

# INTRODUCTION AUX SYSTÈMES D'EXPLOITATION

---



Pr. Hajar IGUER,  
[hajar.iguer@uic.ac.ma](mailto:hajar.iguer@uic.ac.ma)

# Pourquoi un système d'exploitation?

- L'ordinateur est considéré comme un système complexe et difficile à comprendre,
- Pour les programmeurs, il serait bien difficile de créer des programmes tout en prenant en compte la gestion du matériel
- Afin de gérer tous les composants de l'ordinateur, il fallait une interface ou couche intermédiaire.

# Définitions

- Ordinateur: Machine électronique à calculer dotée de mémoire pour traiter l'information reçue ayant la capacité de résoudre des problèmes
- Programme: Un ensemble de d'instructions séquentiels pour que l'ordinateur puisse résoudre un problème donné
- Un processus: L'aspect dynamique d'un programme en cours d'exécution
- Instruction: Une information élémentaire contenant une commande et la description de l'action entreprise.

# INTRODUCTION

Avant de parler systèmes d'exploitation, on remarque tous que lors de l'achat d'un ordinateur avec des caractéristiques de 500 Go de disque dur, 3Go de mémoire. On voudrait remarquer l'exactitude de ces informations.



**On a tous été surpris de voir que le disque dur fait beaucoup moins que 450 Go ou que la mémoire qui passe à 2,5 Go!!!!!!**

# INTRODUCTION

Un **système informatique** moderne comprend un ou plusieurs processeurs, de la mémoire principale, des disques, des imprimantes, un clavier, un écran, des interfaces réseaux et autres périphériques d'E/S.



**C'est pour cela que les ordinateurs sont équipés d'une couche logicielle appelée système d'exploitation**



# OBJECTIF

- Ce cours vise à vous familiariser avec les concepts et les techniques fondamentales des systèmes d'exploitation. Entre autres les éléments suivants:
  - Comprendre le fonctionnement d'un ordinateur en liaison avec le système d'exploitation
  - Expliquer et analyser le fonctionnement des différents modules liés à la gestion de processus et de tâches temps réel,
  - La virtualisation,
  - La compréhension du fonctionnement de la mémoire ainsi que sa gestion
  - La gestion des fichiers
  - Utilisation du MS-DOS et ses différentes commandes
  - Ainsi que les solutions adoptées dans les systèmes modernes (UNIX, LINUX et WINDOWS).

# PLAN



**CADRE GÉNÉRAL**



**HISTORIQUE**



**COMPOSITION**



**EXEMPLES – MS DOS**

# CADRE GÉNÉRAL

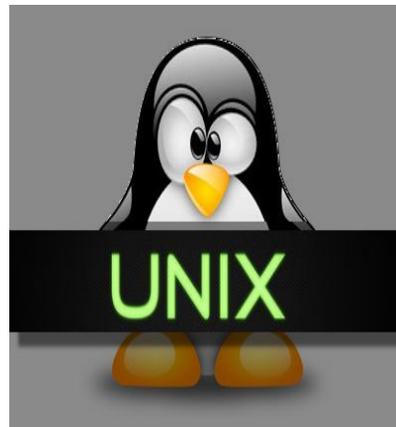
---

# Qu'est ce qu'un système d'exploitation?

- Un système peut être défini comme un ensemble de programmes qui contrôle et dirige l'utilisation d'un ordinateur à travers différentes applications.
- En autres, c'est un système qui **exploite** les ressources matérielles d'un ordinateur et assure la liaison entre les utilisateurs et les applications.
- C'est le premier programme exécuté lors du démarrage de l'ordinateur.
- Le système d'exploitation peut être noté SE ou OS (Operating System)

# Différents Types D'exploitation

Mac-OS Debian  
Symbian QNX Ubuntu  
OpenBSD LynxOS  
BSD Solaris  
Globus  
Red-Hat  
HP-UX Plan9  
Nucleus VMS  
FreeBSD BeOS  
Amoeba  
Windows  
UNIX  
Linux  
Android  
MS-DOS



# Différents Types D'exploitation: Exemples

CP/M (depuis 1974), Digital Research

UNIX (depuis 1969-1979), premier par AT&T

MS-DOS (depuis 1981), Microsoft

MacOS (depuis 1984), Apple

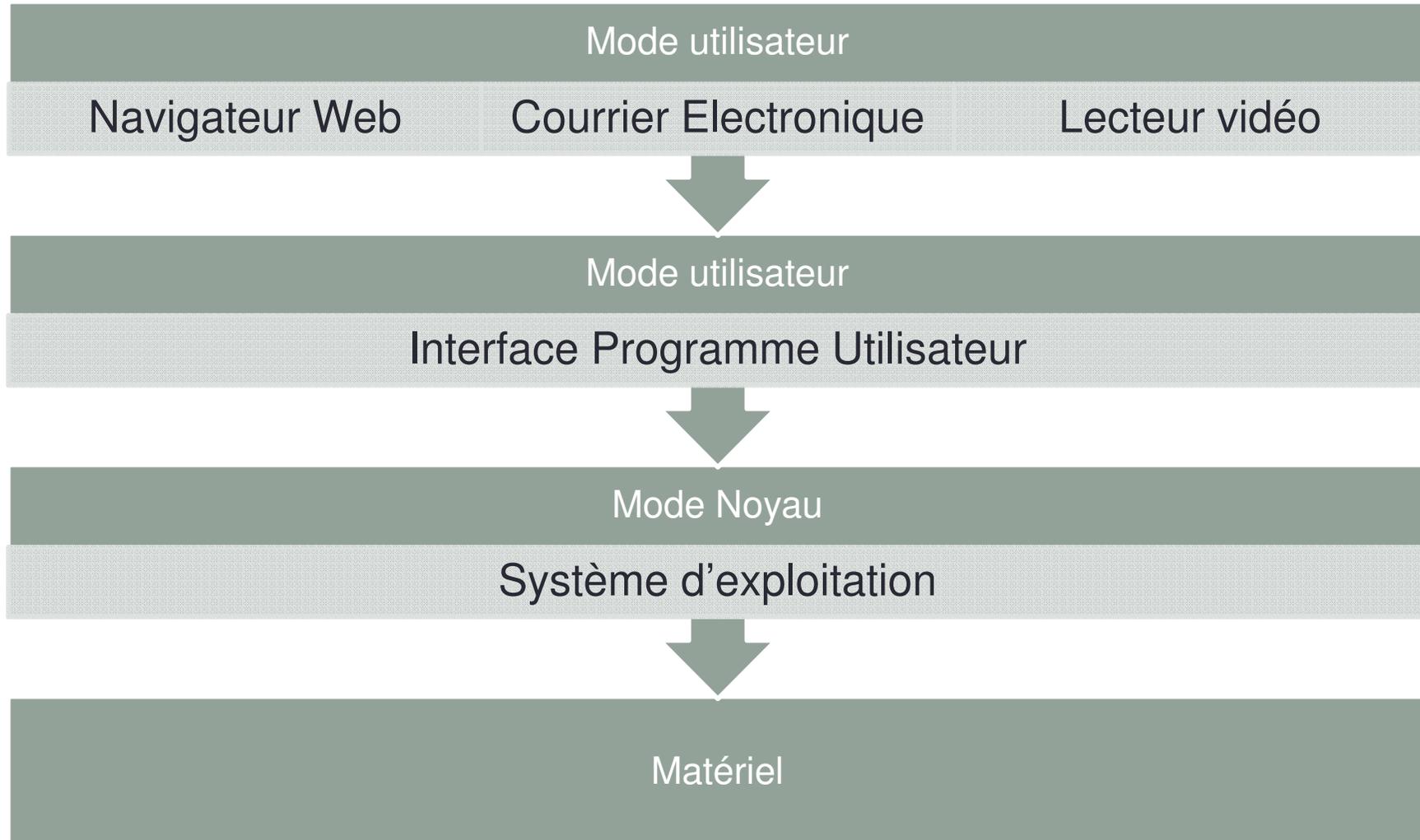
Windows (depuis 1991), Microsoft

Linux (depuis 1992), OpenSource

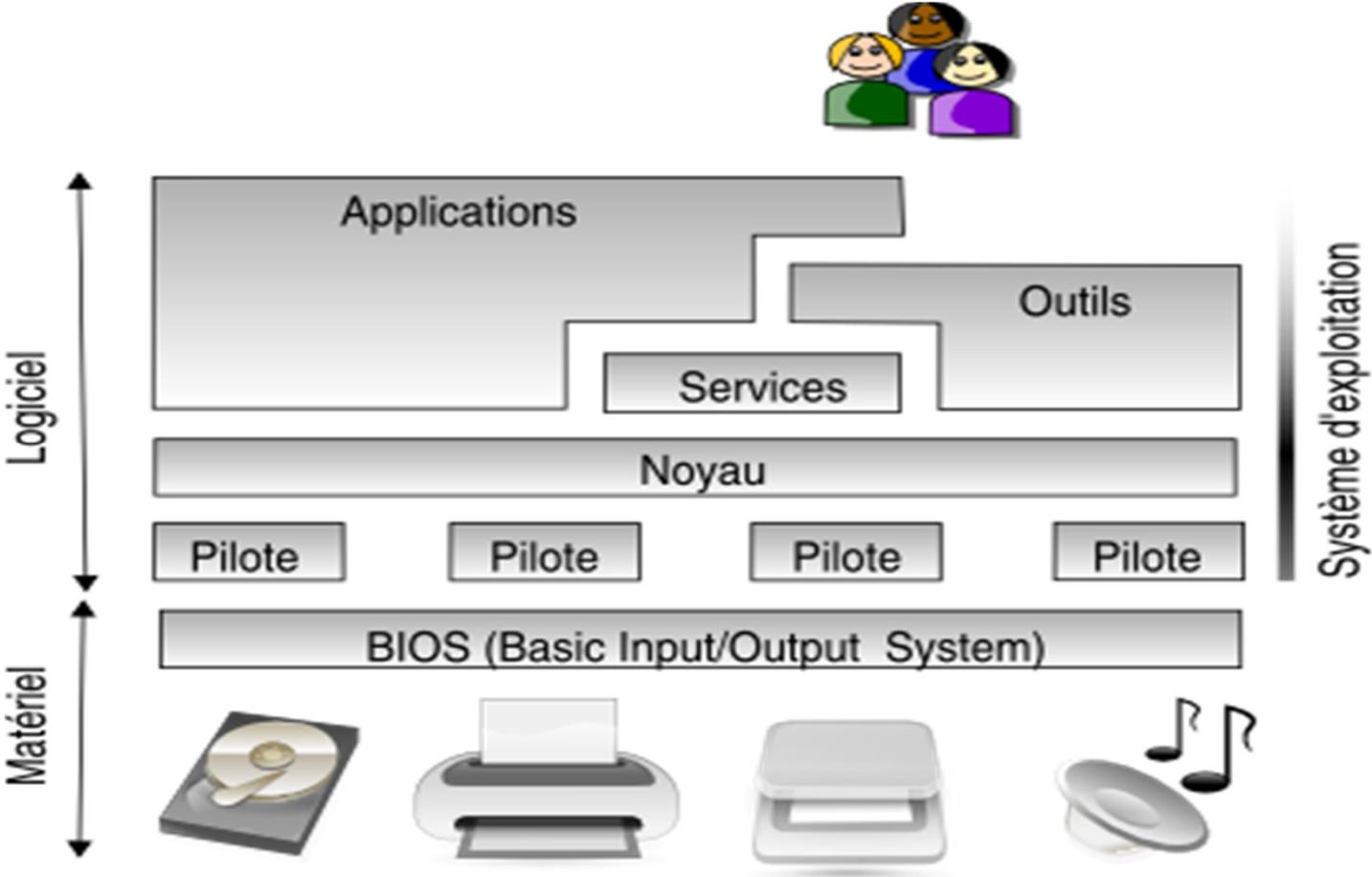
# Définitions - Abstraction

- Machine réelle: L'unité centrale et l'ensemble des périphériques
- Machine abstraite: Le fonctionnement de la machine réelle par un système d'exploitation
- Machine Utilisable: Machine Abstraite + application

# Emplacement du système d'exploitation dans un ordinateur



# Illustration d'un système d'exploitation



# HISTORIQUE : TYPOLOGIE

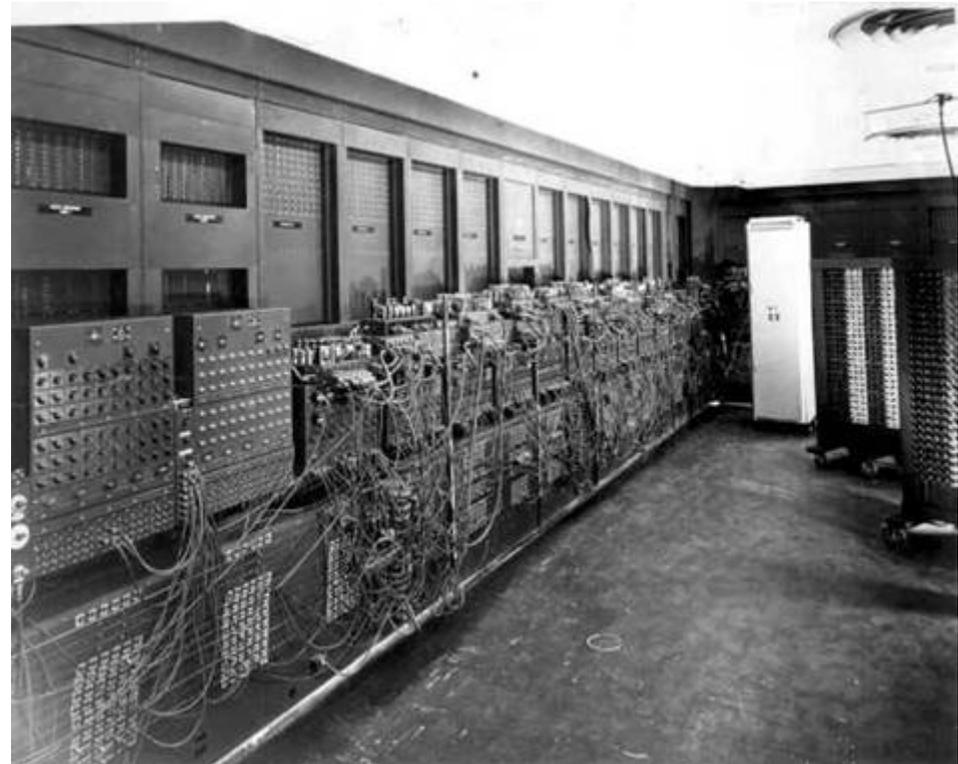
---

# Typologie

- Il existe à ce jour cinq générations de système d'exploitation
- Première Génération : Tubes à vide (Jusqu'à 1955)
- Deuxième Génération: Traitement par lots
- Troisième Génération : Circuits intégrés et Multiprogrammation
- Quatrième Génération: Ordinateurs personnels
- Cinquième Génération : Systèmes Répartis

# Première Génération : Tubes à vide (Jusqu'à 1955)

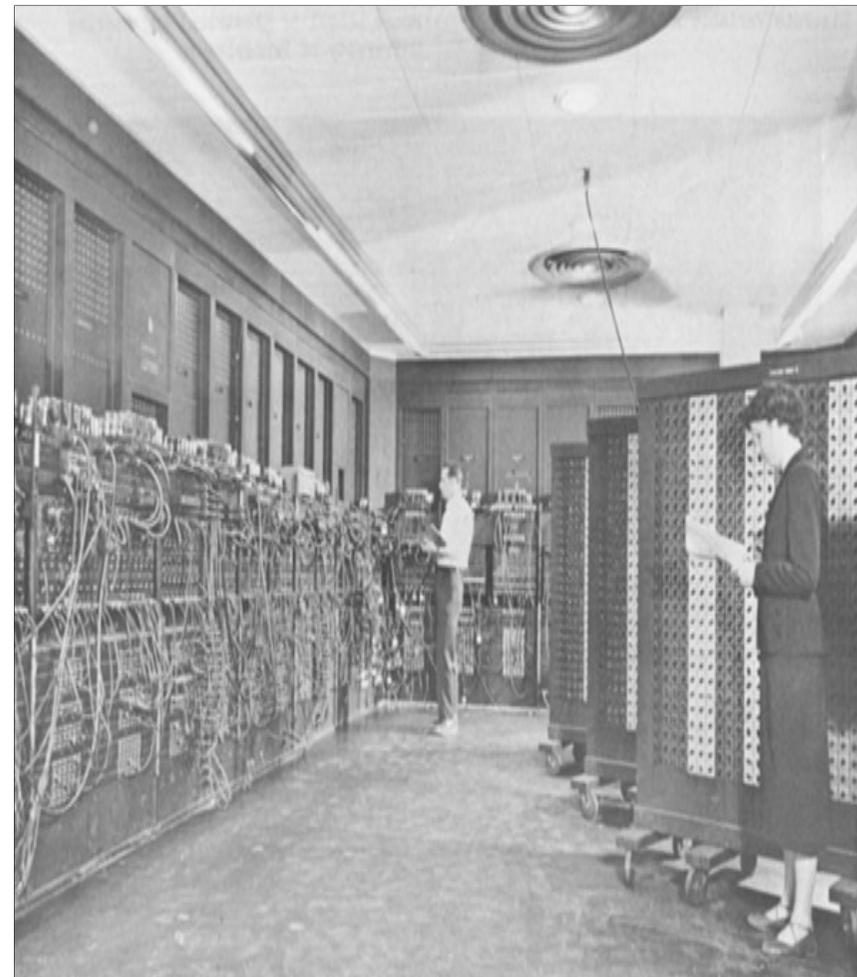
- Vers le milieu des années 1940, les premières machines à calculer sont construites au moyen de tubes électroniques (tubes à vides);
- Ces machines énormes remplissaient des salles et étaient moins rapides qu'une calculatrice moderne;
- Toute la préparation se faisait manuellement;
- Les E/S étaient limitées;



# Première Génération : Tubes à vide (Jusqu'à 1955)

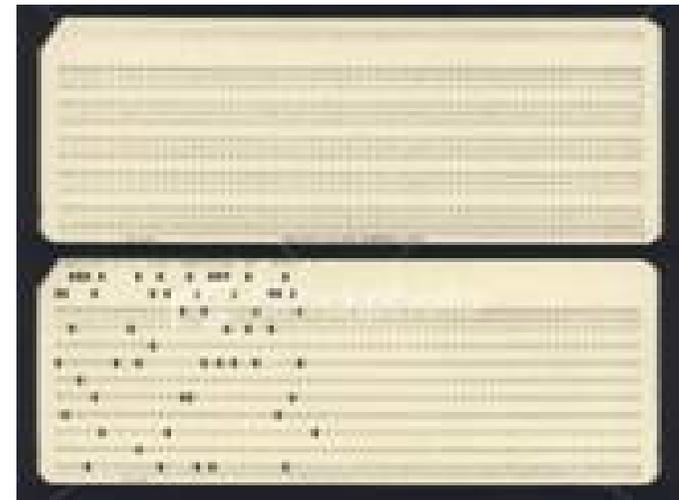
- Le programmeur opère avec la machine par interaction directe (beaucoup de manipulation et beaucoup de lourdeur);
- Concepteur = constructeur = programmeur = opérateur;
- Un programme dispose toujours de toute la mémoire disponible;
- Programmer l'E.N.I.A.C. consistait à câbler des branchements et à ajuster des boutons électriques.

**L'ENIAC 30 tonnes**  
Electronic **N**umerical Integrator  
and **C**omputer



# Deuxième Génération: Traitement par lots

- Le traitement par lots a constitué les bases des systèmes d'exploitation
- Apparition des cartes perforées
- Séparation des rôles : le programmeur n'est pas l'opérateur;
- Enchaînement automatique des commandes (« jobs ») dans une queue d'exécution de programmes, c'est le traitement par lots (« batch processing »);



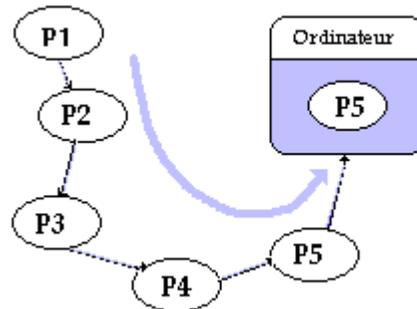
# Deuxième Génération: Traitement par lots

- Sous-programmes d'E/S (chargement et déchargement des cartes);
- Apparition des assembleurs;
- Le système d'exploitation gère la mémoire, les processus et les E/S;
- Gestion par limitation de temps;



# Deuxième Génération: Traitement par lots

- Introduction de la notion d'ordonnancement : dans quel ordre sont effectuées les jobs;
- Lors des E/S, le processeur est inactif;
- " Monoprogrammation " dans lesquels un seul utilisateur est présent et a accès à toutes les ressources de la machine pendant tout le temps que dure son travail.



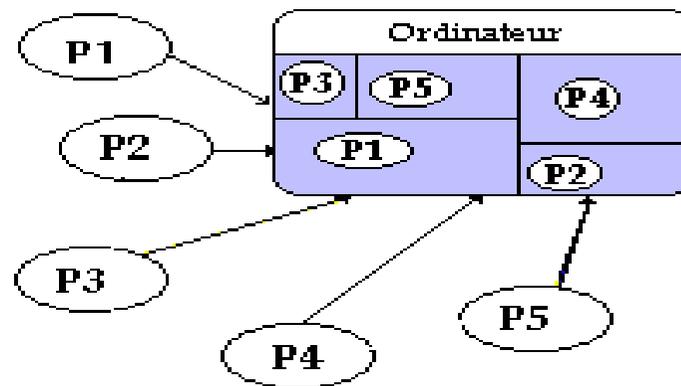
# Troisième Génération : Circuits intégrés et Multiprogrammation (1965 – 1980)

- Apparition des circuits intégrés;
- Selon la loi de Moore, le nombre de transistors d'un circuit intégré double tous les 18 mois,
- Apparition des disques magnétiques;
- Notion de familles d'ordinateurs compatibles (IBM System/360);



# Troisième Génération : Circuits intégrés et Multiprogrammation

- **Multiprogrammation** : pendant qu'un processus effectue ses E/S, le processeur est inactif, il peut donc travailler sur un autre processus;
- Multiprogrammation rendue possible par le « spooling », qui met à la disposition du CPU un ensemble de jobs (*pool*) prêtes à être exécutées;

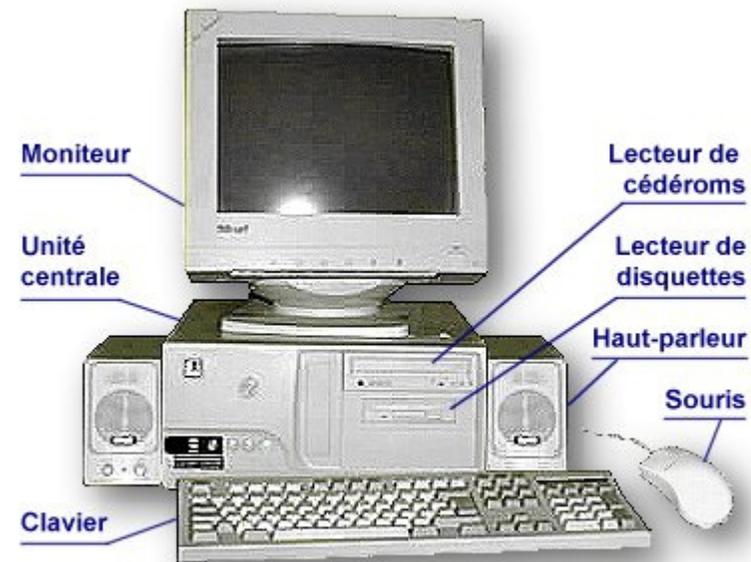


## Quatrième Génération: Ordinateurs personnels (De 1980 - 2000)

- C'est la génération des PCs (Personal Computer)
- Où l'électronique est miniaturisé et l'Apparition des micro-ordinateurs grand public (IBM PC, p.ex.), de MS-DOS, des consoles de jeu;
- IBM conçoit le PC autour du processeur Intel 8080 sans OS
- Bill Gates à combiné l'utilisation du langage BASIC avec le système d'exploitation DOS conçu par Seattle Computer Products.
- Bill Gates et Tim Patterson fondent Microsoft et Tim finalise MS-DOS

# Quatrième Génération: Ordinateurs personnels (De 1980 - 2000)

- Les micro-processeurs
- Motorola 68000;
- L'utilisation d'Internet;
- Création d'applications multimédia;
- Maturation des systèmes d'exploitation vers ceux que l'on connaît aujourd'hui;



**Micro ordinateur**

# Quatrième Génération: Ordinateurs personnels (De 1980 - 2000)

- **PC vs Mac**
- Steve Jobs, co-inventeur d'Apple, visite un jour la société Xérox et voit une interface graphique (prototype)
- Il développe un système d'exploitation et le matériel qui va avec : Lisa (un flop) puis le Macintosh (un succès)
- Evolutions de MS-DOS influencées par le Macintosh avec l'ajout d'une interface graphique

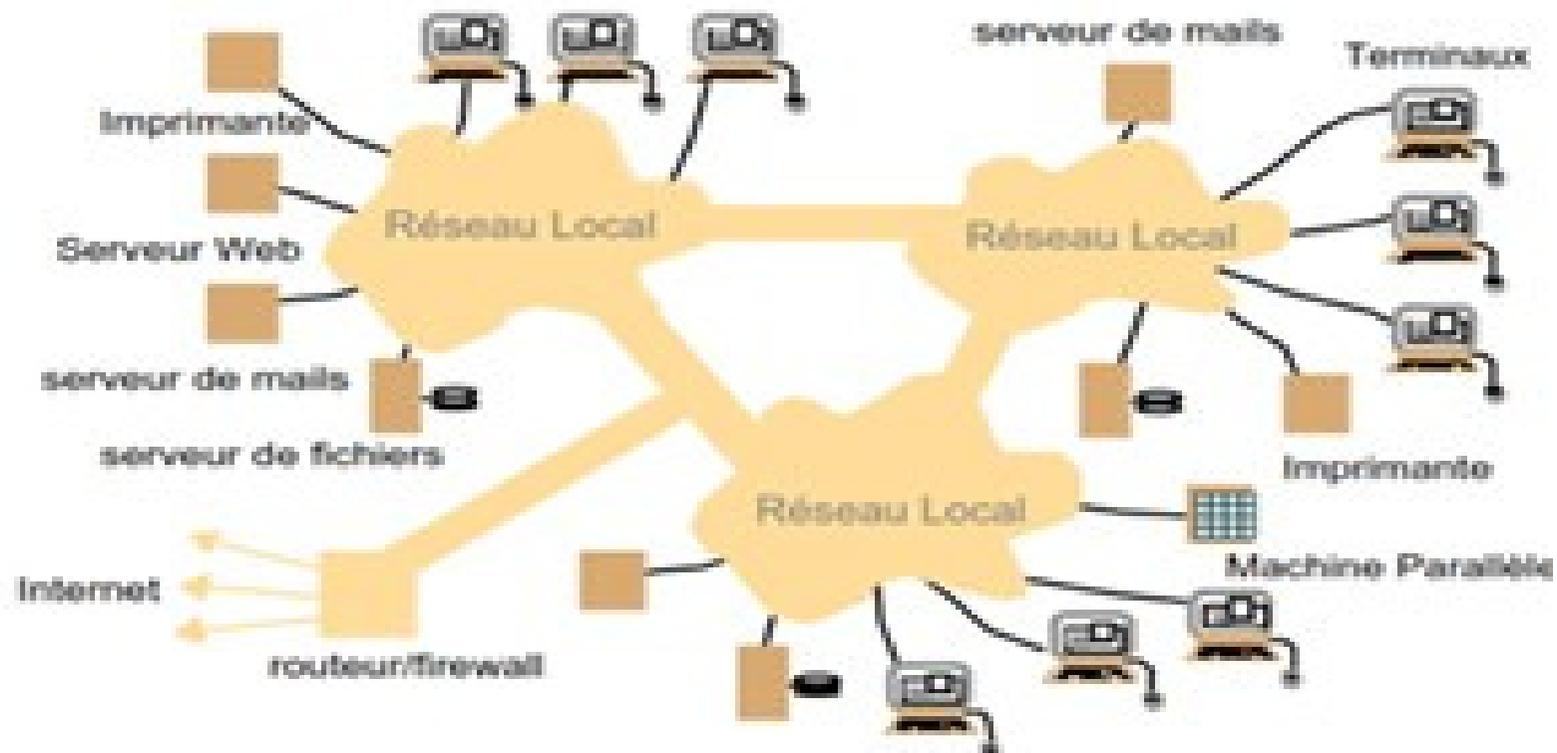


# Systemes Répartis

- L'informatique répartie s'oppose à l'informatique centralisée (Gros ordinateurs), à l'informatique individuelle (micro-ordinateur).
- Elle Compte plusieurs avantages:
  - Partage des données
  - Partage des applications
- Plusieurs processeurs sur un même appareil;
- Plusieurs processeurs disponibles sur des appareils différents pour un même programme;
- Plusieurs « cores » sur un même processeur;
- Systemes de client/serveur;
- Connectivité inter-serveurs;
- Exemple: Un intranet

# Cinquième Génération : Systèmes Répartis

- Un système réparti est un ensemble de sites reliés par un réseau, comportant chacun une ou plusieurs machines.
- Exemple de Systèmes Répartis: Un intranet



# Systeme Temps Réel

- Exécute et synchronise des applications en tenant compte du temps
- Par exemple:
  - **Un système gérant une chaîne de production de charcuteries doit tenir compte des délais d'une pièce boules de charcuterie à la machine de boyau pour l'emballage**
  - **L'arrivée du tramway à une des stations d'arrêts**

# Systeme Embarqué

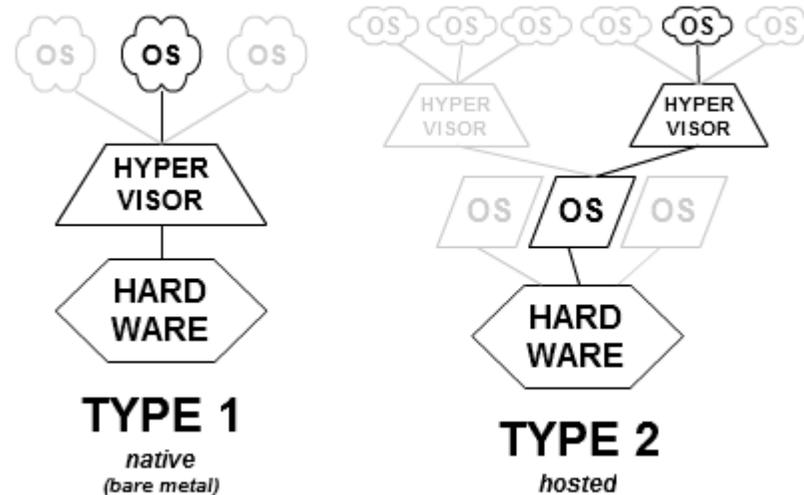
- Un système d'exploitation dédié à des applications en nombre restreint et identifiées;
- Par exemple
  - Un système pour assistant personnel de poche,
  - Un système pour téléphone portables se connectant à internet
  - Un système de navigation dans une voiture
  - Un système pour la robotique

# Hyperviseur

- Hyperviseur : logiciel qui permet d'exécuter un ou plusieurs systèmes d'exploitation (superviseurs) dans un environnement matériel virtuel (ou « machine virtuelle »).
- Exemples :
  - [VMWare](#)
  - [VirtualBox](#)
  - [Parallels](#)
  - [Xen](#)
- Les hyperviseurs ne sont pas des *émulateurs* : l'architecture du processeur virtuel est la même que celle du processeur réel → performances
- Les processeurs modernes offrent des fonctions spécialisées pour la virtualisation (mode utilisateur / superviseur / **hyperviseur**)

# Hyperviseur : Types

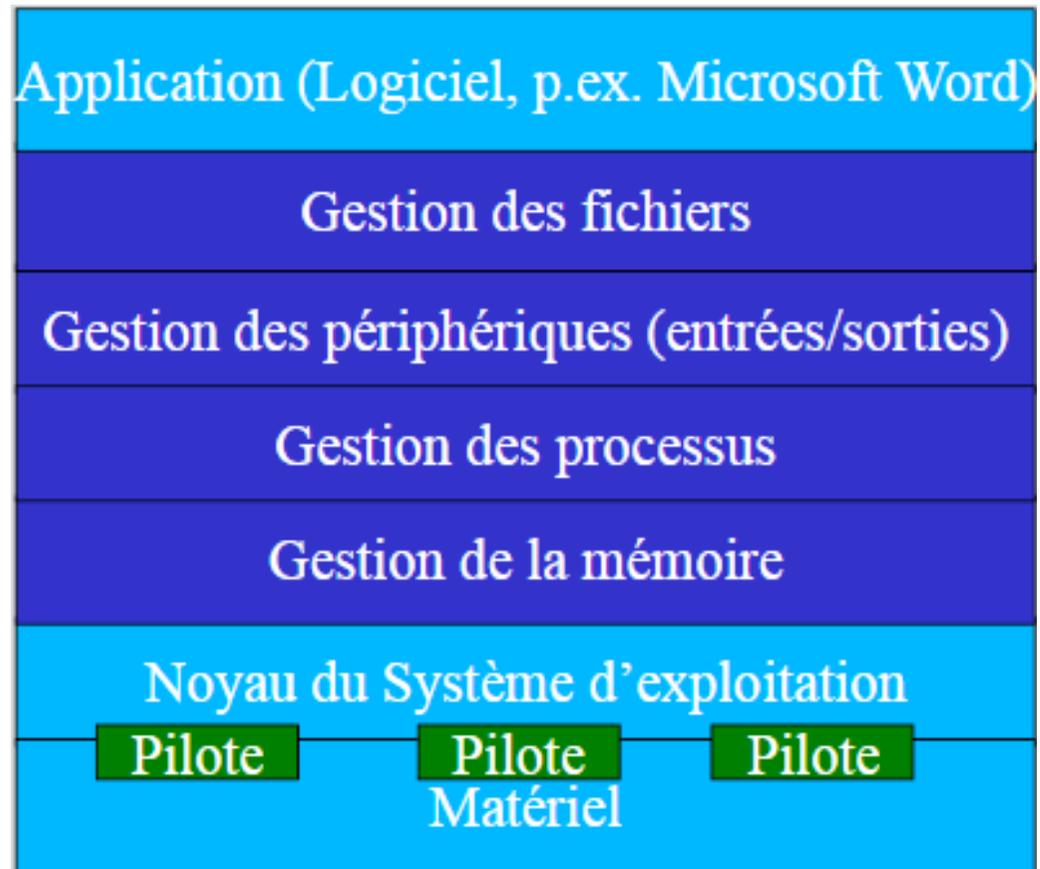
- Type 1: Nommé natif ou « bare-métal » est un logiciel qui se lance directement sur une plateforme matérielle.  
Exemple: Hyper-V de Microsoft, ESX Server de Vmware,..
- Type 2: est un logiciel qui s'exécute directement dans un autre système d'exploitation. Exemple: Microsoft Virtual PC, Vmware Workstation.....



# COMPOSITION

---

# SE – Modèle en Couches



# Le noyau du système d'exploitation

- Appelé en anglais le Kernel, il représente les fonctions fondamentales du système d'exploitation telles que:
- La gestion de la mémoire,
- La gestion des processus,
- La gestion des fichiers,
- La gestion des entrées-sorties

# Le Processeur/ Processus

- Le « cerveau » de l'ordinateur ou UC ( Unité de Commande)
- Extrait des instructions de la mémoire et les exécute
- Un processus est un concept clé dans tout système d'exploitation et peut être schématisé comme un programme en cours d'exécution
- Exemple:
- Courrier électronique, la compression d'un fichier, le navigateur web

# La performance du Processeur

- Les concepteurs d'ordinateurs ont cherché à évaluer et à comparer la vitesse d'exécution des processeurs, grâce à des unités de mesure comme:
- le nombre d'instructions par seconde (IPS),
- le nombre d'opérations par seconde (OPS)
- le nombre d'opérations flottantes par seconde (FLOPS) et leurs multiples (kilo, méga, téra...). Ultérieurement furent définis les dhrystones, les whetstones...
- Les MOPS ou les MFLOPS (lire méga-FLOPS) s'utilisent pour les programmes contenant de nombreux calculs scientifiques.

# La performance du Processeur

- Whetstone et Dhrystone sont des bancs d'essai synthétiques qui reproduisent le comportement de véritables programmes.
  - Whetstone convient pour les applications comprenant de nombreux calculs
  - Dhrystone donne une indication sur l'universalité (*general purpose*) des processeurs modernes.
- Tous deux contribuent à l'écriture de compilateurs efficaces.

# Gestion des processus

- Le SE est responsable de:
  - Allocation de ressources aux processus
  - Création, terminaison des processus
  - Suspension, reprise des processus
  - Synchronisation, communication entre processus

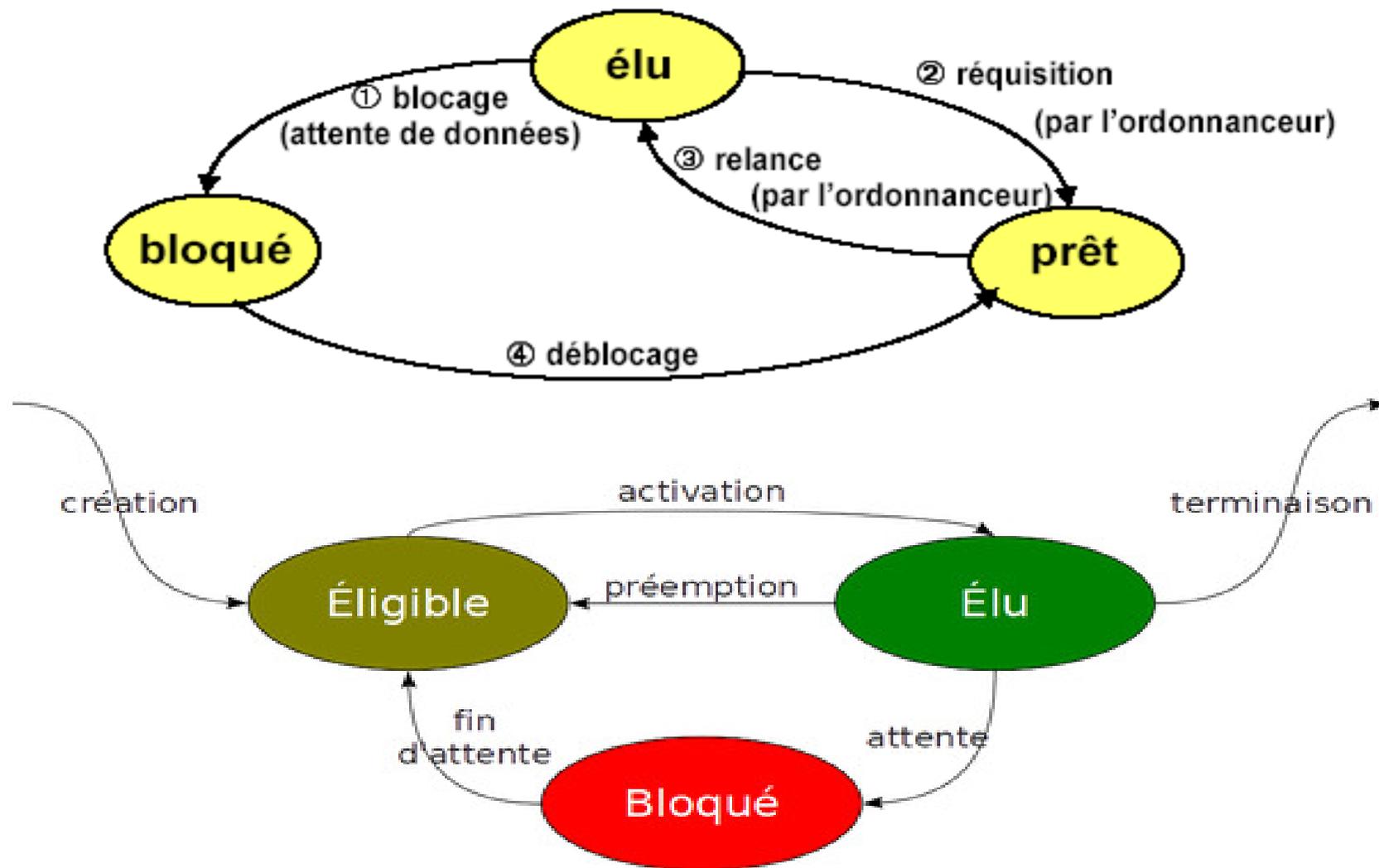
# Table des processus

- La table des processus est un tableau de structures contenant toute l'information sur chaque processus en cours d'exécution sur le système
- Périodiquement, le système d'exploitation décide d'interrompre un processus pour en exécuter un autre. Il doit alors:
  - Suspendre l'exécution d'un processus A
  - Sauver l'information de A dans la table des processus
  - Choisir un nouveau processus B
  - Récupérer l'information du processus B
  - Démarrer l'exécution du processus B

# Etats d'un Processus

- Un processus peut être dans 3 états possibles :
  - **Elu** (Actif) - Processus actif, Processeur en marche
  - **Prêt** (Suspendu) – processus actif, processeur occupé par un autre processus
  - **Bloqué** (En attente d'un évènement extérieur) Processus non actif, même si le processeur est disponible.

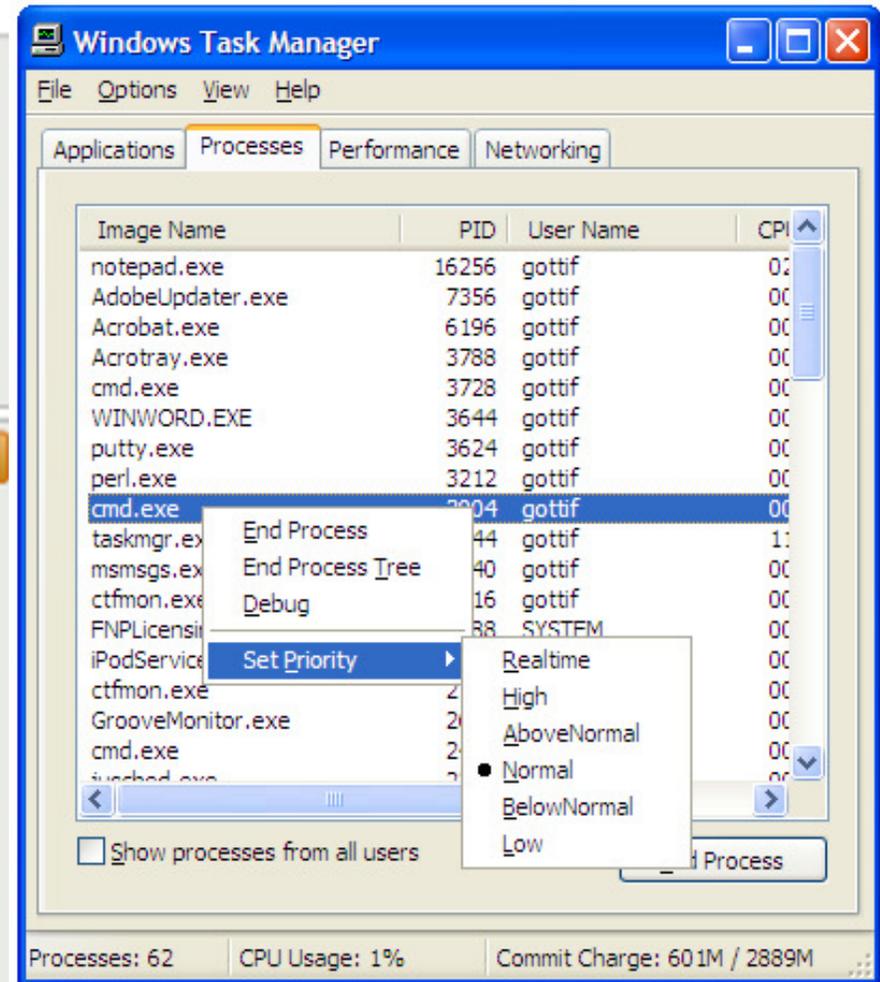
# Etats d'un processus



# Ordonnanceur (*Scheduler*)

- Partie du SE qui détermine si un processus doit poursuivre son exécution ou être temporairement arrêté pour être remplacé par un autre
- Le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes grâce à un **algorithme d'ordonnement**.
- Le type d'ordonnanceur est **totallement dépendant du système d'exploitation**, en fonction de l'objectif visé.
- Doit choisir le prochain processus à être exécuté ce qui revient à gérer des Priorités.

# Gestion des priorités



# Moniteur Système Linux

System Monitor

System Processes Resources File Systems

Load averages for the last 1, 5, 15 minutes: 0.25, 0.42, 0.46

Process Name	ID	User	Status	% CPU	CPU Time	Started	Memory	Virtual Memory
gnome-system-monitor	11873	pa	Running	4	1:20.17	Today 06:00 PM	7.7 MiB	361.0 MiB
compiz	1803	pa	Sleeping	4	2:58:15	Sat 11:49 AM	372.5 MiB	1.0 GiB
firefox-bin	3598	pa	Sleeping	0	1:04:54	Sat 08:30 PM	683.5 MiB	1.5 GiB
gwibber-service	2056	pa	Sleeping	0	2:50.75	Sat 11:50 AM	10.5 MiB	385.2 MiB
gwibber-service	2055	pa	Sleeping	0	2:52.19	Sat 11:50 AM	10.8 MiB	385.7 MiB
gwibber-service	2054	pa	Sleeping	0	2:49.69	Sat 11:50 AM	10.4 MiB	385.2 MiB
gwibber-service	2053	pa	Sleeping	0	2:51.40	Sat 11:50 AM	10.6 MiB	385.3 MiB
empathy	10812	pa	Sleeping	0	0:24.31	Today 11:26 AM	25.5 MiB	672.5 MiB
thunderbird-bin	6065	pa	Sleeping	0	21:06.27	Yesterday 11:01 A	154.8 MiB	814.2 MiB
nautilus	1816	pa	Sleeping	0	3:24.80	Sat 11:49 AM	82.2 MiB	866.7 MiB
dropbox	1860	pa	Sleeping	0	1:04.21	Sat 11:49 AM	36.4 MiB	662.8 MiB
evince	10949	pa	Sleeping	0	0:17.49	Today 05:48 PM	27.0 MiB	520.6 MiB
evince	10173	pa	Sleeping	0	0:06.50	Today 04:00 PM	56.1 MiB	520.4 MiB
gwibber-service	1930	pa	Sleeping	0	3:24.17	Sat 11:50 AM	16.8 MiB	486.7 MiB

End Process

# Multi-tâches et Multi-processeurs

- **Multitâches:**

- Assurer l'exécution de plusieurs processus en même temps
- Chaque processus doit bénéficier de la capacité du processeur. Plusieurs processus concurrents
- D'où l'utilité de l'ordonnancement

- **Multiprocesseurs:**

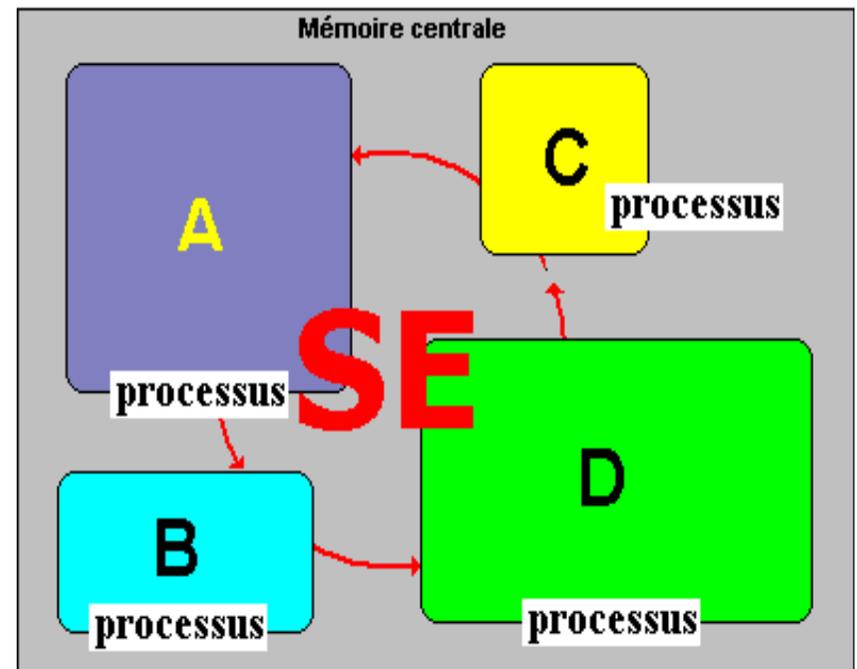
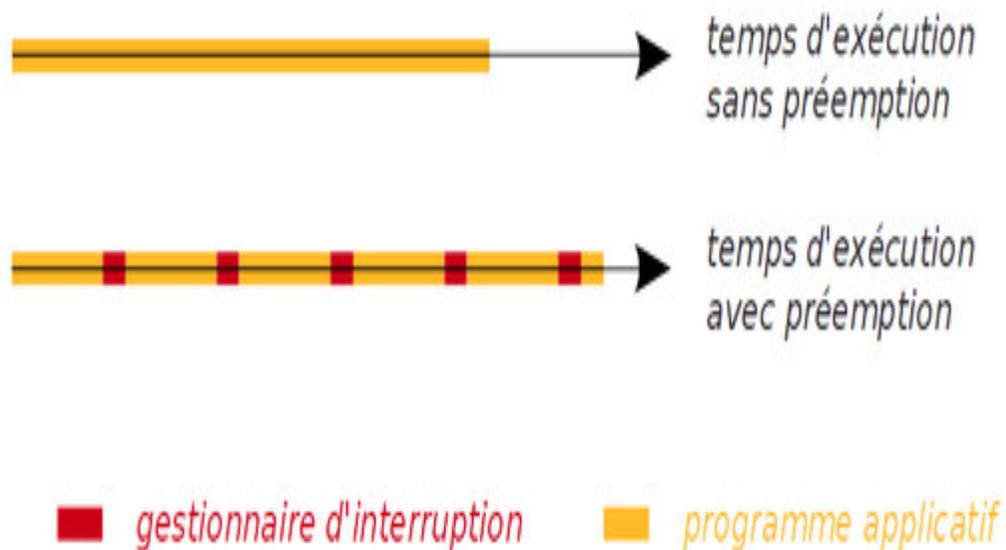
- La gestion multitâches réelle
- L'exécution d'autant de processus qu'il y a de processeurs

# Ordonnancement

- Le système doit faire un choix :
  - Équité : Chaque processus doit avoir du temps processeur.
  - Efficacité : Le processeur doit être utilisé à 100 %
  - Temps de réponse : L'utilisateur devant sa machine ne doit pas attendre.
  - Temps d'exécution : Une séquence d'instructions ne doit pas trop durer.
  - Rendement : Il faut faire le plus de choses possibles dans le temps le plus court possible.

# Gestion des processus

- La préemption est la mise en attente forcée d'un processus
- C'est l'ordonnanceur qui se charge de la gestion de la préemption
- Interruption d'horloges pour permettre la préemption par le SE;



# Multithreading et Circuits intégrés multicoeurs

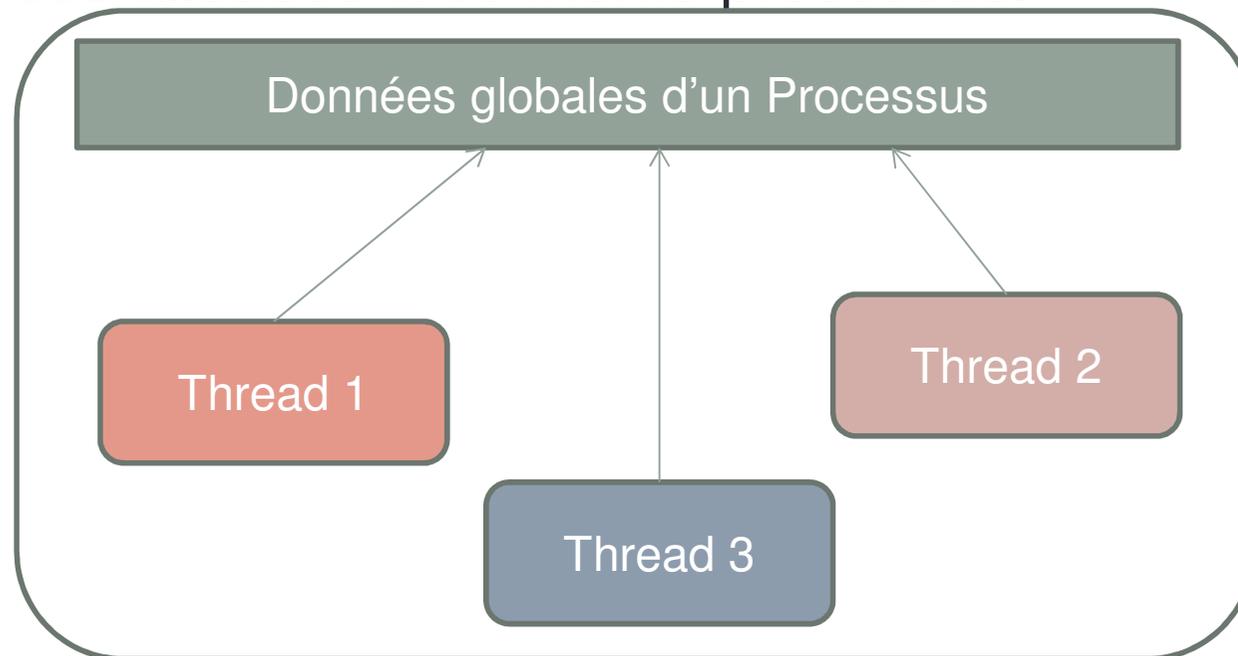
- Une approche de la duplication des unités fonctionnelles et des unités de commande a donné lieu au multithreading
- Un thread est une sorte de processus léger, qui correspond à un programme exécutable par le processeur.
- Une UC dotée de la technique du multithreading peut passer très rapidement à l'autre processus pour éviter l'attente.
- Pour le multithreading , un seul processus est exécuté à la fois . Exemple : Pentium 4

# Thread

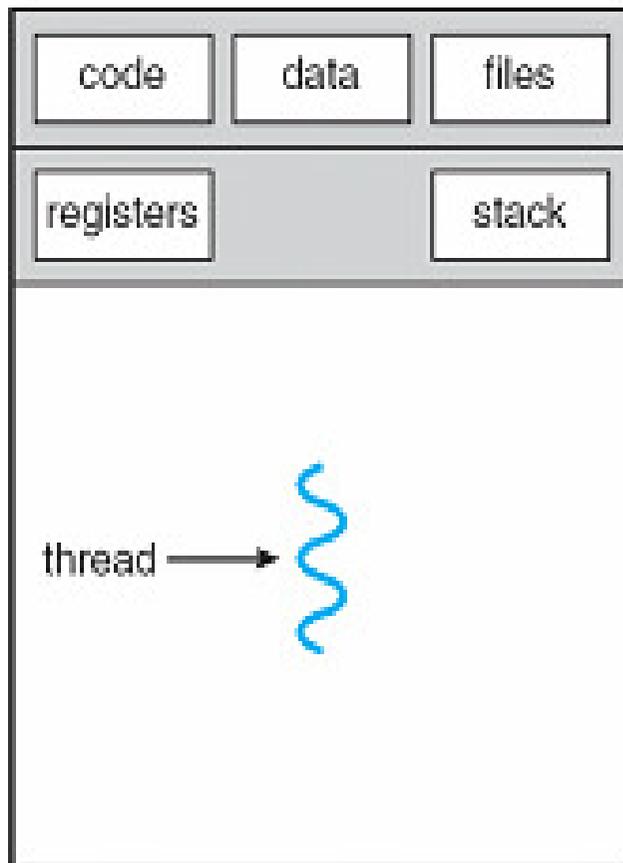
- Afin de séparer les tâches d'un processus, il a été mis en place la notion de **processus léger**.
- Chaque processus peut fonctionner comme un SE en lançant des sous tâches internes au processus. On l'appelle le **multithreading**
- Ces sous-tâches sont nommées **processus léger** ou **thread**.
- Un processus peut comporter plusieurs threads.

# Thread

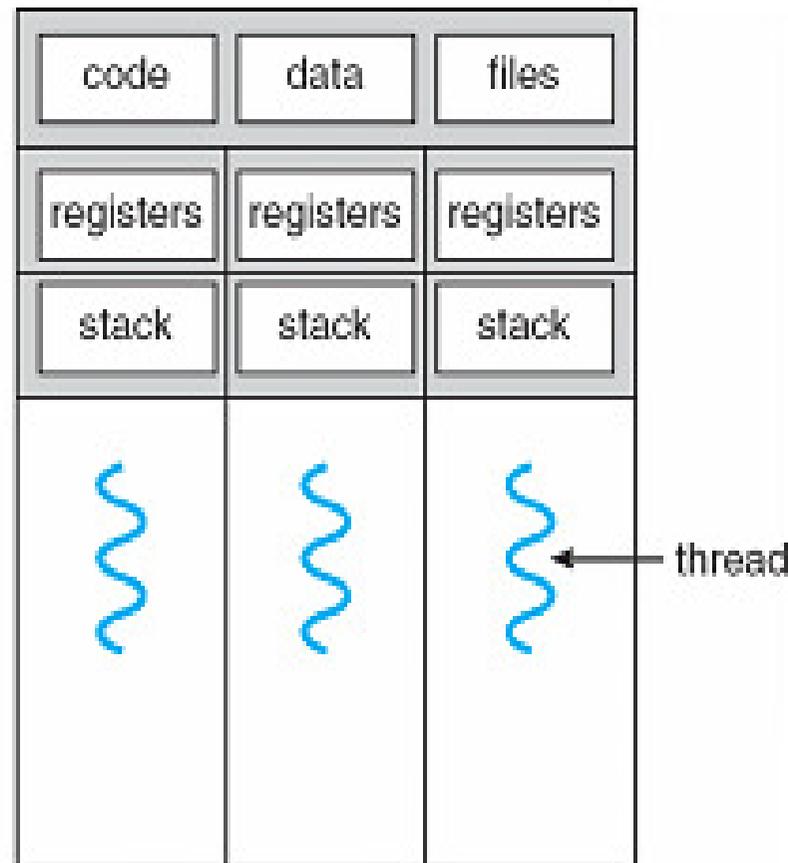
- Un processus travaille et gère, pendant le quantum de temps qui lui est alloué, des ressources et exécute des actions sur et avec ces ressources.
- Les threads situés dans un même processus partagent les mêmes variables générales de données et les autres ressources allouées à ce même processus.



# Thread vs Multithreading



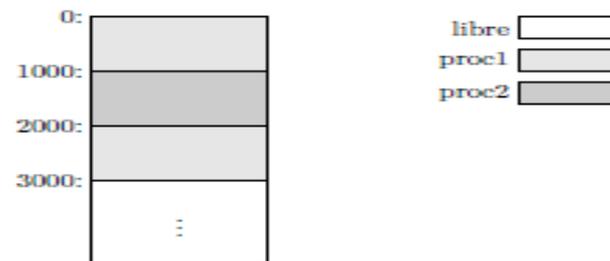
single-threaded process



multithreaded process

# Gestion de la Mémoire

- Un processus doit être chargé dans la mémoire centrale pour être exécuté
- Une fois le processus est terminé, l'espace mémoire qui lui est alloué est donc libéré.
- Ce processus (proc1) sera alors situé (par exemple) aux adresses physiques 0 à 999.
- Un deuxième processus se verra attribué à espace mémoire correspondant aux adresses physiques 1000 à 1999



# Gestion de la Mémoire

- En cas de demande d'extension de la mémoire par le processus 1, le système d'exploitation ne pourra répondre que par un autre espace d'adressage **non contigu** à celui attribué lors de la création du processus.
- Ainsi l'extension peut se situer en les adresses physiques 2000 et 2999.
- À cet effet, la mémoire se trouve fragmentée en une **multitude de petites zones difficilement utilisables**.

# Gestion de la Mémoire

- C'est alors pour ce but qu'on essaye de défragmenter la mémoire pour qu'elle soit correctement utilisée.
- La méthode proposée est la défragmentation de la mémoire ou le tassage de la mémoire qui consiste à éliminer toutes les zones libres et à déplacer les zones de mémoire occupées vers les adresses basses.

# Adresse Physique/ Adresse Virtuelle

- Chaque zone mémoire possède deux adresses :
  - Une adresse dite physique, qui correspond à la position matérielle de la zone en mémoire.
  - Une adresse dite virtuelle qui est un nombre arbitraire auquel il est possible de faire correspondre une adresse physique.
- Pendant l'exécution d'un processus, la conversion des adresses virtuelles utilisées par un processus en adresses physiques connues par les couches matérielles.
- Cette conversion doit donc être très rapide (puisque'elle conditionne la vitesse d'exécution d'un processus) et est assurée par un composant électronique logé dans le processeur : la MMU

# Gestion de la Mémoire

- La mémoire physique sert de zone de stockage temporaire pour les programmes et données que vous utilisez.
- De façon générale, plus la quantité de mémoire est importante, plus vous pouvez lancer d'applications simultanément.
- D'autre part, plus celle-ci est rapide plus votre système réagit vite,

# Types de Mémoire

- La mémoire physique sur un système se divise en deux catégories:
- Mémoire vive: composée de circuit intégrés, donc très rapide. Mais sa gestion requiert la participation avec la mémoire auxiliaire
- Mémoire de masse: composée de supports magnétiques (disque dur, bandes magnétiques...). Dispose d'un grand espace mais lente par rapport à son temps d'accès.

# Gestion de la Mémoire

- La gestion de la mémoire est un difficile compromis entre les performances (temps d'accès) et la quantité (espace disponible).
- La gestion de la mémoire doit de plus remplir les fonctions suivantes :
  - **Permettre le partage de la mémoire: Système multitâches**
  - **Permettre d'allouer des blocs de mémoire aux différentes tâches ;**
  - **Protéger les espaces mémoire utilisés**
  - **Optimiser la quantité de mémoire disponible, notamment par des mécanismes d'extension de la mémoire**

# Gestion de la Mémoire

- Afin de gérer les ressources de la mémoire et de les affecter aux différents processus en cours d'exécution, le SE utilise certains mécanismes dont:
  - La pagination
  - La segmentation
  - La mémoire virtuelle

# Unité de Gestion de la Mémoire (MMU)

- Les adresses virtuelles référencées par l'instruction en cours d'exécution doivent être converties en adresses physiques
- Cette conversion d'adresse est effectuée par le **MMU**, qui sont des circuits matériels de gestion.
- Le **MMU** vérifie si l'adresse virtuelle reçue correspond à une adresse en mémoire physique.
- Le gestionnaire de mémoire est chargé d'assurer la gestion optimale de la mémoire.

# Unité de Gestion de la Mémoire (MMU)

- Si c'est le cas, le **MMU** transmet sur le bus de la mémoire l'adresse réelle, sinon il se produit un défaut de page. Un défaut de page provoque un déroutement (ou **TRAP**) dont le rôle est de ramener à partir du disque la page manquante référencée. La correspondance entre les pages et les cases est mémorisée dans une table appelée **Table de pages ( TP)**

# Propriétés du MMU

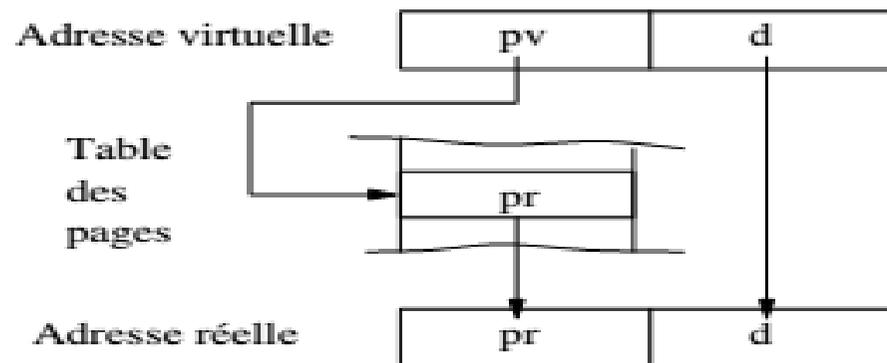
- Le contrôle du tampon
- La protection de la mémoire
- La traduction d'adresses logiques en adresses linéaires par l'unité de segmentation
- La traduction d'adresses linéaires en adresses physiques par l'unité de pagination
- L'arbitrage du bus
- La protection de la mémoire
- La commutation de banque (dans des architectures informatiques plus simples comme les systèmes 8 bit)

# La Pagination

- Cela consiste à découper la mémoire en pages de longueur fixes et de mettre les programmes dans les pages vides.
- En crainte de l'éparpillement des programmes en mémoire, la mise en place d'une table pour reconstituer l'ordre logique de l'emplacement des différentes parties d'un programme est nécessaire.
- La taille d'une page est une puissance de 2 et s'élève en général à quelques K-octets (4, 8 ou 16)
- Cela reste limité à la taille actuelle de la mémoire

# La Pagination

- La conversion d'adresses en pages se fait de la manière suivante:
- Toute adresse virtuelle est décomposable en deux champs : un numéro de page et un déplacement dans cette page.
- La conversion d'une telle adresse en adresse réelle reposera sur une table des pages qui mémorise pour tout numéro de page virtuelle valide, le numéro de page réelle correspondant.



# La Pagination

## Avantages

- Les pages étant de taille fixe en mémoire réelle, l'allocation de l'espace mémoire réel est simple. Il suffit de trouver un nombre suffisant de pages libres pour la taille de mémoire virtuelle.
- La conversion d'adresse est simple

## Inconvénients

- Le principal inconvénient est le découpage arbitraire en tranches du programme "paginé". Il y a donc perte d'espace.
- La structure du programme en section(s) de code, de données, n'est pas connue de la pagination.

# La Segmentation

- La segmentation repose sur les règles suivantes :
  - L'espace d'adressage virtuel est découpé en blocs de taille variable appelés segments ;
  - L'espace d'adressage réel n'est pas structuré et reste un espace de mots ;
  - La taille d'un segment est une puissance de 2 et s'élève en général à quelques centaines de K-octets (64, 128, 256 par exemple).

# La Segmentation : Table des Segments

- Toute adresse virtuelle est décomposable en deux champs : un numéro de segment et un déplacement dans ce segment.
- La conversion d'une telle adresse en adresse réelle reposera sur une table des segments qui mémorise pour tout numéro de segment virtuel valide, l'adresse de début et la longueur effective du segment alloué en mémoire réelle

# La Segmentation

## Avantages

- Permettre un découpage du programme en segments correspondant logiquement au découpage en sections du programme. Ceci optimise l'allocation mémoire
- Puisque l'espace alloué est exactement l'espace nécessaire utilisé.

## Inconvénients

- L'allocation de la mémoire réelle est plus délicate puisqu'il faut allouer des blocs de taille variable.
- C'est pourquoi, on a souvent recours à une technique mixte exploitant les deux techniques à la fois.

# La technique mixte: Segmentation-Pagination

- L'espace d'adressage virtuel est découpé en blocs de taille fixe appelés segments ;
- Les segments sont découpés en pages ;
- L'espace d'adressage réel est découpé en pages
- On combine donc les avantages des deux techniques au prix d'une conversion d'adresse un peu plus complexe et de tables un peu plus nombreuses
- Ainsi , on aura des tables de pages associées à chaque segment

# La technique mixte: Segmentation-Pagination

- Toute adresse virtuelle est décomposable en trois champs : un numéro de segment, un numéro de page et un déplacement dans cette page
- La conversion d'une telle adresse en adresse réelle reposera sur les tables des segments qui mémorise pour tout numéro de segment virtuel valide, l'adresse de début et la longueur effective du segment alloué en mémoire réelle.

# La technique mixte: Segmentation-Pagination

## Avantages

- Il permet de concilier le respect de la structure logique du programme et l'allocation simple de la mémoire réelle

## Inconvénients

- Le mécanisme de segmentation-pagination est évidemment plus coûteux que la simple pagination au point de vue conversion d'adresse et espace mémoire