

# Systèmes d'exploitation des ordinateurs

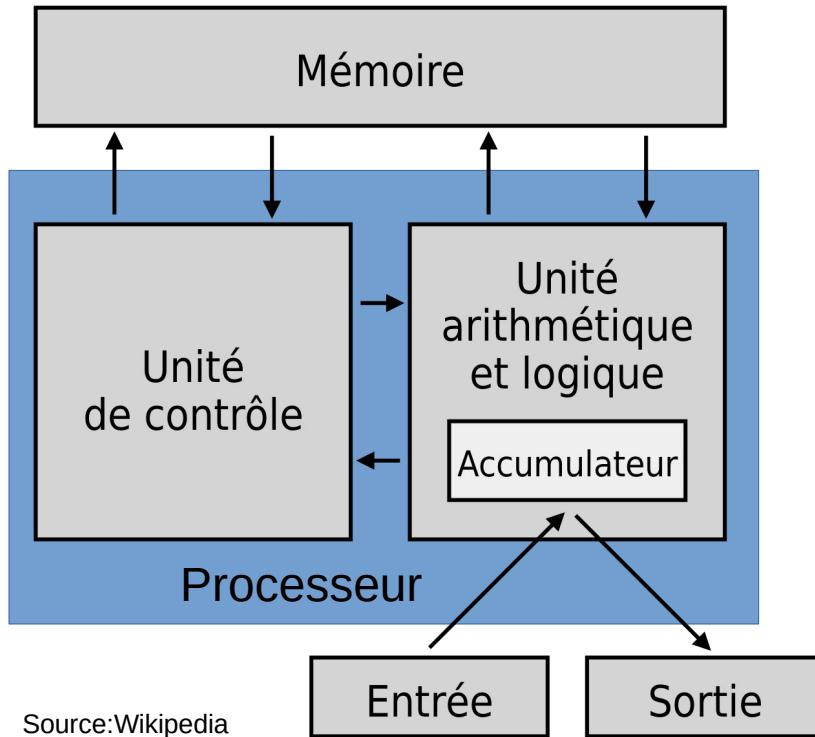
Licence MIASHS

Jérôme David  
2023-2024

# C'est quoi un ordinateur

- Système de traitement d'information programmable
- Fonctionnement:
  - Lit **séquentiellement** des **instructions** qui lui font exécuter des opérations logiques et arithmétiques
  - Ensemble d'instructions : un **programme**

# L'architecture de von Neumann



- **UAL** : traitement de base (opérations, lectures, écritures)
- **UC** : séquencement des opérations, décode les instructions
- **Mémoire** : pour stocker à la fois le programme et les données.
- **Entrées/Sorties** : pour communiquer avec l'extérieur (écran, clavier, imprimante, etc.)
- Les flèches représentent des circuits de communications appelées **bus**

Lecture :<https://interstices.info/le-modele-darchitecture-de-von-neumann/>

# La mémoire

- Il existe plusieurs types de mémoire dans un ordinateur
  - Volatile : registres processeurs, la mémoire cache, la mémoire vive (RAM)
  - Persistante ou mémoire de masse : disques dur, SSD, clés USB, etc.
- On pourrait la représenter comme une commode où les tiroirs :
  - contiennent des données (des 0/1)
  - sont numérotés (leur adresse)



Source:Wikipedia

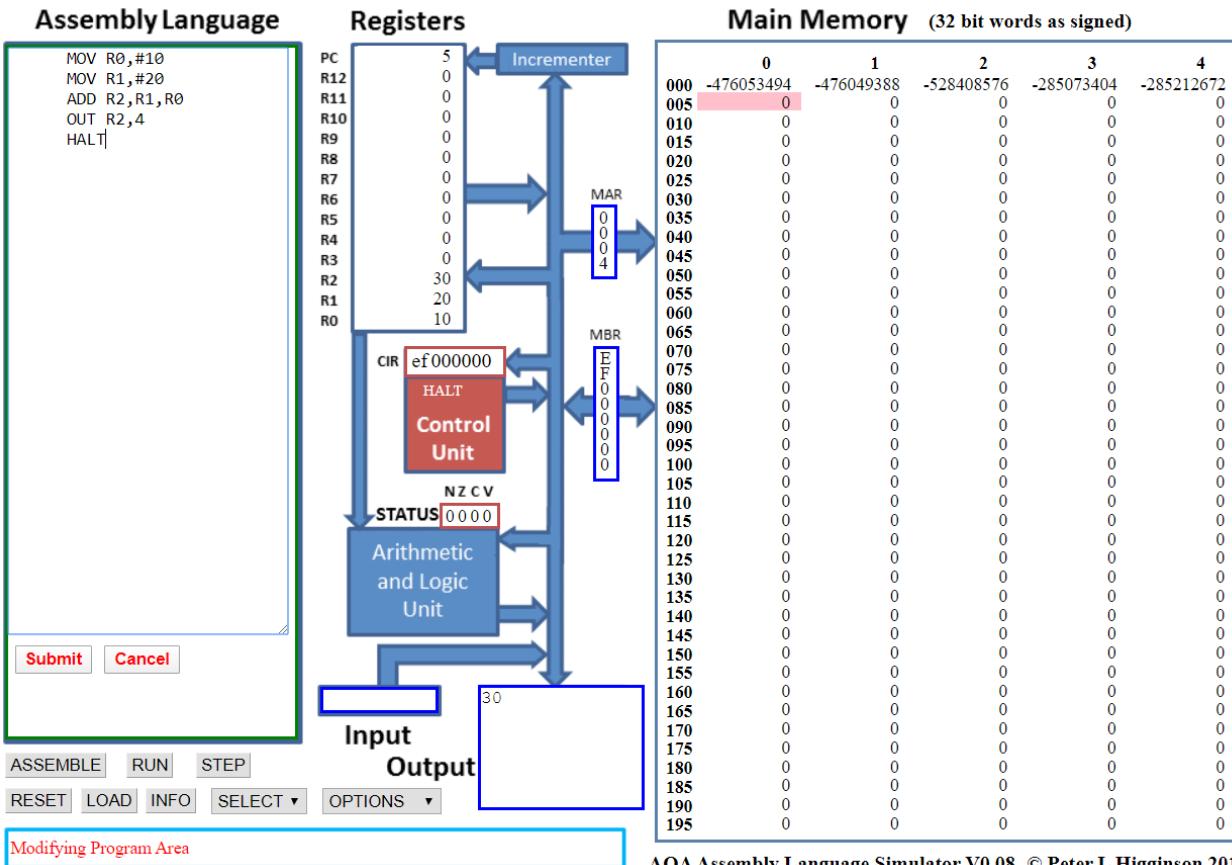


# Le microprocesseur

- CPU (Central Process Unit) est un composant électronique contenant des millions de transistors
- Il est composé (pour simplifier) :
  - D'une horloge : le métronome qui donne le rythme
  - De l'unité de contrôle : le chef d'orchestre
  - De l'unité arithmétique et logique : pour faire les calculs
  - De registres : mémoires spéciales
    - Accumulateur (AC) : pour stocker les résultats de l'UAL
    - Program Counter (PC) : l'adresse de la prochaine instruction à exécuter
    - Current Instruction Register : l'instruction en cours d'exécution
    - Etc.

# Simulation de fonctionnement

- <http://www.peterhigginson.co.uk/AQA/>



Testez ce programme :

```
MOV R0,#10
MOV R1,#20
ADD R2,R1,R0
OUT R2,4
HALT
```

Que fait-il ?

Modifiez le pour stocker le résultat dans la mémoire juste après le programme :

→ instruction :  
STR Rx,adresse\_mémoire

# Les interruptions

- Mécanisme permettant d'interrompre le processeur
  - Utilisé par les périphériques pour prévenir que l'action est terminée (une lecture sur disque par exemple)
  - Utilisé pour annoncer des erreurs de matériel
  - Utilisé par les systèmes pour « partager » le processeur entre plusieurs tâches.

# Langage de programmation

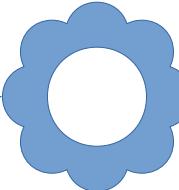
- Langage machine & assembleur :
  - Assembleur : c'est une version lisible du langage de la machine
  - Ne contient que les instructions du processeur
  - Très bas niveau
- Les langages haut-niveau
  - Plus proche de langue naturelle
  - Plus proche de la démarche de résolution de problème que du fonctionnement de la machine
  - C'est donc beaucoup plus facile de résoudre des problèmes complexes avec ce type de langage
  - Il existe plusieurs centaines de langages haut niveau !
- Comment passer d'un langage haut niveau vers le langage machine ?

# La compilation

- Code source
  - Texte écrit dans un langage de haut niveau
  - On dit aussi « programme »
- Compilateur
  - Programme qui transforme un code source en code objet (i.e. en langage machine)
- Les étapes de compilation
  - 1) Analyse lexicale : découper le texte en une suite de « mots » (i.e. reconnaître les mots)
  - 2) Analyse syntaxique : Vérifie que la liste de mots est grammaticalement correcte vis à vis du langage de programmation (i.e reconnaître les phrases bien formées)
  - 3) Analyse sémantique : vérification des types, des portées, construction de la table des symboles, etc.
  - 4) Génération du code machine exécutable

```
#include <stdio.h>
int main() {
    printf("Bonjour DCISS\n");
    return 0;
}
```

Compilation



# Interprétation

- Interpréteur
  - Programme qui analyse un code écrit dans un langage de haut niveau et l'exécute
  - C'est quasiment pareil que la compilation sauf que la génération d'un code machine est remplacé par l'exécution du code
- Langages compilés : C, C++, Java
- Langages interprétés : Python, JavaScript
- Les interpréteurs sont aussi utilisés par les langages de commande
  - Des langages utiliser pour dialoguer avec l'ordinateur

# Les composants d'un PC

- Une alimentation électrique
  - Chaque composant a des besoins de tension différente, et donc il faut plein de fils...
- La carte mère
  - Composant essentiel où tous les autres composants sont « branchés »
  - Elle intègre des composants comme ceux pour le son, le réseau (wifi, ethernet), graphique (affichage), les ports usb, etc.
- Le processeur
  - On a vu, c'est lui qui calcule, lit et écrit dans la mémoire.
- La mémoire vive ou RAM
  - Mémoire de travail pour les applications ouvertes. Elle est rapide mais volatile, lorsque l'on éteint l'ordinateur, tout est perdu.
- La mémoire de masse : permanente pour sauvegarder les données (fichiers, etc)
  - Disque dur : mémoire magnétique.
  - SSD : mémoire « flash » (des puces électronique), beaucoup plus rapide qu'un disque dur
  - Les autres : clés usb, carte mémoire, etc.
- La carte graphique
  - Elle s'occupe de l'affichage et des calculs vectoriel pour les jeux/3D.
  - Peut être intégrée au processeur parfois.
  - Si elle est indépendante, elle possède sa propre mémoire « vive »

# Les microprocesseurs PC

- Ils sont caractérisés :
  - Par leur fréquence d'horloge : 1Ghz = 1 milliard de cycles par seconde
  - Leur nombre de coeurs : aujourd'hui il y a souvent 4 ou plus de processeurs dans une seule puce
  - Leur mémoire cache : c'est une mémoire interne au processeur très rapide. Plus il en a meilleur c'est : au moins 10MB (i.e. une capacité de stockage de  $10 * 8$  millions de 0 ou de 1).
  - Leur architecture/jeu d'instruction : aujourd'hui x64 sur PC (ARM sur smartphone)
  - Leur marque et modèle
- Deux marques sur le PC :
  - Intel : le principal constructeur
    - Les gammes : Atom, Celeron, Pentium, Core (i3, **i5**, **i7**, i9), Xeon (serveurs)
  - AMD
    - Les gammes : Athlon, **Ryzen**, Epyc (serveurs)
    - Fabrique aussi des processeurs graphique au de gamme

# La mémoire vive (RAM)

- Mémoire de travail à accès direct et rapide
- RAM = Random Access Memory
- Caractéristiques :
  - Capacité : 8Go et 32Go
  - Type d'aujourd'hui : DDR-SDRAM (DDR4 et bientôt DDR5)
  - 2 format de barrettes:
    - DIMM (pc classiques)
    - SO-DIMM (portables)
    - On peut connecter plusieurs barrettes sur une carte mère (entre 2 et 8)
  - La latence : CAS (au alentour de 10ns)
  - La fréquence de fonctionnement : 200Mhz à 400MHz
  - Le débit (fonction de la fréquence de la mémoire et de son type) : 1600 à 3200 MT/s (nb transferts/seconde),



Barrette DDR format DIMM

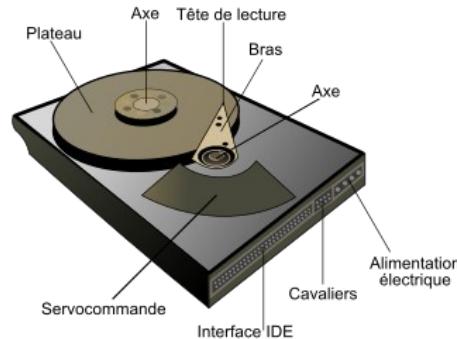
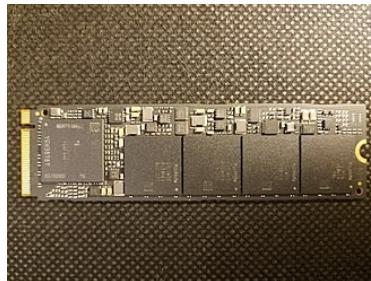


Barrette DDR format SO-DIMM

Source des images : Wikipedia

# Les mémoires de masse

- Mémoires non volatiles, de grande capacités
- Deux technos principales :
  - disque du magnétique
  - SSD (Solid State Drive)
- Caractéristiques
  - Capacité de stockage : aujourd'hui se compte en To
  - Temps d'accès aux informations
    - 5 millisecondes pour disque dur
    - 0,05 milliseconde pour SSD
  - Débit de transfert des informations
  - La taille / format physique : 2.5", 3.5", M.2 (SSD)
  - Coût initial d'acquisition ; coût par information ; coût d'usage
    - Les SSD coûtent beaucoup plus cher que les disques dur
  - Durée de vie des informations stockées
    - Avantage du disque dur jusqu'à présent.



Source des images : Wikipedia

# Exercice

- Lister l'ensemble des caractéristiques de votre ordinateur :
  - Processeur : marque, modèle, fréquence, nb de coeurs, taille du cache
  - Quantité de RAM, nombre de barrettes, type, fréquence ou taux de transfert
  - Mémoire de masse : SSD ou/et disque dur, taille, marque, modèle
- Constituer sur un site de matériel informatique votre PC de bureau pour moins de 1000 euros à partir de pièce détachées

# Les logiciels de base sur un PC

- Le BIOS (Basic Input Output System)
  - Ensemble de programmes permettant d'effectuer des opérations de base (initialisations) à de l'allumage de l'ordinateur
  - Stocker dans une mémoire flash sur la carte mère du PC
  - Permet ensuite l'amorçage du Système d'Exploitation
- Le système d'exploitation (Operating System, OS)

# Système d'exploitation

- Définition :
  - C'est ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs
  - Abréviation : OS (Operating System)
- Objectifs :
  - Faciliter la programmation et l'utilisation de la machine en fournissant des services permettant d'abstraire le matériel
  - Gérer les ressources matérielles

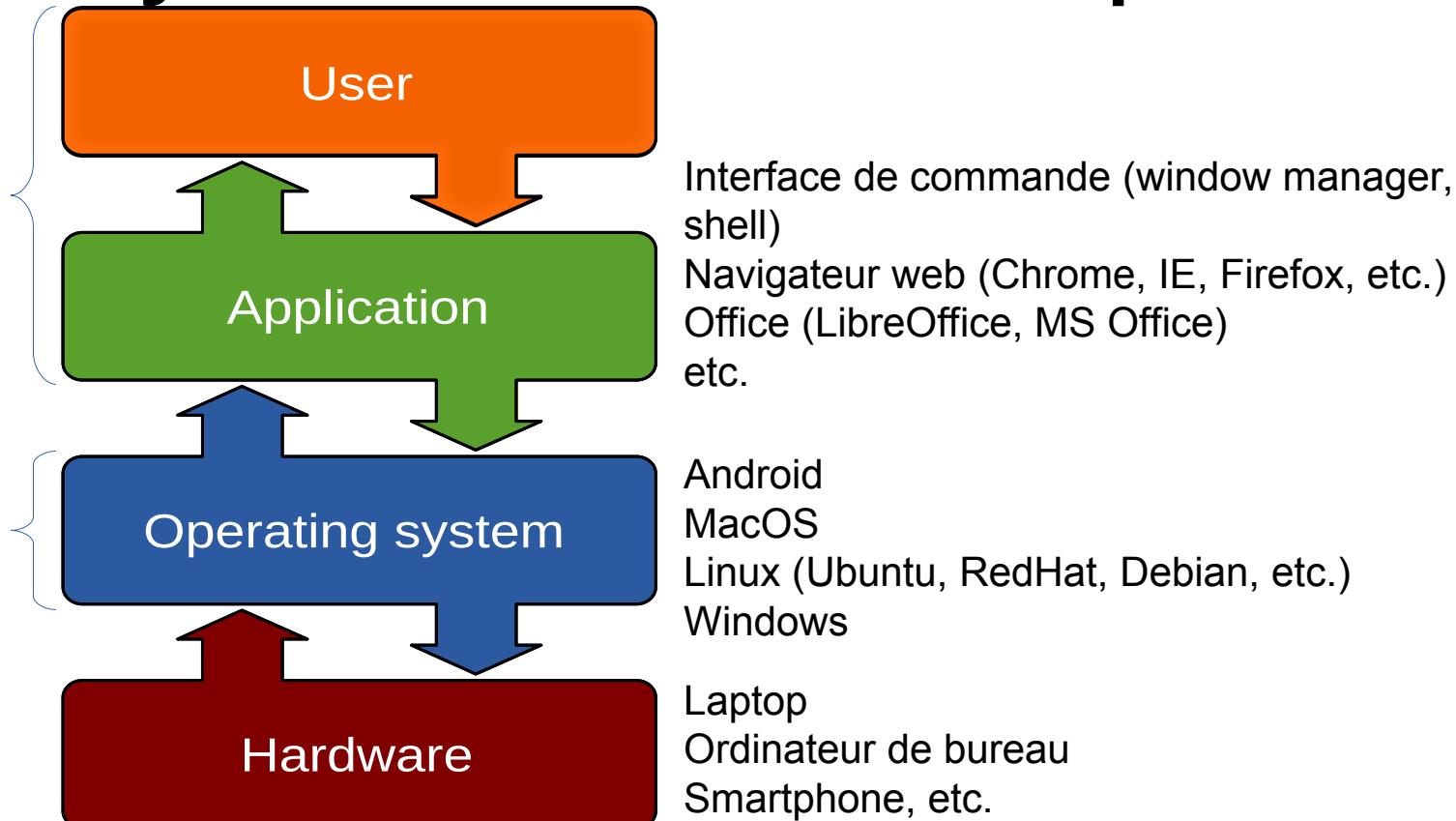
# Les Systèmes d'exploitation

- Quelques OS actuels :
  - Windows : Microsoft, Propriétaire
  - MacOS : Apple, mélange de Propriétaire et Libre
  - GNU/Linux : Libre (GNU GPL)
  - Pour téléphones
    - IOS : Apple, Propriétaire
    - Android : Google , Libre (Apache v2 et GNU GPL v2)
- On distingue souvent deux familles :
  - Les UNIX : MacOS, GNU/Linux, etc.
  - Les windows

# Système informatique

Mode utilisateur  
→ accès à une abstraction du matériel

Mode noyau  
→ accès à tout le matériel



Source :[https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_d%27exploitation#/media/  
File:Operating\\_system\\_placement-fr.svg](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation#/media/File:Operating_system_placement-fr.svg)

# Fonctionnalités

- Gestion des périphériques (ou des entrées/sorties)
  - Abstraire la communication vers les périphériques (écran, claviers, webcam, imprimante, réseau) via des pilotes
- Gestion des données
  - Fournir une vue d'accès aux données sous forme d'un système de fichiers
  - Permettre de contrôler l'accès aux fichiers
- Gestion des ressources
  - Gestion et partage du CPU via une stratégie d'ordonnancement (multiplexage dans le temps)
  - Gestion et partage de la mémoire vive (multiplexage dans l'espace)

# Fonctionnalités (suite)

- Gestion de l'exécution des applications (processus)
  - Affecter et partager les ressources entre différentes applications qui s'exécutent
  - Gérer le cycle de vie des application (démarrage, fin, pause, etc.)
- Gestion des utilisateurs
- Fournir une interface de commande et de programmation
  - Permettre à l'utilisateur de contrôler le système via une interface graphique ou en ligne de commande
  - Faciliter la tâche du programmeur en offrant une API (interface de programmation abstraite)

# Utilisations

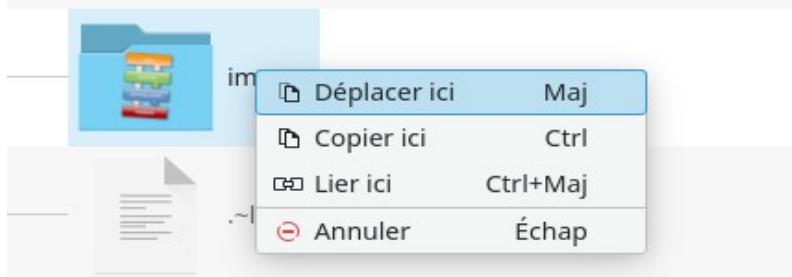
- L'interaction avec l'OS se fait via :
  - Son interface de programmation
    - L'ensemble des appel systèmes (fonctions) que le programmeur peut utiliser
    - c.f. POSIX la norme d'interface de programmation des systèmes UNIX
- L'interpréteur de commande (shell)
  - Ne fait pas partie de l'OS mais permet à l'utilisateur de dialoguer avec l'OS (faire des appel systèmes) en mode “texte”
- L'interface graphique
  - Idem que l'interpréteur de commande mais plus intuitif car en mode graphique (souris)

# Exemples d'utilisation d'interfaces

Avec l'interpréteur de commandes

```
cp chose.txt truc.txt
```

Avec l'interface graphique



En langage C

```
#include <stdio.h>
main() {
    FILE *fp1, *fp2;
    char ch;

    // ouvrir le fichier en lecture
    fp1 = fopen("file1.txt", "r");
    // ouvrir le fichier en écriture
    fp2 = fopen("file2.txt", "w");

    // Lire le contenu du fichier
    while((ch = getc(fp1)) != EOF)
        putc(ch, fp2);

    fclose(fp1);
    fclose(fp2);
    getch();
}
```

# Objectifs et contenu du cours

- Manipuler un OS en ligne de commande
  - Cours sur les commandes Unix et shell Bash
- Appréhender quelques mécanismes de base des OS
  - La notion de processus
    - Mécanismes d'exécution : multiprogrammation, ordonnancement
  - Les problèmes de concurrence d'accès et synchronisation
    - L'exclusion mutuelle
    - La programmation multithread