

# Chapitre :

## Architecture et systèmes d'exploitation



### I. Architecture Von Neumann

---

⊗ **Activité** : Simulateur de machine de Von Neumann pages 10-11

⊗ **Activité** : Ordinateur Cardiac et exécution d'un programme.

Dans un système informatique on trouve des composants physiques (la partie matérielle, en anglais hardware) et des programmes ou logiciels (le software) qui s'exécutent sur cette partie matérielle. Les programmes sont plus ou moins efficaces suivant les caractéristiques du matériel. Il est donc important de connaître le fonctionnement interne d'une machine pour optimiser les algorithmes et les rendre efficaces, rapides, etc.

L'architecture des ordinateurs actuels repose sur le modèle de Von Neumann, basé sur l'idée de programme enregistré. Cela signifie que la mémoire de l'ordinateur, dans laquelle sont stockées les données, contient également les programmes qui s'exécutent. Autrement dit, programmes et données se partagent l'espace mémoire de l'ordinateur.

Une machine contient :

- Une **mémoire** qui contient à la fois les données et les programmes ;
- Une **unité arithmétique et logique** (UAL) qui effectue les calculs et les tests logiques ;
- Une **unité de contrôle** qui gère la séquentialité des opérations ;
- Des dispositifs d'**entrée et de sortie** qui permettent la communication depuis et vers l'extérieur.

Le tout est rythmé par une **horloge interne** qui détermine la fréquence du processeur.

Entre ces différentes composantes, la communication de données, d'adresses, etc. se fait par des **bus**. Il existe différents types de mémoire, qui peuvent être placées à différents endroits dans l'ordinateur. On peut retenir que généralement les mémoires les plus proches du processeur (les **registres**) sont les plus rapides et les plus petites, quand les mémoires les plus éloignées sont les plus lentes et les plus grandes (comme les disques durs).

Le temps d'exécution a une grande importance en informatique, et la vitesse de calcul des machines n'a cessé d'augmenter. Pour cela, le principal levier était la fréquence d'horloge. Cependant cela implique un échauffement du processeur. Depuis quelques années, on augmente le nombre d'unités de calcul, autrement dit le nombre de processeurs (ou de cœurs). Un **cœur** est en fait un ensemble composé d'unités de calculs et de commandes avec des registres. Il faut alors écrire des programmes capables d'utiliser plusieurs cœurs pour partager les temps de calcul (le **parallélisme**).

⊗ **Activité** : Réalisation de circuits logiques (nécessite le logiciel logisim)

## II. Réseau

---

Pour ce thème, lire les ressources disponibles sur Internet et les noter sur feuille de cours pour les retenir.

Par exemple, sur ces sites de cours de spécialité NSI : le site [numerique-sciences-informatiques.fr](http://numerique-sciences-informatiques.fr) et le site [pixees.fr](http://pixees.fr) (dans la section Architectures matérielles et systèmes d'exploitation, en commençant par la partie Introduction réseau).

Utiliser éventuellement d'autres sources.

On s'attachera à retenir les informations concernant le programme officiel de la spécialité, à savoir :

- Transmission de données dans un réseau ;
- Protocoles de communication ;
- Architecture d'un réseau ;
- Mettre en évidence l'intérêt du découpage des données en paquets et de leur encapsulation ;
- Dérouler le fonctionnement d'un protocole simple de récupération de perte de paquets (bit alterné). Voir pour cela sur le site [pixees.fr](http://pixees.fr) ;
- Simuler ou mettre en œuvre un réseau. On trouve une vidéo d'utilisation du logiciel Filius sur le site [pixees.fr](http://pixees.fr), qu'il suffira en première de regarder. Le logiciel pourra être utilisé en terminale.

En particulier, les éléments suivants seront à comprendre et à savoir :

- Les différentes tailles de réseau (PAN, LAN,...) ;
- Les différentes topologies de réseau (étoile, maillé,...) ;
- Les deux modèles pour les réseaux : TCP/IP et OSI. Connaître un peu leurs couches, et en particulier leur nombre ;
- À propos de l'encapsulation, savoir ce que sont et comment s'imbriquent trames, paquets et segments ; connaître un peu leur contenu, en particulier quels types d'adresses sont données dans chacun. Savoir dans quelles couches du modèle OSI ils sont traités ;
- Savoir ce quelle est la forme d'une adresse IP, en particulier IPv4. La connaissance de la forme des adresses IPv6 n'est pas requise mais conseillée ;
- Savoir également ce qu'est une adresse MAC et savoir à quelle couche du modèle OSI elle est utilisée (et le savoir aussi pour l'adresse IP) ;
- Savoir ce qu'est un masque de sous-réseau et savoir calculer le nombre de machines que l'on peut connecter sur un réseau étant donné son masque de sous-réseau. Savoir en particulier ce qu'est l'adresse de broadcast ;
- Connaître ce que sont les protocoles TCP et UDP et leurs différences. Savoir à quelle couche du modèle OSI ils correspondent ;
- Connaître la différence entre les hubs, les switchs (commutateurs) et les routeurs ;
- Connaître le protocole du bit alterné.

Une fois ces informations lues et notées, faire la fiche d'exercice sur les réseaux.

Un devoir surveillé avec des questions sur les points à connaître permettra de vérifier que les connaissances ont bien été retenues et comprises.

# III. Systèmes d'exploitations

---

Comme nous avons pu le voir, programmer en langage assembleur (langage machine) n'est pas facile, bien que ce soit le langage que comprend la machine. De la nécessité d'avoir un intermédiaire entre l'humain et la machine qui gère les ressources de la machine naît le système d'exploitation.

Un **système d'exploitation** (SE en français, OS en anglais pour Operating System) a pour tâches de :

- Fournir une interface entre l'humain et la machine ;
- Gérer les ressources de l'ordinateur (mémoire, processeur, périphériques, etc.) ;
- Gérer les utilisateurs, en particulier leurs droits d'accès et d'exécution des fichiers ;

Un système d'exploitation rend concret ce qui ne l'est pas : un fichier est par essence abstrait, c'est seulement une suite de 0 et de 1 placés quelque part dans la mémoire de la machine, mais le système d'exploitation nous permet d'en avoir une notion concrète, par une adresse.

Il existe beaucoup de systèmes d'exploitation, tels Windows 10, linux, mac OS X.

Chaque système d'exploitation possède un **terminal**, qui est un programme permettant d'exécuter des tâches à l'aide de commandes données au clavier.

Il y a des commandes qui sont généralement communes aux systèmes d'exploitation comme :

`ls` qui liste les fichiers et répertoires du répertoire courant.

`cd` qui change de répertoire (Change Directory)

`rm` qui supprime un fichier ou répertoire (ReMove)

D'autres commandes ne sont pas communes à tous les systèmes d'exploitation.

Nous utiliserons ici un système d'exploitation libre, à savoir linux, dont il existe de nombreuses distributions, et qui est basé sur le système UNIX (sur lequel est également basé mac OS X).

## IV. Entrées et sorties

---

Les dispositifs d'entrée/sortie sur un ordinateur sont constitués de périphériques connectés par câble sur des ports ou via une technologie sans fil (bluetooth, wifi, etc.). Les exemples les plus fréquents pour les ordinateurs sont les claviers, moniteurs, souris, imprimantes, etc.

L'utilisateur de ces périphériques envoie un signal à l'ordinateur (en entrée) qui réagit en fonction d'un algorithme, selon le principe de gestion des événements comme nous avons pu le voir par exemple avec JavaScript.

Pour faire cela, l'ordinateur fait en permanence fonctionner un programme qui boucle tant que l'ordinateur est allumé, et qui est à l'écoute des événements en provenance de ses périphériques, et qui réagit en fonction des données reçues et des programmes qui écoutent ces événements.

Dans d'autres cas, l'exécution d'un algorithme produit en sortie des données, par exemple sur un moniteur ou une imprimante.

Certains périphériques portent plutôt le nom de capteurs ou d'actionneurs. Ils équipent les [systèmes embarqués](#) comme les robots et les objets connectés.

Il s'agit par exemple de capteurs de son, de lumière, de température, etc. et les actionneurs sont par exemple des moteurs ou des diodes.

Autrement dit, un capteur est un dispositif d'acquisition de données (physiques), et un actionneur est un dispositif permettant au système d'agir.

Ces éléments sont mis en œuvre dans les robots, qui sont des objets équipés de systèmes embarqués, éventuellement connectés par wifi à des ordinateurs qui peuvent les programmer.

Généralement, chaque système embarqué a son propre mode de programmation, mais le principe est toujours le même.

Il y a des fonctions qui interrogent les capteurs et d'autres qui commandent les actionneurs. Des algorithmes permettent alors d'établir le comportement souhaité en fonction des données captées.

## V. Interface homme machine

---

Une interface homme machine (IHM) est une interface utilisateur permettant à une personne de dialoguer avec un système. Il peut s'agir d'un écran qui affiche des informations lors du fonctionnement de la machine. Une interface graphique donne des moyens à l'utilisateur d'agir (en cliquant sur des boutons, en tapant du texte), mais aussi à la machine d'informer de l'état du fonctionnement.

Les premières interfaces homme machine se trouvent sous la forme des [télétypes](#) (ou téléscripteurs), qui sont essentiellement un système d'un clavier pour les entrées et une imprimante pour la sortie (il n'y avait pas d'écran). L'influence de ces systèmes est telle que sous Linux, les périphériques de type Console (ou Terminal) sont nommés, tty en référence à ces interfaces.

Les premiers environnements graphiques, avec écran, datent de 1968, où Douglas Engelbart, à Stanford, présenta un pointeur permettant d'ouvrir ou de fermer des fenêtres. C'est l'invention, autrement dit, de la souris.

À partir de 1983 naissent les premières interfaces utilisateur graphiques (GUI pour Graphical User Interface en anglais). Elles rendent l'utilisation plus simple des ordinateurs, permettant ainsi leur démocratisation.

Depuis quelques années se développent des interfaces naturelles (NUI pour Natural User Interface) : on utilise le toucher, la voix, le mouvement, etc. pour commander les machines.

Lorsque l'on doit réaliser une IHM, il faut commencer par établir un cahier des charges. Autrement dit on détaille les spécificités du système, les besoins en capteurs et en actionneurs pour réaliser les

tâches souhaitées, les conditions nécessaires au bon fonctionnement, les moyens d'interagir avec la système, et la description des tâches que la machine doit effectuer.

Dans certains cas, on peut tester des systèmes, comme par exemple des robots, par émulateurs. Autrement dit on teste le fonctionnement dans un environnement numérique qui simule une réalité physique dans lequel on place une représentation du robot que l'on laisse agir comme il le ferait en vrai, selon les instructions codées par le programmeur.