Rappels cours
Architecture et Systèmes d'Exploitation

MaDSI 1 - 2020

B. Diop - UGB

Adapté du cours de Hugues DELALIN

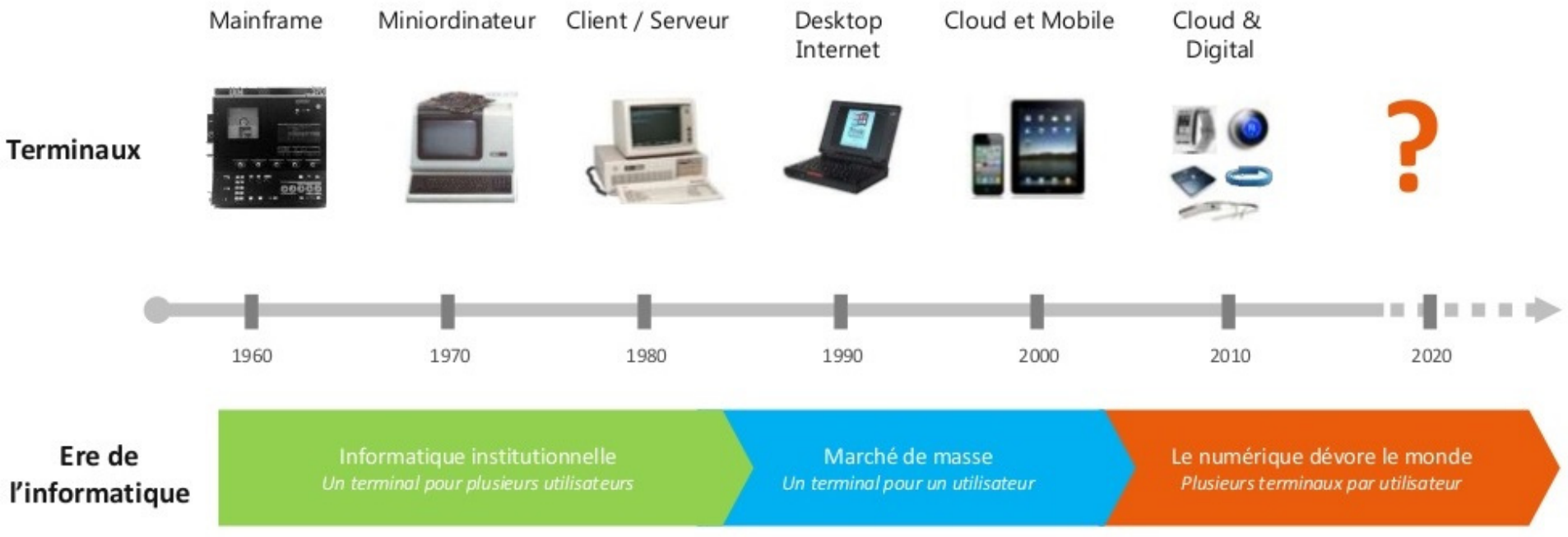
Departement Service et Réseaux de Communication IUT de Lens - Université d'Artois

But de ce cours

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

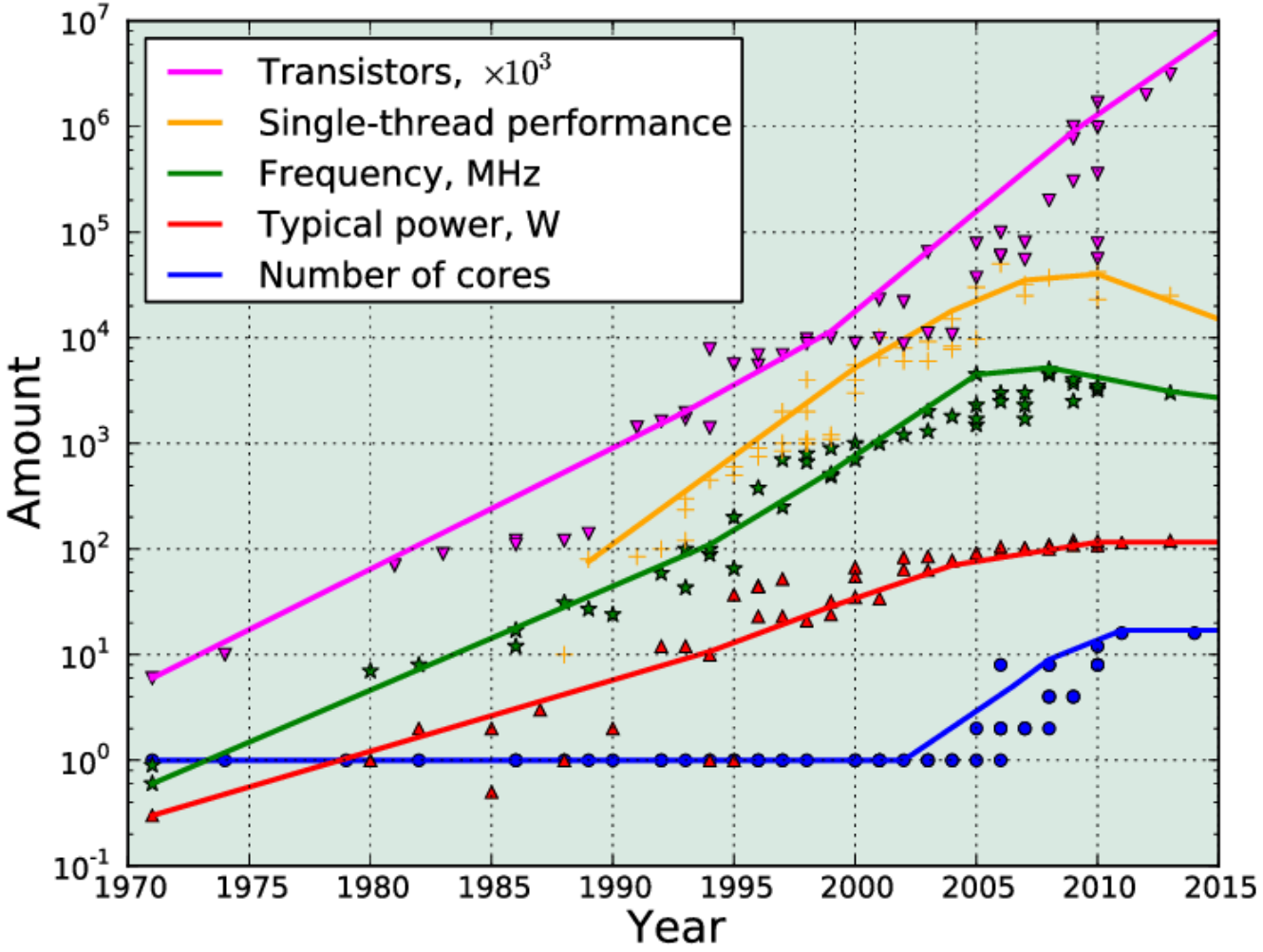
Plan

Chapitre I : évolution des ordinateurs



https://connectemoi.wordpress.com/2013/12/17/histoire/

Chapitre I : évolution des CPU



https://www.researchgate.net/figure/Evolution-of-CPU-characteristics_fig1_315932768

Chapitre II

Architecture des ordinateurs

But

Comprendre le fonctionnement d'un ordinateur à bas niveau.

Plan

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

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 héxadécimal, $b = 16$, $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$;

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 :
  $a_n a_{n-1} \ldots a_1 a_0 = \sum_{i=0}^{n} a_i * b^i$ 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 :
  $a_n a_{n-1} \ldots a_1 a_0 a_{-1} a_{-2} \ldots a_{-p} = \sum_{i=-p}^{n} a_i * b^i$

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.
  $44/2 = 22$ reste 0.
  $22/2 = 11$ reste 0.
  $11/2 = 5$ reste 1.
  $5/2 = 2$ reste 1.
  $2/2 = 1$ reste 0.
  $1/2 = 0$ 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.
  $(44)_{10} = (101100)_2$ $0.25 * 2 = 0.50 => a_{-1} = 0$
  $0.50 * 2 = 1.00 => a_{-2} = 1$
  $0.00 * 2 = 0.00 => a_{-3} = 0$

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 = héxadécimal. Exemple : $(10011011)_2 = (9B)_{16}$
- On manipule souvent des nombres formés de 8 bits, les octets. Représentation=2 chiffres héxadécimaux.

Plan

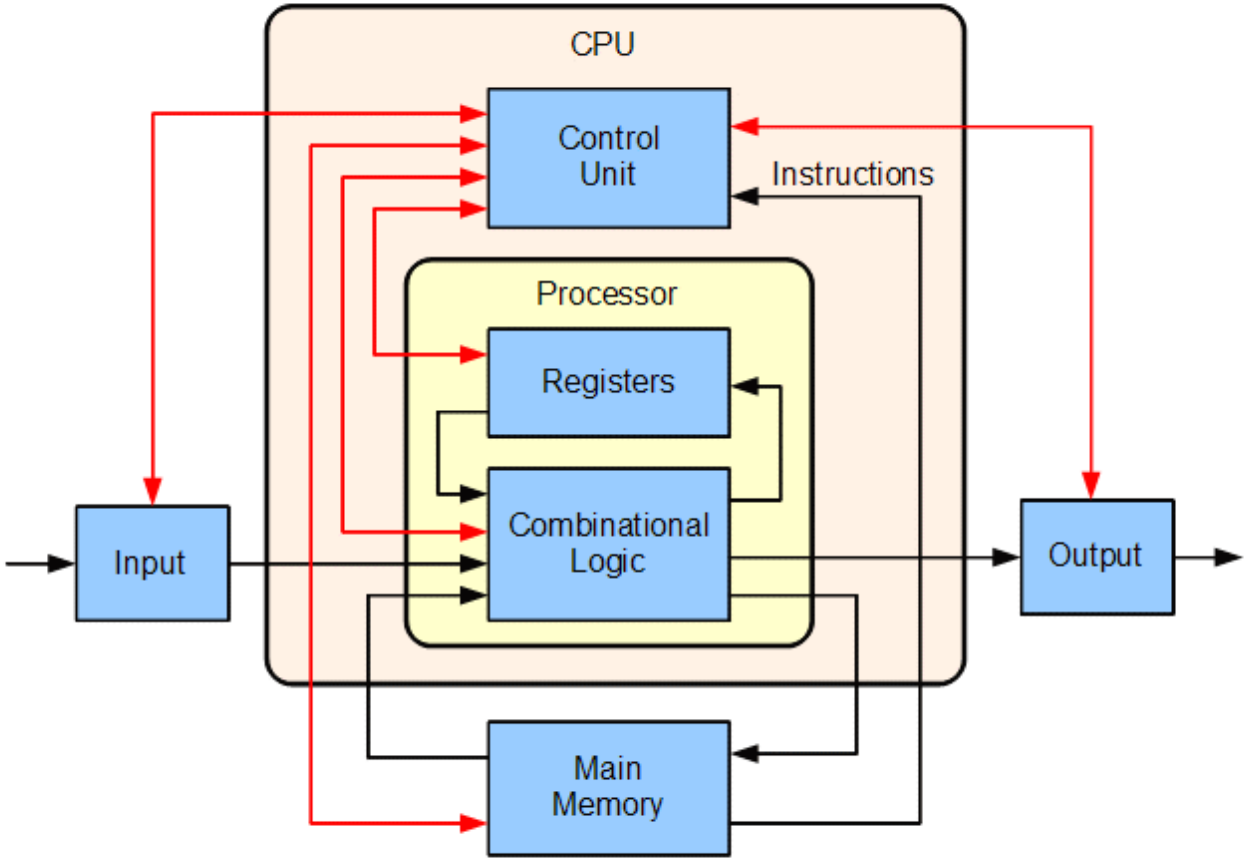
Opérations de base

- 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.

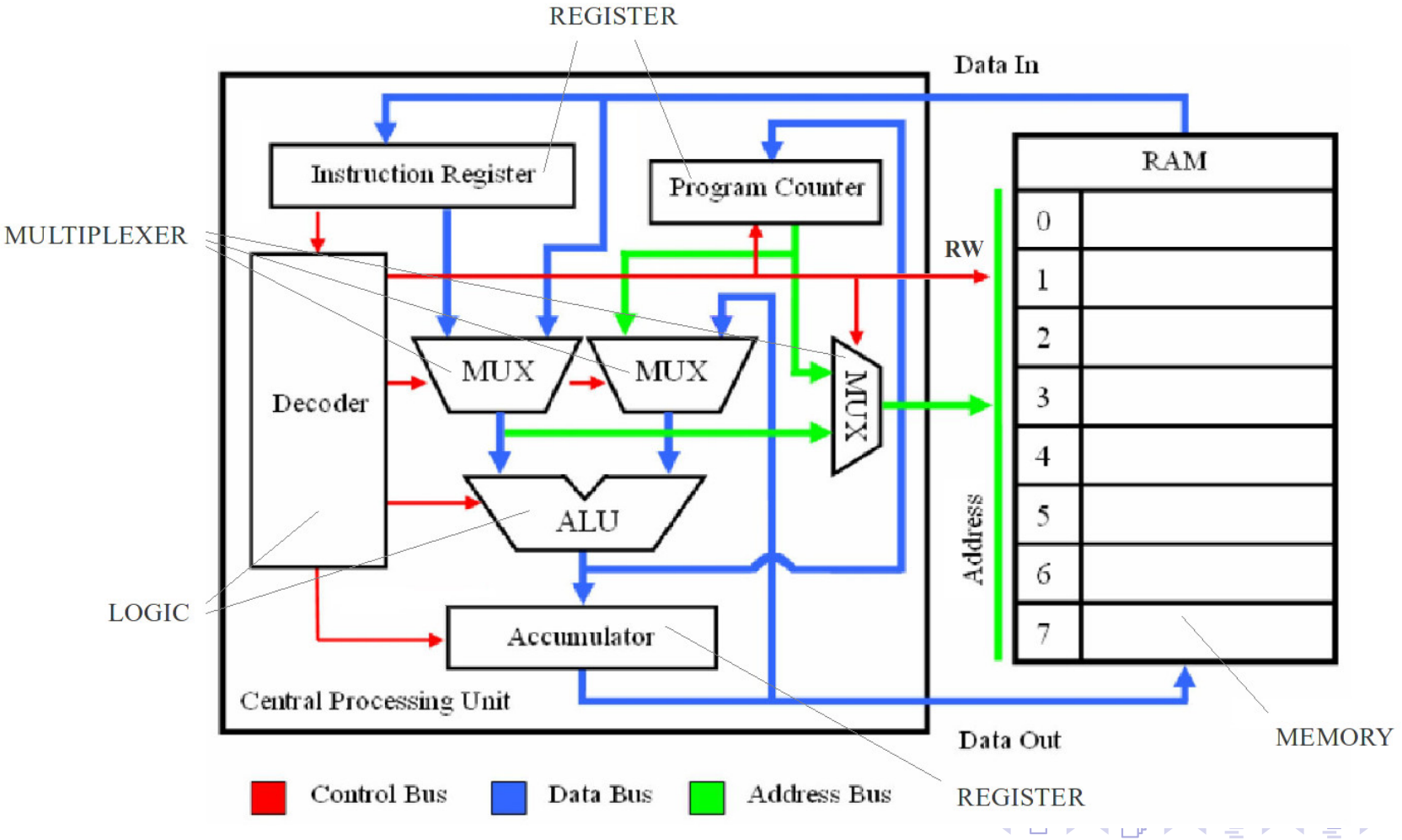
Plan

Plan

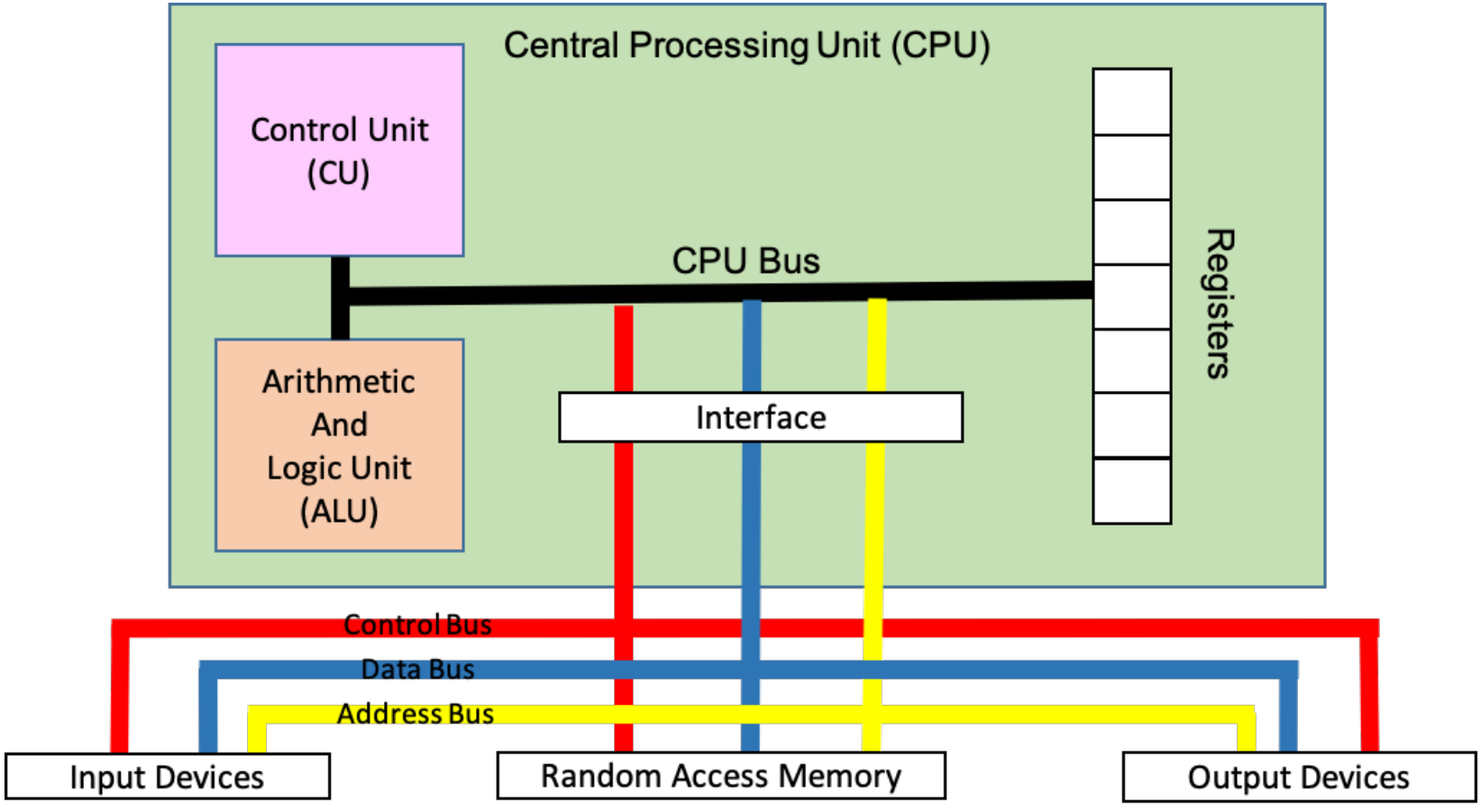
Architecture Von Neumann



Architecture Von Neumann



Architecture Von Neumann



Architecture Von Neumann

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 controleurs 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 ou 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)

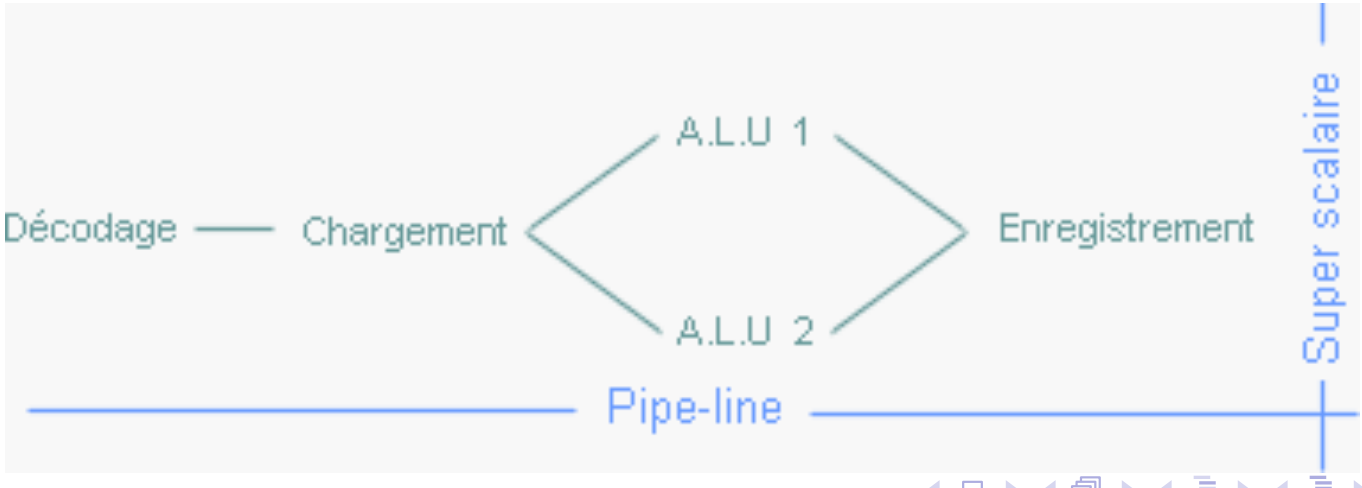
Unités d’Entrée / Sortie

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

Plan

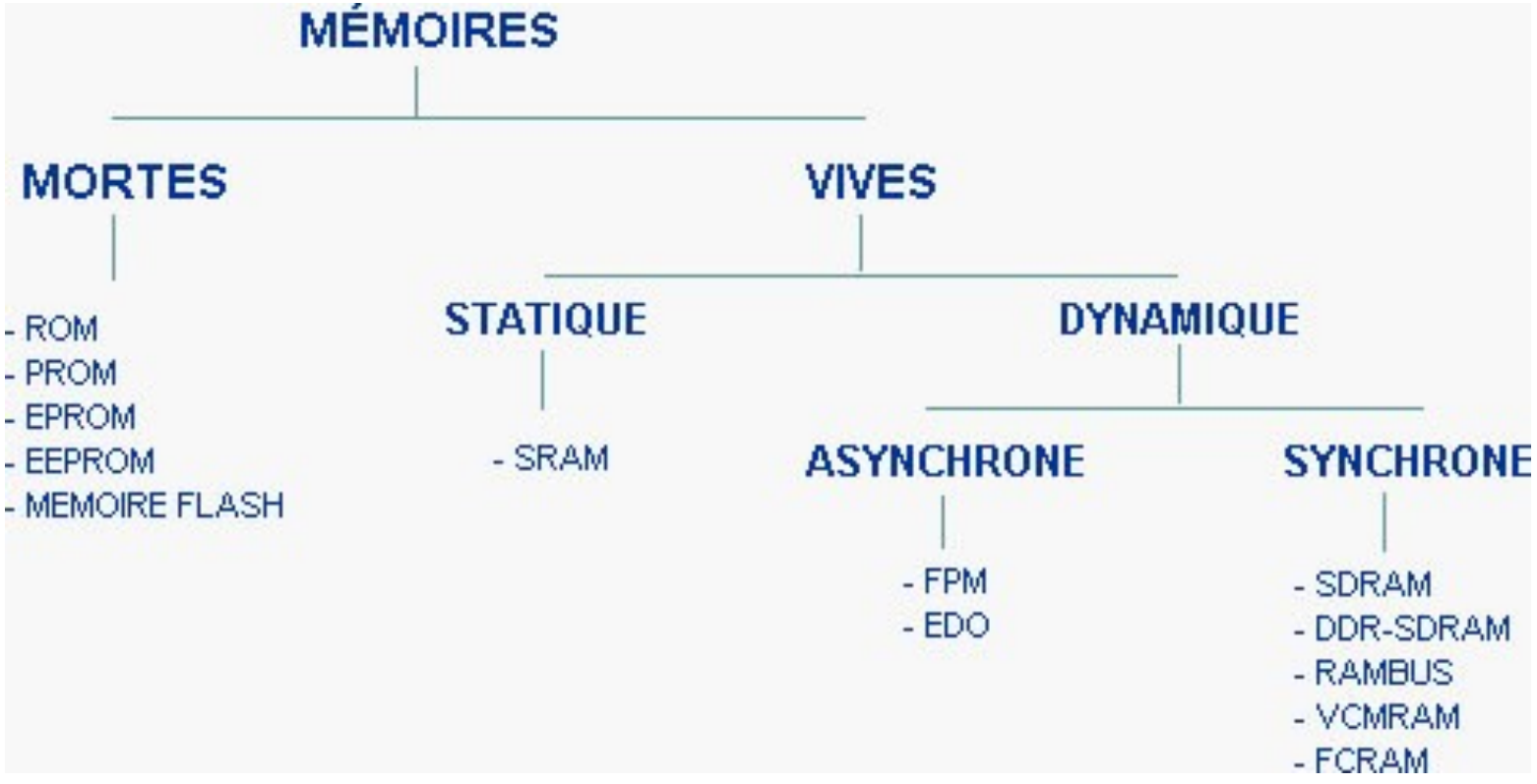
Processeur

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



Plan

Mémoires


MÉMOIRES
MORTES
VIVES
- ROM
- PROM
- EPROM
- EEPROM
- MEMOIRE FLASH
STATIQUE
- SRAM
DYNAMIQUE
ASYNCHRONE
- FPM
- EDO
SYNCHRONE
- SDRAM
- DDR-SDRAM
- RAMBUS
- VCMRAM
- FCRAM

Plan

Bus

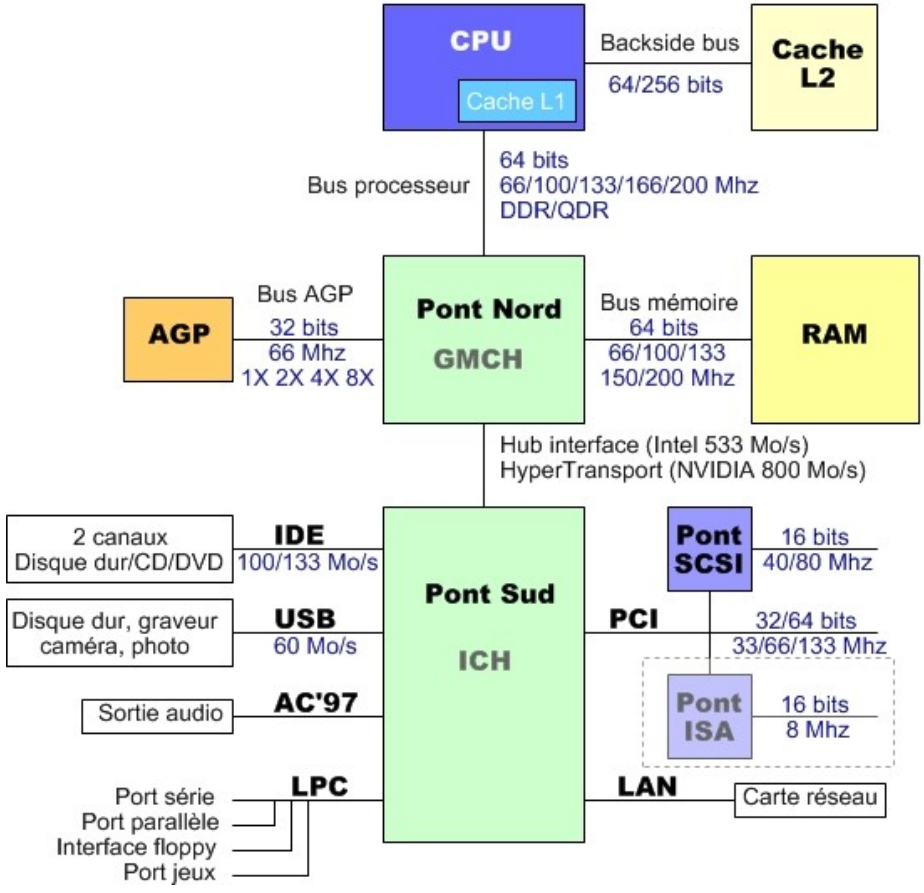
- De sa largeur en bits et de sa fréquence dépend sa rapidité de la communication entre les unités de l'ordinateur.
- Un bus connecte l'unité centrale à sa mémoire principale (accès bus) et à la mémoire résidant sur les unités de contrôle des périphériques.
- Un bus permet de transférer des données entre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

Bus d'extension

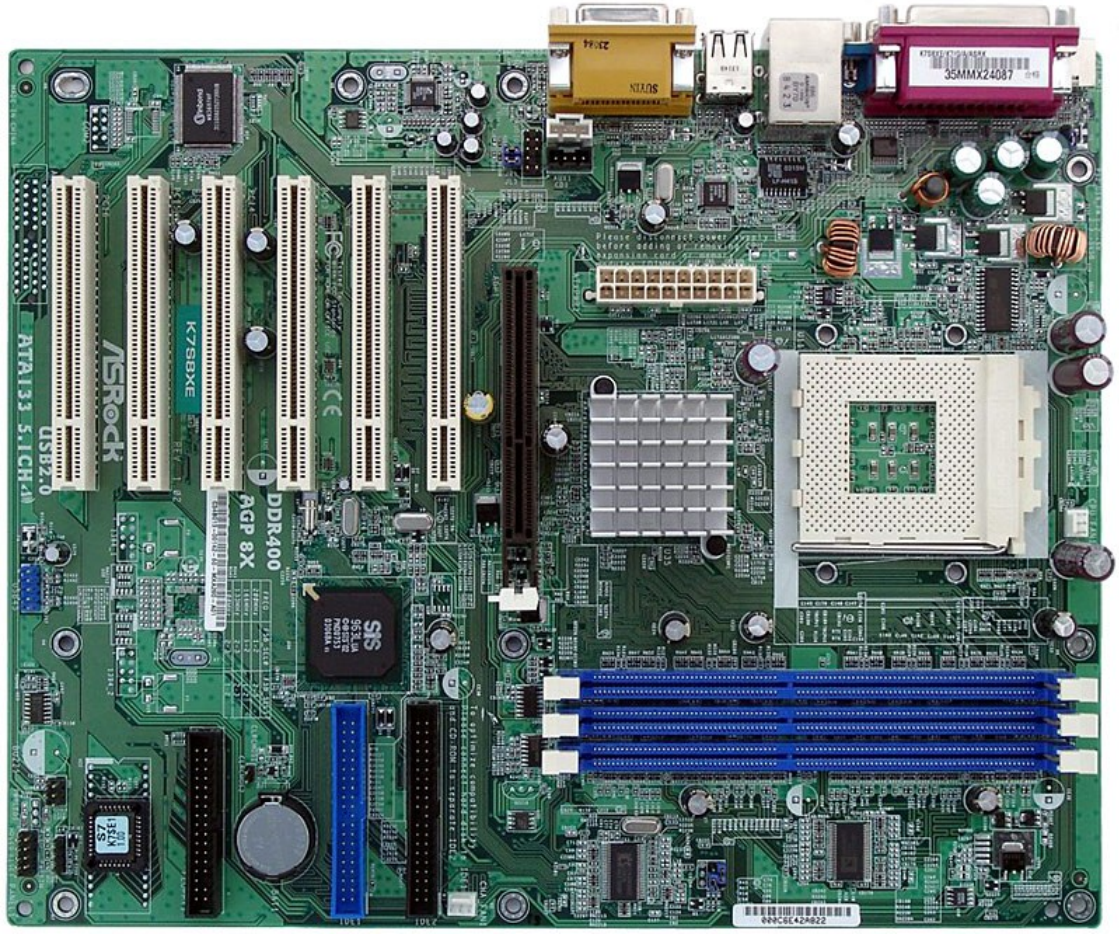
Un bus d'extension permet d'étendre un système PC à l'aide de cartes, en permettant aux données de circuler entre la carte et l'unité centrale.
Il existe différents bus d'extension :

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

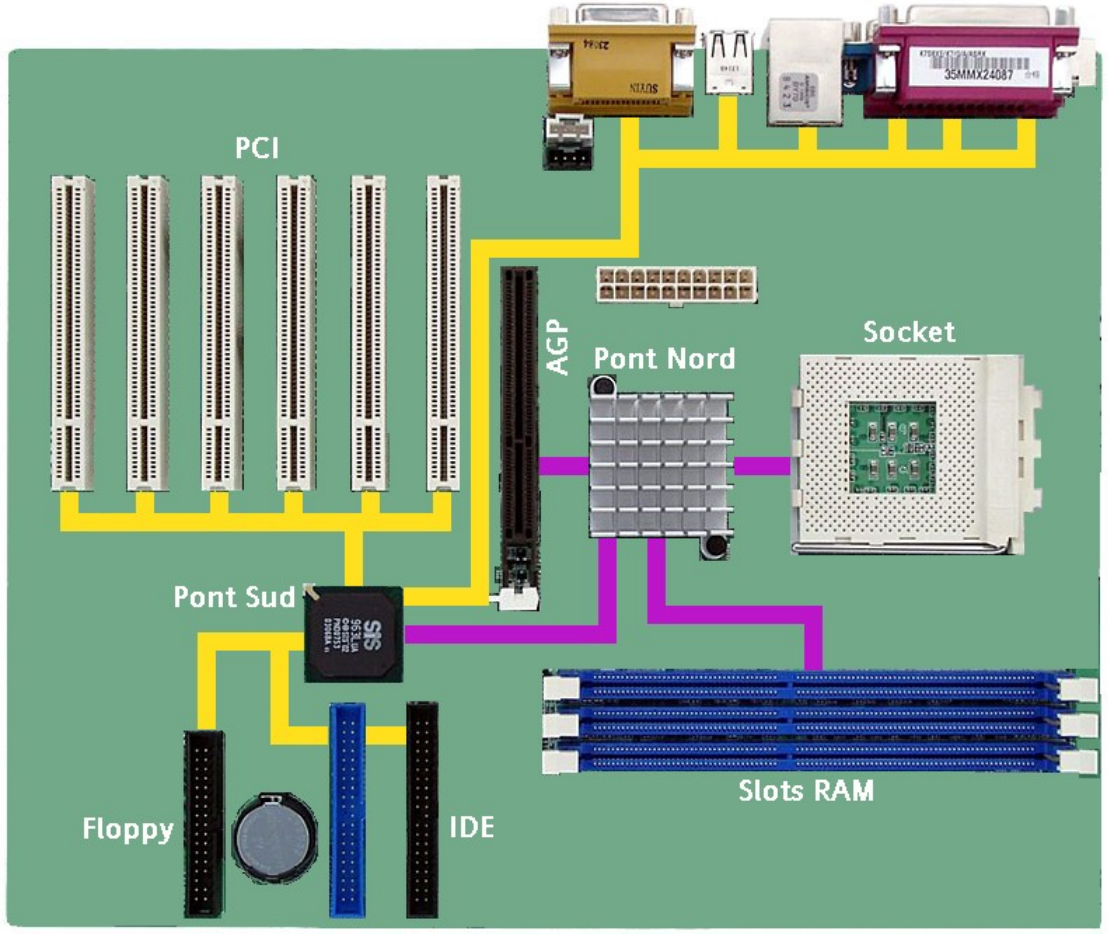
Fonctionnement



Concrètement



Concrètement



Plan

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

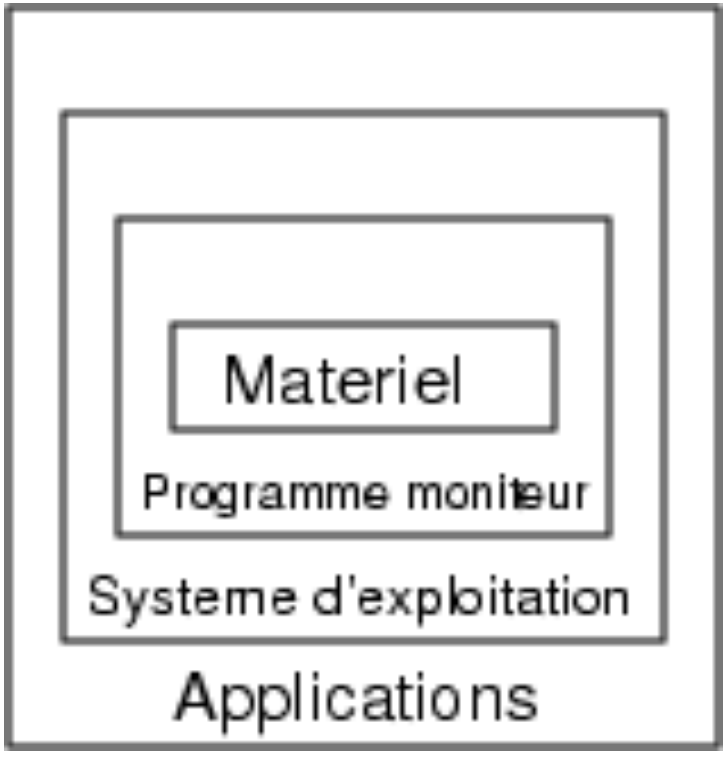
Définition

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


Materiel
Programme moniteur
Systeme d'exploitation
Applications

Plan

Définition

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

- un espace d'adressage qui contient :
  - le programme executable
  - 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 pocessus 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 clône 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 évênement 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'ordonnancement

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

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 memoire 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

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 electronique, le **controleur de périphérique** (ex : controleur 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 travaille du processeur.
- Reduit le nombre d’interruptions.

Les disques magnétiques

- Organisation en cylindres
- Chaque cylindre contient 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

Stockage à long terme d'informations

- Enegistrement 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 voie 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...