/CAS_latency#Memory_timing_examples)

1.5.4. Format DIMM

Les mémoires RAM pour les desktops sont au format DIMM (Dual Inline Memory Module) et le format pour les laptops est le SO-DIMM (Small Outline DIMM). Elles sont plus fines et possèdent un peu moins de broches.

1.5.5. Parité de la RAM (ECC, buffered)

La mémoire pour un poste de travail (desktop) est généralement **non ECC** et **non buffered**, contrairement à la mémoire vive qu'on met dans les serveurs (plus chère). Une mémoire **ECC** (Error Correcting Code) permet de corriger certaines erreurs de mémoire ; c'est une précaution supplémentaire qui est prise lorsqu'on ne peut pas se permettre d'erreurs. Une **mémoire buffered** est une **mémoire vive à registre**, ce qui permet de réduire la charge électrique sur le contrôleur de mémoire du processeur et permet donc encore une meilleure stabilité.

1.5.6. Dual/Quad Channeling

La plupart des contrôleurs mémoire (pour rappel, ce sont les CPU qui se charge de ça dans les desktops, de nos jours) proposent un **fonctionnement en double canal** (*en: double channeling*). Il s'agit d'exploiter des modules de mémoire vive par paire afin d'en tirer un maximum de bande passante. C'est pour cela qu'il est impératif de mettre des **barrettes de RAM identiques** (même marque, même fréquence) par groupe de deux dans un ordi. Les processeurs qui acceptent jusqu'à du **quad channeling** requièrent donc 4 barrettes identiques.

1.6. Disques durs

en: hard drive

Les périphériques de stockage (à mémoire de masse) permettent de stocker durablement des données. Les vitesses d'exécution sont beaucoup plus lentes que celles de la RAM.

Parmi les périphériques de stockage de masse, on peut citer : les disques durs à plateaux et les SSD ; les clés USB ; les lecteurs/graveurs de DVD et disques Blu-Ray ; les cartes SD ; les disquettes...

1.6.1. Normes de connecteurs

Il en existe 5, dont deux sont en fin de vie :

- IDE ou Parallel-ATA (ancien, pour les PC) ;
- DCII (ancien, pour les serveurs) ;
- SAS (actuel, pour les serveurs) ;
- SATA (Serial ATA), la norme actuelle la plus classique pour les PC ;
    - A subi des révisions, la plus récente étant le SATA 3.5 (juillet 2020).
    - Le mSATA existe aussi (mini-SATA).
        - Attention : les connecteurs SATA ont tendance à être fragile et sensibles à la torsion.
- M.2 (Mini PCI Express), connecteur qui permet de slotter (notamment) du stockage de masse directement sur la carte-mère via du PCIe (aucun câblage nécessaire).

1.6.2. Disques à plateaux (HDD)

Les disques durs à plateaux sont historiquement les plus anciens, mais sont encore très utilisés aujourd'hui.

Un **disque dur à plateaux** fonctionne avec des plateaux en rotation permanente. S'il tombe durant le fonctionnement, il sera irrémédiablement cassé et irréparable. En revanche, s'il est bien entretenu, un disque dur à plateaux peut théoriquement écrire et effacer des données à l'infini. Il y a très peu d'usure matérielle lors de l'écriture et de l'effacement. Après une défragmentation et un formatage, un disque dur à plateaux sera presque comme neuf.

1.6.3. Disques SSD

Les **disques durs SSD** (Solid State Drive) sont plus récents, plus rapide et sont devenus très abordables. Un SSD ne contient aucune pièce mobile, ce qui le rend plus rapide (mémoire flash) et plus résistant aux chocs. Il existe actuellement quatre types de mémoires flash SSD :

- SLC NAND – Single Level Cell (1 bit par cellule) ~ 100 000 cycles
- MLC NAND – Multi Level Cell (2 bits par cellule) ~ 10 000 cycles

- TLC NAND – Triple Level Cell (3 bits par cellule) ~ 3 000 cycles
- QLC NAND – Quadruple Level Cell (4 bits par cellule) ~ 1 000 cycles

Contrairement à la mémoire d'un disque à plateaux, **la mémoire flash s'use à chaque écriture**. Moins il y a de cellules, plus le disque sera durable, mais moins on aura d'espace de stockage. Les disques SLC sont également beaucoup plus chers, et sont de facto utilisés uniquement dans les milieux professionnels. Cependant, pour un usage privé, même intensif, un MLC récent peut en théorie tenir 300 ans et un TLC plus de 100 ans ! (voir https://www.compuram.de/blog/en/the-life-span-of-a-ssd-how-long-does-it-last-and-what-can-be-done-to-take-care/).

On ne **défragmente JAMAIS un SSD**, car cela génère des écritures/effacements néfastes au disque.

Attention aux **SSD DRAM-less** qui n'ont pas de mémoire RAM. En effet, les SSD possèdent normalement une puce DRAM qui permet au système d'exploitation de localiser facilement les données sur le disque. L'absence de cette puce fait baisser le prix du SSD, mais aussi drastiquement ses performances et sa durée de vie. Évitez donc cette économie qui n'en est pas une.

Attention aussi à l'appellation « **SLC cache** » ou « **SLC caching boosts** » qui n'indique absolument pas qu'il s'agit d'un disque SLC ; c'est en réalité une technologie d'écriture récente qui est utilisée sur les MLC et TLC.

1.6.4. Non-Volatile Memory Express (NVMe)

Il existe aujourd'hui une spécification pour accéder à du stockage de masse en SSD via des ports PCI-Express : le NVMe. Grâce à lui, de nombreuses opérations peuvent être réalisées en parallèle au niveau du stockage (un peu comme un CPU qui travaille avec plusieurs cœurs), ce qui permet des vitesses de transfert jusqu'à 3 Go/s (5x plus rapide qu'un SSD en SATA). Pour les desktops, le format de connecteurs utilisé pour cette technologie est le M.2 (toutes les cartes-mères récentes ont au moins un connecteur M.2 ; le M.2 tend même à remplacer le mSATA dans les laptops).

Lors d'un achat, faites bien attention à acheter du NVMe M.2, car du SATA M.2 existe aussi et ne fournira aucun bénéfice significatif comparé à un SSD SATA classique. Le M.2 n'est en effet qu'un type de connecteur, il peut y avoir différents types de bus et de spécifications derrière qui transportent les données : **M.2 n'est donc pas toujours égal à NVMe** !

1.6.5. Systèmes de fichiers

Sous Windows, 3 types de « files systems » coexistent :

- FAT (File Allocation Table ; obsolète, date de 1977) : permettait la gestion de disques de maximum 2 Go.

- FAT32 : permet la gestion de disques jusqu'à 2 To, mais des fichiers de max 4 Go. Encore utilisé pour les supports ammovibles.
- NTFS (New Technology File System) : jusqu'à 256 To de disque et il n'y a plus de limite de taille de fichiers. C'est le FS par défaut sous Windows.
- exFAT (extended FAT) : évolution de FAT pour les supports externes.

Sous Linux, les types de file systems sont :

- Ext3 : apporte des facultés de journalisation.
- Ext4 : permet d'atteindre 1 Eo de stockage (1 000 000 To).
- SWAP : pour de la mémoire virtuelle.
    - Tmpfs : système de fichiers stockés en mémoire RAM

1.6.6. Taille réelle des disques durs

À cause d'une confusion sur la signification des multiplicateurs K, M, G, T, on ne parle pas de la même quantité d'octets quand il s'agit de mémoire vive et de mémoire disque.

Pour la mémoire vive, on parle en binaire et donc en puissance de 2. Un Ko = 2^{10} = 1 024 octets. Chaque multiplicateur de plus rajoute 10 à l'exposant de 2 (1 Mo = 2^{20}…). En revanche, chez les fabricants de disques, ces multiplicateurs sont utilisés avec des puissances de 10, comme le système international le prévoit…

Unités		Base 2	Total	Base 10	Total
Octet	o	2^0	1	10^0	1
Kilooctet	Ko	2^{10}	1024	10^3	1000
Mégaoctet	Mo	2^{20}	1 048 576	10^6	1 000 000
Gigaoctet	Go	2^{30}	1 073 741 824	10^9	1 000 000 000
Téraoctet	To	2^{40}	1 099 511 627 776	10^{12}	1 000 000 000 000

Résultat, quand on branche un disque de « 1 To », il n'a pas la capacité attendue si on en croit notre ordinateur ! Sans présenter l'équation, sachons qu'il faut diviser la capacité présentée par les fabricants par 1,073741824 pour avoir la capacité « réelle » d'un disque.

Annoncée	Réelle
500 Go	465,66 Go
1 To	931,32 Go
2 To	1,862 To

1.6.7. Privilégier à l'achat

Pour l'installation de l'OS et des programmes gourmands en input/output, on privilégiera un SSD, tandis que pour le stockage de données importantes, les disques à plateaux restent meilleur marché (même si les SSD MLC/TLC de plusieurs To se démocratisent). Il est utile d'acheter toujours deux disques séparés pour conserver des données précieuses si le disque principal venait à lâcher.

Niveau marque, on restera sur du fiable avec Western Digital, Seagate, Samsung…

1.7. Écran

`en: monitor`

Il existe de nombreux types d'écrans, avec des spécificités qui dépassent le domaine de l'informatique pure. On entre dans de la physique précise, qui ne nous intéressera pas pour ce cours. Pour l'aspect guide d'achat, vous trouverez tout de même ci-dessous quelques infos utiles.

1.7.1. Ecrans incurvés (en: curved monitors)

Une mode pour des écrans incurvés est née il y a quelques années. Il faut savoir quelques détails importants les concernant :

- La courbure est plus notable et profitable lorsqu'on est **proches** de l'écran (très peu d'intérêt pour des télévisions, par exemple) et qu'on est installé de façon **parfaitement perpendiculaire** par rapport à l'écran ;
- la courbure permet une meilleure **immersion** et donne un **champ de vision accru** grâce à une répartition plus uniforme de la surface (par rapport à nos yeux), donnant l'impression que l'écran est légèrement plus grand que ce qu'il n'est vraiment ;
- cette courbure devient vraiment utile pour les écrans **très** larges (à partir d'un ratio de 21:9), car les coins sont ainsi rapprochés de nos yeux.
- une courbure faible n'augmentera que très peu le champ de vision (environ 1°), tandis qu'une courbure trop agressive pourrait carrément diminuer le champ de vision (on fera confiance aux fabricants pour éviter que ça ne se produise…) ;
- un écran incurvé a tendance à être beaucoup plus cher, on comparera donc intelligemment pour savoir si le faible gain en confort vaut les € supplémentaires.

Voici une vidéo très informative à ce sujet : https://youtu.be/-3aK0k2Is6Q

1.7.2. Connecteurs

Les écrans peuvent être connectés à la carte-mère ou au GPU via différents types de connecteurs. Voici les plus utilisés (écrit en janvier 2021) :

VGA

- Devenu obsolète, car ne suffit plus aux écrans actuels ; à éviter.

DVI-D/DVI-I

- Assez ancien, mais tient encore la route.
- Connecteur très versatile, car avec des adaptateurs, on peut y brancher des câbles VGA ou HDMI.
- Max résolution de 2560x1600 à 144 Hz.
- Ne permet de transmettre que des signaux vidéo.

HDMI (2.1)

Existe en version « mini » et même « micro », ce qui ne change en rien la qualité.

- Permet de connecter presque tous les appareils vidéo de nos jours.
- Transmet les données vidéo, audio et trames Ethernet.
- Compatibles avec de très nombreuses résolutions et fréquences (depuis HDMI 2.0).
    - Atteint 4K à 144 Hz (HDMI 2.1).
- Ne présente pas de mécanisme de verrouillage à au niveau du connecteur.

DisplayPort (1.4)

Le DisplayPort a l'avantage d'être pensé pour l'usage avec des ordinateurs, contrairement aux normes précédentes qui se veulent plus généralistes.
Existe en version « mini », ce qui ne change en rien la qualité.

- Connecteur 100 % compatibles avec toutes les autres normes citées, moyennant un adaptateur.
- Transmet vidéo, audio, données réseau et encore d'autres.
- Compatibles avec virtuellement tous les formats d'images et fréquences.
    - Atteint 4K à 144 Hz, 8K à 120 Hz.
- Permet de brancher plusieurs écrans sur le même port (un écran est connecté, et les écrans additionnels sont connectés entre eux = « daisy configuration »).
- Présente un mécanisme de verrouillage au niveau du connecteur.

En gros, si vous achetez un écran dernier cri, le DisplayPort est le connecteur le plus adapté.

Bref…

Faites attention en achetant un écran qu'il présente bien les connecteurs adaptés à la qualité que vous souhaitez atteindre et que votre GPU possède également ces connecteurs.

1.7.3. Taux de rafraichissement (en: refresh rate)

Gardez aussi à l'esprit que des écrans avec des fréquences de rafraichissement incroyables de genre 240 Hz sont souvent inutiles pour le gaming, puisque le hardware peine à atteindre autant de frames par seconde dans les jeux modernes, surtout quand on veut de la 4K. Si vous préférez un max de fps (un affichage fluide) plutôt qu'une densité de pixels la plus élevée possible (une image nette), privilégiez le 1440p (voire le 1080p).

1.7.4. Temps de réponse (en: response time)

Une dernière considération à l'achat concerne le temps de réponse de l'écran, exprimé en milliseconde. Ce temps de réponse représente le temps qu'une LED met à changer totalement de couleur et à revenir à la couleur de départ (on mesure souvent « de gris à gris »). Cependant, les méthodes de mesure ne sont pas très uniformisées…

Un temps de réponse court a tendance à résulter en une image plus floue. Par contre, un temps de réponse très rapide peut résulter en une image moins lumineuse et moins colorée. Ainsi, pour du gaming multijoueur, viser 1 ms ou moins est idéal, mais pour l'achat d'une télévision, un temps de réponse plus élevé est même préférable.

1.7.5. Paramétrage

Une fois que votre écran est connecté à votre GPU et que tous les drivers et apps de monitoring sont installés, vérifiez que les paramètres d'affichage sont bien corrects. En effet, il n'est pas rare que Windows laisse un taux de rafraîchissement par défaut de 60 Hz ou que l'outil de gestion de votre GPU (p.ex. NVidia Gefore Experience) n'ait pas pris en compte certains détails.

1.8. Cartes d'extension

en: expansion cards

Une carte d'extension ajoute des fonctionnalités à un ordinateur/serveur : carte son, périphérique vidéo pour plus d'écrans, carte réseau, carte TV,… Pour la bureautique, de nos jours, la carte-mère intègre tout ce qui est réellement nécessaire.

1.8.1. Cartes graphiques (GPU)

Les cartes graphiques (GPU ; Graphical Processing Unit) ne sont plus nécessaires pour faire fonctionner un ordinateur, puisque le processeur (CPU) principal est capable de gérer l'affichage vidéo. En revanche, pour du gaming ou de la modélisation 3D, le GPU fait office d'ouvrier très spécialisé et réalise ces tâches beaucoup mieux qu'un CPU (qui est avant tout construit pour être extrêmement polyvalent).

Les cartes graphiques viennent avec leur propre mémoire RAM, qu'on appelle de la mémoire graphique ou de la mémoire vidéo (VRAM). Une bonne quantité de VRAM est essentielle pour faire une bonne carte graphique.

Les cartes graphiques fournissent des sorties vidéos supplémentaires en face arrière, incluant de nos jours des DVI et/ou HDMI et/ou DisplayPort. Pour profiter de votre GPU, **c'est là qu'il faudra impérativement brancher votre écran** ! Les ports vidéo de la carte-mère n'utilisent pas du tout le GPU.

Niveau marque, on privilégiera uniquement nVidia et AMD pour les GPU, associé à des partenaires fiables et connus pour les circuits imprimés comme ASUS, MSI et Gigabyte (il y en a plein d'autres, mais vérifiez vous-même la fiabilité). Par contre, pas besoin de prendre le modèle le plus cher avec des RGB partout et des ventilos de data center.

Pour checker les GPU les plus puissants, différents « benchmarks » existent : https://www.videocardbenchmark.net/high_end_gpus.html

Pour des chiffres en jeu, voir aussi la chaîne PC Benchmark : https://www.youtube.com/channel/UCpc_uUd25ObT3pQGbr5MpHg

2. Matériel Serveur

Un ordinateur traditionnel est utilisé comme **poste de travail**. Un ou deux utilisateurs vont s'en servir et il fonctionnera quelques heures par jour. Pour un **serveur**, c'est une toute autre histoire. Il tourne souvent tout le temps, est utilisé par de nombreuses personnes, et on y stocke des données très sensibles. Sa conception doit être rigoureuse et utiliser du hardware robuste pour éviter un maximum les pannes.

Un serveur peut servir à de nombreuses choses : contrôleur de domaine, serveur web (héberge des sites), serveur FTP (héberge des fichiers)…

Ce qui différencie réellement un serveur d'un desktop, ce sont **les logiciels qu'on y installe**. Être un **serveur** est avant tout un rôle qu'un ordinateur prend pour rendre des services à des **clients**. C'est pour cela que du matériel desktop peut suffire à créer un serveur, mais on repassera pour la fiabilité ! Le hardware classique des desktops n'est pas conçu pour gérer de nombreux inputs ou pour tourner 24h/24 à fond.

2.1. Composants de base

On retrouve les mêmes composants que dans un PC : un (gros) boitier, une (ou des) alimentation(s), une carte mère, un (ou des) processeur(s), de la RAM, des disques durs, éventuellement des lecteurs/graveurs, et parfois des cartes périphériques additionnelles.

Contrairement à un PC, on va rarement choisir les pièces une à une. On choisit une marque (HP, Dell, IBM…) et on prend tout chez elle.

2.2. Types de serveurs

La plupart des serveurs sont constitué de racks de 19'' (48,26 cm) que l'on range dans un des trois formats suivants :

2.2.1. Armoire serveur

Il existe des armoires verticales de différentes tailles (unités, notées U ; le nombre d'étages de rangement ; ces étages mesures 1,75'' de haut = 4,445 cm) dans lesquelles on vient slotter ces différents racks à l'horizontale. Tout le matériel réseau pro se conforme à ces normes de taille. Certains modules peuvent mesurer plusieurs U de haut.

Une armoire classique mesure 42U de haut.

2.2.2. Serveurs Blade

On trouve également des serveurs blades (lames) qui ont l'avantage d'être moins encombrant. Ils permettent de mutualiser les ressources : alimentations, connectique réseau… Les racks blade sont slottés à la verticale.

2.2.3. Format tour

Il est possible de faire de petits serveurs avec des boitiers « classiques », mais il faudra surtout avoir assez de baies de 5,25'' pour pouvoir racker suffisamment de disques durs.

2.3. Front panel

La majorité du matériel installé dans un serveur est connecté directement au réseau. Cela demande énormément de connectique. Pour rendre le travail propre et clair, on va mettre un **front panel** (panneau de brassage, ou encore patch panel) sur lequel on fixe toutes les arrivées de câbles.

2.4. Alimentation

Dans les serveurs, on met souvent plusieurs **alimentations redondantes**. Cela permet d'en avoir de rechange si l'une venait à tomber en panne. Les alims peuvent également être « rackable à chaud », c'est-à-dire être remplacée sans arrêter le système.

Les alims se rackent à l'avant ou à l'arrière, ça dépend du matériel.

2.4.1. Uninterruptible power supply (UPS)

De nombreux serveurs sont équipés d'un UPS qui prend le relais en cas de coupure d'électricité. Cela permet d'assurer le service pendant un certain temps même sans alimentation par le réseau électrique.

2.5. Carte-mère

La carte-mère d'un serveur est généralement propriétaire. Elle accueille bien souvent deux processeurs, mais certaines peuvent en accueillir quatre ou plus. Le nombres de slots pour accueillir de la mémoire vive est aussi plus élevé que dans un PC standard.

Comme pour les carte-mères de PC, on trouve aujourd'hui sur celles des serveurs beaucoup de connecteurs qui n'étaient disponibles que via des cartes d'extension par le passé : KVM, USB, cartes réseau, interface de contrôle et de monitoring à distance…

2.6. Processeurs

Les serveurs utilisent des modèles de processeurs spécifiques, qui ont quasiment toujours des sockets différents des processeurs qu'on trouve sur les desktops. Les familles **Intel Xeon** et **AMD Epyc** sont par exemple réservées à un usage serveur. Ils ne gèrent aucune interface graphique. Ils ont plus de cores, on en voit parfois jusqu'à 64.

2.6.1. Refroidissement

Les processeurs des serveurs dégagent encore plus de chaleur que ceux des PC. Il faut donc une ventilation plus importante (ce qui génère beaucoup de bruit) et des radiateurs de meilleure qualité.

Bien souvent, comme les alimentations, les ventilateurs sont remplaçables à chaud.

2.7. Mémoire

Sans surprise, on retrouve la même chose que sur les PC :

- De la mémoire de travail processeur (cache) qui coûte cher ;
- de la mémoire vive avec la RAM qui est rapide, mais volatile ;
- du stockage de masse, généralement sous forme de disques durs, qui peuvent être des HDD ou des SSD.

2.7.1. FB-DIMM (RAM)

La mémoire RAM d'un serveur doit être encore plus fiable que celle d'un PC. Elle est « **fully buffered** », c'est-à-dire qu'elle utilise une mémoire tampon entre le contrôleur mémoire et le module. Cela augmente ainsi la fiabilité et permet une correction des erreurs.

Il y a aussi des **ECC unbuffered**, qui contiennent des bits de contrôles sans tampon.

On parlera aussi de **Registered ECC** (RECC), de la RAM qui possède un registre tampon pour encore augmenter la fiabilité d'accès aux données.

Ces barrettes sont bien plus chères que celles des desktops, et elles ne en général sont pas compatibles avec les cartes-mères et les CPU de ceux-ci.

2.8. Mémoire de masse

On utilise rarement la norme SATA pour connecter des disques durs à un serveur, mais plutôt du **SCSI** ou du **SAS** :

Norme	Bande passante (Mo/s)	Nombre périph par contrôleur
SCSI-1	5	7
SCSI-2	10	7
SCSI-2-WIDE	20	15
SCSI-2-Ultra Wide	40	15
SCSI-3-Ultra-2 Wide	80	15
SCSI-3-Ultra-160 Wide	160	15
SCSI-3-Ultra-320 Wide	320	15
SCSI-3-Ultra-640 Wide	640	15
SAS	375	128 (théorie)
SAS 2.0	750/1500	128 (théorie)

Les disques durs viennent à plusieurs dans un serveur et sont rangés en baies. La majorité des serveurs permettent un monitoring de « l'état de santé » des disques durs pour prévoir des remplacements (qui peuvent aussi se faire à chaud). Les disques durs sont disponibles en face avant et indiquent généralement leur état de santé avec un LED (rouge = pas cool).

2.8.1. Norme SCSI

Les bus SCSI permettent un transfert rapide de données, mais gèrent uniquement les HDD. Il faut un contrôleur dédié pour la gestion des disques durs ; ce ne sont pas les processeurs des serveurs qui s'en occupent (contrairement à ceux des desktops).

2.8.2. Norme SAS

La norme SAS (Serial Attached SCSI) combine le meilleur du SCSI et du SATA. Permet d'éviter que la bande passante ne soit partagée entre les périphériques (contrairement au SCSI). Permet également d'augmenter les débits, alors qu'avec le SCSI, on est limité par la dégradation des signaux (à cause de l'augmentation de fréquence).

On peut venir brancher un disque SATA sur un port SAS, mais pas inversement. Cela permet d'économiser un peu d'argent, mais on réservera les disques SATA à des usages moins intensifs (type back-up).

On trouve maintenant des disques SSD au format SAS pour être intégrés directement dans les baies de disques des serveurs.

2.8.3. RAID (Redundant Array of Independent Disks)

Le RAID est une technologie de virtualisation permettant de faire fonctionner de multiples disques durs comme s'ils ne formaient qu'une seule entité. De nos jours, on utilise majoritairement les technologies **RAID1** et **RAID5**.

Le **RAID1** fonctionne en miroir. Il permet de recopier intégralement les données sur deux disques distincts. On a donc une sauvegarde des données si l'un des disques venaient à périr. L'inconvénient est évidemment qu'on ne profite que de la moitié de l'espace de stockage total.

Le **RAID5** fonctionne comme un « volume agrégé à parité répartie ». Pour fonctionner, il faut au moins quatre disques et un disque de rechange (*en: spare*) qui prend le relai si l'un des disques lâche. On ne peut avoir qu'un seul disque défectueux à la fois, il faut rapidement changer un disque en panne.

2.8.4. Back-ups

La technologie RAID ne s'occupe que du stockage, il n'y a **aucune sécurité** intégrée dedans. Il est important de faire des **back-ups** des données, au cas où on rencontrerait des problèmes (virus, inondation, incendie...). Il est aussi important de faire des back-ups distants ; si on fait des back-ups sur place et que tout prend feu, on n'est pas avancés. On peut par exemple faire des back-ups en **cloud**, mais les entreprises s'en méfient pour des raisons d'espionnage industriel. C'est pourtant très pratique, car une entreprise n'a pas à se soucier de l'aspect logistique de son stockage et de ses back-ups. L'inconvénient, c'est qu'elle alors dépend d'Internet pour accéder à ses données ; elle n'est pas maître du produit.

Il n'est pas rare, encore aujourd'hui, de faire des back-ups sur **bandes magnétiques** ; les données restent intactes même si le système d'écriture est endommagé et on peut sortir facilement les données en cas de besoin. En revanche, la vitesse est très lente sur des bandes !

Pour la sauvegarde de données des clients, on peut utiliser un **NAS** (Network-Attached Storage) qui est un serveur de fichiers autonome accessible via le réseau. Il connaît de multiples protocoles et la sauvegarde des données peut être automatisée.

La diminution du prix des HDD et l'augmentation de leur fiabilité a permis la mise au point de baies de stockage au format 19'' en utilisant une interface SCSI, SAS ou du **Fiber Channel**.

2.8.5. Fiber Channel

Le **Fiber Channel** (FC) est un protocole permettant une connexion haut-débit. Il permet de transporter plusieurs protocoles comme IP et SCSI. C'est un protocole série qui peut fonctionner sur de la fibre optique, de la paire torsadée (RJ45 ; *en: twisted pair*) ou du coaxial.

Génération	Débit (Mb/s)	Disponibilité
FC 1 Gb	125	1997
FC 2 Gb	250	2001
FC 4 Gb	500	2004
FC 8 Gb	1000	2005
FC 10 Gb	1250	2008
FC 16 Gb	2000	2011
FC 32 Gb	4000	2016
FC 128 Gb	16000	2016

2.9. KVM (Keyboard, Video, Mouse)

Un serveur n'a pas d'écran, ni de clavier. Il faut venir brancher un kit spécifique dessus, une console, pour interagir avec lui et le programmer. Des switchs permettent de mutualiser un même clavier/écran pour agir sur plusieurs serveurs dans une même armoire, sans devoir débrancher puis rebrancher la console. Il existe des modèles de KVM rackable prenant 1U.

2.10. Interface de contrôle à distance

Chaque marque possède son **interface de contrôle**. Elles permettent de gérer les serveurs à distance via une console SSH (Secure Shell) et même de les redémarrer, voire réinstaller, sans être sur place.

2.11. Salles serveurs

Étant donné le bruit et la chaleur dégagée par des serveurs (qui sont rarement seuls, d'ailleurs), on va les placer dans des pièces dédiées. Cela sert aussi à protéger physiquement le matériel (on ne voudrait pas que Jacky renverse son Coca dessus) et cela protège les données qui sont sur les disques (il n'y a pas besoin d'être un grand hacker pour voler des disques durs). Sans parler du fait que le matériel serveur coûte cher et qu'on aimerait autant éviter de se le faire voler.

3. Systèmes d'exploitation

Les **systèmes d'exploitation** (*en: operating systems; OS*) sont des logiciels qui font le lien entre la **couche utilisateur** (les logiciels applicatifs) et la **couche matériel** (le hardware). Les OS sont indispensables, car ils permettent la gestion des ressources de la machine et représentent la base pour le développement et l'exécution des programmes.

Le rôle précis d'un OS est d'allouer les ressources matérielles (mémoire, temps processeur) et de s'assurer que les processus ne se gênent pas les uns les autres par un système de priorités et d'**ordonnancement**. L'OS doit aussi gérer la communication inter-processus.

Les premiers systèmes informatiques n'avaient pas d'OS. Il fallait donc gérer les instructions machine pour chaque programme que l'on créait ! Malheureusement, ces instructions sont différentes pour chaque processeur, ce qui rendait la tâche très peu efficace.

3.1. BIOS

Avant d'arriver sur un OS, c'est le BIOS qui travaille. Le BIOS (ou Basic Input Output System) est un **micrologiciel** (*en: firmware*) stocké dans une puce située sur la carte-mère. Son rôle est de contrôler des fonctions de base telles que l'alimentation et le bouton d'allumage. Il se lance dès l'allumage de la machine et effectue les opérations suivantes :

- identification et initialisation du hardware branché sur la carte-mère ;
- diagnostics pour voir si le hardware fonctionne correctement (POST) ;
    - stabilité de l'alimentation ;
    - initialisation correcte du processeur ;
    - intégrité du BIOS ;
    - test mémoire ;
    - test clavier…
- appel de l'OS (boot) à des sources prédéfinies.

Ensuite, le BIOS passe le relai à l'OS et son travail s'arrête là.

3.1.1. Mémoire CMOS

Le BIOS a une mémoire (qui contient le firmware), mais on ne peut pas écrire dessus. Les paramètres du BIOS (heure, date, séquence de boot, paramètres hardware…), sont en fait stockés dans la **mémoire CMOS** (une autre puce). Cette puce est volatile, il lui faut donc de l'énergie pour garder ces infos ; pour cela, il y a une pile pour l'alimenter en permanence (une sorte de pile de montre) qui permet de conserver les settings même sans alimentation secteur.

Si on retire cette pile, ou si elle est épuisée, tout est reset aux settings d'origine. Cela peut s'avérer utile pour débugger certaines machines en repartant de zéro : on appelle cela un **reset CMOS**.

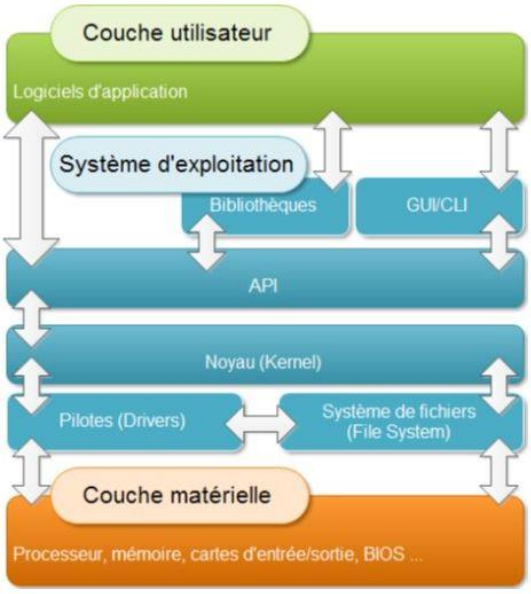
Les mémoires CMOS n'ont plus toujours une puce dédiée, elles sont souvent intégrées à l'horloge du pont sud du chipset.

3.1.2. UEFI (Unified Extensible Firmware Interface)

L'UEFI est le type le plus récent de BIOS, qu'on trouve sur à peu près toutes les cartes-mères. Il permet des interfaces plus user-friendly, permet d'utiliser la souris, et affiche même

des animations simples (ventilateurs, notamment). Il possède quelques fonctions de protection, notamment il évite le chargement de drivers foireux ou de malware.

3.2. Architecture des systèmes d'exploitation



On définit un **programme** (*en: software*) comme étant « une suite d'instructions » (souvent stockée sur un support de masse).

Le **processeur** est « l'agent qui exécute les instructions d'un programme » quand l'OS lui demande.

Les **processus** sont « les données des programmes en cours d'exécution ». Un processus peut avoir de nombreux sous-processus tourant dans le même contexte ; on les appelle **threads**.

Les **pilotes** (*en: drivers*) sont des « programmes permettant d'interagir avec un périphérique » (souris, clavier…).

Le **noyau** (*en: kernel*) est le cœur de l'OS. Il contient toutes les fonctions fondamentales de la gestion mémoire, la gestion des processus…

L'**API** (Application Programmable Interface) est l'interface de programmation qui contient des fonctions permettant à l'utilisateur d'interagir avec le système.

Les **bibliothèques** constituent l'ensemble des fonctions pouvant être utilisées dans les applications utilisateurs.

Les **file systems** (voir 1.6.3.) s'occupent d'organiser le stockage de masse.

3.2.1. GUI vs CLI

La **GUI** (Graphical User Interface) et la **CLI** (Command Line Interface) sont les deux environnements de travail (des affichages) graphique ou textuel permettant d'utiliser un programme cible. Certains OS comme Linux peuvent être lancés directement en CLI. Il n'y a pas que les OS qui ont des CLI, toutes sortes d'équipements informatiques peuvent en avoir une.

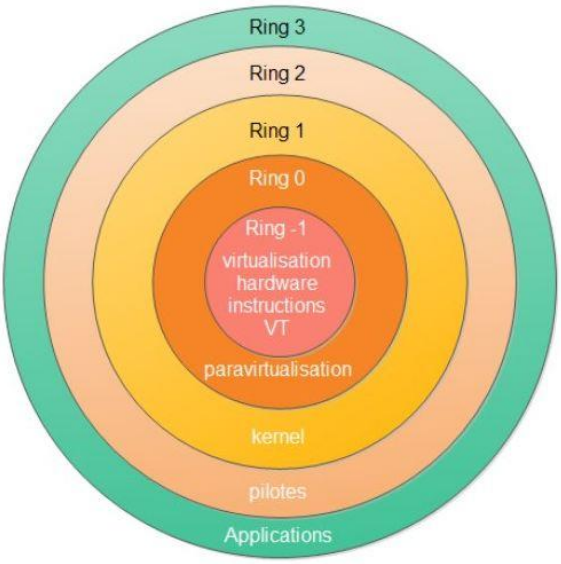
3.3. Mode privilégié

Pour avoir un accès total aux ressources de la machine (mémoires, processeur…), les OS s'exécutent en **mode privilégié**, qu'on appelle aussi **mode kernel**.

Les OS modernes des PC ont tendance à faire fonctionner les applications (couche user) en **mode non-privilégié** pour éviter de nombreux plantages ; elles ne peuvent donc pas écrire dans la zone mémoire utilisée par le kernel.
Ce n'était cependant pas le cas sous DOS, par exemple.

Attention : ces modes n'ont rien à voir avec le « mode administrateur ».

Les niveaux de privilège peuvent être représentés comme sur le schéma ci-contre.



La **paravirtualisation** optimise les performances de la virtualisation (machines virtuelles), mais exige un noyau adapté (concerne surtout les OS libres).

Anciennement, le noyau était au rang 0.

3.3.1. OS à un seul rang

Dans le monde des systèmes embarqués, on s'encombre de moins de choses (pas de mémoire virtuelle, p. ex.). Ce choix est fait pour optimiser l'utilisation de la place disponible et l'accès aux ressources. Tout s'exécute alors dans le noyau.

La programmation doit être pensée au cas par cas en fonction de la cible.

3.4. Types d'OS

De très nombreux OS ont été créés, ne fusse que durant les dernières décennies. Ils peuvent être rangés en **quatre catégories** :

- mono-tâche ;
  - Un seul processus est exécuté à la fois (ex : DOS).
- mono-utilisateur ;
  - Un seul utilisateur peut se connecter (ex : DOS).
- multitâche ;
  - Plusieurs processus peuvent fonctionner « en même temps » (Windows, Linux, MacOS…).
- multi-utilisateurs.
  - Peut gérer plusieurs utilisateurs simultanément (de facto, il y a plusieurs tâches en même temps, donc ces OS sont également multi-tâches).

3.4.1. Ordonnancement

L'ordonnancement (*en : task scheduling*) est la capacité de l'OS à déterminer dans quel ordre effectuer les tâches, de façon à profiter au mieux du processeur (géré par des algorithmes).

On distingue deux sous-catégories d'OS : les **OS en temps réel** (*en : Real-Time Operating System ; RTOS*) et les **OS à temps partagé** (qui sont les plus répandus de nos jours dans les PC).

Temps partagé

Le « temps partagé » s'occupe de permettre à plusieurs utilisateurs de travailler simultanément (ici « utilisateur » peut signifier un « utilisateur logique », pas forcément une personne réelle). La différence avec le « multi-tâche » est subtile, car dans le cas du multi-tâche, il s'agit de faire fonctionner plusieurs processus d'une même application simultanément. Dans la suite de cette synthèse, c'est du « multitâche » qu'on parlera essentiellement.

Temps réel

On trouve encore aujourd'hui des RTOS, mais plutôt dans les systèmes embarqués (thermostats programmables, robotique, automates industriels…). Exemples de RTOS : Windows CE, RTLinux, Linux CNC… Avec ces OS, on a une **garantie de temps d'exécution**, mais le fonctionnement sera plus lent.

3.4.2. Multitâche coopératif (obsolète)

Avec ce type d'OS, chaque tâche donne explicitement la main au processus suivant. Ainsi, si un processus plante, tout est planté. Le partage des ressources est également assez peu efficace.

3.4.3. Multitâche préemptif

Avec ce système, du temps de processeur est attribué à chaque processus (quelques nanosecondes). Si le processus n'est pas terminé à la fin du temps imparti, le **scheduler/system agent** (l'ordonnanceur ; l'algorithme qui sert de « superviseur des processus ») sauve son état et le replace dans la file et exécute le programme suivant. L'état du processus doit être rechargé dans le processeur pour que le code soit exécuté à nouveau : on appelle ça la **commutation de contexte**. Cette opération se passe *très* vite. À tel point qu'on a l'impression de nos jours que toutes les tâches sont simultanées, mais ce n'est pas le cas ! On parle de **pseudo-parallélisme**. S'il y a plusieurs cœurs dans le processeur, le scheduler peut réellement faire tourner deux processus simultanément ; on parle alors de **parallélisme**. Il peut évidemment y avoir des files d'attente de processus sur les deux cœurs en même temps : on combine alors parallélisme et pseudo-parallélisme.

Avec ce système, si un processus plante, il ne fait pas planter toute la machine puisque ce n'est pas lui qui passe la main au suivant.

Certaines tâches sont prioritaires par rapport à d'autres. Elles peuvent être exécutées plus souvent, ou plus longtemps. Il existe aussi un système d'interruption qui permet d'exécuter une nouvelle tâche immédiatement (la saisie clavier, par exemple ; tout s'arrête pour qu'on puisse afficher à l'écran le caractère entré).

3.4.4. Multithreading

Le **multithreading** est une subdivision d'un processus en plusieurs processus. Cette division est réalisée au niveau de l'application ; il faut coder l'appli pour qu'elle fonctionne ainsi.

NB : le multitâche est réalisé au niveau de l'OS, mais le multithreading est un choix qui est fait par le programmeur qui prévoit le partage des ressources.

Chaque thread fonctionne comme un processus isolé (en parallèle ou pseudo-parallèle, selon le nombre de cœurs). Si une application est monothread, quand on doit accéder à une ressource, toute l'application est bloquée pendant le temps nécessaire à cet accès. Avec du

multithread, l'accès à la ressource peut se faire en parallèle de l'utilisation normale de l'appli, par exemple ; cela optimise donc encore davantage l'utilisation du processeur (mais il faut bien coder !).

Exemple de multithread : MS Word qui fait en pseudo-parallèle l'affichage de la frappe au clavier, le correcteur orthographique et la sauvegarde automatique. S'il y a plusieurs cœurs dans le processeur, ces trois processus peuvent être réalisés vraiment simultanément.

3.5. Les processus

On l'a vu, un processus est un programme en cours d'exécution sur un ordinateur. Il a besoin de temps processeur, d'espace mémoire et de certaines ressources. Son exécution prend un certain temps et peut être déclenchée par : un utilisateur, un évènement ou un autre processus. Les applications utilisateurs sont des ensembles de processus.

3.5.1. Création d'un processus

Un processus doit d'abord être créé. Il passe de l'état passif à l'état actif **en étant chargé dans la RAM**. Il est alors placé dans la file d'attente par l'ordonnanceur et il attend son tour pour être exécuté durant le temps qui lui est imparti.

1. Processus créé (naissance) ;
2. Passage en RAM et mis en file d'attente (attente) ;
3. Élection par l'ordonnanceur pour utiliser du temps processeur (exécution) ;
4. Fin d'exécution (mort) ou remise en fil d'attente si non terminé.

3.5.2. Interruption d'un processus

Les processus peuvent être interrompus. Ils passent dans un état « bloqué » (*en: sleep*) et passent en mémoire SWAP (voir 3.6.). Après déblocage, ils retournent dans la file d'attente. Cet état bloqué arrive quand un programme demande l'input d'un utilisateur, par exemple.

3.5.3. Process identifier (PID)

Pour chaque processus, l'OS attribue un PID (un nombre aléatoire) de manière identifier chaque processus sans faille (on peut voir le PID dans le gestionnaire des tâches sous Windows).

3.5.4. Processus père et fils

Certains processus dépendent d'un processus qui les a engendrés : on parle d'une hiérarchie père-fils. Si le processus père se termine, il peut se passer deux choses :

- tous les processus fils se terminent, car ils n'ont aucune raison d'exister seuls ;
- un des processus fils prend le relai et sert de père pour les autres.

Sur de nombreux OS, il y a un processus père au-dessus de tous les autres (SYSTEM, sous Windows ; INIT sous Linux) ; c'est lui qui coordonne tout. Si on tente de l'arrêter (pour autant qu'on puisse), cela rend le système instable et il va en général instantanément redémarrer (ou planter totalement).

On peut arrêter (« tuer ») des processus à la main, mais on ne réservera cette pratique que pour les processus plantés ; ce n'est pas anodin d'arrêter un processus qui est en pleine écriture.

3.5.5. Communication inter-processus

Les processus peuvent communiquer entre eux grâce à deux mécanismes : les **pipes** et les **sémaphores**.

Un pipe est une file d'attente FIFO (first in, first out) ; très pratique dans l'enchaînement de commandes.

Un sémaphore est une sorte de bâton de parole. Les programmes attendent que le bâton soit libre pour le prendre, mais il n'y a pas de bagarre pour le récupérer.

Il est aussi possible de faire communiquer des processus entre eux via le **réseau interne 127.0.0.1** (*en: localhost*). Ce réseau n'est accessible que sur l'ordi où l'on travaille et toutes sortes d'échanges sont possibles via cette adresse. Pendant qu'un processus **écrit**, les autres **écoutent**. L'information peut donc circuler entre de nombreux processus.

3.6. Mémoire SWAP

La **mémoire swap** est une mémoire lente qui est mise en place par l'OS sur le disque dur. Elle permet de stocker des processus endormis pour soulager la mémoire vive (RAM) ; c'est l'ordonnanceur qui s'occupe des échanges entre SWAP et RAM. La taille en Go de la mémoire swap doit être égale à la RAM. Windows stocke ces données dans de gros fichiers qu'on peut voir sur le disque C:\ ➔ **pagefile.sys** et **swapfile.sys**.

Ensemble, la RAM et la SWAP forment ce qu'on appelle la **mémoire virtuelle** (parfois « VRAM » ; à ne pas confondre avec la Video RAM des GPU !). Elle permet donc d'exécuter simultanément plus de processus que la mémoire vive seule ne pourrait en contenir. Chaque processus doit rarement manipuler toutes les données simultanément, c'est donc très pratique de pouvoir ne mettre que celles qui sont utiles dans la RAM. Quand un processus a besoin d'informations stockées dans la SWAP, elles sont remises en RAM pour être traitées.

3.6.1. Pages de mémoire SWAP

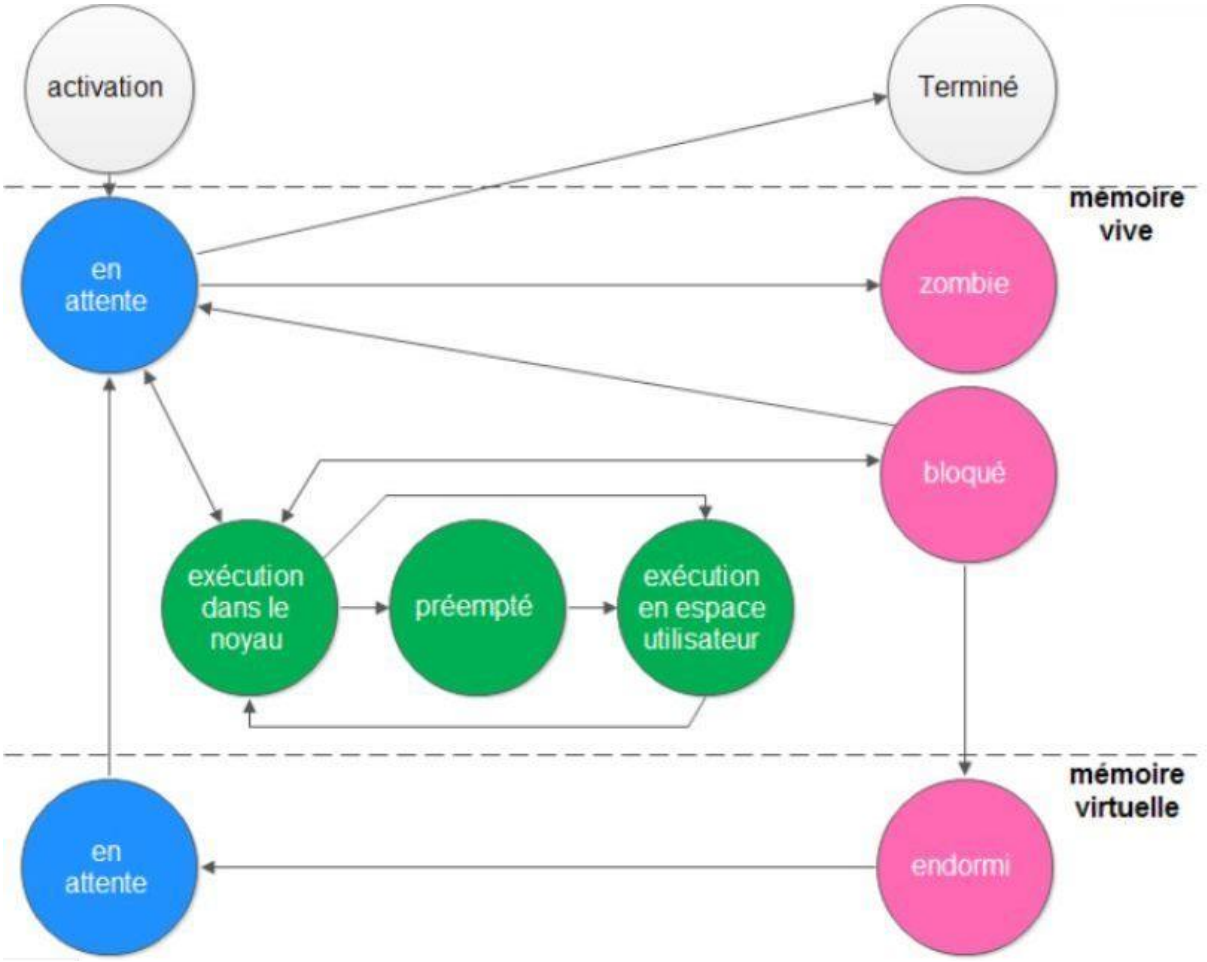
La mémoire SWAP est découpée en pages (de 4 Ko sous Linux). Ces pages ont un numéro propre et sont rangées à un numéro d'emplacement (adresse = deux numéros).

3.6.2. Cadres de mémoire RAM

La mémoire vive est séparée en cadres de tailles identiques. Une adresse physique est composée d'un numéro de cadre et d'un emplacement sur ce cadre.

3.7. Etats d'un processus

Dans un OS actuel, les différents états d'un processus peuvent être schématisés de la façon suivante :



Un processus **exécuté dans le noyau** a accès à l'entièreté de la mémoire ; il n'y a en théorie que l'OS qui a ce privilège.

Un processus **préempté** est suspendu par l'ordonnanceur et copié en SWAP pour libérer de la RAM.

Les processus venant d'applications sont **exécutés en espace utilisateur** ; ils sont limités à l'espace mémoire qui leur est attribué.

Un processus devient un **zombie** si on n'arrive pas à le décharger de la mémoire alors qu'il n'a plus de raison d'exister. Il faut le kill au plus vite. Dans le pire des cas, un reboot fera l'affaire.

3.8. Méthodes d'ordonnancement

Il existe de nombreuses méthodes d'ordonnancement des processus. Nous en verrons trois : le **round-robin**, le **SJF** et le **CFS**.

3.8.1. Round-Robin

Le round-robin (*fr : tourniquet*) est un algorithme qui attribue le même temps à chaque processus, sans priorités. Les processus sont sur un « carrousel » qui va à une vitesse fixe.

Un nouveau processus est ajouté en fin de liste. L'utilisation par un processus ne peut jamais dépasser un certain laps de temps. Un processus qui a consommé son temps est soit tué (il a fini), soit mis en fin de liste et préempté (swappé).

Logiquement, avec ce système, plus il y a de processus, plus le fonctionnement est lent. En effet, la commutation de contexte n'est pas instantanée : il faut remettre en place l'espace de travail pour qu'un processus suspendu reprenne dans l'état dans lequel il était. Si on accorde trop de temps à chaque processus, c'est lent parce que les processus attendent longtemps pour être exécutés. Si on accorde trop peu de temps à chaque processus, on perd plein de temps en commutation.

3.8.2. Shortest Job First (SJF)

Cet algorithme va exécuter en priorité le processus le plus court. Quand un nouveau processus arrive, si son temps d'exécution est plus court que tous les autres présents dans la file d'attente, il est placé en tête de file.

Cet algorithme est réservé aux environnements spécialisés, car il faut savoir précisément combien de temps chaque processus prend à être exécuté.

3.8.3. Completely Fair Scheduler (CFS)

Depuis 2007, c'est l'ordonnanceur du kernel Linux. Cet algorithme trie régulièrement les processus suivant le manque de temps d'allocation qu'ils ont par rapport à une allocation « idéale ». L'ordonnance fera donc exécuter en priorité celui qui est le plus en manque.

3.9. Systèmes distribués

Un système distribué est généralement un ensemble de machines autonomes connectées les unes aux autres afin d'effectuer des opérations contribuant à un projet global. Au niveau de l'utilisateur, le système fonctionne comme une seule machine : on parle de **clustering**. Il faut un **middleware** pour rassembler les données.

Internet permet de créer des systèmes distribués à travers le monde entier.

3.9.1. Clustering de serveurs

De la même façon, on peut regrouper plusieurs serveurs pour n'en faire « qu'un seul » et profiter d'une puissance de calcul énorme sans avoir besoin de hardware très cher.

4. Introduction aux lignes de commande

Les lignes de commandes sont tapées dans une **CLI**. Elles sont basiques, mais permettent du **scripting** très facilement ; une suite de lignes de commande à effectuer à la chaîne. Tous les OS ont une CLI, même Android (elle n'est juste pas facilement accessible).

Sur Windows, il y a deux CLI : l'invite de commande (souvenir de DOS) et PowerShell (plus puissant et fait tout ce que sait faire l'invite classique) ; voire même PowerShell ISE, qui a le mérite de gérer l'Unicode (entre autres). Powershell permet notamment de faire du scripting (pas l'invite de commandes) et de la programmation objet.

4.1. Commandes essentielles sous Windows

Les commandes ne sont pas sensibles à la casse. Le système de fichiers est basé sur des lecteurs auxquels on accède via une lettre, comme « C: » pour le disque dur principal.

4.1.1. Commutateur

Les commandes ont souvent des options. On les active grâce à un **commutateur**, souvent `/` suivi d'une lettre ou d'un mot. On peut utiliser plusieurs commutateurs en même temps : `dir /b/p`

4.1.2. Complétion automatique

Quand on tape une ligne de commande, on peut afficher la fin automatiquement **en appuyant sur la touche TAB**. S'il y a plusieurs commandes possibles sur base de ce qu'on a déjà taper, la première dans l'ordre alphabétique va apparaître. En continuant de taper sur TAB, on parcourt les possibilités suivantes.

On peut remplacer une ou plusieurs lettres d'une chaine par le caractère `?` : `dir wi??ows`

On peut remplacer une chaine de caractères, entière ou en partie, par le « joker » `*` : `dir win*` ou `dir *.txt`

4.1.3. Liste des commandes

- `c:\` : racine d'un disque ; l'arborescence est basée sur cette racine.
  - Exemple : `c:\windows\system32`
- `help` : donne la liste des commandes supportées
  - `/?` : permet d'obtenir de l'aide pour une commande précise
    - Exemple : `cd /?`
- `dir` : affiche les fichiers et dossier du dossier en cours. On peut aujourd'hui aussi utiliser `ls`, venu de la pratique sous Linux.
- `cd` ou `chdir` permet de changer de dossier.
  - `cd nom` : permet de se déplacer dans le sous-dossier « nom ». Si le nom du sous-dossier a plus d'un mot, il faut le mettre entre double guillemets : `cd "program files"`
  - `cd ..` : permet de remonter d'un niveau
  - `cd \` : permet de remonter à la racine du lecteur
  - …
- `md` ou `mkdir` permet de créer un dossier ou une arborescence de dossier
  - `md \nom` : crée un dossier à la racine
  - `md nom` : crée un sous-dossier dans le dossier en cours
  - `md \nom1\nom2` : crée à la racine le dossier « nom 1 » et dans ce dossier, le dossier « nom 2 ».

4.2. Commandes essentielles Sous Linux

Pas de concept de disques sous Linux. Les commandes sont sensibles à la casse.

4.2.1. Commutateur

Les options des commandes sous Linux utilisent le commutateur `-` qui est précédé d'un espace : `ls -l`. Plusieurs options peuvent être utilisées en écrivant ceci `ls -l-t` ; ou ceci `ls -lt`. Le double commutateur `--` permet de spécifier une option qui est un mot : `ls --option`

4.2.2. Complétion automatique

Quand on tape une commande, on peut afficher la fin automatiquement **en appuyant sur la touche TAB**. Si rien ne s'affiche, c'est qu'il y a plusieurs commandes possibles sur base de ce qu'on a déjà taper (ou bien aucune). S'il y a plusieurs possibilités, on peut appuyer **une seconde fois sur TAB** pour avoir la liste des possibilités.

Le caractère `.` remplace n'importe quel caractère avec `grep`. Le caractère `?` remplace n'importe quel caractère avec les commandes `find`, `cat`… Le caractère `*` (le « joker ») remplace une chaîne de caractère quelconque.

4.2.3. Liste des commandes

- `/` : racine ; si plusieurs disques durs sont présents, ils sont intégrés au système de fichier.
    - L'accès aux lecteurs en eux-même se fait en montant une partition dans un dossier : `mount /dev/cdrom` pour accéder au lecteur CD.
- `--help` : permet d'obtenir l'aide pour une commande précise
    - Exemple : `ls –help`
- `man` : Une aide plus complète encore
    - Exemple : `man ls`
- `ls` : affiche les fichiers et dossier du dossier en cours.
    - `ls -l` : permet de voir le contenu du dossier en cours de façon détaillée.
        - `ls -l /exemple` : permet de voir le contenu d'un autre dossier sans pour autant aller dedans.
    - `ls -l /` : permet de voir le contenu du répertoire racine.
- `cd` permet de changer de dossier.
    - `cd /nom` : permet de se déplacer dans le dossier « nom » si c'est un dossier de la racine.
    - `cd nom` : permet de se déplacer dans le sous-dossier « nom » à partir du dossier actuel.
    - `cd ..` : permet de remonter d'un niveau.
    - `cd \` : permet de remonter à la racine du lecteur.
- `mkdir` permet de créer un dossier.
    - `mkdir nom` : crée un sous-dossier dans le dossier en cours
    - `mkdir /nom1/nom2` : crée le dossier « nom 2 » dans « nom 1 », mais « nom 1 » doit déjà exister.