

Résumé de l'architecture élémentaire d'un PC

I - Matériel (hardware)

1) Unité centrale

a) Le processeur (CPU : Central Processing Unit)

Le processeur effectue les opérations élémentaires : calcul, comparaison, choix...

Il comprend :

- Une unité de commande constituée d'un compteur ordinal (CO ou PC (Program Counter) ou encore IP (Instruction Pointer)) qui stocke l'adresse de l'instruction, et d'un registre d'instruction (RI ou IR (Instruction Register)) qui stocke l'instruction à exécuter.
- Une unité de traitement constituée de registres qui stockent les données de l'opération à traiter, ainsi que son résultat, et d'une unité arithmétique et logique (UAL ou ALU : Arithmetic and Logical Unit) capable d'effectuer les opérations élémentaires. En général, elle possède également une unité dédiée au calcul avec une virgule flottante (FPU : Floating Point Unit) et une autre dédiée au chargement/stockage (LSU : Load Storage Unit).

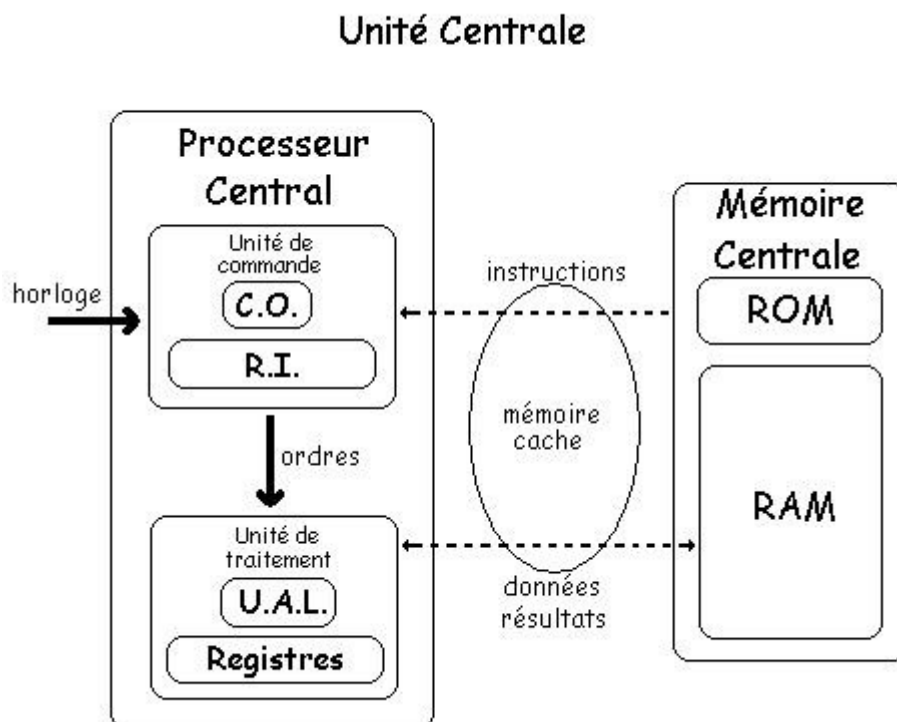
b) La mémoire centrale (main memory)

Elle contient les programmes et les données utilisées et/ou produites par ces programmes, elle est constituée de « cases » contenant des mots mémoires repérées par une adresse mémoire. On distingue deux parties :

- La mémoire morte R.O.M. (Read Only Memory), non modifiable, elle contient les instructions de base, le BIOS (Basic Input Output System), la procédure de démarrage : le POST (Power On Self Test).
- La mémoire vive R.A.M. (Random Access Memory), modifiable, le terme d'accès aléatoire veut dire qu'on peut accéder directement à n'importe quel endroit (contrairement à un accès séquentiel). C'est une mémoire volatile : les données ne sont pas conservées lorsqu'elle est mise hors tension.

Les opérations internes au processeur sont plus rapides que les transferts registres-mémoire centrale. Pour palier cela on introduit une mémoire intermédiaire, la mémoire cache (ou antémémoire), qui contient les instructions les plus fréquentes (proximité temporelle) ou proches de l'instruction en cours (proximité spatiale). Il peut y avoir plusieurs niveaux de cache (cache L1, L2...).

Illustration :



Architecture de Von Neumann

Cycle d'exécution d'une instructions :

- Phase 1 : l'adresse de l'instruction courante est stockée dans le CO, l'instruction est stockée dans le RI, après avoir transitée via le bus « instructions ».
- Phase 2 : l'instruction courante est décodée, l'unité de traitement va chercher les données nécessaires au calcul dans la RAM grâce aux adresses de ces données via le bus « données/résultats ».
- Phase 3 : l'U.A.L. Effectue l'opération et transfère le résultat de son registre résultat via le bus « données/résultats ».

2) Les périphériques

On distingue :

- Les périphériques d'entrée : introduction de l'information dans l'ordinateur (clavier, souris, scanner, micro, webcam, capteurs...)
- Les périphériques de sortie : transmission de l'information de l'ordinateur vers l'utilisateur (moniteur, haut-parleurs, imprimante, traceur...)
- Les périphériques de sauvegarde : stockage de l'information (disque dur, CD, DVD, clé USB...)

II - Logiciels (software)

1) Logiciels de base

Ce sont les systèmes d'exploitation, éditeurs de textes, compilateurs, éditeurs de liens...

Système d'exploitation :

Le système d'exploitation (UNIX, Windows, OS, Linux, ...) est la couche logicielle qui est entre le matériel et le programme utilisateur.

C'est lui qui traduit un ensemble d'instructions et de programmes de base en langage directement compréhensible par la machine (par exemple affichage à l'écran d'une lettre saisie par le clavier, ...). Il doit également :

- gérer le processeur (temps partagé...)
- Gérer la mémoire (répartition entre différents programmes en cours, traduction des adresses utilisées dans un programme en adresses physiques, ...)
- Gérer les entrées/sorties
- Gérer les fichiers (emplacement physique, table d'allocation, ...)
- Sécuriser (ne pas permettre la suppression du fichier d'un autre utilisateur...)

Il peut proposer à l'utilisateur des utilitaires (gestionnaires de fichiers, ajout de périphériques...)

Langage de programmation :

Le langage assembleur est un langage comprenant des instructions simples (LOAD charger, ADD additionner...).

Les langages les plus évolués (Fortran, Pascal, C, ...) contiennent des instructions plus complexes (répétitions d'opérations, utilisation directe de tableaux, opérations arithmétiques complexes, ...).

Les compilateurs sont des traducteurs qui transforme un programme écrit (ou source) en langage machine (i.e. compréhensible par le processeur).

Ils comprennent :

- un analyseur lexical (reconnaît les mots BEGIN, FOR, ...)
- un analyseur syntaxique (vérifie la cohérence des phrases)
- un analyseur sémantique (vérifie par exemple que les opérations portent sur des types compatibles).

Un programme peut être composé de plusieurs modules compilés séparément. Avant d'obtenir un exécutable, il y a donc une phase où on relie les différents codes, issus des différentes compilations, elle est assurée par l'éditeur de liens. Le fichier obtenu contient trois parties : une contenant les données statiques connues à la compilation, une contenant les instructions à exécuter, un espace mémoire appelé la pile destiné à recevoir les résultats intermédiaires, les paramètres d'appel d'une fonction, etc...

2) logiciels d'application

- Bureautique : traitement de textes (Word), tableur (Excel), présentation (Power Point)
- Dessin (D.A.O., C.A.O.)
- SGBD : systèmes de gestion de base de données (DB2, Oracle, Access, MySQL)
- Navigateurs (Netscape, Internet Explorer)
- PGI : progiciels de gestion intégrée, pour gérer finances, personnel, production, logistique...
- Calcul formel (Mathematica, Maple, Mathlab)
- jeux!