

# Cours de Systèmes d'Exploitation

Hugues DELALIN

Departement Service et Réseaux de Communication  
IUT de Lens - Université d'Artois

Année 2005/2006 - SRC 1<sup>ère</sup> année

## But de ce cours

Comprendre le fonctionnement système, matériel et logiciel d'un ordinateur dans son contexte de travail.

# Déroulement

- Cours : 3\*2h - Historique, architecture, systèmes d'exploitation
- TD : 3\*2h - Application du cours et pratique
- TP : 6\*2h - Découverte de Linux et du langage de commande C-shell

# Plan

- **Chapitre I : Histoire de l'informatique**
- Chapitre II : Architecture des ordinateurs
- Chapitre III : Systèmes d'exploitation

# Chapitre I : Histoire de l'informatique

- 1 La «Préhistoire» : -40000 à 1945
- 2 Les premiers ordinateurs : 1946 à 1955
- 3 La mini-informatique : 1956 à 1970
- 4 La micro-informatique : 1971 à nos jours

# Plan

- Chapitre I : Histoire de l'informatique
- **Chapitre II : Architecture des ordinateurs**
- Chapitre III : Systèmes d'exploitation

## Chapitre II : Architecture des ordinateurs

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 Matériel
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Plan

- Chapitre I : Histoire de l'informatique
- Chapitre II : Architecture des ordinateurs
- **Chapitre III : Systèmes d'exploitation**



## Chapitre III : Systèmes d'exploitation

- 7 Définition
- 8 Processus
- 9 Mémoire
- 10 Entrées/Sorties
- 11 Systèmes de fichier

La «Préhistoire» : -40000 à 1945  
Les premiers ordinateurs : 1946 à 1955  
La mini-informatique : 1956 à 1970  
La micro-informatique : 1971 à nos jours

# Chapitre I

## Histoire de l'informatique

La «Préhistoire» : -40000 à 1945

Les premiers ordinateurs : 1946 à 1955

La mini-informatique : 1956 à 1970

La micro-informatique : 1971 à nos jours

# Plan

- 1 La «Préhistoire» : -40000 à 1945
- 2 Les premiers ordinateurs : 1946 à 1955
- 3 La mini-informatique : 1956 à 1970
- 4 La micro-informatique : 1971 à nos jours

## Préhistoire

- 9000 **Calculi** : billes, cônes et bâtonnets Sumériens.
- 3000 Empereur chinois **Fou-Hi** : octogone à trigramme.
- 1100 Invention du **boulier** en Chine.
- 300 Aristote définit ce qu'est la logique.
- 1615 Bâtons de **Neper** (Napier)
- 1623 Wilhelm Schickard invente une **horloge calculante**.
- 1632 La **règle à calcul** par Oughtred.
- 1642 La **Pascaline** est mise au point.
- 1679 L'arithmétique binaire est mise au point par **Leibnitz**.
- 1694 Machine à calculer (multiplications et divisions).
- 1728 **Falcon** : métier à tisser par cartes perforées.
- 1820 Charles-Xavier Thomas de Colmar : l'**arithmomètre**.

# Préhistoire

- 1833 **Babbage** : concepts de ce que sera un ordinateur.
- 1840 **Ada Lovelace** processus logique d'exécution = algorithme.
- 1854 **Boole** : processus logique = suite opérations logiques 2 états
- 1884 Tabulatrice à cartes perforées pour recensement (Hollerith).
- 1886 **Felt** : Comptometer (touches).
- 1889 **Léon Bollée** : machine à multiplication directe.
- 1892 **Burroughs** : machines à calculer de bureau.
- 1896 **Hollerith** : Tabulating Machine Corporation.

La «Préhistoire» : -40000 à 1945  
Les premiers ordinateurs : 1946 à 1955  
La mini-informatique : 1956 à 1970  
La micro-informatique : 1971 à nos jours

## Préhistoire

- 1904 Invention du 1<sup>er</sup> tube à vide, la diode, par **Flemming**.
- 1907 Invention de la **triode** par Lee De Forest.
- 1919 Circuit bi-stable, basculeur d'**Eccles** et **Jordan**.
- 1924 Tabulating Machine Corporation devient **IBM**.
- 1935 **IBM 601**, une multiplication par seconde.
- 1937 **Machines de Turing**.
- 1938 Thèse de **Shannon**, définition du chiffre binaire : le bit.
- 1939 **Atanasoff** et **Berry** : additionneur 16 bits par tubes à vide.
- 1940 Calculateurs pour déchiffrer les messages allemands (**Turing**).
- 1941 Calculateur à algèbre de Boole : ABC (**Atanasoff Berry**).
- 1941 **Zuse** conçoit le Z3, 1<sup>er</sup> ordinateur (programme en mémoire).
- 1943 **ASCC Mark 1** par Howard Aiken (électro-mécanique).
- 1945 Un insecte bloque Mark 1 : **Bug**.
- 1945 **John Von Neuman** décrit l'EDVAC (architecture ordinateurs)

# Plan

- 1 La «Préhistoire» : -40000 à 1945
- 2 Les premiers ordinateurs : 1946 à 1955
- 3 La mini-informatique : 1956 à 1970
- 4 La micro-informatique : 1971 à nos jours

## Premiers Ordinateurs

1946 **ENIAC** (19000 tubes, 30 tonnes, 72m<sup>2</sup>, 140kW)

- plus rapide
- même équipe conçoit, programme et administre la machine
- pas de langage de programmation

1950 Invention de l'**assembleur** par M. V. Wilkes.

1951 Mémoire de masse de 1 Mbits : **ERA 1101**.

1951 Compilateur **A0** par G. M. Hopper.

1953 Mémoire à tores de ferrite.

1955 **IBM 704** dédié aux calculs scientifiques (5 kFLOPS).



# Plan

- 1 La «Préhistoire» : -40000 à 1945
- 2 Les premiers ordinateurs : 1946 à 1955
- 3 La mini-informatique : 1956 à 1970**
- 4 La micro-informatique : 1971 à nos jours

## Mini-informatique

1956 **TRADIC** : 1<sup>er</sup> ordinateur à transistors.

- seconde génération d'ordinateurs.

1956 IBM : 1<sup>er</sup> disque dur, le **RAMDAC 305**.

1957 **TX0** : 18 bits, 3500 transistors, 83000 instr./s, 1000 Watts, mémoire de 65536 mots, clavier, écran graphique...

1957 Langage de programmation universel : **FORTRAN** par Backus.

1958 Démonstration du 1<sup>er</sup> circuit intégré par **Texas Instrument**.

1959 **ATLAS I** : mémoire virtuelle et multiprogramation.

1961 **Fairchild Semiconductors** : série de circuits intégrés.

## Mini-informatique

1964 **IBM 360** : circuits intégrés.

- troisième génération d'ordinateurs.
- lignée d'ordinateurs compatibles entre eux.
- Système d'exploitation : OS/360.

1964 le **BASIC** par Kurtz et Kemeny.

1964 Création du code **ASCII** normalisé en 1966 par l'ISO.

1964 **CDC 6600** par Seymour Cray (3 MIPS).

1964 Début du projet **MULTICS**, multi-utilisateurs.

1965 **Illiac IV**, ordinateur à architecture vectorielle (200MIPS).

1969 Mise au point d'**UNIX** sur Dec PDP 7.

1970 Développement du langage B pour UNIX.

1970 Première puce mémoire développée par Intel.

La «Préhistoire» : -40000 à 1945  
Les premiers ordinateurs : 1946 à 1955  
La mini-informatique : 1956 à 1970  
La micro-informatique : 1971 à nos jours

# Plan

- 1 La «Préhistoire» : -40000 à 1945
- 2 Les premiers ordinateurs : 1946 à 1955
- 3 La mini-informatique : 1956 à 1970
- 4 La micro-informatique : 1971 à nos jours**

## Micro-informatique

- 1971 Intel vend le micro-ordinateur **MCS-4** basé sur le 4004.
- 1971 Ritchie reprend le langage B -> langage C sur PDP/11.
- 1972 microprocesseur 8 bits par Intel : **8008**.
- 1972 Fondation d'ATARI par Dabney, borne de jeu d'arcade.
- 1972 Création d'un langage orienté objet : **SmallTalk**.
- 1973 Premier système d'exploitation pour micros : **CP/M**.
- 1975 Compagnie de B. Gates et P. Allen devient **Micro-Soft**.
- 1976 Création d'**Apple** par Jobs et Wozniak (programme moniteur).
- 1976 Microprocesseur 16 bits par Texas Instrument.
- 1977 **TRS 80** de Tandy (Z80, 4 Ko de ROM et RAM, clavier).
- 1977 Atari présente la console de jeux **VCS** (Atari 2600).
- 1978 **Intel 8086** : 16 bits, 4,77MHz, 1Mo de RAM.
- 1979 Taito lance le célèbre jeu **Space Invaders**.

## Micro-informatique

- 1979 **Apple ][ Plus** 48 ko de RAM pour 1195\$.
- 1981 Commodore présente le **VIC 20**, processeur 6502A.
- simple à utiliser grâce au Basic
  - meilleur marché (300\$)
  - applications disponibles en grand nombre
- 1981 Sinclair annonce le ZX 81 (200\$).
- 1981 IBM lance son **5150 Personal Computer**.
- 1982 **Intel 80286**, 16 bits, 6 MHz, 134000 transistors, 0.9 MIPS.
- 1982 Microsoft commercialise MS/DOS 1.1 pour IBM/PC.
- 1982 Sortie du **Commodore 64**.
- 1982 **Compaq Portable PC**, compatible PC d'IBM.
- 1982 Thomson commercialise le **TO 7**.

La «Préhistoire» : -40000 à 1945  
Les premiers ordinateurs : 1946 à 1955  
La mini-informatique : 1956 à 1970  
La micro-informatique : 1971 à nos jours

## Micro-informatique

- 1983 **IBM PC XT** avec MS/DOS 2.0.
- 1983 Développement du **C++** par Stroustrup.
- 1984 **Apple Macintosh** : interface graphique pilotée à la souris.
- 1984 Microsoft présente **Interface Manager** (qui deviendra Windows).
- 1984 Sortie du **PC AT** d'IBM basé sur le 80286.
- 1984 Amstrad commercialise le **CPC 464**, 8 bits, familial.
- 1984 Thomson quant à lui sort le **MO 5**.
- 1985 Sortie des Atari 130 ST et 520 ST.
- 1985 Sortie de **Microsoft Windows 1.0**.
- 1985 Sortie de l'**Amiga 1000**.
- 1985 Sortie du processeur 32 bits d'Intel : **80386DX**, 16MHz.
- 1985 **Chips & Technologies** : 5 composants (chipset) pour PC AT.
- 1986 Sortie du **Deskpro 386** par Compaq.

## Chapitre II

# Architecture des ordinateurs



# But

Comprendre le fonctionnement d'un ordinateur à bas niveau.

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 Matériel
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Introduction

- Informations de différents types.
- Représentation et manipulations en binaire (bits).
- Codage = changement de représentation.
- Binaire car réalisé à l'aide de bistables (transistors).
- Opérations arithmétiques faciles à exprimer en base 2.

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 Matériel
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Changement de base

- Système décimal pour nombres.
- 10 symboles distincts : les chiffres.
- En base  $b$ , on utilise  $b$  chiffres.
  - En décimal,  $b = 10$ ,  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ;
  - En binaire,  $b = 2$ ,  $\{0, 1\}$ ;
  - En hexadécimal,  $b = 16$ ,  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ ;

# Représentation des nombres entiers

- En base 10, on a par exemple :  
$$1996 = 1 * 10^3 + 9 * 10^2 + 9 * 10^1 + 6 * 10^0$$
- Dans le cas général, en base  $b$ , on a :  
$$a_n a_{n-1} \dots a_1 a_0 = \sum_{i=0}^n a_i * b^i$$
 où  $a_0$  (resp.  $a_n$ ) est le chiffre de poids faible (resp. fort).
- Exemple : En base 2,  $(101)_2 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 5$

# Représentation des nombres fractionnaires

- Nombres à chiffres après la virgule.

- En décimal, on écrit :

$$12.346 = 1 * 10^1 + 2 * 10^0 + 3 * 10^{-1} + 4 * 10^{-2} + 6 * 10^{-3}$$

- En général, en base  $b$  :

$$a_n a_{n-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-p} = \sum_{i=-p}^n a_i * b^i$$

## Passage d'une base quelconque à la base 10

- Il suffit d'écrire le nombre comme suit et d'effectuer les calculs en décimal. Exemple en hexadécimal :  
 $(AB)_{16} = 10 * 16^1 + 11 * 16^0 = 160 + 11 = (171)_{10}$  (en hexadécimal, A=10, B=11, ..., F=15).



## Passage de la base 10 à une base quelconque

- Nombres entiers : On procède par divisions successives par la base. Exemple de  $(44)_{10}$  converti en binaire.

$$44/2 = 22 \text{ reste } 0.$$

$$22/2 = 11 \text{ reste } 0.$$

$$11/2 = 5 \text{ reste } 1.$$

$$5/2 = 2 \text{ reste } 1.$$

$$2/2 = 1 \text{ reste } 0.$$

$$1/2 = 0 \text{ reste } 1.$$

$$(44)_{10} = (101100)_2$$

## Passage de la base 10 à une base quelconque

- Nombres fractionnaires : Pour la partie entière, on procède comme pour les entiers. Pour la partie fractionnaire, on multiplie la partie fractionnaire par la base jusqu'à ce qu'elle soit nulle (ou que l'on ait atteint la précision voulue).

Exemple de  $(44.25)_{10}$  converti en binaire.

$$(44)_{10} = (101100)_2 \quad 0.25 * 2 = 0.50 \Rightarrow a_{-1} = 0$$

$$0.50 * 2 = 1.00 \Rightarrow a_{-2} = 1$$

$$0.00 * 2 = 0.00 \Rightarrow a_{-3} = 0$$

## Cas des bases 2, 8 et 16

- Ces bases correspondent à des puissances de 2 ( $2^1$ ,  $2^3$  et  $2^4$ ).
- Très utilisées en informatique car représentation compacte des configurations binaires.
- base 8 = octal, base 16 = hexadécimal. Exemple :  
 $(10011011)_2 = (9B)_{16}$
- On manipule souvent des nombres formés de 8 bits, les octets. Représentation=2 chiffres hexadécimaux.

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 Matériel
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Opérations de base

- Mêmes méthodes en base quelconque qu'en base 10.
- Une retenue ou report apparaît quand on atteint ou dépasse la valeur  $b$  de la base.

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - **Architecture de Von Neumann**
  - Processeur
  - Mémoire
  - Bus
  - Périphériques E/S

# Architecture de Von Neumann

Architecture de John von Neumann et ses collaborateurs.  
Les ordinateurs sont structurés en unités séparées,  
fonctionnellement différentes :

- l'unité de calcul (Unité Arithmétique et Logique)
- l'unité de contrôle
- la mémoire interne (programme et données)
- les unités d'Entrées / Sorties).



# L'Unité Arithmétique et Logique

3 parties :

- Les registres qui sont des unités de stockage.
- Les circuits de l'UAL qui effectuent les opérations (calcul ou logique).
- Les voies de circulation dans l'UAL (bus de commandes et de données).

## Unité de contrôle

- Chercher dans la mémoire l'instruction suivante d'un programme (pointeur ordinal).
- Décoder et déterminer ce qui doit être fait.
- Envoyer les bonnes commandes à l'UAL, la mémoire et les contrôleurs d'entrée/sortie.
- Mémoriser le résultat.

## Unité de contrôle

- Les instructions exécutées par l'ordinateur sont exprimées en langage machine (code binaire) :
  - op codes, opération (instruction) à effectuer.
  - adresse, endroit en mémoire où doit s'effectuer l'opération.
- Les instructions en langage machine sont organisées avec le op code en premier, suivi des adresses mémoire.

L'ensemble des opérations qu'un processeur peut effectuer est appelé **jeu d'instructions**.

# Memoire

Le programme et les données y sont stockés.

Idéalement :

- Plus rapide que le temps d'exécution d'une instruction.
- Disponible en grande quantité.
- Peu onéreuse.

En pratique, hiérarchie de couches :

- les registres
- la mémoire cache (lignes de cache), différents niveaux de cache.
- la mémoire principale, appelée RAM.
- disques durs (aspect mécanique)

## Unités d'Entrée / Sortie

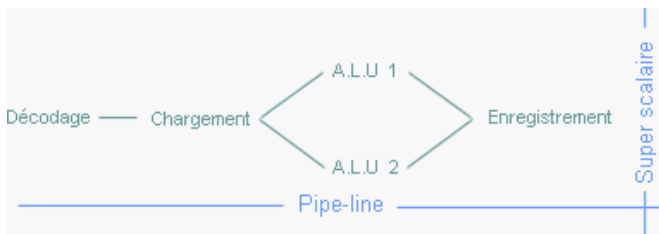
Sous-système qui permet à l'ordinateur d'interagir avec d'autres périphériques et de communiquer avec le monde extérieur.

# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - Architecture de Von Neumann
  - **Processeur**
  - Mémoire
  - Bus
  - Périphériques E/S

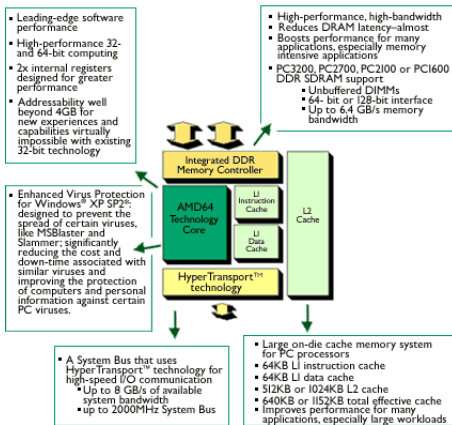
# Processeur

- Brique de base : transistor.
- Regroupe l'UAL et l'unité de contrôle.
- FPU et instructions multimédia
- Architecture :
  - Pipeline
  - Superscalaire



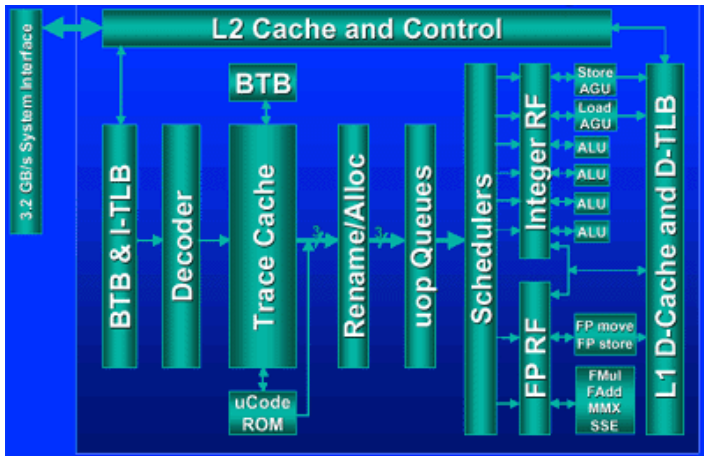
# AMD Athlon 64

## AMD Athlon™ 64 Processor Architecture





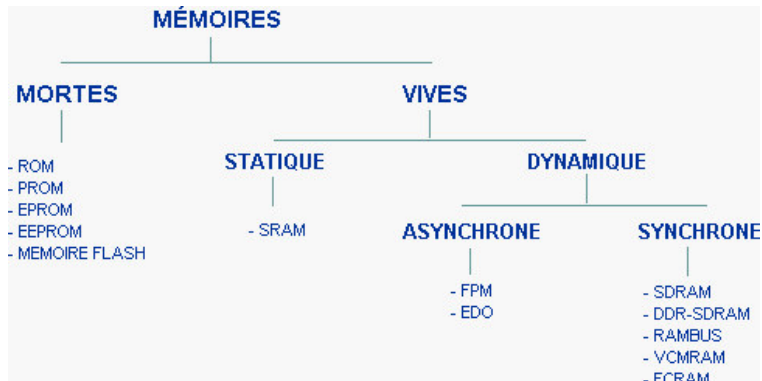
# Intel Pentium 4



# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - Architecture de Von Neumann
  - Processeur
  - **Mémoire**
  - Bus
  - Périphériques E/S

# Mémoires



# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - **Bus**
  - Périphériques E/S

# Bus

- De sa largeur en bits et de sa fréquence dépend sa rapidité de la communication entre les unités de l'ordinateur.
- Un bus connecte l'unité centrale à sa mémoire principale (accès bus) et à la mémoire résidant sur les unités de contrôle des périphériques.
- Un bus permet de transférer des données entre la carte mère et les périphériques qui s'y connectent.

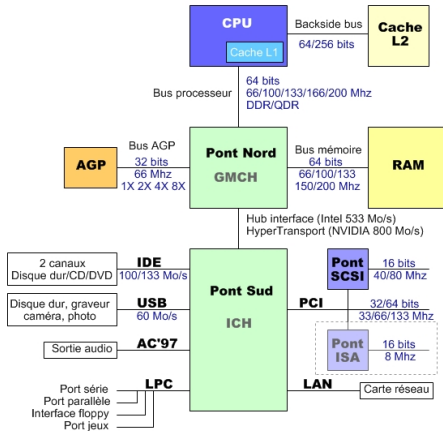
## Bus d'extension

Un bus d'extension permet d'étendre un système PC à l'aide de cartes, en permettant aux données de circuler entre la carte et l'unité centrale.

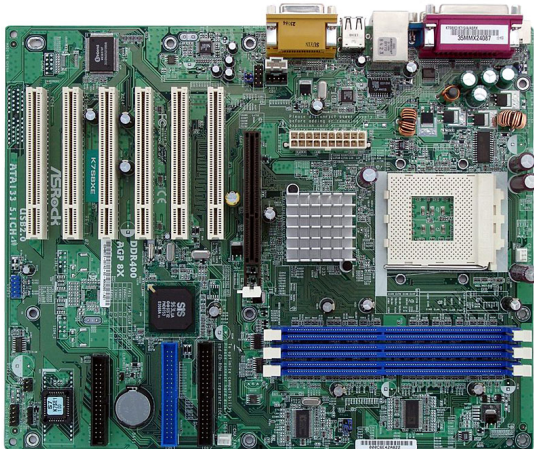
Il existe différents bus d'extension :

- AGP pour les cartes graphiques
- PCI pour les cartes d'extension
- USB
- Firewire
- PCI-Express

# Fonctionnement

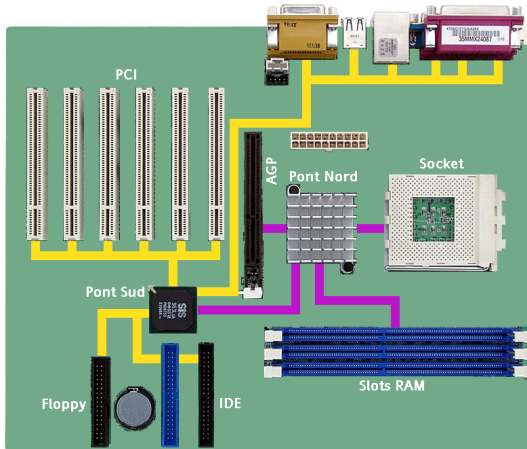


# Concrètement





# Concrètement



# Plan

- 5 Principe de fonctionnement d'un ordinateur
  - Codage de l'information
  - Opérations de base
  
- 6 **Matériel**
  - Architecture de Von Neumann
  - Processeur
  - Mémoire
  - Bus
  - **Périphériques E/S**

## Entrées / Sorties

- Périphériques d'entrée :
  - clavier
  - scanner
- Périphériques de sortie :
  - écran
  - imprimante
- Périphériques d'entrée/sortie :
  - disque dur
  - carte son

## Chapitre III

# Systèmes d'exploitation

# Introduction

Pour qu'un ordinateur soit capable de faire fonctionner un programme informatique (appelé parfois application ou logiciel), la machine doit être en mesure d'effectuer un certain nombre d'opérations préparatoires afin d'assurer les échanges entre le processeur, la mémoire, et les périphériques.

# Plan

- 7 Définition
- 8 Processus
- 9 Mémoire
- 10 Entrées/Sorties
- 11 Systèmes de fichier

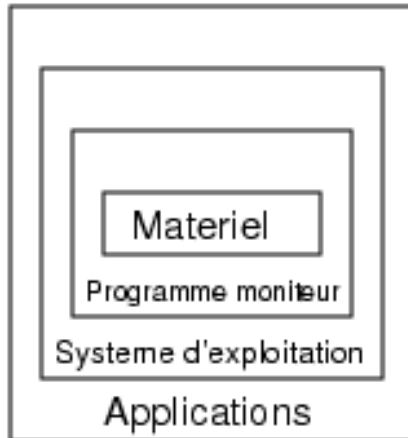
## Définition

Le système d'exploitation (noté SE ou OS, abréviation du terme anglais Operating System), est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications.

Deux tâches :

- Fournir à l'utilisateur une machine étendue ou virtuelle, plus simple à programmer.
- Gestion des ressources. Deux dimensions du partage (multiplexage) :
  - temps
  - espace

# Définition





# Plan

- 7 Définition
- 8 Processus**
- 9 Mémoire
- 10 Entrées/Sorties
- 11 Systèmes de fichier

## Définition

C'est un programme en cours d'exécution. Chaque processus possède :

- un espace d'adressage qui contient :
  - le programme exécutable
  - ses données
  - sa pile
- un ensemble de registres dont :
  - le compteur ordinal
  - le pointeur de pile
- d'autres registres matériels et informations nécessaires.

# Pseudo-parallélisme

- Les ordinateurs sont capables de faire plusieurs choses en même temps.
- Le processeur bascule constamment d'un processus à l'autre : **multiprogrammation**.
- Différence processus / programme

## Création d'un nouveau processus

Evénements conduisant à la création d'un nouveau processus :

- Initialisation du système
- Exécution d'un appel système de création de processus par un processus en cours.
- Requête utilisateur sollicitant la création d'un nouveau processus
- Initiation d'un travail en traitement par lots

## Création d'un nouveau processus

- Sous UNIX :
  - Appel système : **fork** qui crée un clone du processus appelant.
  - Les processus père et fils ont alors la même image mémoire et les mêmes fichiers ouverts.
  - Le processus enfant exécute alors **execve** par exemple pour modifier son image mémoire et exécuter un nouveau programme.
- Sous Windows :
  - Appel à la fonction Win32 **CreateProcess**.
  - Prise en charge de la création du processus et du chargement du programme approprié.
  - Dizaine de paramètres : programme à exécuter, paramètres du programme, attributs de sécurité et bits de contrôle (héritage, priorité, fenêtre), etc.

## Fin d'un processus

- Arrêt normal (volontaire) (exit)
- Arrêt pour erreur (volontaire)
- Arrêt pour erreur fatale (involontaire)
- Le processus est arrêté par un autre processus (involontaire) (kill)

# Hiérarchisation des processus

- Pas le cas sous Windows.
- Sous UNIX :
  - lorsqu'un processus en crée un autre, le père et l'enfant continuent d'être associés.
  - l'enfant peut lui-même créer d'autres processus
  - formation d'une hiérarchie de processus (init)
  - Un processus et l'ensemble de ses descendants est appelé un **groupe** de processus

# Etats

Un processus peut prendre un de ces 3 états :

- En cours d'exécution (le programme utilise le processeur)
- Prêt (exécutable, temporairement arrêté pour laisser un autre processus)
- Bloqué (ne peut pas s'exécuter tant qu'un événement externe ne se produit pas)

Le passage de `En cours` à `Prêt` et inversement est géré par l'**ordonnanceur** de processus



# Système d'exploitation

Ensemble de processus qui :

- exécutent des programmes qui comprennent les commandes saisies par l'utilisateur
- gèrent des tâches telles que le transport de requêtes pour le service de fichiers ou la gestion des détails de l'exécution d'un disque.

## Gestion des interruptions

- **Vecteur d'interruption** stocke l'adresse des routines associées à chaque type d'interruption.

Numéro d'interruption	Gestionnaire
0	Horloge
1	Disque
2	Terminaux
3	Autres périphériques
4	Logiciel (trap)
5	Autres

- Numéros d'interruptions différents pour les périphériques du système (routines différentes)
- Parfois possible de paramétrer les numéros : IRQ.

## Traitement d'une interruption

- Le matériel place dans la pile le compteur ordinal, etc.
- Le matériel charge un nouveau compteur ordinal à partir du vecteur d'interruptions.
- La routine de traitement en langage machine sauvegarde les registres
- Elle définit une nouvelle pile
- Le service d'interruption en C s'exécute
- La procédure C retourne au code en langage machine
- La procédure en assembleur exécute une instruction de retour de procédure.

# Communication inter-processus

- Certains processus ont besoin de coopérer :
  - communication
  - synchronisation (accès concurrent)
- D'autres entrent en compétition pour les ressources :
  - nature physique de la ressource
  - opérations qui peuvent provoquer des incohérences ou des interblocages

Solutions : Sections critiques, masquage des interruptions, ...

## Algorithme d'ordonnement

Choix dépend de l'utilisation, plusieurs critères :

- équité
- efficacité
- minimisation du temps de réponse pour les utilisateurs
- minimisation du temps d'exécution (traitement par lots)
- rendement (nombre de travaux réalisés par unité de temps maximal)

Solutions : tourniquet, priorité, plus court d'abord, ...

# Plan

- 7 Définition
- 8 Processus
- 9 Mémoire**
- 10 Entrées/Sorties
- 11 Systèmes de fichier

# Gestion de la mémoire

- Hiérarchisation de la mémoire (cache, RAM, disque dur).
- Coordination de la manière dont sont utilisées les différentes mémoires.

# Gestionnaire de mémoire

Son rôle :

- conserver la trace de la mémoire en cours d'utilisation ou pas
- allouer la mémoire aux processus qui en ont besoin
- gérer le va-et-vient (swapping) entre mémoire principale et disque.



# Monoprogrammation

Exemple : MS DOS

- Un seul processus en mémoire à la fois.

En pratique :

- Partie de l'espace d'adressage réservée au système d'exploitation (ROM + SE chargé au démarrage)
- A la fin du programme, retour à l'interpréteur de commande qui demande le prochain programme à lancer.

# Multiprogrammation

- Facilite développement de programmes en les fractionnant en processus indépendants.
- Elle permet une maximisation de l'utilisation des ressources processeur.
- Problème : Comment organisé la memoire de façon la plus efficace possible.

## Multiprogrammation avec partitions fixes

- Division de la mémoire en partitions (si possible inégales).
- Quand une tâche arrive, elle est placée dans une file d'attente :
  - une file d'attente par partition de mémoire
  - une seule file d'attente pour toute les partitions
- réallocation
- protection

## Va et vient

- Mémoire insuffisante pour contenir tous les processus courant.
- Nécessité de placer certains de ces processus sur le disque.

Il faudra donc ramener régulièrement des processus sur le disque en mémoire centrale et inversement. C'est ce qu'on appelle le **va-et-vient** ou swapping.

## Mémoire virtuelle

La taille de l'ensemble formé par le programme, les données et la pile peut dépasser la capacité de mémoire disponible.

- Le SE conserve les parties de programme en cours d'utilisation dans la mémoire principale, et le reste sur le disque.

## La technique de la pagination

- A l'intérieur des processeurs actuels, il y a une **unité de gestion de la mémoire** (MMU) qui fait correspondre des adresses virtuels à des adresses physiques.
- Un programme travaille sur un espace d'adressage virtuel.
- Cet espace est divisé en unités appelées **pages** (512 à 4096 octets) et les unités correspondantes en mémoire physique sont appelées **cadres de pages** (page frames).
- La MMU dispose d'une table d'indirection des pages pour convertir les adresses virtuelles en adresses physiques.
- Les pages virtuelles qui ne sont pas mappées en mémoire centrale sont identifiées grâce à un **bit de présence/absence**.
- Les transferts RAM/disque se font par pages entières.

## Pagination

Espace d'adressage virtuel plus grand que la mémoire physique.

Si un processus essaie de faire appel à une page non présente en mémoire physique :

- Déroutement du processeur (**défaut de page**) pour rendre la main au SE.
- Le SE sélectionne un cadre de page peu utilisé et sauve son contenu sur le disque.
- transfère la page demandée dans le cadre de page libéré.
- modifie la correspondance.
- recommence l'instruction déroutée.

Gestion par un algorithme de remplacement de pages.

# Plan

- 7 Définition
- 8 Processus
- 9 Mémoire
- 10 Entrées/Sorties**
- 11 Systèmes de fichier



# Entrées/Sorties

Le SE a la tâche importante de contrôler les périphériques d'entrées/sorties (E/S).

- Fonctions :
  - Emission des commandes vers les périphériques.
  - Interception des interruptions.
  - Gestion des erreurs.
- But :
  - Fournir une interface simple entre les périphériques et le système.
  - Interface identique pour tous les périphériques.

## Les unités d'entrées/sorties

Deux catégories :

- périphériques par bloc : informations stockées par blocs de taille fixe, chacun possédant sa propre adresse. (ex : disque)
- périphériques par caractères : information circule sous la forme d'un flot de caractères, sans aucune structure de bloc. (ex : clavier, imprimante, souris).

Deux parties dans une unité :

- un composant mécanique, le périphérique (ex : disque).
- un composant électronique, le **contrôleur de périphérique** (ex : contrôleur IDE).

## Communication

- Interface entre contrôleur et périphérique de très bas niveau.
- Le contrôleur possède des registres qui permettent la communication avec le processeur.
  - Ecriture dans ces registres : le SE ordonne au périphérique de délivrer des données, d'en accepter ou d'effectuer une action donnée.
  - Lecture : le SE peut connaître l'état du périphérique, savoir s'il est capable d'accepter une nouvelle commande.
- Certains périphériques sont équipés d'un tampon de données que le SE peut lire ou écrire.

## Les E/S mappées en mémoire

Le processeur communique avec les registres de contrôle et les tampons de données, deux cas possibles :

- un numero de **port d'E/S** est assigné à chacun des registres de contrôle et il existe des instructions spéciales pour lire et/ écrire sur ces ports.
- **E/S mappées en mémoire**, chaque registre de contrôle se voit attribuer une adresse mémoire unique à laquelle aucune mémoire n'est assignée.

Dans les ordinateurs de bureau actuels, les E/S sont mappées en mémoire, les adresses entre 640ko et 1Mo sont par exemple marquées comme étant destinées au bus PCI et non à la mémoire. La puce de pontage PCI filtre les adresses.

# Les interruptions

Pour permettre au processeur de réaliser d'autres opérations pendant qu'il attend la réalisation d'une E/S, on fait appel aux **interruptions**.

## L'accès direct à la mémoire (DMA)

- Disponible uniquement s'il y a un contrôleur DMA.
- Le contrôleur DMA a accès au bus système sans dépendre du processeur -> E/S programmée qui fait le travail du processeur.
- Réduit le nombre d'interruptions.

# Les disques magnétiques

- Organisation en cylindres
- Chaque cylindre contient autant de pistes que de têtes empilées verticalement.
- Les pistes sont divisées en secteurs.

On appelle cette organisation **géométrie**.

Sur les disques durs actuels, la géométrie spécifiée peut être différente du format physique réel.

# Plan

- 7 Définition
- 8 Processus
- 9 Mémoire
- 10 Entrées/Sorties
- 11 Systèmes de fichier**



## Stockage à long terme d'informations

- Enregistrement d'une grande quantité d'informations.
- Informations conservées après la fin du processus qui les utilise (**persistance**).
- Plusieurs processus doivent pouvoir avoir accès simultanément à une information.

# Fichiers

- Mécanisme d'abstraction (utilisateur ne voit pas où et comment sont stockées les informations).
- Subdivision des fichiers par types en fonction de leur nature :
  - typage fort : le type de fichier est défini par son **extension** (MS DOS)
  - typage déduit : les extensions des fichiers ne sont qu'indicatives, le système détermine la nature du fichier par inspection du contenu (UNIX).

# Catalogues

Nommés aussi **répertoires** ou **dossiers**. Système à repertoire hiérarchique :

- permet regroupement logique des fichiers
- notion de chemin d'accès :
  - chemin d'accès absolu (depuis la racine)
  - chemin d'accès relatif (depuis le répertoire courant)

## Système de fichier

- Disques divisés en partitions pouvant contenir différents systèmes de fichiers.
- Secteur 0 du disque = Master Boot Record (MBR) qui comprend la table de partitions. Boot sur la partition marquée comme active.
- L'organisation d'une partition varie fortement d'un système de fichier à un autre. Cependant, présence d'un bloc de boot et souvent d'un superbloc qui contient les informations sur le type de système de fichier.
- Différentes méthodes d'implantation des fichiers (allocation contigüe, listes chaînes,...).

# Chapitre IV

## Conclusion

# Conclusion

- Fonctionnement d'un ordinateur et de son système d'exploitation.
- Pas si simple que ça...