

Cours: Introduction aux systèmes d'exploitation



Pr. Hajar IGUER,
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 de la gestion du matériel
- Afin de gérer tous les composants de l'ordinateur, il fallait une interface ou une couche intermédiaire.



Définitions : Bases

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



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
 - Ainsi que les solutions adoptées dans les systèmes modernes (UNIX, LINUX et WINDOWS).



Organisation du cours

- 12 séances de 2 heures
- 2 Contrôles Continus
- 1 Examen Final
- 1 Note de Participation



PLAN



CADRE GÉNÉRAL



HISTORIQUE



COMPOSITION



CADRE GÉNÉRAL



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

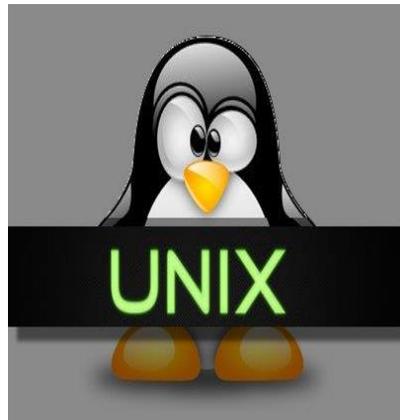
- Un système peut être défini comme un ensemble de programmes qui **contrôle** et **dirige** l'utilisation d'un ordinateur à travers différentes applications.
- En autres, c'est un système qui **exploite** les ressources matérielles d'un ordinateur et assure la liaison entre les utilisateurs et les applications.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



Différents Types de Systèmes D'exploitation

Exemples

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



MS-DOS

Microsoft®



Différents Types de Systèmes 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

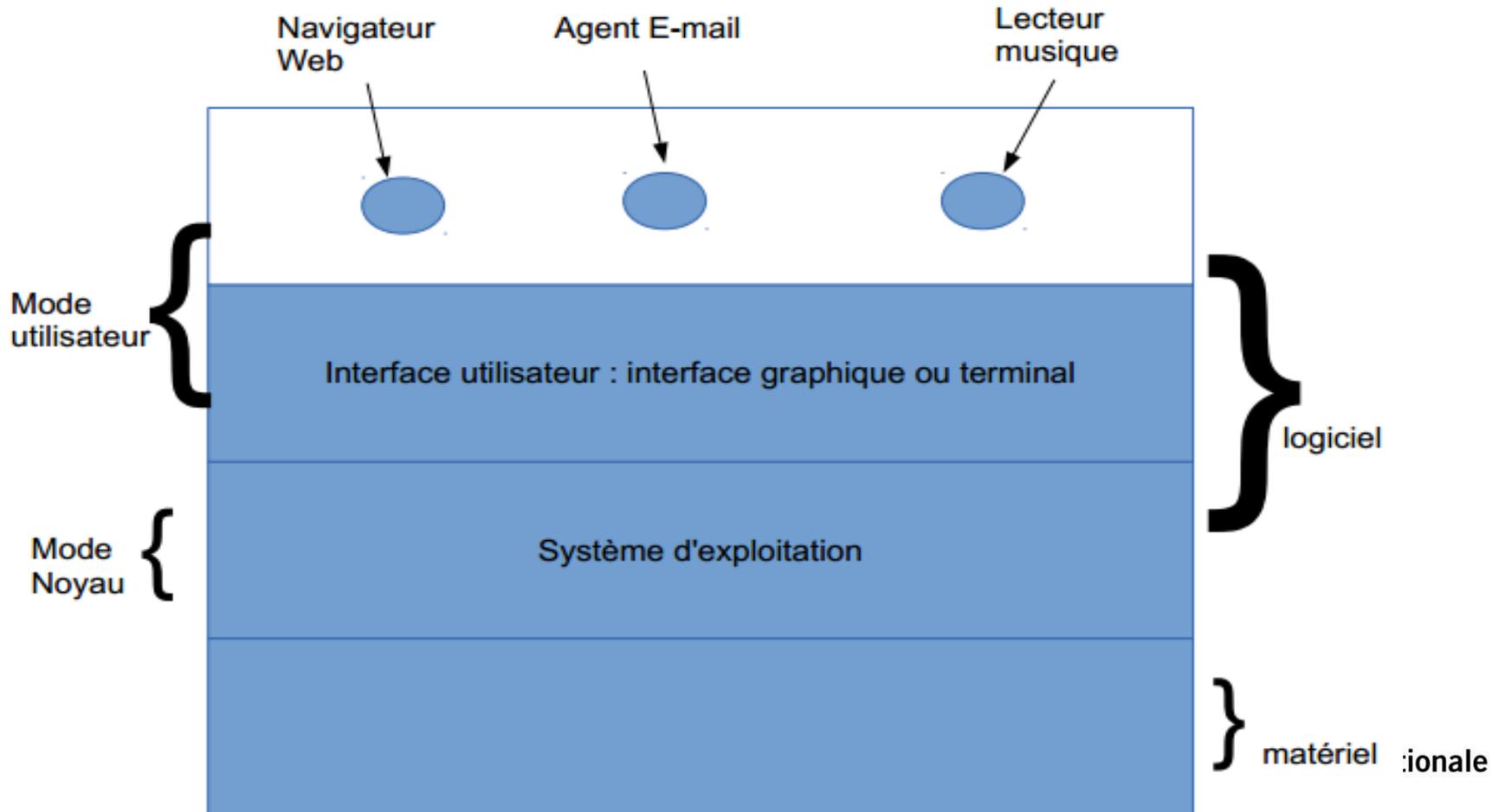


Définitions - Abstraction

- **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



Emplacement du système d'exploitation dans un ordinateur

- **Mode Noyau:**
 - Manière de fonctionnement du système d'exploitation
 - Inclus la gestion du matériel, et l'exécution de toutes les instructions comprises par le processeur
- **Mode Utilisateur:**
 - Correspond au mode de fonctionnement de tous les programmes
 - Par contre, aucun accès direct au matériel n'est autorisé



Emplacement du système d'exploitation dans un ordinateur

- Interface Utilisateur:

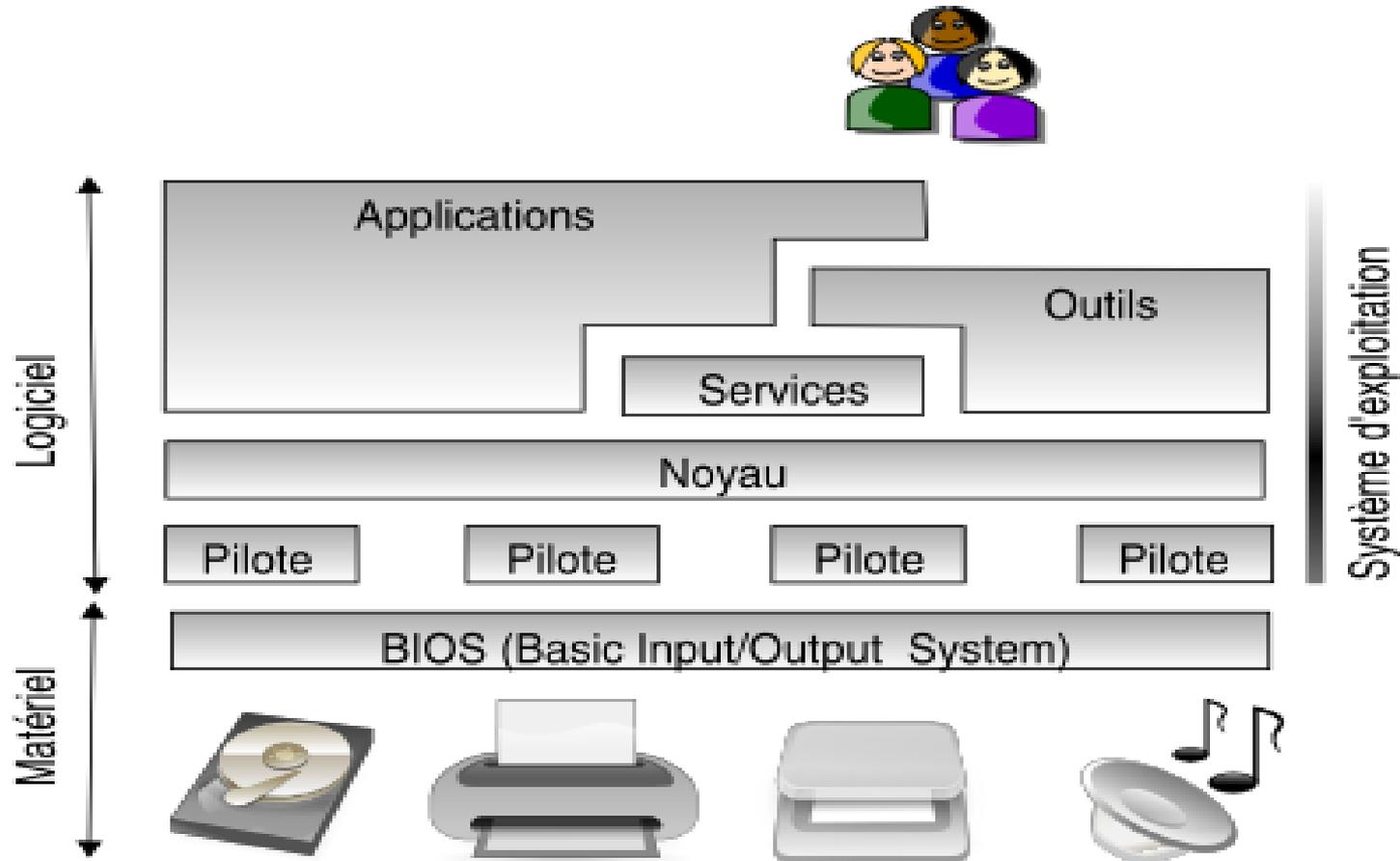
- Le niveau bas des programmes utilisateurs
- Utilisé pour démarrer d'autres programmes : navigateur
- Interface graphique ou textuelle

- OS:

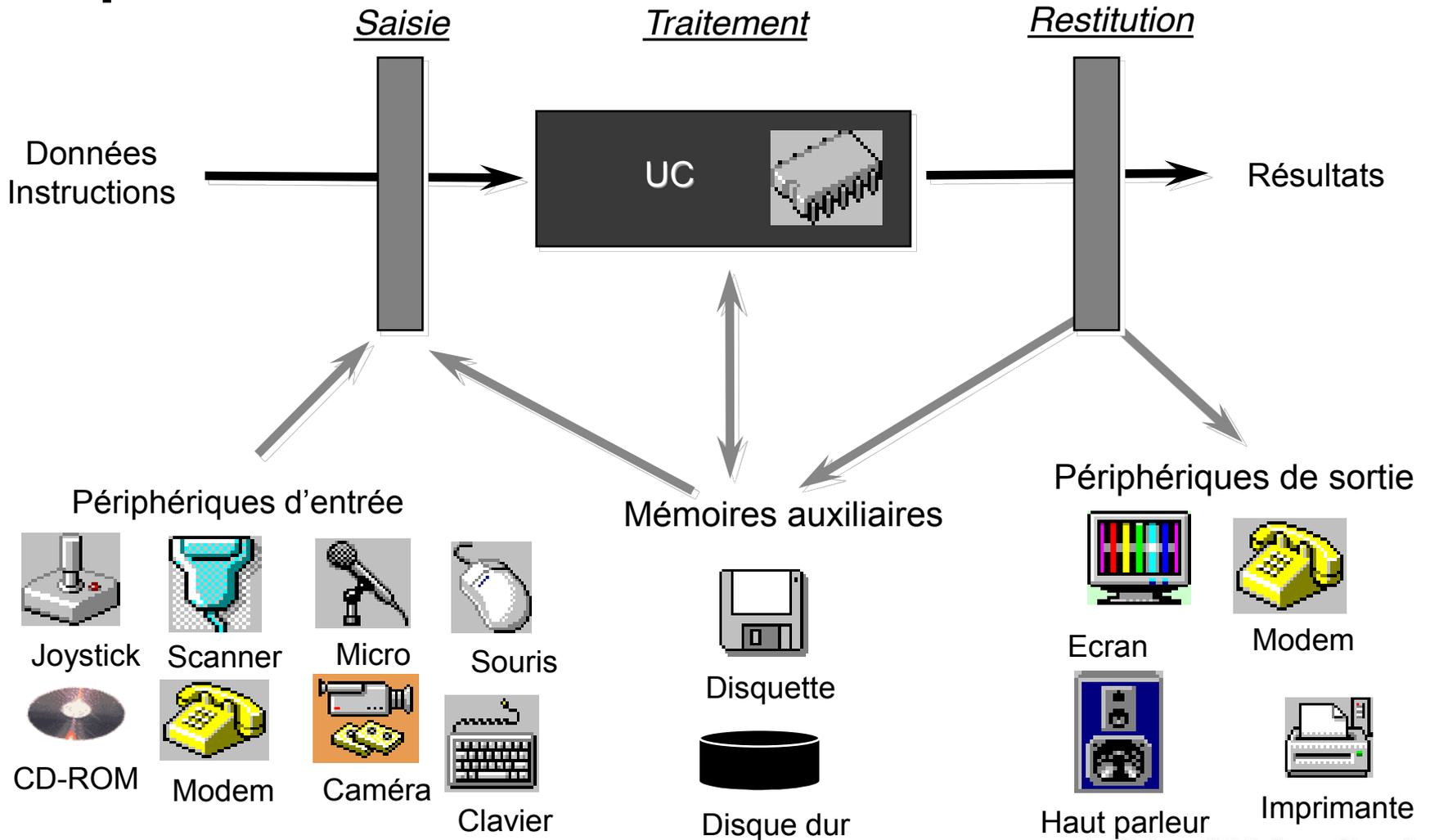
- Linux ou Windows: Plusieurs millions de codes
- Temps de vie supérieur au à celui des programmes



Illustration d'un système d'exploitation



Composants d'un ordinateur



HISTORIQUE : TYPOLOGIE



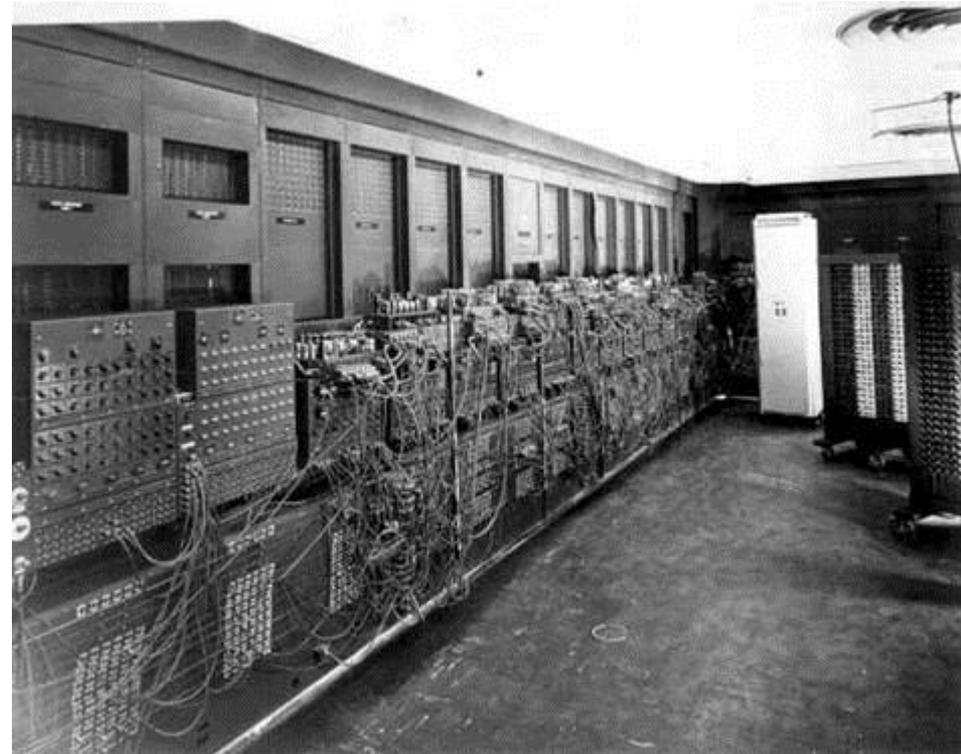
Typologie

- Il existe à ce jour quatre générations de d'ordinateurs
- Première Génération : Tubes à vide (Jusqu'à 1955)
- Deuxième Génération: Traitement par lots
- Troisième Génération : Circuits intégrés et Multiprogrammation
- Quatrième Génération: Ordinateurs personnels



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

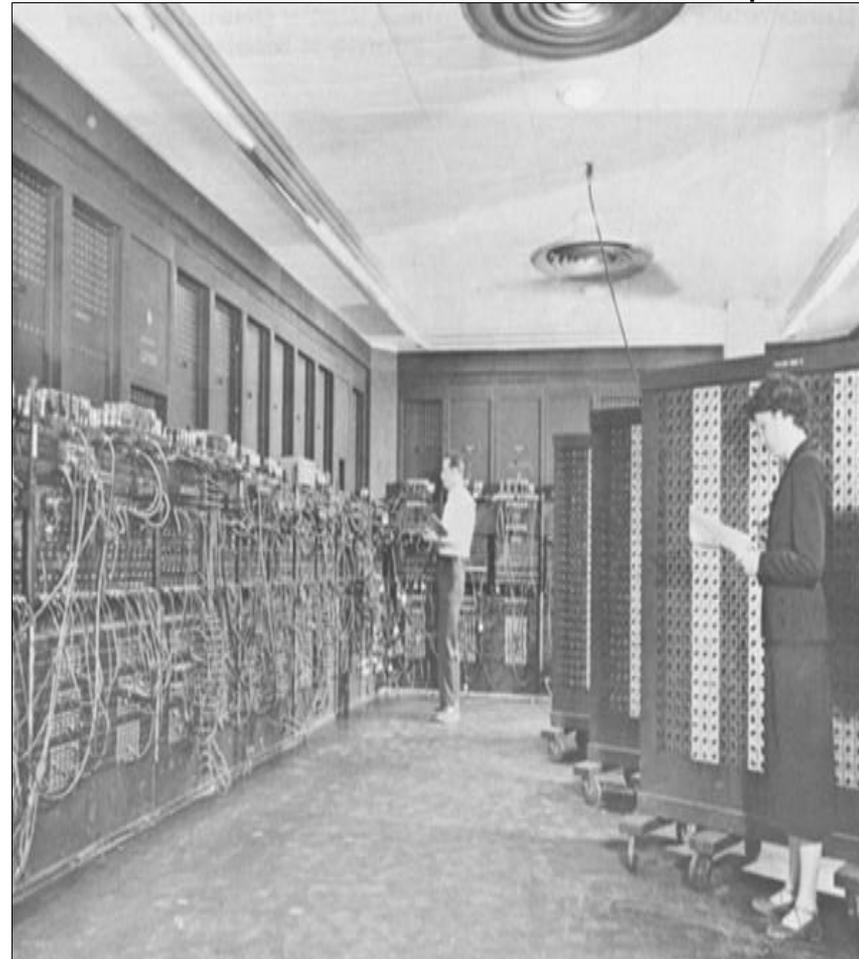
- Vers le milieu des années 1940, les premières machines à calculer sont construites au moyen de tubes électroniques (tubes à vides);
- Ces machines énormes remplissaient des salles et étaient moins rapides qu'une calculatrice moderne;
- Toute la préparation se faisait manuellement;
- Les E/S étaient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



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

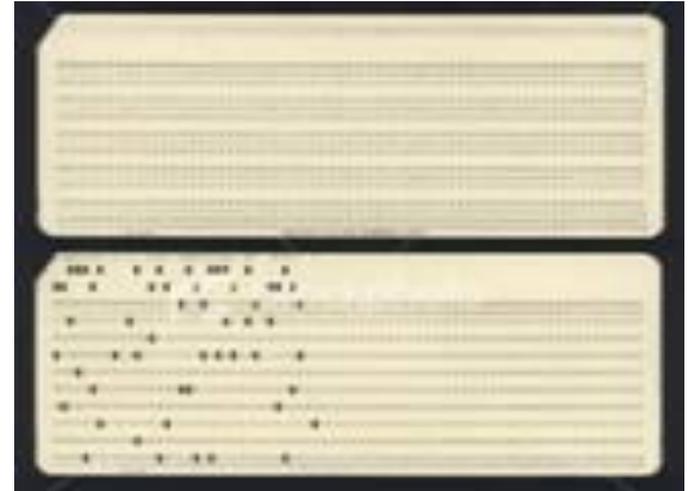
L'ENIAC 30 tonnes
Electronic Numerical Integrator
and Computer

- 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.



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 »);



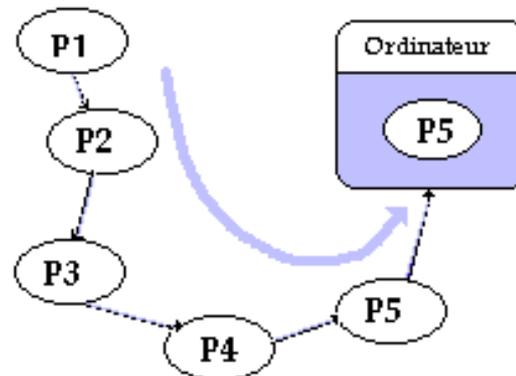
Deuxième Génération: Traitement par lots

- 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.



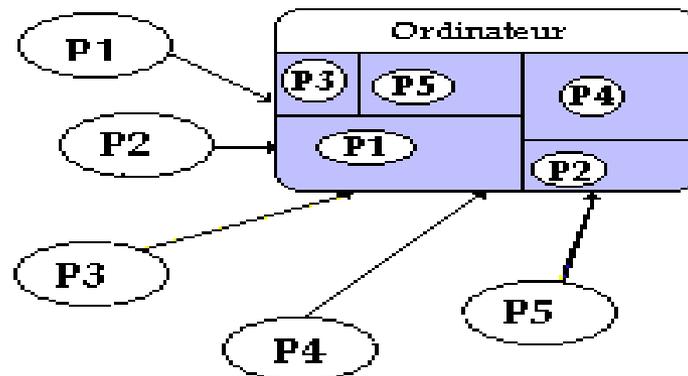
Troisième Génération : Circuits intégrés et Multiprogrammation (1965 – 1980)

- 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



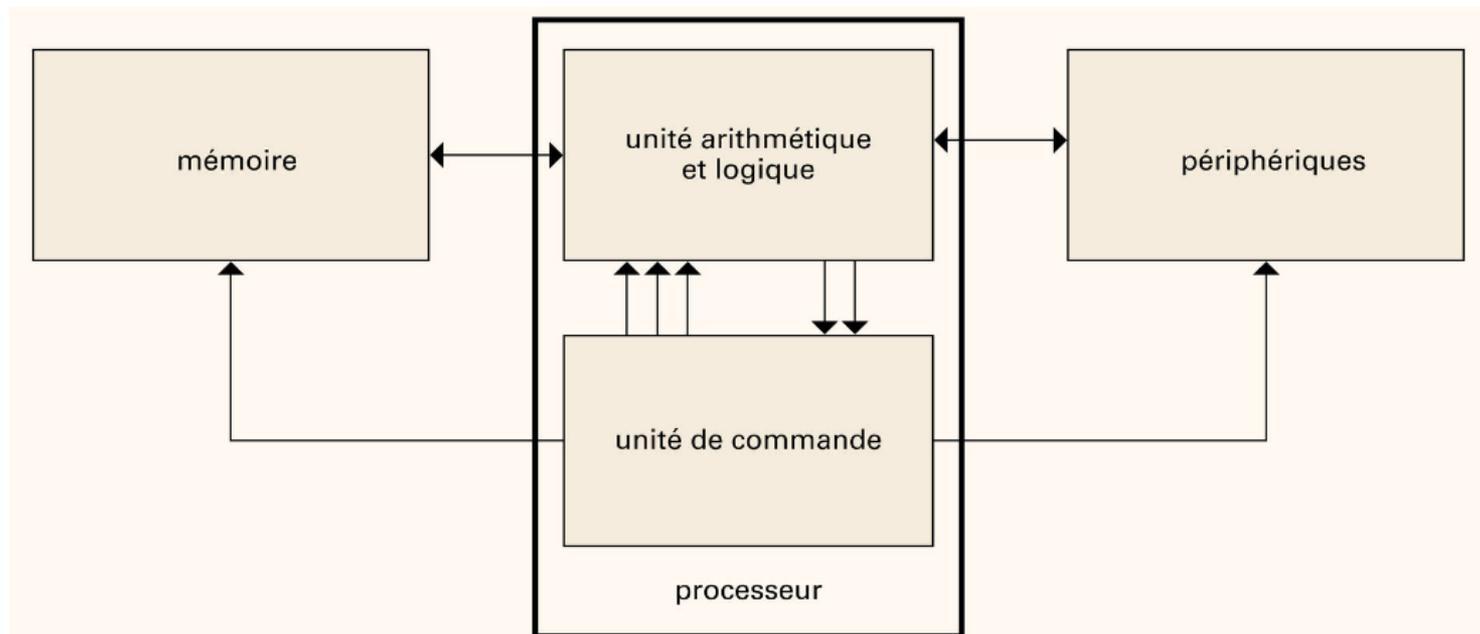
Quatrième Génération: Ordinateurs personnels (De 1980 - 2000)

- **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

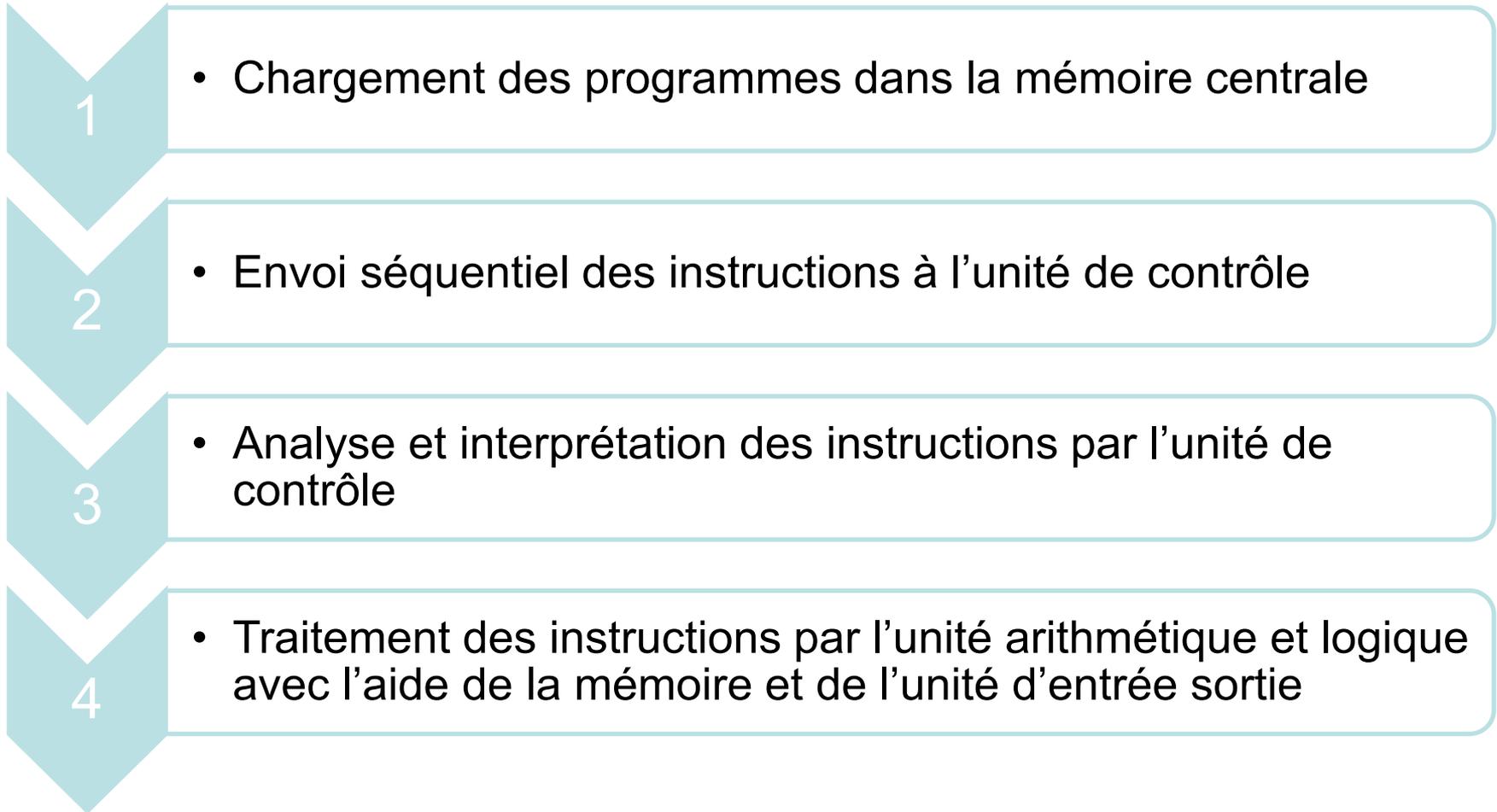


Architecture Von Neumann

- John von Neumann a proposé une structure universelle de machine à calculer et en a défini les constituants de base. La machine est composée des éléments suivants :



Etapes d'Exécution d'un programme



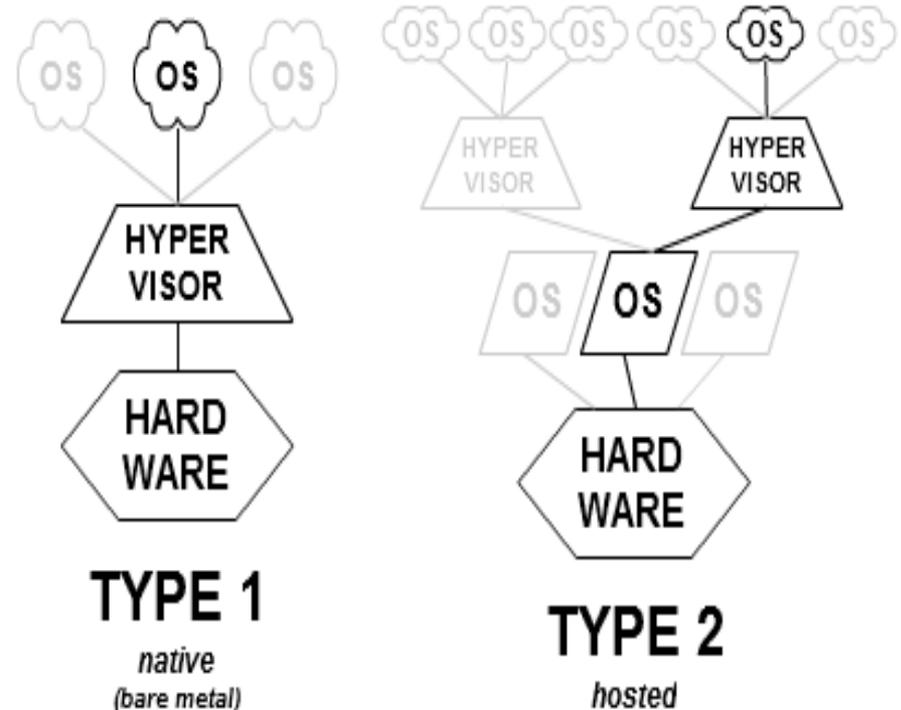
Hyperviseur

- 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

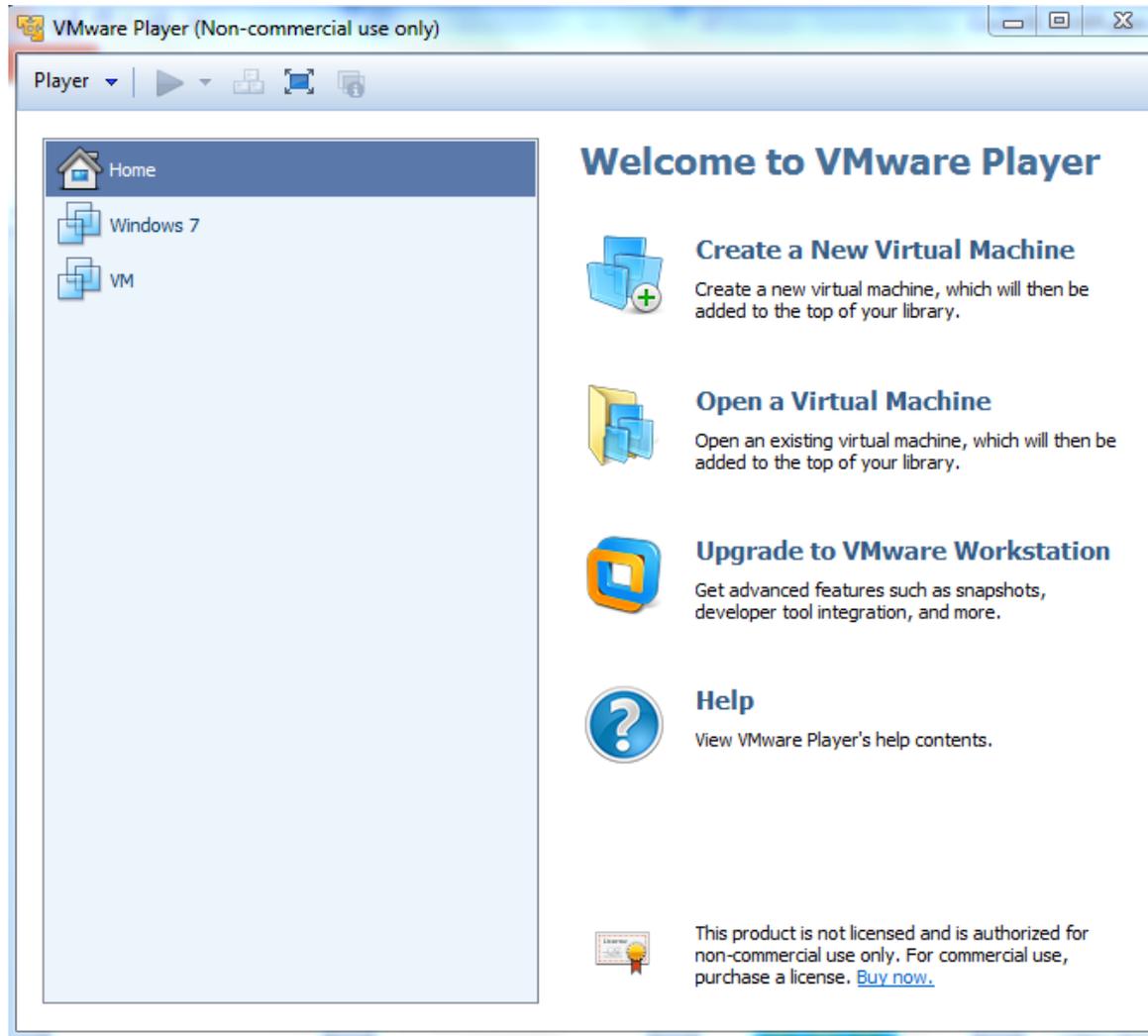
- **Type 1:** Nommé natif ou « bare-métal » est un logiciel qui se lance directement sur une plateforme matérielle. Exemple: Hyper-V de Microsoft, ESX Server de VMware,...
- **Type 2:** est un logiciel qui s'exécute directement dans un autre système d'exploitation. Exemple: Microsoft Virtual PC, VMware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



Exemples

Nous innovons pour votre réussite !

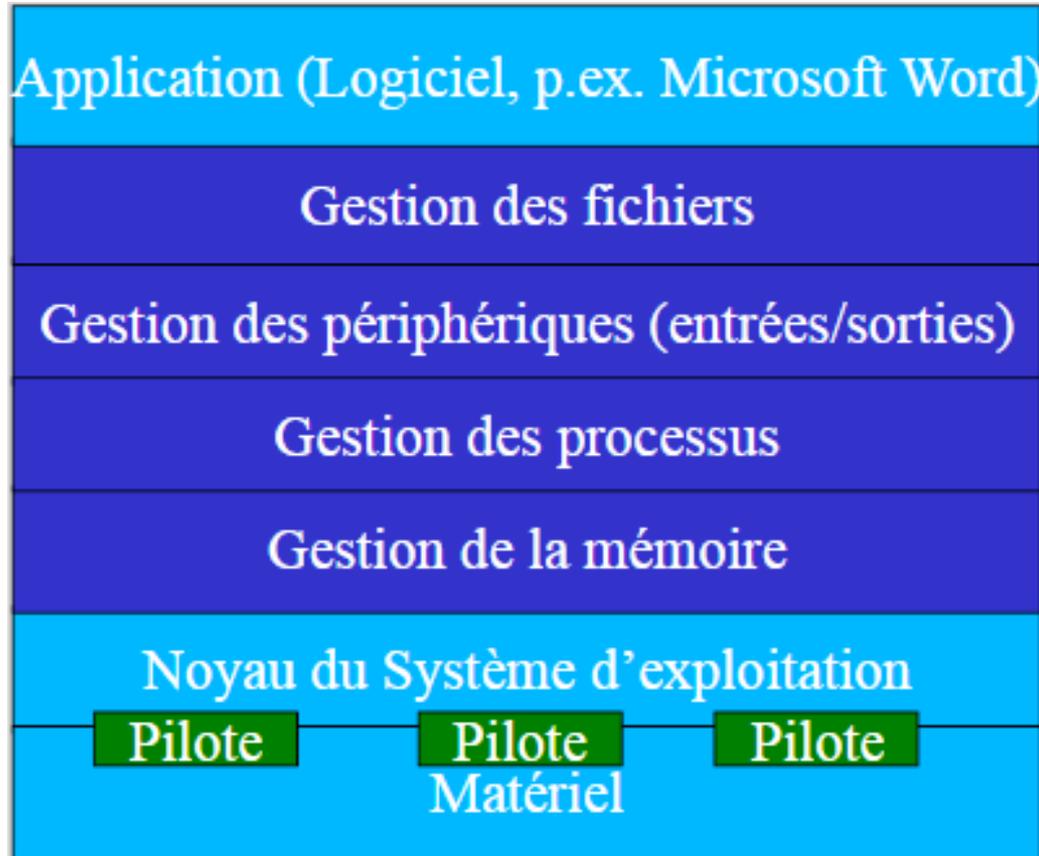
- Hyperviseur Type 2 : VMware Player



COMPOSITION



Rappel: SE – Modèle en Couches

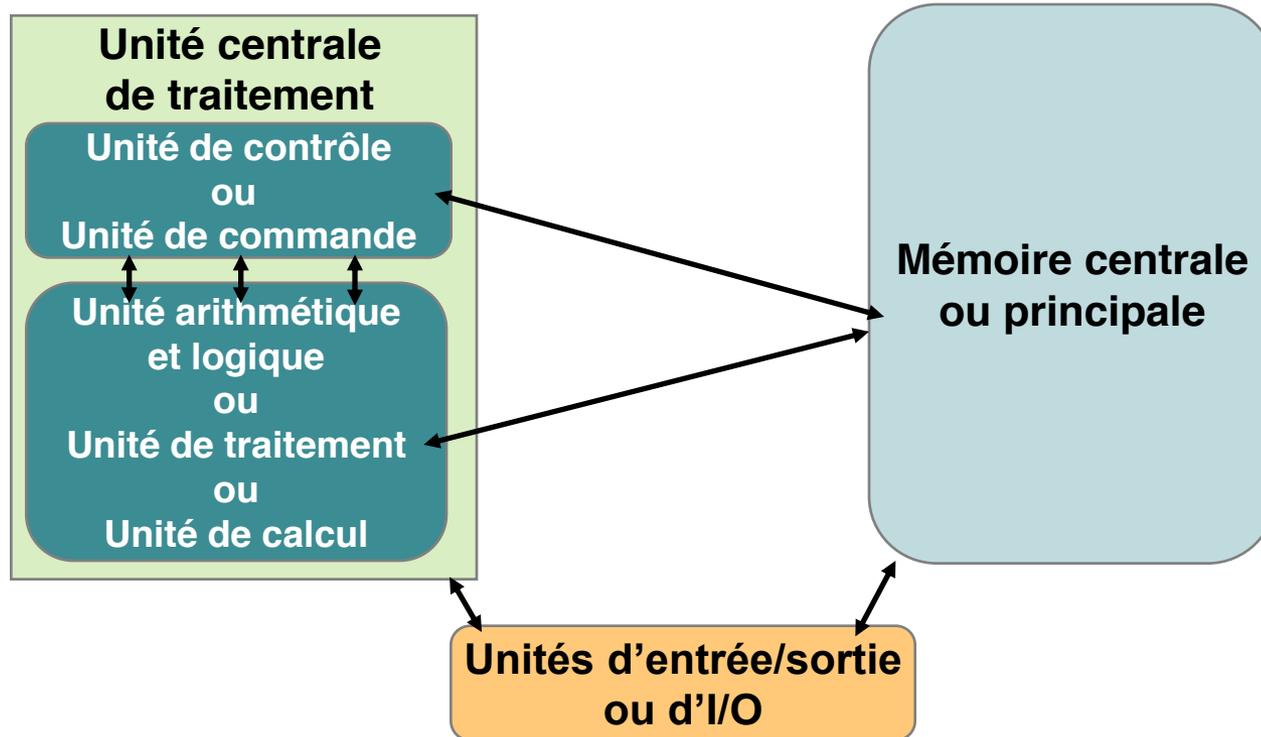


Le noyau du système d'exploitation

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

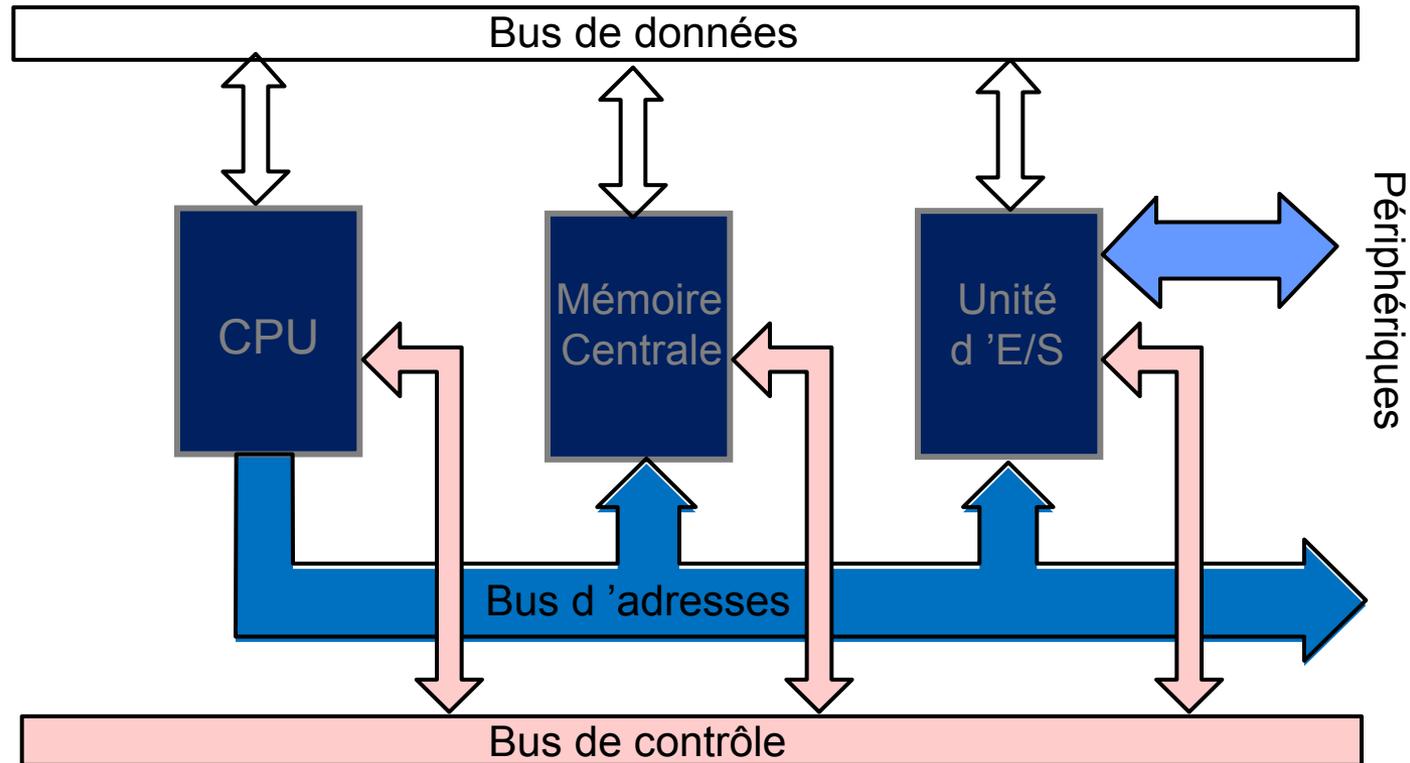


Principes de base



Principes de base

- Interconnexion par des **systèmes de câblage transportant des signaux électriques**.
- Pour éviter de relier chaque unité à chacune des autres, on fait usage de lignes exploitées en commun.
- **On appelle bus ces ensembles de lignes**



GESTION DES PROCESSUS



Le Processeur

- Le « cerveau » de l'ordinateur ou UC (Unité de Commande)
 - Extrait des instructions de la mémoire et les exécute
 - il traite et fait circuler les instructions et les données bruts
- Exemple de Produits
 - Intel (Pentium 166Mhz), Cyrix (P200+), AMD (K6200)



Le Processeur

Nous innovons pour votre réussite !

- ❖ Composé des éléments suivants:
 - **Unité Arithmétique et Logique (UAL)**
 - Ensemble de circuits qui exécutent les opérations arithmétiques et logiques de base
 - **Unité de contrôle (ou de commande)**
 - Son rôle est d'extraire une instruction du programme en MC, de la faire exécuter par l'UAL ou un périphérique et de chercher l'instruction suivante
 - Elle décode les instructions et trouve les données pour l'UAL
 - Reliée à l'horloge système, sa cadence de fonctionnement dépend de la fréquence



Le Processeur

■ L'horloge

- Elle contrôle et synchronise le microprocesseur et les composants associés
- Sa vitesse (fréquence) est exprimée en mégahertz (MHz) c'est-à-dire en million de cycles par seconde
- L'efficacité du microprocesseur est directement proportionnelle à la fréquence de l'horloge : une fréquence élevée est donc souhaitable

■ Exemples

- Intel Pentium 100Mhz
- Intel Pentium 200Mhz



Exemple:

Evolution des microprocesseurs Intel

Date	Nom	Nombre de transistors	Finesse de gravure (µm)	Fréquence de l'horloge	Largeur des données	MIPS
1971	4004	2 300		108 kHz	4 bits/4 bits bus	
1974	8080	6 000	6	2 MHz	8 bits/8 bits bus	0,64
1979	8088	29 000	3	5 MHz	16 bits/8 bits bus	0,33
1982	80286	134 000	1,5	6 à 16 MHz (20 MHz chez AMD)	16 bits/16 bits bus	1
1985	80386	275 000	1,5	16 à 40 MHz	32 bits/32 bits bus	5
1989	80486	1 200 000	1	16 à 100 MHz	32 bits/32 bits bus	20
1993	Pentium	3 100 000	0,8 à 0,28	60 à 233 MHz	32 bits/64 bits bus	100
1997	Pentium II	7 500 000	0,35 à 0,25	233 à 450 MHz	32 bits/64 bits bus	300
1999	Pentium III	9 500 000	0,25 à 0,13	450 à 1 400 MHz	32 bits/64 bits bus	510
2000	Pentium 4	42 000 000	0,18 à 0,065	1,3 à 3,8 GHz	32 bits/64 bits bus	1 700
2004	Pentium 4D « Prescott »	125 000 000	0,09 à 0,065	2,66 à 3,6 GHz	32 bits/64 bits bus	9 000
2006	Core 2™ Duo	291 000 000	0,065	2,4 GHz (E6600)	64 bits/64 bits bus	22 000
2007	Core 2™ Quad	2*291 000 000	0,065	3 GHz (Q6850)	64 bits/64 bits bus	2*22 000 (?)
2008	Core 2™ Duo (Penryn)	410 000 000	0,045	3,33 GHz (E8600)	64 bits/64 bits bus	~24 200
2008	Core 2™ Quad (Penryn)	2*410 000 000	0,045	3,2 GHz (QX9770)	64 bits/64 bits bus	~2*24 200
2008	Intel Core i7 (Nehalem)	731 000 000	0,045 (2008) 0,032 (2009)	2,66 GHz (Core i7 920) 3,33 GHz (Core i7 Ext. Ed. 975)	64 bits/64 bits bus	?
2009	Intel Core i5/i7 (Lynnfield)	774 000 000	0,045 (2009)	2,66 GHz (Core i5 750) 2,93 GHz (Core i7 870)	64 bits/64 bits bus	?
2010	Intel Core i7 (Gulftown)	1 170 000 000	0,032	3,33 GHz (Core i7 980X)	64 bits/64 bits bus	?

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.



Le Processus

Nous innovons pour votre réussite !

- Un processus est un concept clé dans tout système d'exploitation et peut être schématisé comme un programme en cours d'exécution
- Un processus contient toute l'information qu'il est nécessaire de conserver pour poursuivre l'exécution d'un programme interrompu.
- Exemple:
- Courrier électronique, la compression d'un fichier, le navigateur web



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
 - Sauvegarder 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

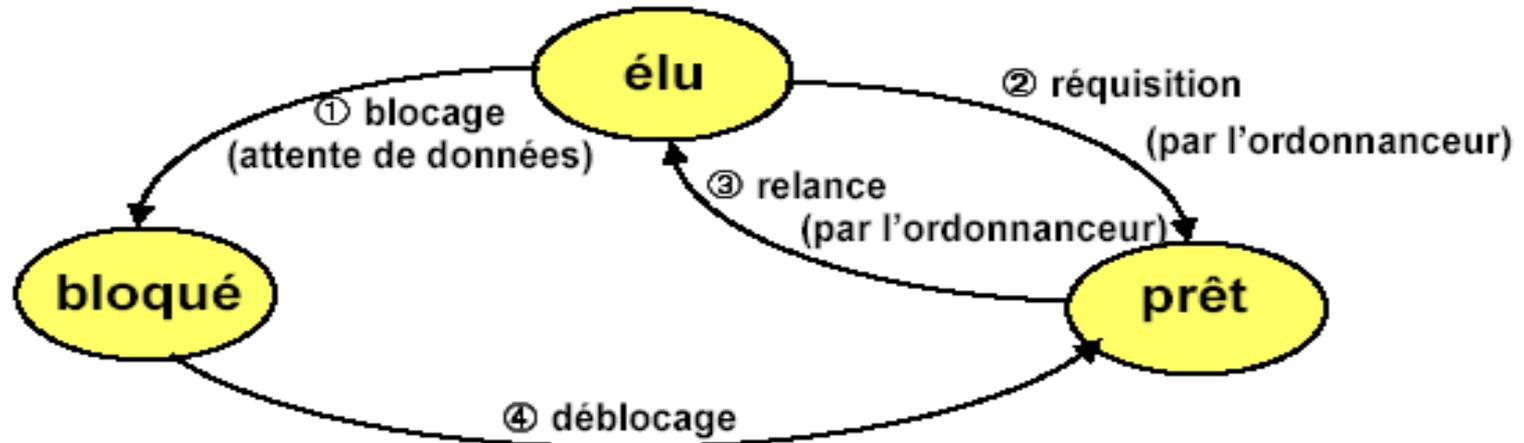


Etats d'un Processus

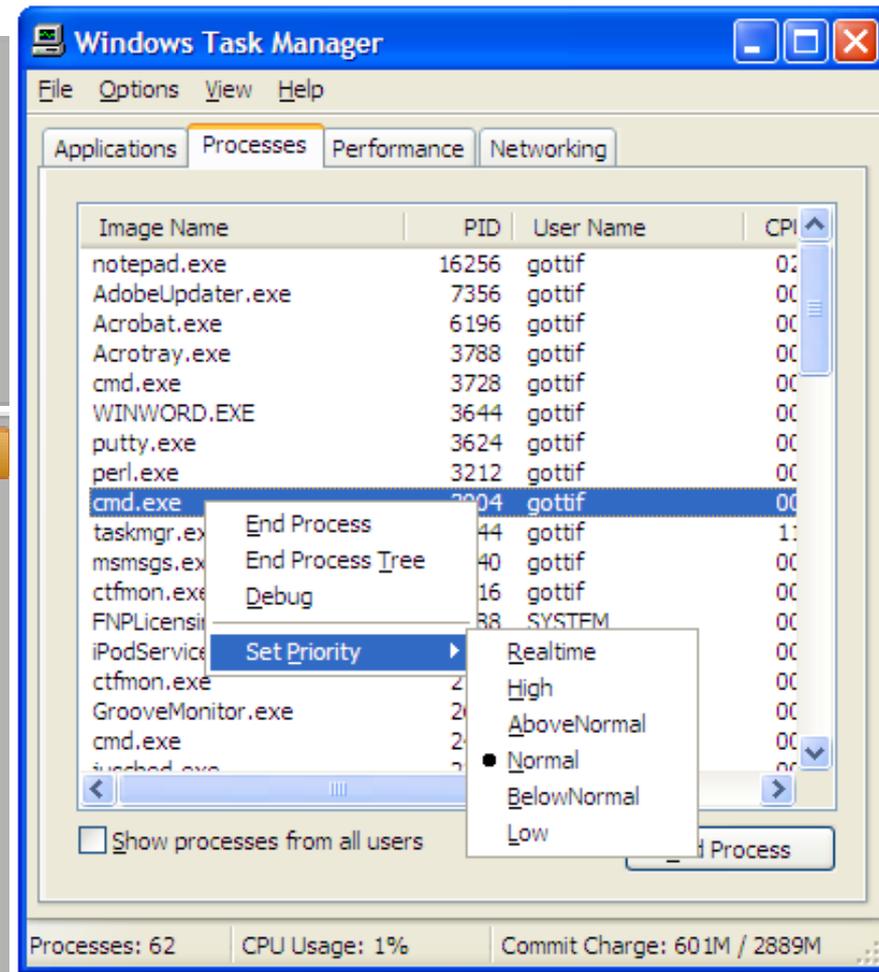
- Un processus peut connaître 3 états possibles :
 - **Élu** (Actif) - Processus actif, Processeur en marche
 - **Prêt** (Suspendu) – processus actif, processeur occupé par un autre processus
 - **Bloqué** (En attente d'un évènement extérieur) Processus non actif, même si le processeur est disponible.



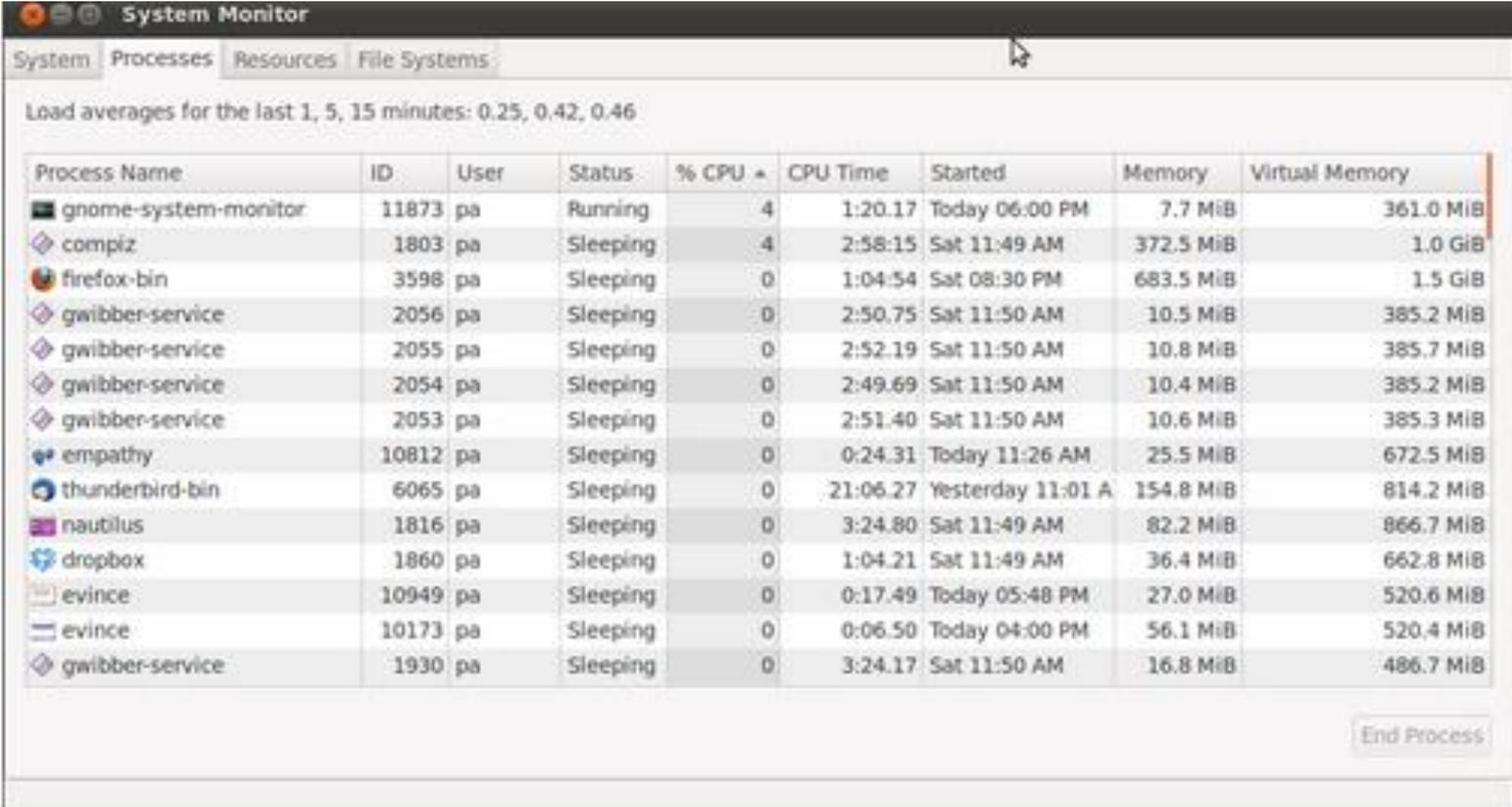
Etats d'un processus



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



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'ordonnancement**.
- Le type d'ordonnanceur est **totalelement 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.



Hiérarchie entre les processus

- Les SE qui font appel au concept de processus doivent permettre de créer et détruire dynamiquement les processus
- 2 exemples : Unix et MS-DOS
- Unix :
 - Les processus sont créés par l'appel système `fork`
 - Le `fork` crée une copie conforme du processus appelant
 - À la suite du `fork`, le processus père continue à s'exécuter en « parallèle » avec son fils
 - Le processus père peut créer d'autres fils et ces processus fils peuvent eux-mêmes avoir des fils → **Arborescence** de processus → hiérarchie
- MS-DOS :
 - Un appel système pour charger un fichier binaire en mémoire et l'exécuter en tant que processus fils
 - Contrairement à Unix, MS-DOS suspend le père jusqu'à ce que le fils ait terminé son exécution → **pas de pseudo-parallélisme**



Hiérarchisation des processus

- 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 sont appelés un **groupe de processus**



Informations sur un Processus

- Identificateur du processus (PID)
 - **numéro unique (à un instant t, mais réutilisable)**
- Informations de généalogie
 - **processus parent (PPID)**
 - **processus enfants**
- Information de droits
 - **utilisateur propriétaire du processus**
 - **utilisateur effectif (augmentation ou diminution de droits)**
- Information sur la mémoire utilisée
- Information sur le temps passé
- Information sur les fichiers ouverts (ressources)

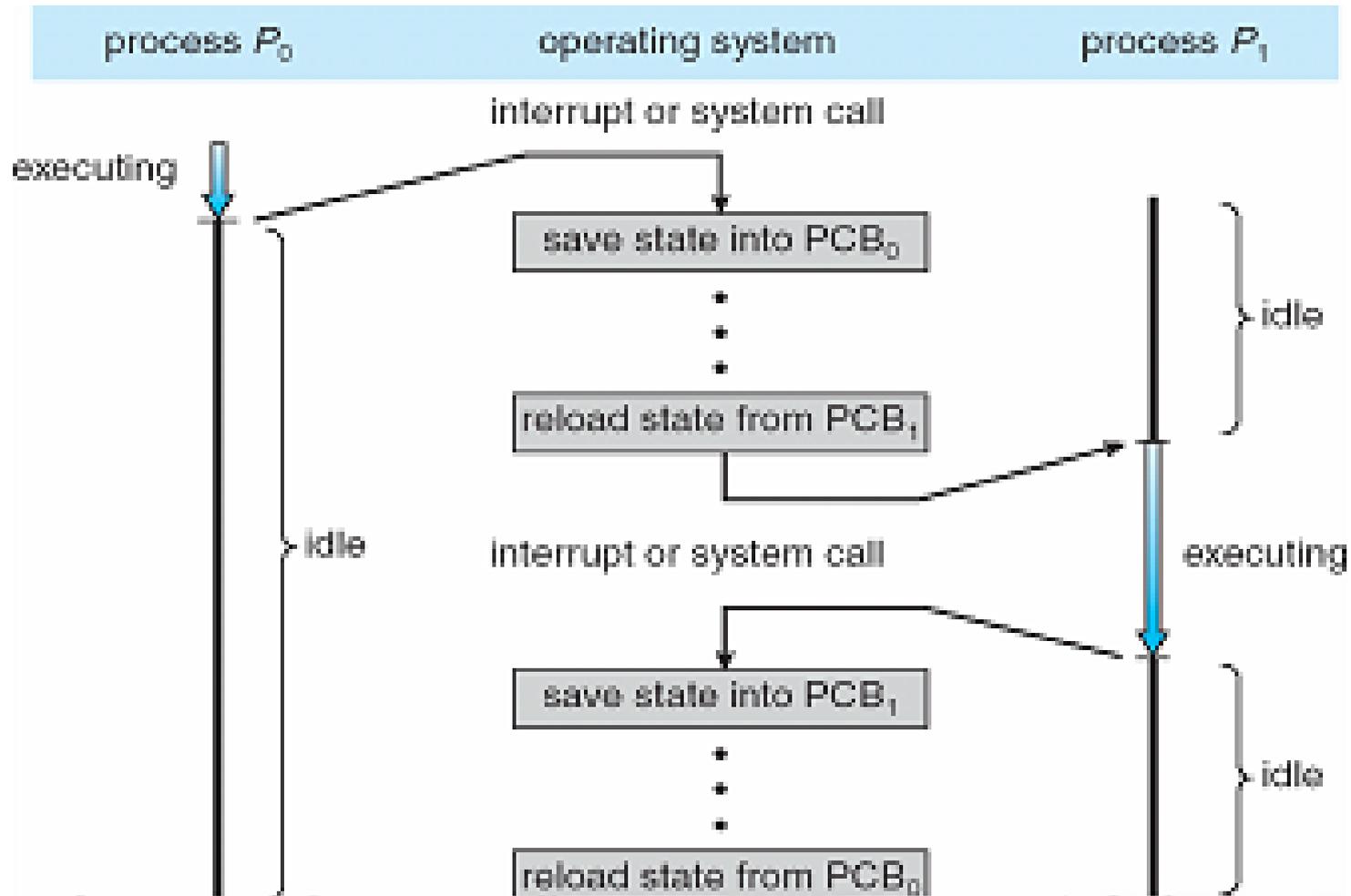


Process Control Block (PCB)

- Structure de données du SE contenant les informations sur un processus.
- Ces informations sont dans une zone mémoire accessible uniquement par le noyau du SE.
- Pour obtenir des informations sur les processus, il est donc nécessaire de passer par des appels systèmes sous UNIX:
- getpid, getppid, getuid, geteuid, setuid...



Process Control Block (PCB)



Les différents états d'un processus

- Les processus, bien qu'étant des entités indépendantes doivent parfois interagir avec d'autres processus (ex : `cat fic1 fic2 | grep mot`)
- L'ordre et la vitesse d'exécution des processus peut amener un des processus à se bloquer (ex : le `grep` est prêt à s'exécuter mais ne peut le faire faute de données)
- Un processus se **bloque** lorsqu'il ne peut pas, pour une raison logique, poursuivre son exécution (ex : attente de données)
- Un processus **élu** peut être arrêté et mis en situation d'attente (on dit qu'il est **prêt**), même s'il peut poursuivre son exécution (ex : le SE décide d'allouer le processeur à un autre processus)
- Ces 2 situations sont totalement différentes :
 - Dans le premier cas, la suspension est inhérente au problème
 - Dans le second cas, il s'agit d'une caractéristique technique du système

1. **Élu** (en cours d'exécution)

☀️ 3 états possibles : 2. **Prêt** (suspendu pour permettre l'exécution d'un autre processus)

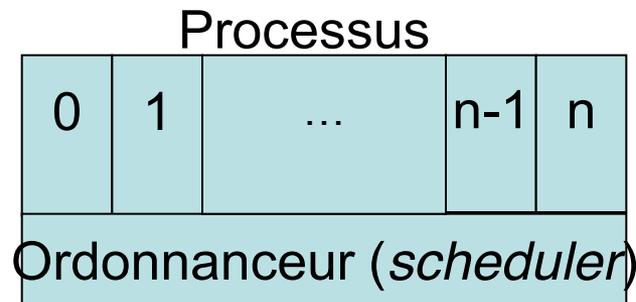
3. **Bloqué** (attendant un événement extérieur pour pouvoir continuer)



Le modèle des processus

- Le modèle des processus permet de mieux comprendre ce qui se passe à l'intérieur du SE :
 - Une partie des processus exécutent les programmes des utilisateurs
 - Les autres processus font partie du système et gèrent les tâches telles que les requêtes au gestionnaire de fichiers ou les opérations sur disques

🔦 Le nouveau modèle obtenu :



🔦 L'ordonnanceur constitue le niveau le + bas du SE

🔦 Il est surmonté d'une multitude de processus

🔦 La gestion des interruptions, la suspension et la relance des processus sont déportées dans l'ordonnanceur

🔦 Le reste du SE est structuré sous forme de processus



La réalisation des processus

- Pour mettre en œuvre pratiquement le modèle de processus, le SE a une table (tableau de structures) : la **table des processus** (TP) dont chaque entrée correspond à un processus particulier
- Chaque entrée comporte des informations sur :
 - l'état du processus
 - son compteur ordinal
 - son pointeur de pile
 - son allocation mémoire
 - l'état des fichiers ouverts
 - ... et tout ce qui doit être sauvé lorsqu'un processus passe de l'état **élu** à l'état **prêt**
- Le contenu de la table des processus varie d'un système à un autre
- En général, les informations concernent la :
 - La gestion du processus
 - La gestion mémoire
 - La gestion des fichiers



La table des processus Unix

Quelques champs typiques de la table des processus:

Gestion des processus	Gestion de la mémoire	Gestion des fichiers
Registres, compteur ordinal	Pointeur sur le segment de code	Masque <code>UMASK</code>
Mot d'état du programme	Pointeur sur segment de données	Répertoire racine
Pointeur de pile	Pointeur sur segment BSS	Répertoire de travail
État du processus	Statut de fin d'exécution	Descripteurs de fichiers
Date de lancement du processus	Statut de signal	<code>uid</code> effectif
Temps CPU utilisé	Identificateur de processus (<code>pid</code>)	<code>gid</code> effectif
Temps CPU des fils	Processus père	Paramètres des appels systèmes
Date de la prochaine alarme	Groupe du processus	Divers indicateurs
Pointeurs sur files de message	<code>uid</code> réel, <code>uid</code> effectif	
Bits des signaux en attente	<code>gid</code> réel, <code>gid</code> effectif	
Identificateur de processus (<code>pid</code>)	Table de bits des signaux	
Divers indicateurs	Divers indicateurs	



Gestion des processus - Ordonnancement

- **Un bon algorithme d'ordonnancement doit :**
 - ✓ S'assurer que chaque processus reçoit sa part de temps CPU
 - ✓ Utiliser le CPU à 100%
 - ✓ Minimiser le temps de réponse pour les utilisateurs en mode interactif
 - ✓ Minimiser l'attente des utilisateurs qui travaillent en traitement par lots
 - ✓ Maximiser le nombre de travaux effectués en une heure



Ordonnancement

- Le système doit faire un choix selon les critères suivants :
 - 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 - Ordonnancement

- Pour assurer qu'aucun processus ne s'exécute pendant trop de temps, pratiquement tous les ordinateurs ont un **compteur de temps** (*timer*) ou une **horloge** qui génère périodiquement une interruption (un signal).
- A chaque interruption le système reprend la main et décide si le processus courant doit poursuivre son exécution ou s'il a épuisé le temps CPU qui lui était imparti, le processeur est alors alloué à un autre processus.



Gestion des processus - Ordonnancement

- Cette stratégie qui permet de suspendre des processus prêts est appelée **ordonnancement avec réquisition** (*preemptive scheduling*) et s'oppose à la méthode **d'exécution jusqu'à achèvement** (**ordonnancement sans réquisition**) des premiers systèmes à traitements par lots
- Les algorithmes d'ordonnancement sans réquisition sont simples et faciles à mettre en œuvre mais inadaptés à des systèmes généraux à plusieurs utilisateurs



Ordonnancement

Nous innovons pour votre réussite !

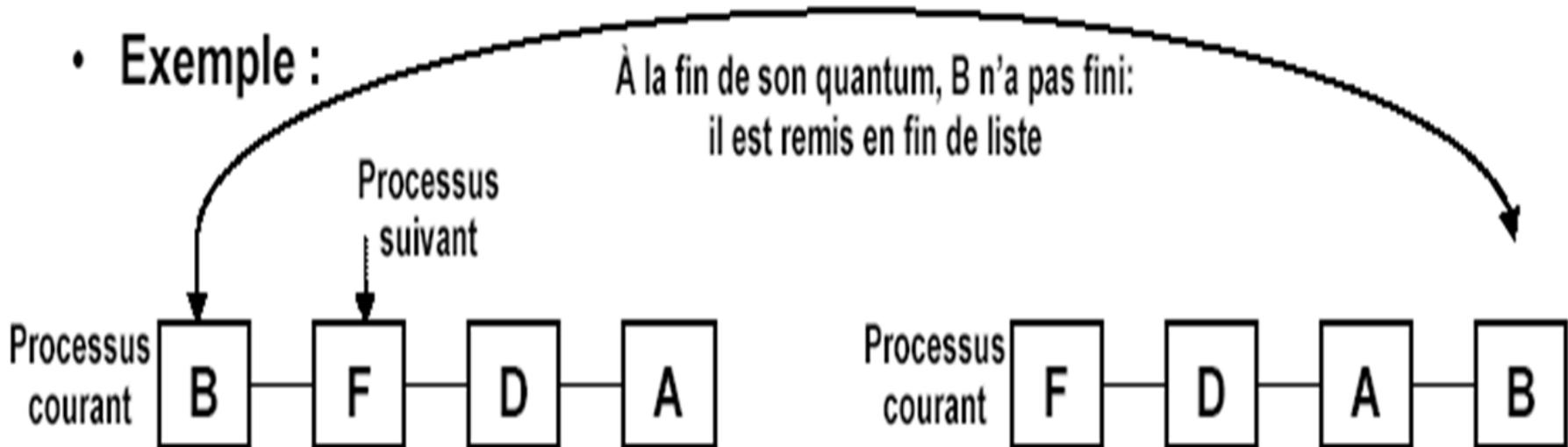
- **Ordonnancement sans réquisition :**
 - Un processus est exécuté jusqu'à la fin.
- **Ordonnancement avec réquisition :**
 - A chaque signal d'horloge, le SE reprend la main, décide si le processus en cours a consommé son quota de temps et alloue éventuellement le processeur à un autre processus.
 - Il existe de nombreux algorithmes d'ordonnancement.



Ordonnancement Circulaire (Tourniquet)

- Chaque processus possède un quantum d'exécution:
 - Si le processus a fini dans cet intervalle, on passe au suivant.
 - S'il n'a pas fini, il passe en fin de liste et l'on exécute le suivant.

- Exemple :



Ordonnancement Circulaire

- Un des algos les plus anciens, les plus simples, les plus fiables et les plus utilisés (**tourniquet** ou *round robin*)
- Chaque processus possède un intervalle de temps, son **quantum**, durant lequel il est autorisé à s'exécuter
- Si le processus se bloque ou se termine avant la fin de son quantum, le CPU est immédiatement alloué à un autre processus
- L'ordonnanceur doit mémoriser une liste de processus prêts , lorsque que le quantum est épuisé le processus est mis en fin de liste



Ordonnancement Circulaire

- Problème : Réglage du Quantum :
 - Quantum trop petit / Commutation :
Le processeur passe son temps à commuter.
 - Quantum trop grand
Augmentation du temps de réponse d'une commande
- Le seul problème est la durée du quantum !
- La **commutation de processus** (ou **commutation de contexte**) requiert un certain tps de gestion (sauvegarde et restitution des registres, ...)
- Durée de commutation : 5 ms et quantum : 20 ms →
20% du temps CPU gâché pour la gestion :
overhead
- Solution : augmenter le quantum à 500 ms (→ 1% pour gérer)



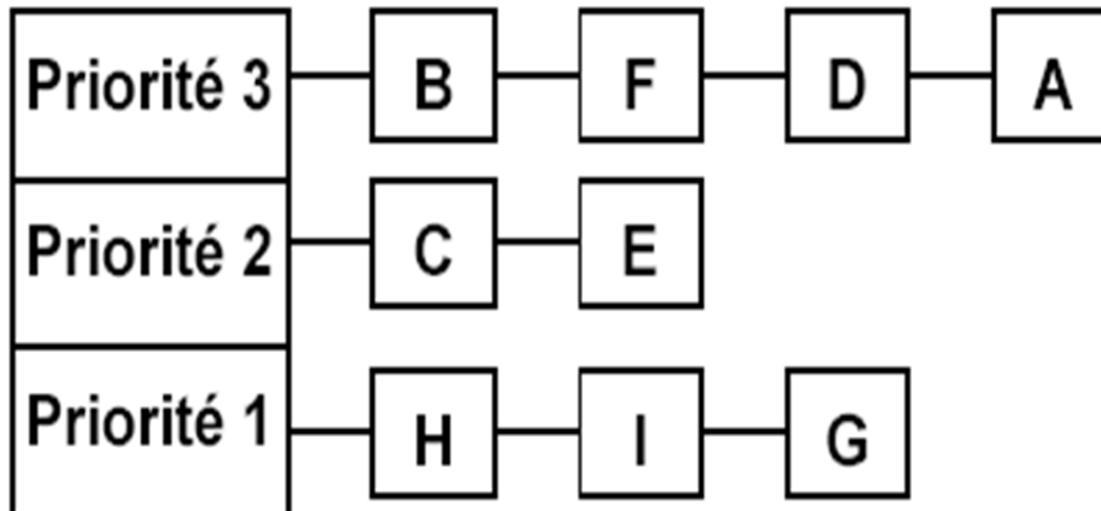
Ordonnancement Circulaire

- PB : 10 utilisateurs interactifs lancent leur programme presque en même temps : le premier est lancé immédiatement, le second devra attendre $\frac{1}{2}$ seconde et ainsi de suite. Le dernier attendra pendant 5 secondes !
- Conclusion : un quantum trop petit provoque trop de commutations de processus et abaisse l'efficacité du processeur, alors qu'un quantum trop élevé augmente les temps de réponse de courtes commandes en mode interactif
- Un quantum de 100 ms est un compromis raisonnable



Ordonnancement avec priorité

- Plusieurs files d'attente avec un niveau de priorité différent
- La priorité d'un processus décroît au cours du temps pour ne pas bloquer les autres files



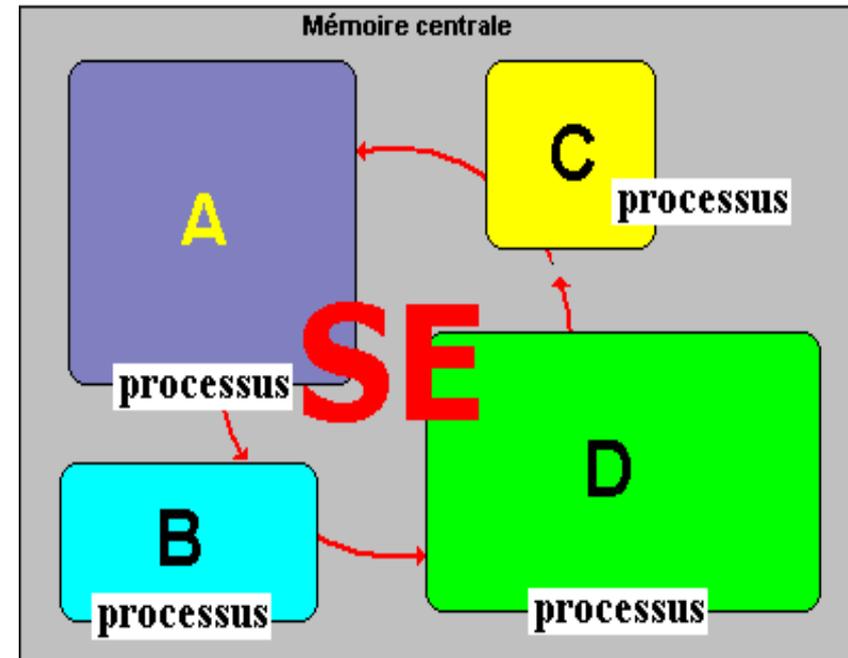
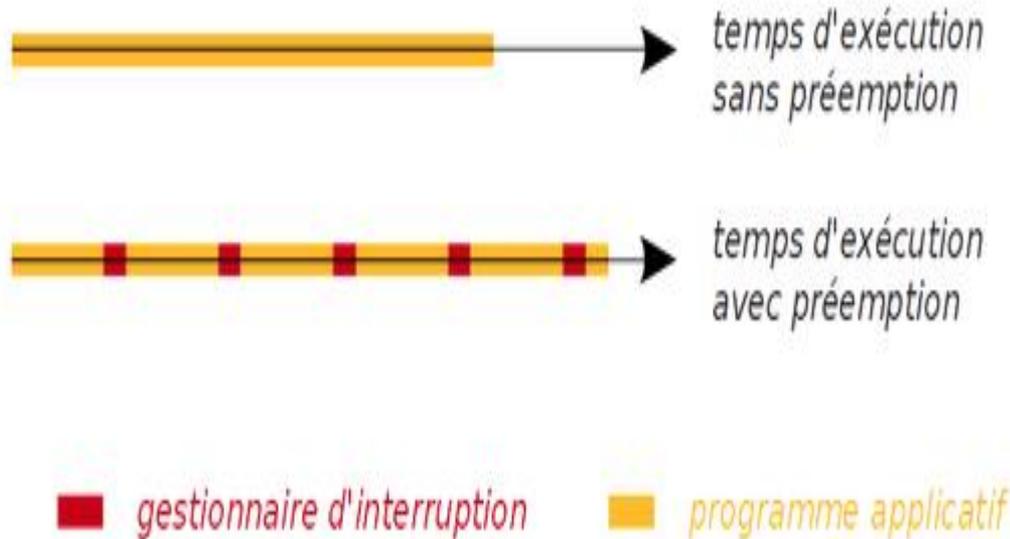
Ordonnancement - Autres algorithmes

- Ordonnancement « plus court d'abord » :
 - Estimation de la durée de chaque processus en attente
 - Exécuter le processus le plus court
- Ordonnancement garanti :
 - Si n utilisateurs connectés, chacun reçoit $1/n$ du temps processeur.



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;



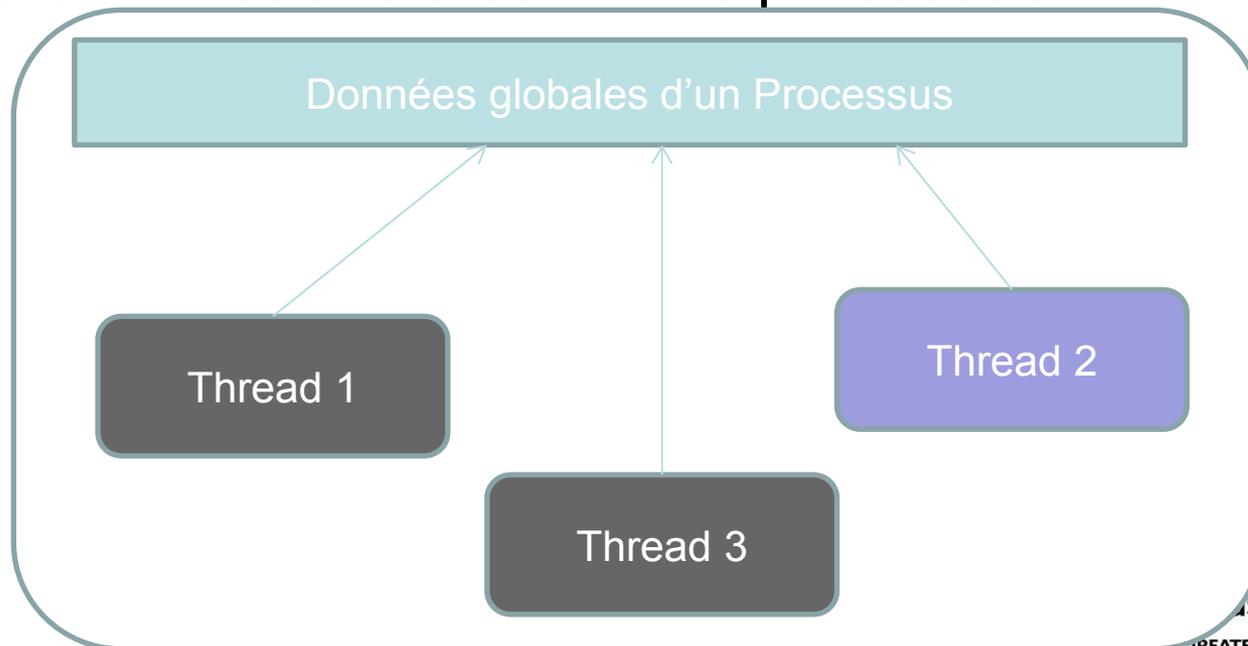
Thread

- 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.

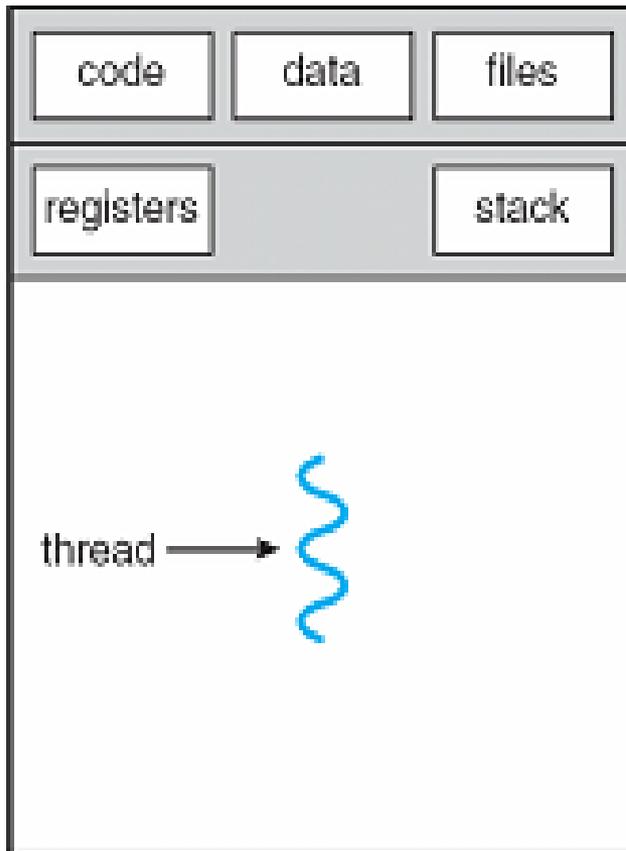


Thread

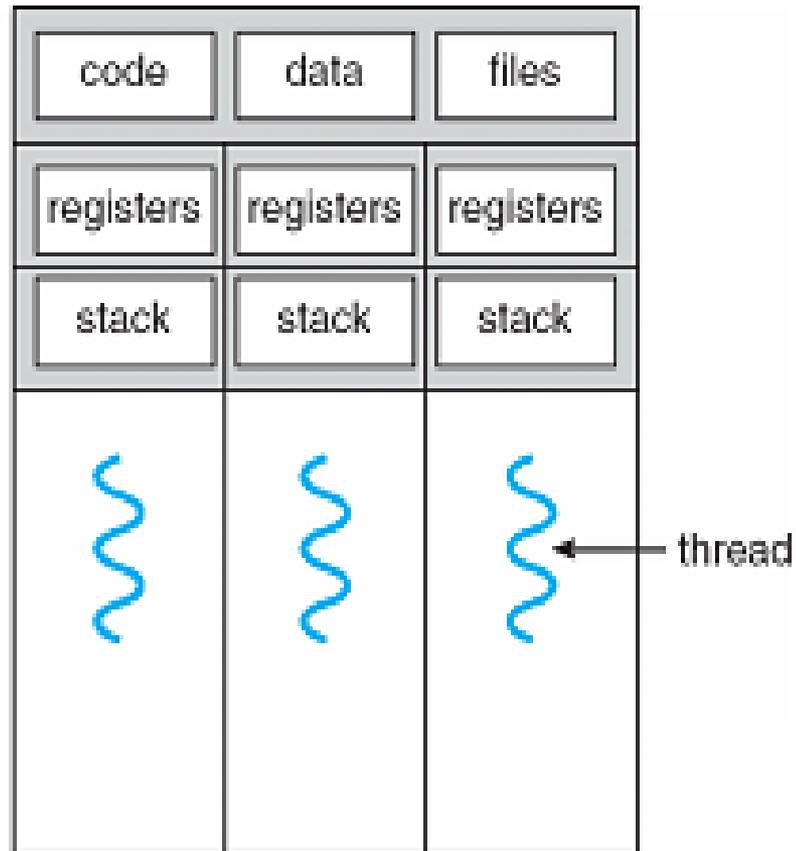
- Un processus travaille et gère, pendant le quantum de temps qui lui est alloué, des ressources et exécute des actions sur et avec ces ressources.
- Les threads situés dans un même processus partagent les mêmes variables générales de données et les autres ressources allouées à ce même processus.



Thread vs Multithreading



single-threaded process



multithreaded process

Systemes multi-utilisateurs

- Permettre à différents utilisateurs d'utiliser les ressources d'un même ordinateur en même temps
 - À distance avec un serveur (ftp, telnet, ssh , ...
 - À travers de le terminal de l'ordinateur
- La nécessité d'introduire la gestion des droits pour éviter la suppression de fichiers ou l'arrêt de processus

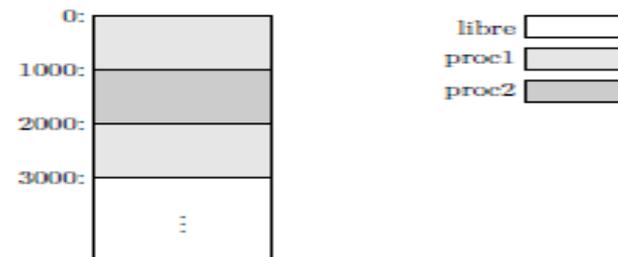


GESTION DE LA MÉMOIRE



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



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.



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)**



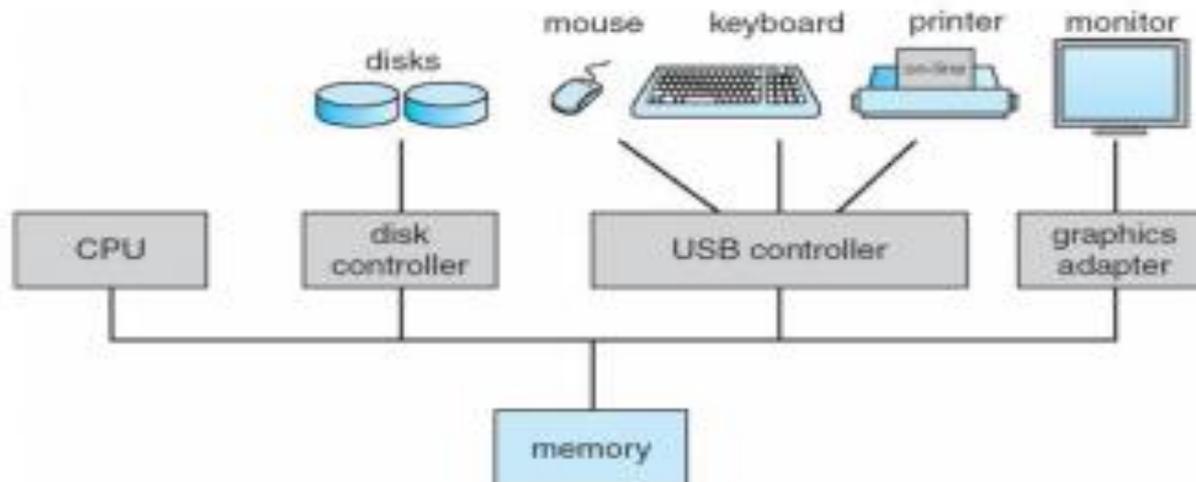
Propriétés du MMU

- 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
- Allocation d'espace mémoire
- Mémoire virtuelle



Rôle du gestionnaire de mémoire

- Conserver la trace de la mémoire en cours d'utilisation
- Allouer la mémoire aux processus qui en ont besoin
- Gérer le va-et-vient (swapping) entre mémoire principale et disque



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,



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 d'extension et de découpage



Extension de la mémoire

- Afin d'étendre l'utilisation des mémoires, il s'avère intéressant d'utiliser des techniques d'extension de mémoire dont:
 - La technique qui consiste à stocker une partie permanente dans RAM et à stocker une autre partie dans le disque dur et y accéder au besoin,
 - La mémoire virtuelle



La Mémoire virtuelle

- Elle consiste à utiliser le disque dur comme mémoire principale et à stocker uniquement dans la RAM, les instructions et les données utilisées par le processeur.
- En réalisant cette opération, le SE crée un fichier temporaire (appelé fichier **SWAP**, traduisez "**fichier d'échange**") dans lequel sont stockées les informations lorsque la quantité de mémoire vive n'est plus suffisante.
- Cela se traduit par une baisse considérable des performances, étant donné que le temps d'accès du disque dur est extrêmement plus faible que celui de la RAM.



La Mémoire virtuelle

- Cette méthode est utilisée pour :
 - Une mémoire insuffisante pour contenir tous les processus courants
 - La nécessité de placer certains de ces processus sur le disque.
- Le swapping peut être défini comme l'action de ramener régulièrement des processus sur le disque en mémoire centrale et inversement
- La mémoire virtuelle permet :
 - D'augmenter le taux de multiprogrammation
 - De mettre en place des mécanismes de protection de la mémoire
 - De partager la mémoire entre processus



Découpage de la mémoire

- L'organisation ou le découpage de mémoire peut se faire de différentes manières:
 - Pagination
 - Segmentation
 - Technique mixte



La Pagination

- Cela consiste à découper la mémoire en blocs et les programmes en pages de longueur fixe 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 limités 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 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



Protection de la mémoire

- La protection de la mémoire est un service essentiel des systèmes d'exploitation.
- Il n'est pas acceptable aujourd'hui qu'une erreur dans un programme d'édition d'images, provoque l'arrêt d'une machine.
- De façon générale, il n'est pas tolérable que le fonctionnement d'une machine (quelle que soit son utilisation) soit mis en cause par des erreurs logicielles.

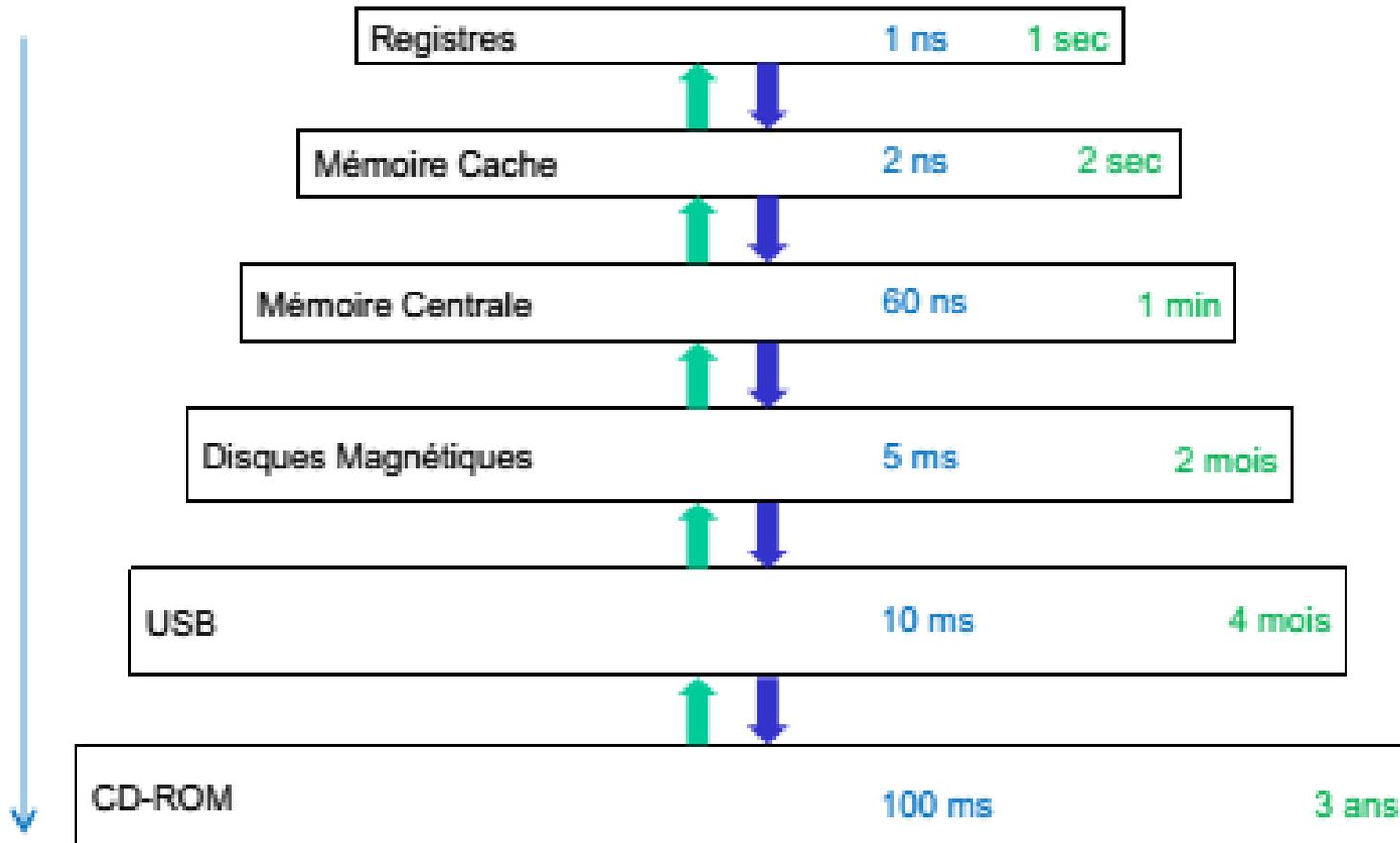


Protection de la mémoire

- Le principe c'est d'empêcher un processus B d'écrire dans une zone mémoire déjà attribuée à un autre processus A et ainsi perturber son exécution.
- Le SE doit assigner ce rôle au MMU pour qu'il ne donne accès au processus qu'aux pages qui lui sont attribuées.
- En cas de tentative, un signal est envoyé au processus fautif . Ce dernier doit demander au SE d'accéder aux données d'un autre processus tant que les opérations demandées sont permises.



Temps d'accès



La technique mixte: Segmentation-Pagination

- L'espace d'adressage virtuel est découpé en blocs 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



Protection de la mémoire

- La protection de la mémoire est un service essentiel des systèmes d'exploitation.
- Il n'est pas acceptable aujourd'hui qu'une erreur dans un programme d'édition d'images, provoque l'arrêt d'une machine.
- De façon générale, il n'est pas tolérable que le fonctionnement d'une machine (quelle que soit son utilisation) soit mis en cause par des erreurs logicielles.

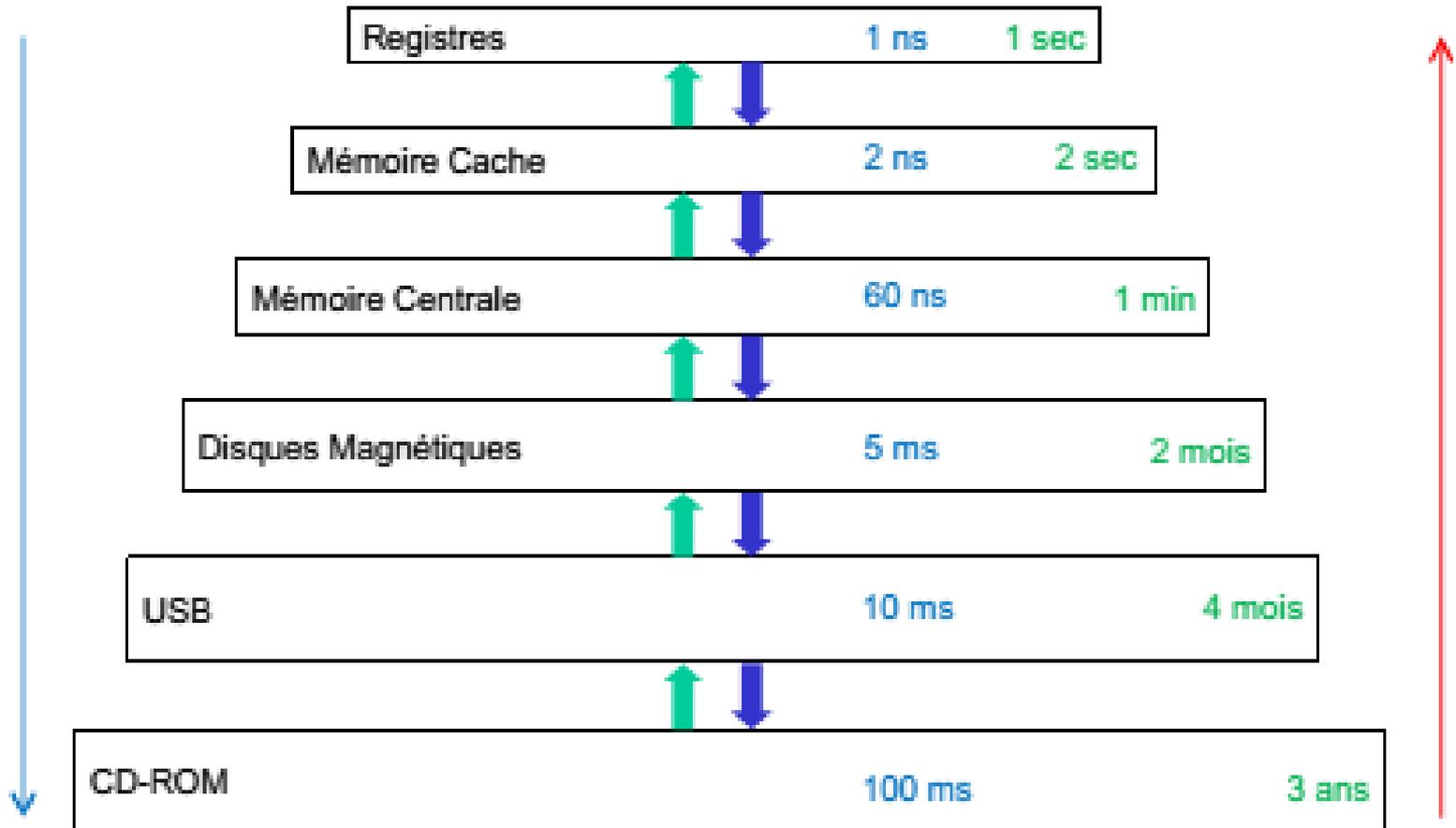


Protection de la mémoire

- Le principe c'est d'empêcher un processus B d'écrire dans une zone mémoire déjà attribuée à un autre processus A et ainsi perturber son exécution.
- Le SE doit assigner ce rôle au MMU pour qu'il ne donne accès au processus qu'aux pages qui lui sont attribuées.
- En cas de tentative, un signal est envoyé au processus fautif . Ce dernier doit demander au SE d'accéder aux données d'un autre processus tant que les opérations demandées sont permises.



Temps d'accès



GESTION DES PÉRIPHÉRIQUES



Les Périphériques d'E/S

- L'ordinateur échange des données à travers des organes externes:
 - Électroniques: Mémoires
 - Magnétiques: Disque
 - Mécaniques: clavier , imprimante
- Il existe deux méthodes de communication entre les périphériques et les programmes
- Scrutation (Polling): L'initiative est au programme
- Interruption: l'initiative est au périphérique



La Gestion des Périphériques

- Le SE a la tâche importante de contrôler les périphériques d'entrées/sorties (E/S) à travers:
 - L'émission des commandes vers les périphériques
 - L'interception des interruptions
 - La gestion des erreurs
- Afin de:
 - Simplifier l'interfaçage entre les périphériques et le système
 - Donner une Interface identique pour tous les périphériques



Unités d'entrées sorties

Il existe deux catégories d'unités entrées sorties:

- **Périphériques par bloc** : informations stockées par blocs de taille fixe, chacun possédant sa propre adresse (ex.. : disque)
- **Périphériques par caractères** : l'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 (e.g. : contrôleur IDE)



La Communication

- Interface entre contrôleur et périphérique de très bas niveau
- Le contrôleur possède des registres qui permettent la communication avec le processeur
 - **Écriture 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



GESTION DES FICHIERS



Définition d'un fichier

- La notion de fichier est l'abstraction fondamentale introduite dans les systèmes d'exploitation pour gérer les données des utilisateurs acquises ou produites via les divers périphériques connectés : disques, écran/clavier, imprimantes, scanners, modems, etc
- Le système de fichiers d'un système d'exploitation est la partie la plus couramment sollicitée par l'utilisateur.



- Dans le cas d'un programme écrit en langage C ou un texte saisi par « Word », par exemple, chaque octet peut représenter un caractère ou participer à cette représentation.
- La plupart des systèmes d'exploitation ne font pas de distinction entre ces fichiers : ce sont les programmes qui les manipulent qui doivent utiliser les bonnes conventions pour interpréter correctement le contenu des fichiers qu'ils utilisent.
- Afin de rendre cette suite d'octets disponible après arrêt de l'ordinateur, cette suite est généralement enregistrée sur un support permanent appelé périphérique de stockage (disque dur dans la plupart des cas, les bandes magnétiques,).



- Chaque fichier est associé à un nom qui sera utilisé ensuite pour l'identifier.
- À l'aide de ce nom, un programme peut demander au système d'exploitation la lecture ou l'écriture du fichier.
- Pour beaucoup d'applications, le chargement complet d'un fichier en mémoire n'est ni utile ni préférable (pour des raisons de coûts, la capacité de la mémoire est très souvent inférieure à celle des périphériques de stockage).
- Dans ce cas, un programme utilisera le fichier comme une bande magnétique, en lisant successivement des petites parties d'un fichier.



LES RÉPERTOIRES

Nous innovons pour votre réussite !

- Un nommage « à plat » des fichiers serait peu pratique à gérer au delà d'une centaines de fichiers, ainsi tous les systèmes de fichiers permettent la création de répertoires qu'un utilisateur utilisera pour rassembler un ensemble de fichiers.

- Pour rendre le classement encore plus efficace, ces répertoires peuvent contenir à leur tour d'autres répertoires, ce qui donne une structure arborescente.

- Les opérations usuelles portant sur les répertoires sont les suivantes :
 - La création d'un répertoire ;
 - La suppression d'un répertoire ;
 - La lecture d'un répertoire, afin d'obtenir la liste des fichiers ; cette opération nécessite généralement plusieurs opérations élémentaires, qui peuvent distinguer une ouverture de répertoire, la lecture (répétée) d'un nom de fichier, et la fermeture ;



Le Système des Fichiers

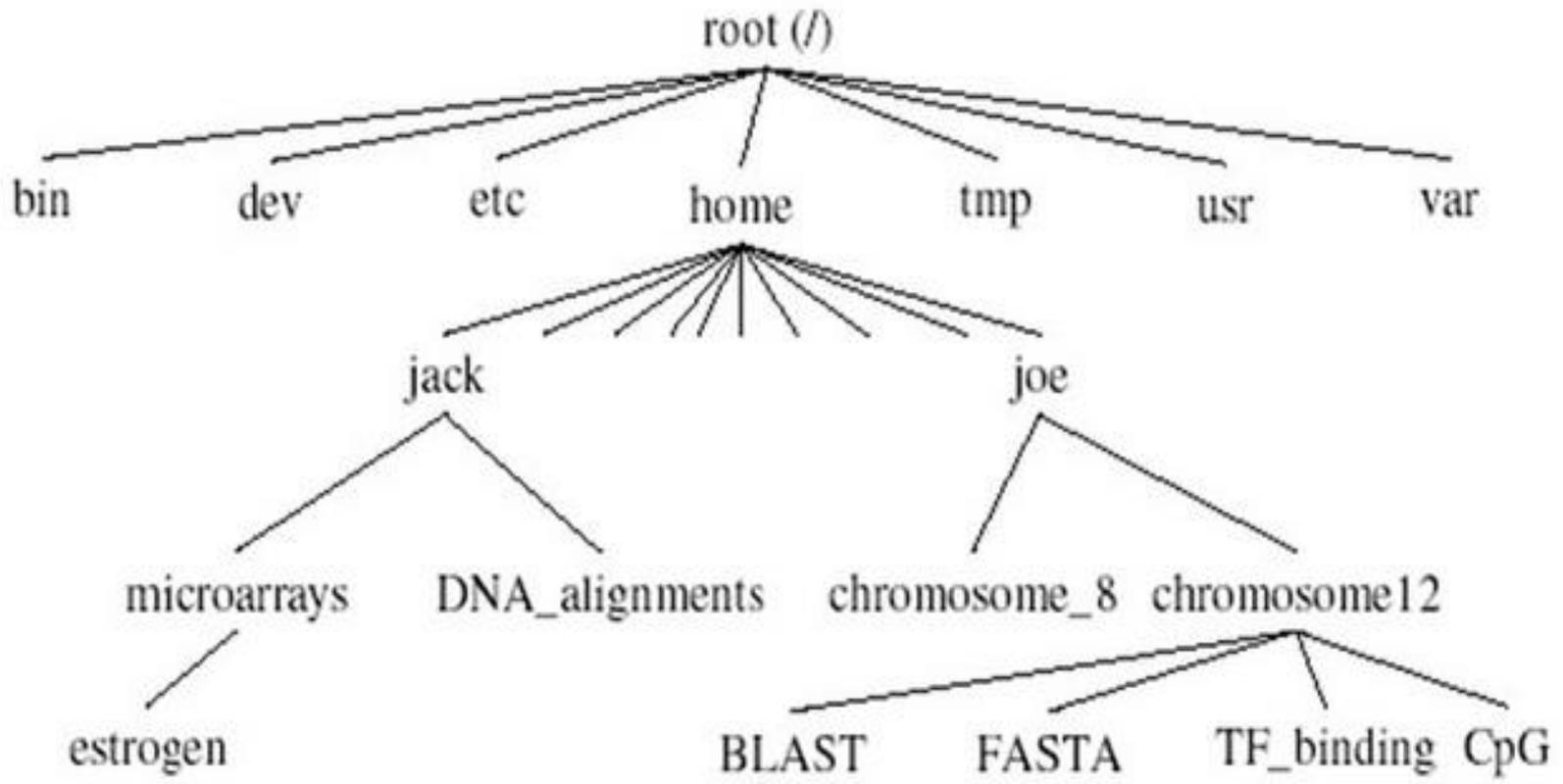
- Composante essentielle d'un système d'exploitation noté (FS : File System), il permet d'enregistrer les fichiers dans une arborescence
- Le système de fichier offre à l'utilisateur une vision structurée et organisée des données et des ressources.
- Le système gère la création des fichiers, leur destruction, leur affectation avec les dispositifs physiques, la gestion des droits sur les fichiers
- L'organisation du système de fichiers est donc primordiale et elle conditionne l'utilisation efficace de la machine.
- Les fichiers sont organisés en une structure arborescente



Structure d'un système de fichier :

Nous innovons pour votre réussite !

Exemple LINUX



Structure d'un système de fichier:

Exemple WINDOWS

1

Disque dur
(partition système)



Vista (C:)

2



Utilisateurs

"Racine du disque"
Dossiers du système



Windows



Programmes

3



Alain



Public

Dossiers des utilisateurs et
Dossier de partage

4



Bureau



Contacts



Documents



Favoris



Images



Liens



Musique



Parties
enregistrées



Recherches



Téléchargement



Vidéos



ntuser.dat