

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**L'université Mohamed Boudiaf - M'Sila –**

**Faculté de mathématiques et d'informatique**

**2<sup>ème</sup> Année Licence (2L)**

# **Systeme d'exploitation 1 (SE 1)**

**Semestre : 04**  
**2023/2024**

**Réalisé par**  
**Dr. DABBA ALI**

➤ **Contactez-nous**

---

[alidabba@gmail.com](mailto:alidabba@gmail.com)

[ali.dabba@univ-msila.dz](mailto:ali.dabba@univ-msila.dz)

- **En cas de problèmes ou de difficultés, me contacter ou contacter votre enseignant TD / TP**
- **Nous sommes à votre disposition pour vous aider**



# CHAPITRE 1

---

## Introduction aux Systèmes d'Exploitation

# Plan

---

- I. Introduction**
- II. Qu'est-ce qu'un système d'exploitation ?**
- III. Objectives des Systèmes d'exploitation**
- IV. Système Informatique (SIQ)**
- V. Les rôles d'un système d'exploitation**
- VI. Les fonctions du système d'exploitation**
- VII. Historique et évolution des systèmes d'exploitation**
- VIII. Taxonomie des systèmes d'exploitation**
- IX. Classification des systèmes d'exploitation**
- X. Mode d'exécution et appels systèmes**
- XI. Exemples de systèmes d'exploitation**



# I. Introduction

---

- Vers les années 40, un seul groupe de personnes concevait, construisait, programmait et utilisait la machine à cause de sa complexité
- La complexité est devenue réduite petit à petit

## Abstraction

Cacher la complexité des machines pour l'utilisateur afin d'utiliser la machine sans savoir ce qui est derrière.

## Objective

L'ordinateur devient à la portée de tout le monde.

# I. Introduction

---

## Structure d'un ordinateur (2 Parties)

- ✓ Matérielle ou Hardware
- ✓ Logicielle ou Software

### ❖ **Partie Hardware**

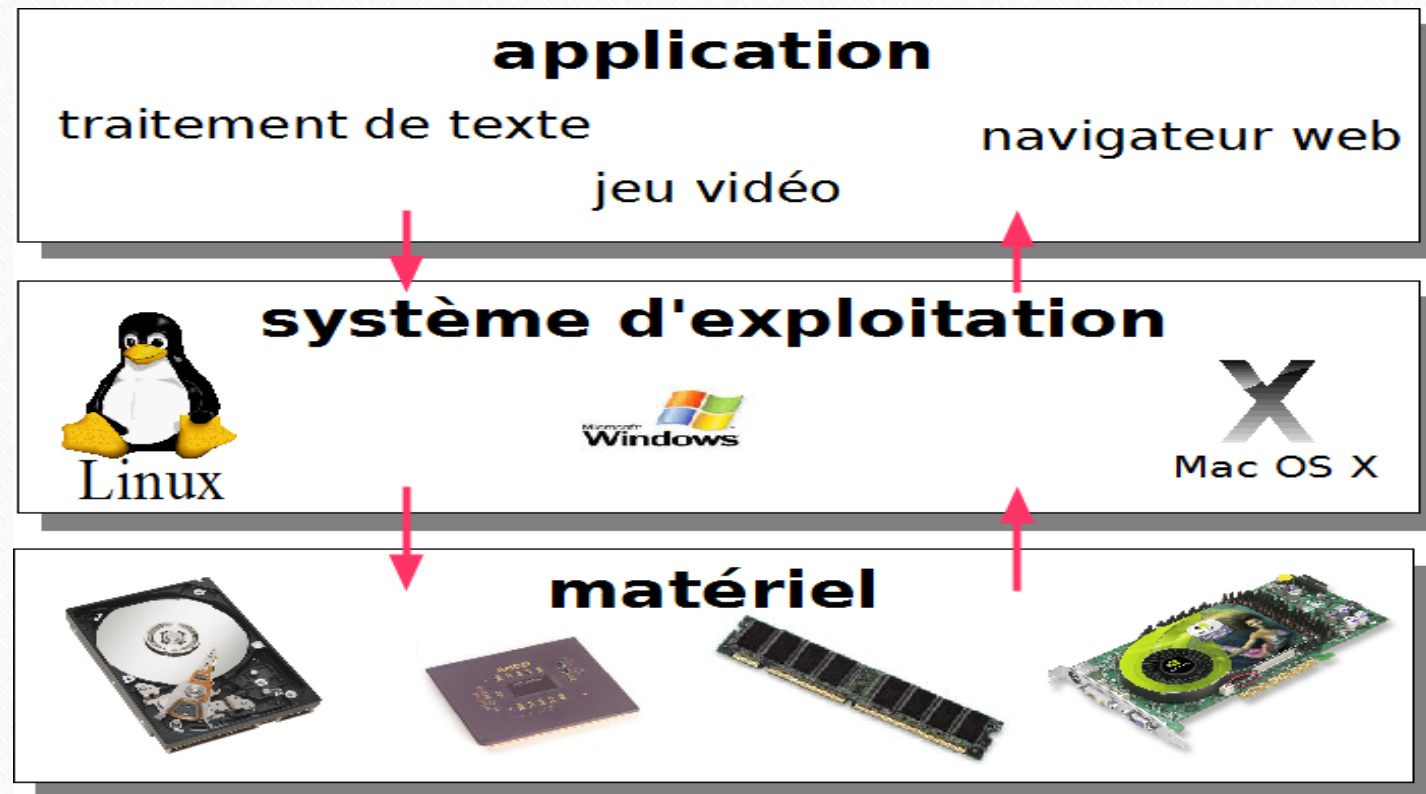
- ✓ Composants physiques (Processeur, Mémoire, Ecran, Disque, ...)

### ❖ **Partie Software**

- ✓ Les programmes système qui permettent le fonctionnement de l'ordinateur.
- ✓ Les programmes d'application qui résolvent les problèmes des utilisateurs.

## II. Qu'est-ce qu'un système d'exploitation ?

Logiciel qui permet une utilisation simple et efficace des ressources d'un ordinateur





## II. Qu'est-ce qu'un système d'exploitation ?

---

### **Définition 1.1**

Un Système d'exploitation (SE) ou (Operating System) est la partie la plus importante d'un Système Informatique (SIQ).

Il est constitué d'un ensemble de programmes agissant comme interface entre l'utilisateur et le matériel. Son but principal est de rendre le système informatique plus pratique à utiliser et qui fonctionne d'une manière efficace et optimale.



## II. Qu'est-ce qu'un système d'exploitation ?

### Définition 1.2

Un système d'exploitation est un ensemble de programmes et de sous programmes (fonctions) qui assurent la gestion des ressources matérielles et logicielles pour coordonner les opérations d'un ordinateur. C'est l'intermédiaire entre les programmes d'applications et le matériel :

il intercepte les demandes des applications et les transmet aux diverses ressources matérielles (mémoire centrale, périphériques d'entrée sortie,...).

Il soustrait le matériel au regard du programmeur et offre une vue agréable à l'ordinateur donc il transforme l'ordinateur en une machine virtuelle facilement manipulable par un simple utilisateur.

## II. Qu'est-ce qu'un système d'exploitation ?

### Définition 1.3

Un Système d'exploitation (SE) est un ensemble de programmes qui coopèrent à la gestion des ressources de la machine (ordinateur). Il doit satisfaire les deux fonctions (principales) suivantes :

- **Partager les ressources** : Une système doit partager les ressources de l'ordinateur entre plusieurs utilisateurs simultanés.
- **Présenter une machine virtuelle à l'utilisateur** : Le système d'exploitation doit convertir un vulgaire matériel en un machine virtuelle.



# III. Objectives des Systèmes d'exploitation

---

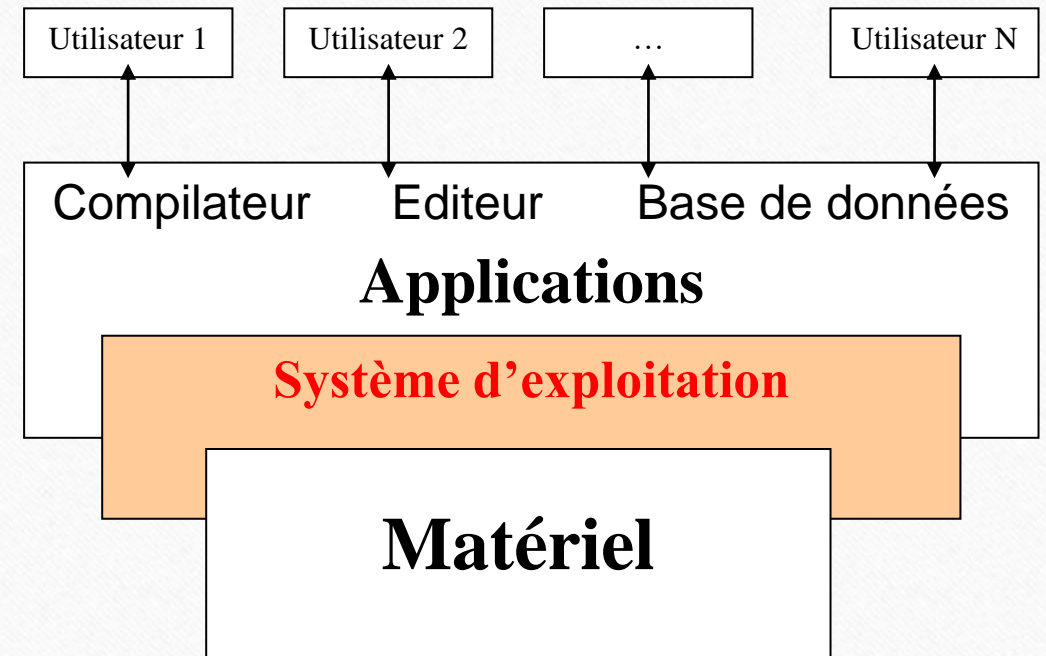
Un SE a deux objectifs principaux :

- ✓ **Présentation** : Il propose à l'utilisateur une abstraction plus simple et plus agréable que le matériel : une machine virtuelle
- ✓ **Gestion** : il ordonne et contrôle l'allocation des processeurs, des mémoires, des icônes et fenêtres, des périphériques, des réseaux entre les programmes qui les utilisent.
  - ❑ Il assiste les programmes utilisateurs.
  - ❑ Il protège les utilisateurs dans le cas d'usage partagé.

# IV. Système Informatique (SIQ)

- ❑ Le matériel (UC, MC, des périphériques d'E/S, ...)
- ❑ Le système d'exploitation (système opératoire)
- ❑ Les programmes d'applications
- ❑ Les utilisateurs

**Un système d'exploitation  
=  
Programme d'allocation  
+  
Programme de contrôle.**





## V. Les rôles d'un système d'exploitation

---

SE est l'intermédiaire entre un ordinateur (ou en général un appareil muni d'un processeur) et les applications qui utilisent cet ordinateur ou cet appareil.

- **Vue par l'utilisateur (avoir un SIQ pratique)**
- **Vue par le système informatique (Avoir un SIQ efficace et optimale)**

## V. Les rôles d'un système d'exploitation

---

### ➤ Vue par l'utilisateur (avoir un SIQ pratique)

- L'objectif c'est de maximiser le travail effectué sur ce système informatique.
- le système d'exploitation est conçu principalement pour faciliter l'utilisation,
- Le SE s'interface entre l'utilisateur et la machine pour fournir un environnement plus pratique
- une Interface Homme Machine (IHM) plus intuitive, plus conviviale et plus facile (principe de la machine virtuelle).



## V. Les rôles d'un système d'exploitation

---

➤ **Vue par le système informatique (Avoir un SIQ efficace et optimale)**

- le SE c'est un programme lié intimement au matériel.
- le SE c'est **un allocateur de ressources**
- le SE joue le rôle **d'un gestionnaire de ressources.**
- Le SE doit garder le SIQ dans un état cohérent en évitant les erreurs et les utilisations incorrectes des programmes de la machine.
- Il doit appliquer également une stratégie de sécurité gardant le SIQ loin de tout accès non autorisé

# VI. Les fonctions du système d'exploitation

---

- Offrir une machine virtuelle simple
- Gestion des processeurs
- Gestion des processus (programmes en cours d'exécution)
- Gérer la mémoire pour exécuter des tâches
- Gérer et conserver l'information
- Assurer l'interaction entre programmes
- Assurer la protection et gérer les erreurs
- Gérer et contrôler les ressources partagées



## VII. Historique et évolution des systèmes d'exploitation

---

*Le développement a été motivé par les objectifs suivants :*

- Décharger le programmeur des tâches de programmations énormes et fastidieuses, et de lui permettre de se concentrer sur l'écriture de son application.
- Protéger le système des usagers et de fausses manipulations.
- Offrir une vue simple, uniforme et cohérente de la machine et de ses ressources.

## VII. Historique et évolution des systèmes d'exploitation

---

*Quatre générations de systèmes peuvent être distinguées :*

- Premiers systèmes
- Systèmes de traitement par lots
- Systèmes multiprogrammes
- Temps partagé (Time Sharing, 1970)



## VII. Historique et évolution des systèmes d'exploitation

---

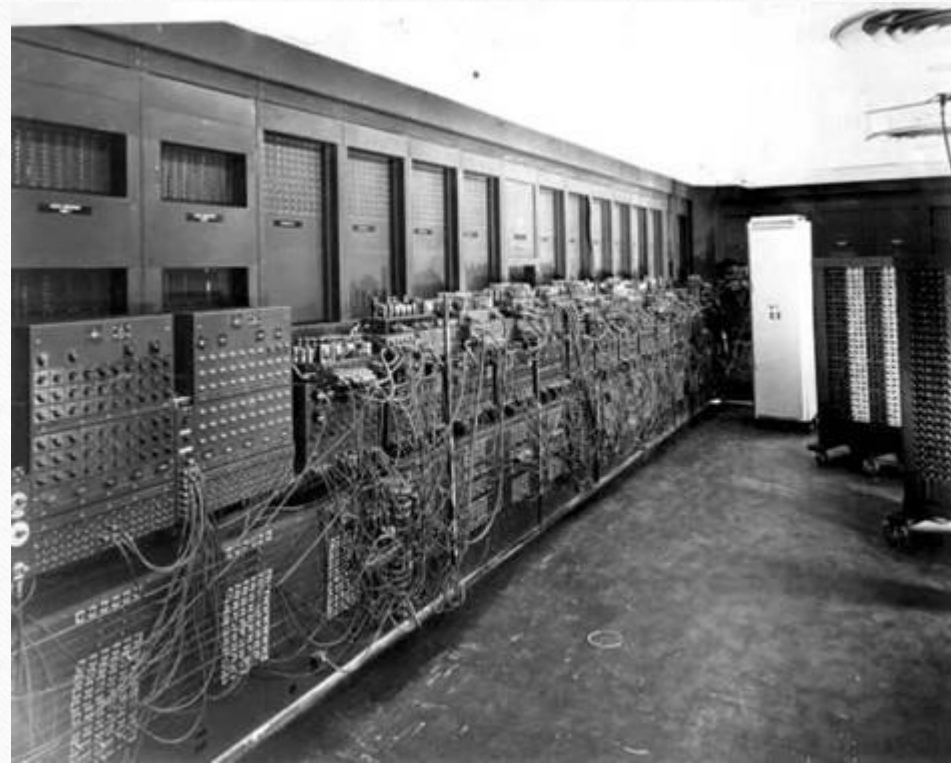
### ➤ Premiers systèmes

- Années 1945-1955
- Des machines à tubes à vide volumineuses, très fragiles et très lentes.
- la programmation en langage machine sur des cartes enfichables ou cartes perforées
- Réaliser des calculs simples
- Elles tournaient sans système d'exploitation et constituaient des systèmes mono-usager.

## VII. Historique et évolution des systèmes d'exploitation

---

### ➤ Premiers systèmes





## VII. Historique et évolution des systèmes d'exploitation

---

### ➤ Premiers systèmes (Inconvénients)

- Temps perdu dans l'attente pour lancer l'exécution d'un programme.
- Vitesse d'exécution de la machine limitée par la rapidité de l'opérateur qui appuie sur les boutons et alimente les périphériques.
- Pas de différences entre : concepteurs ; constructeurs ; programmeurs ; utilisateurs ; mainteneurs.

## VII. Historique et évolution des systèmes d'exploitation

---

### ➤ Systèmes de traitement par lots

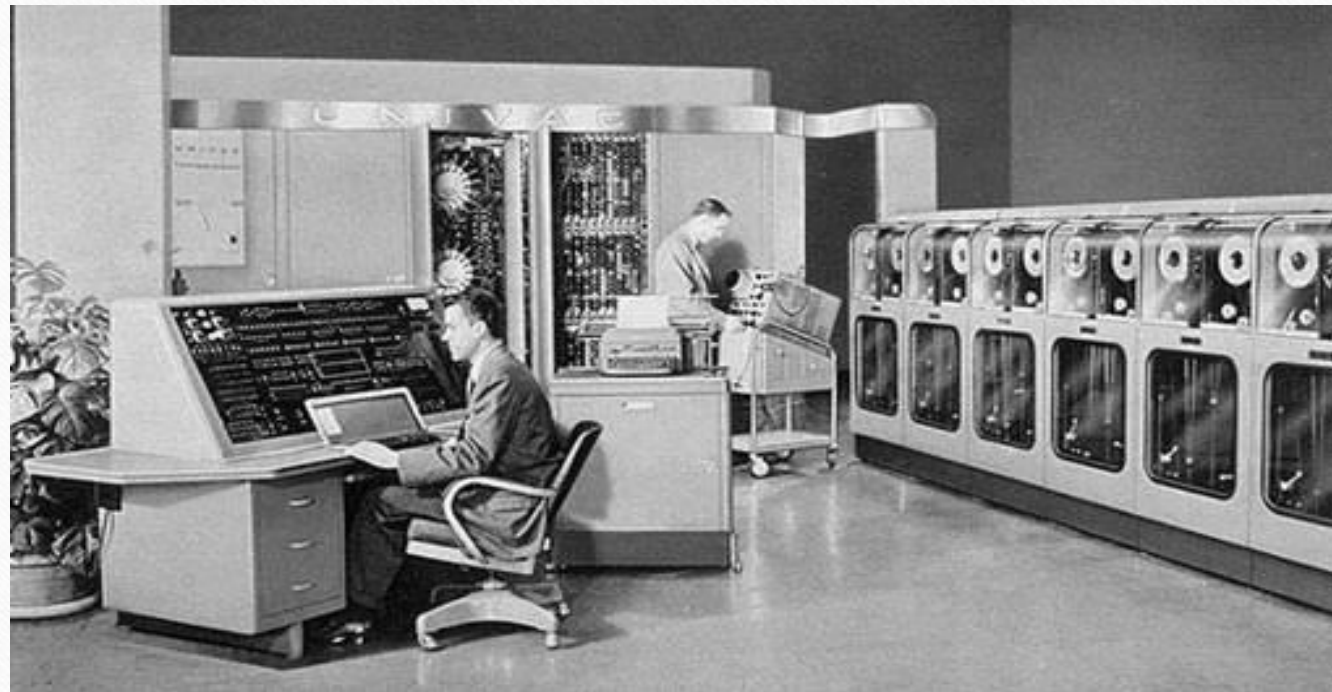
- Années 1955-1965
- Des machines à transistors fiables mais qui demeurent coûteuses.
- la programmation en Fortran et Assembleur
- une distinction nette entre opérateurs de la salle machine, constructeurs, programmeurs et utilisateurs a été faite.
- le séquençement des jobs ou travaux selon l'ordre des cartes de contrôle à l'aide d'un moniteur d'enchaînement.



## VII. Historique et évolution des systèmes d'exploitation

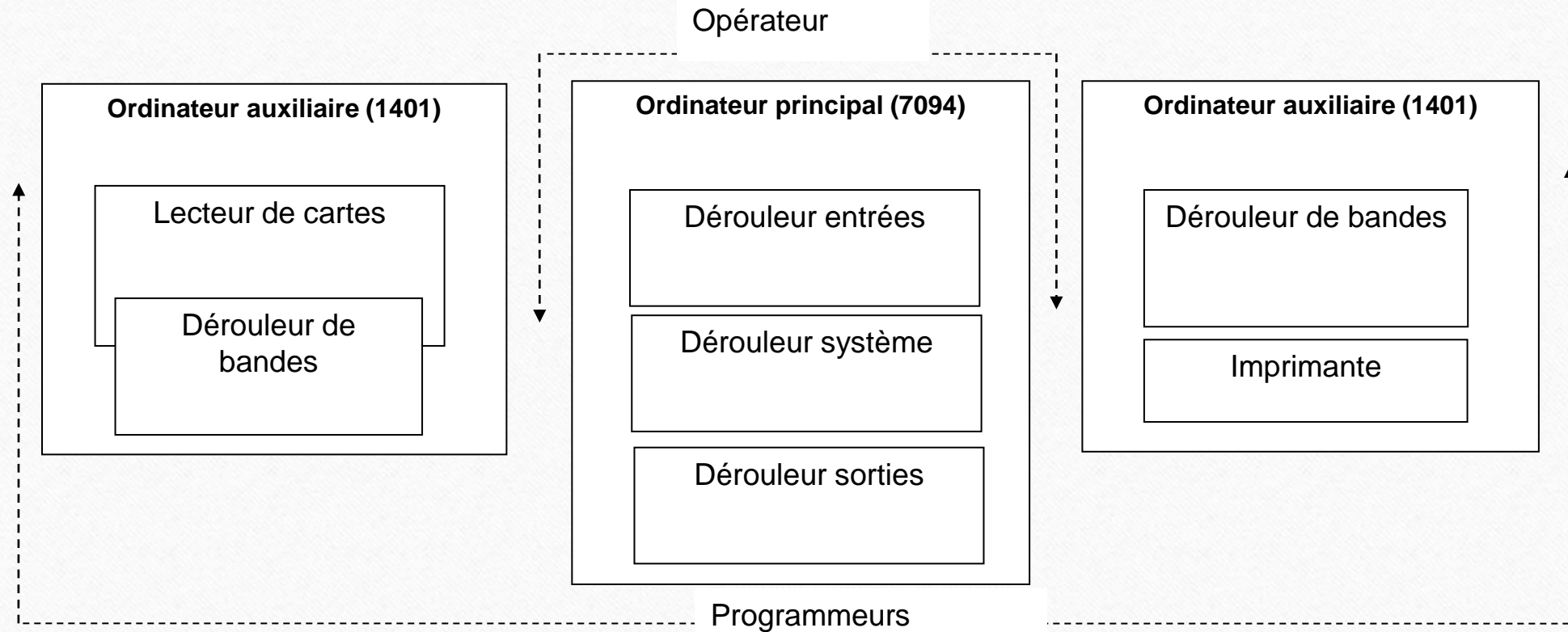
---

### ➤ Systèmes de traitement par lots



# VII. Historique et évolution des systèmes d'exploitation

## ➤ Systèmes de traitement par lots





## VII. Historique et évolution des systèmes d'exploitation

---

### ➤ **Systemes de traitement par lots (Inconvénients)**

- Perte de temps dû à l'occupation du processeur durant les opérations d'E/S. (En effet, le processeur restait trop inactif, car la vitesse des périphériques mécaniques était plus lente que celle des dispositifs électroniques).
- Manque d'interaction entre l'utilisateur et le travail pendant l'exécution
- Les tâches inachevées sont abandonnées.

## VII. Historique et évolution des systèmes d'exploitation

---

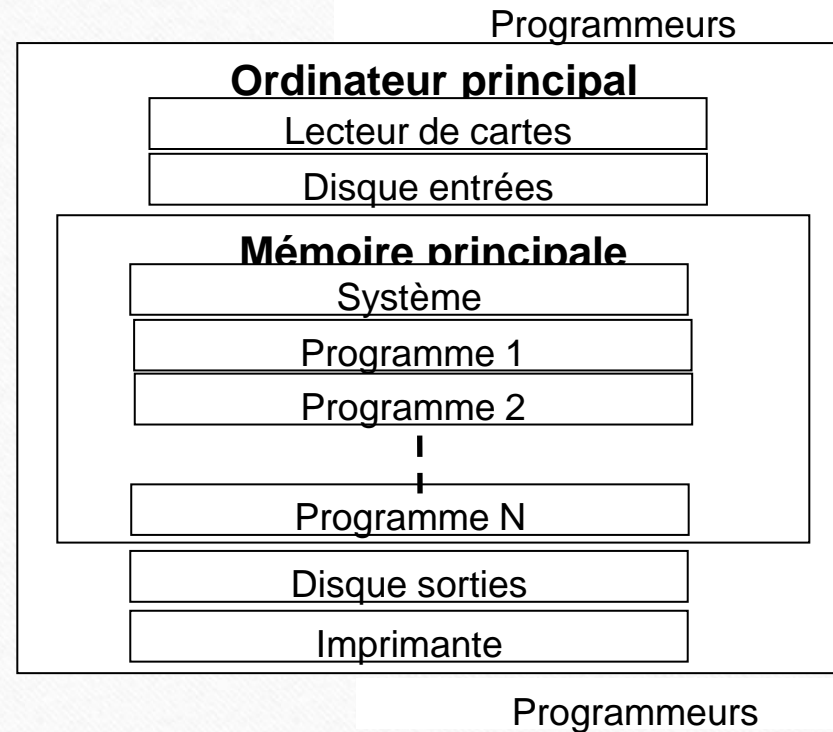
### ➤ Systèmes multiprogrammes

- Années 1965-1980
- Des Machines à circuits intégrés qui sont moins coûteuses et plus performantes
- l'utilisation de la mémoire secondaire au lieu des bandes magnétiques
- l'utilisation d'un seul ordinateur pour le calcul et l'échange des entrées/sorties au moyen d'un canal.
- Un système multiprogrammé maintient plusieurs travaux en mémoire à la fois



# VII. Historique et évolution des systèmes d'exploitation

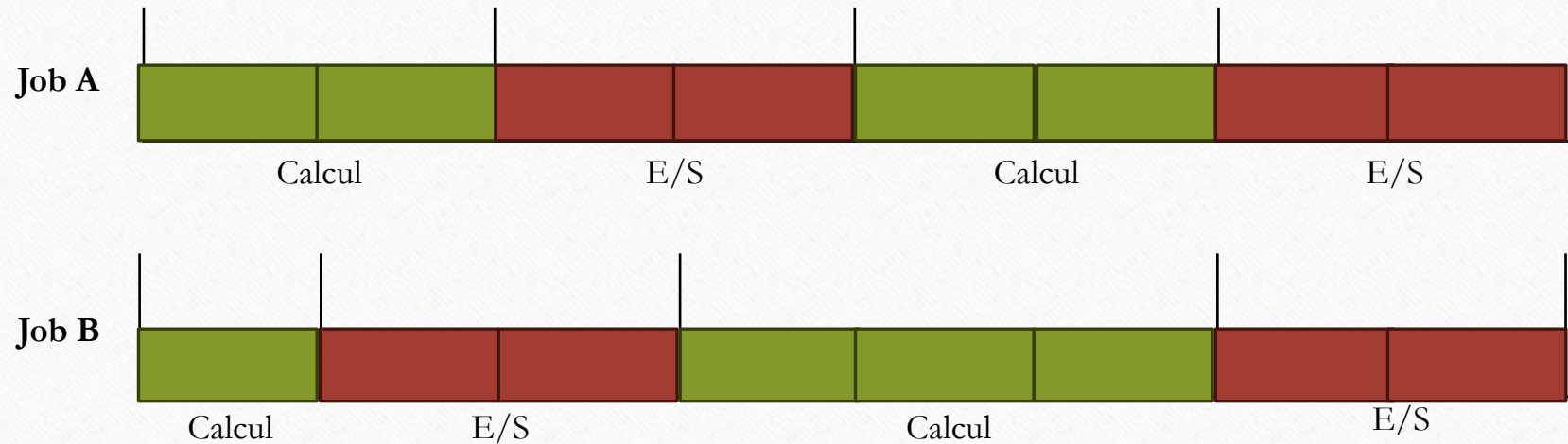
## ➤ Systèmes multiprogrammes



## VII. Historique et évolution des systèmes d'exploitation

### ➤ Systèmes multiprogrammes (Exemple)

Soient les deux programmes A et B suivants :



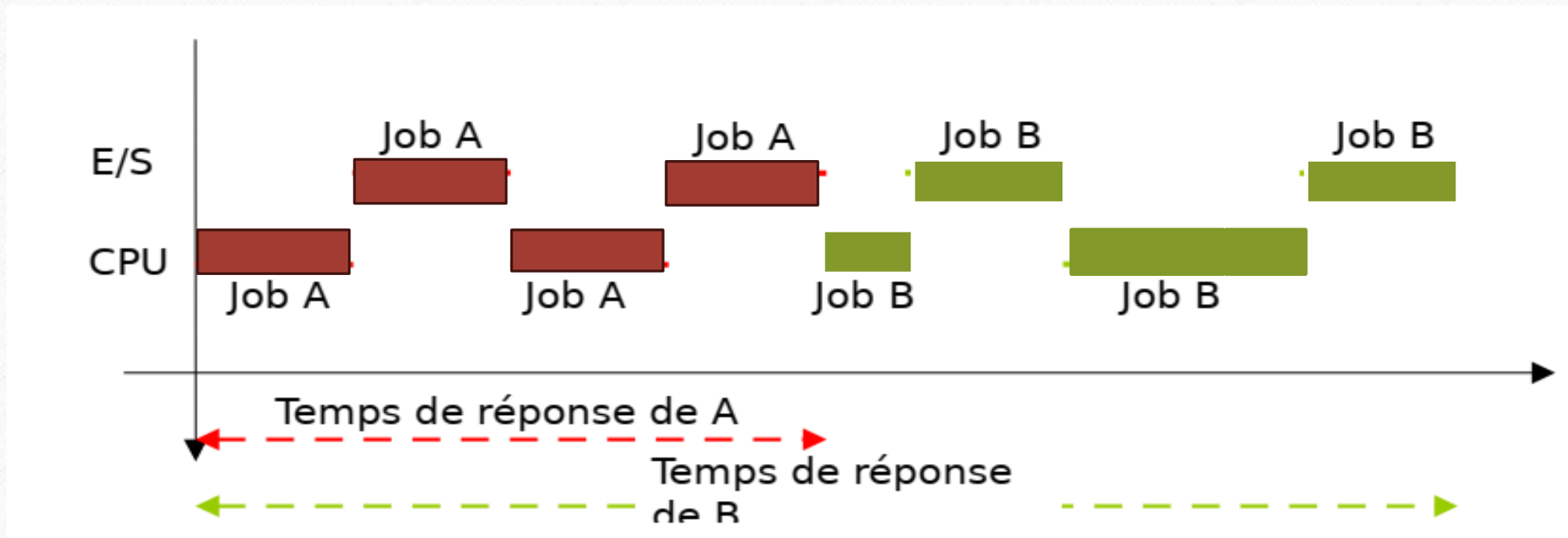
□ On suppose qu'on a un seul périphérique d'E/S.



## VII. Historique et évolution des systèmes d'exploitation

### ➤ Systèmes multiprogrammes (Exemple)

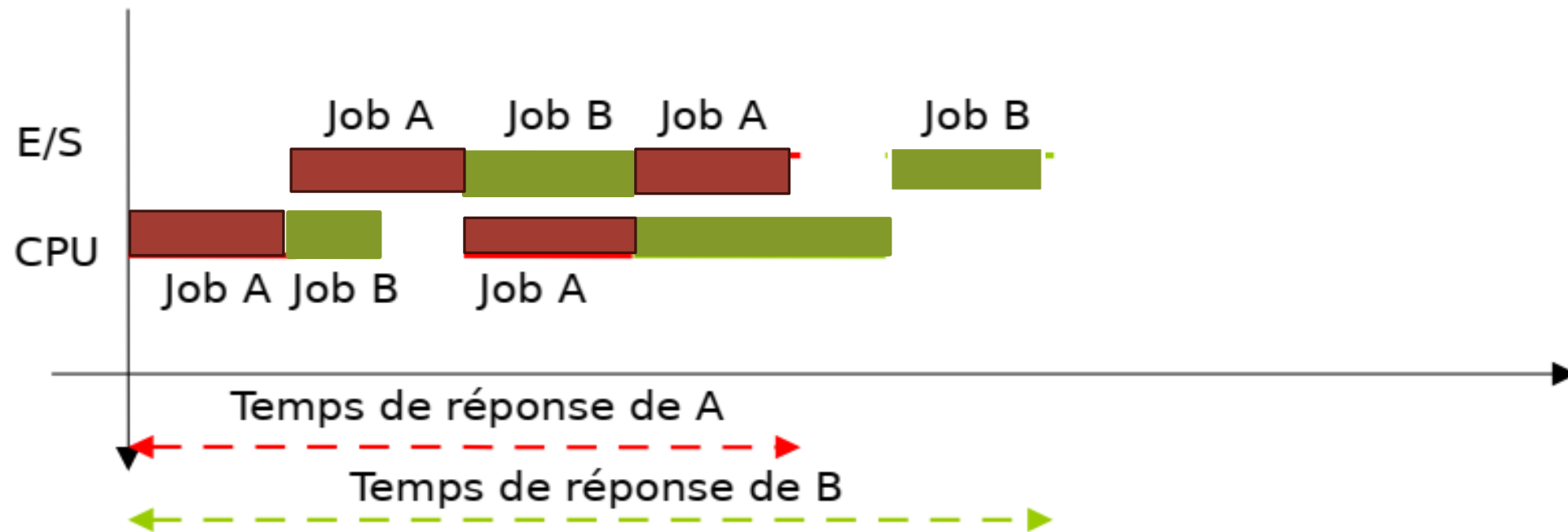
Système mono-programmé



## VII. Historique et évolution des systèmes d'exploitation

### ➤ Systèmes multiprogrammes (Exemple)

Système multiprogrammé





# VII. Historique et évolution des systèmes d'exploitation

## ➤ Systèmes multiprogrammes

### Monoprogrammation

- ✓ Mauvaise utilisation des ressources (processeur, mémoire, E/S,...etc.).
- ✓ Temps de réponse imposé par les jobs très longs.
- ✓ S.E. simple ; seule contrainte : protéger la partie résidente du système des utilisateurs.

### Multiprogrammation

- ✓ Possibilité de mieux équilibrer la charge des ressources.
- ✓ Mieux utiliser la mémoire (minimiser l'espace libre).
- ✓ Possibilité d'améliorer le temps de réponse pour les travaux courts.
- ✓ Protéger les programmes utilisateurs des actions des autres utilisateurs, et protéger aussi la partie résidente des usagers

## VII. Historique et évolution des systèmes d'exploitation

---

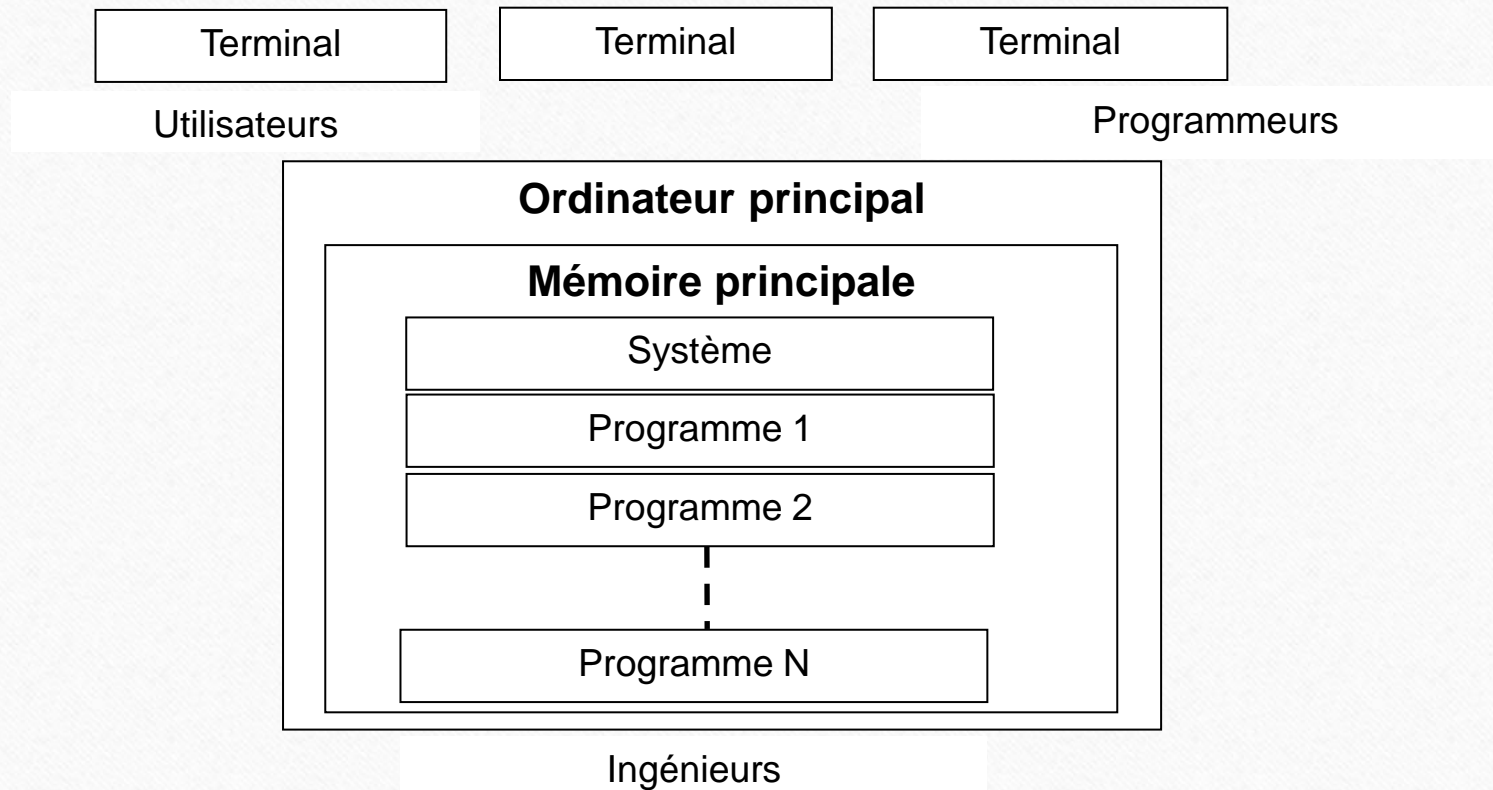
### ➤ Temps partagé (Time Sharing)

- ❑ Années 1970-
- ❑ Le temps partagé est une extension logique de la multiprogrammation
- ❑ le temps CPU est distribué en petites tranches appelées **quantum de temps**.
- ❑ Les systèmes à temps partagé consistent à partager l'unité centrale entre plusieurs tâches par quantum de temps pour assurer leur traitement "simultané"
- ❑ L'objectif est d'offrir aux usagers une interaction directe avec la machine par l'intermédiaire de terminaux de conversation



# VII. Historique et évolution des systèmes d'exploitation

## ➤ Temps partagé (Time Sharing)



## VIII. Taxonomie des systèmes d'exploitation

---

- **Systemes des ordinateurs personnels**
- **Systemes Parallèles (Parallel Systems)**
- **Systemes Distribués et Réseaux**
- **Systemes Temps Réels**
- **Systemes Embarqués (Embedded Systems)**
- **Systemes Multi-cœurs (Multicore Systems)**



# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ Systèmes des ordinateurs personnels

- Les PC sont apparus dans les années 70.
- Les objectifs de ces systèmes (qui n'étaient ni multiutilisateurs, ni multitâches) sont concentrés sur la commodité de l'utilisateur et la rapidité de réaction, la convivialité et l'interactivité avec l'utilisateur, plutôt que de rendre maximale l'utilisation de l'unité centrale et ses périphériques.
- Exemples de systèmes : MS-DOS de Microsoft, Macintosh d'Apple.

# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ Systèmes Parallèles (Parallel Systems)

- ❑ Les systèmes parallèles, appelés aussi systèmes **fortement couplés (Tightly Coupled)**,
- ❑ plus d'un processeur en étroite communication, qui partagent le bus de l'ordinateur, l'horloge et parfois la mémoire et les périphériques.
- ❑ L'avantage de tels systèmes est d'augmenter la **capacité de traitement** et aussi la **fiabilité**



# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ **Systemes Distribués et Réseaux**

Les **systemes réseaux (Network Operating Systems)**, permettant l'accès des utilisateurs à des services communs, trop coûteux pour être individuels : stockage de grandes quantités d'informations, impression des documents, processeurs de grandes puissances, ...etc. Ces services sont gérés par des systèmes appelés serveurs, auxquels s'adressent les utilisateurs clients qui disposent chacun de leurs machines individuelles. Un ordinateur qui exécute un système d'exploitation réseau agit indépendamment de tous les autres ordinateurs présents sur le réseau, bien qu'il ait connaissance du réseau et soit capable de communiquer avec d'autres ordinateurs connectés.

# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ **Systemes Distribués et Réseaux**

Les **systemes distribués (Distributed Systems)** qui consistent en plusieurs ordinateurs pourvus de leur système d'exploitation et reliés entre eux par des canaux de communication. On parle de systèmes répartis ou **faiblement couplés (Loosely Coupled)**. Les S.E. de ces ordinateurs coopèrent et collaborent pour exécuter une application distribuée. Ces systèmes communiquent de manière suffisamment proche pour donner l'illusion qu'il n'y a qu'un S.E. unique qui contrôle le réseau.



# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ Systèmes Temps Réels

Ce type de système est un S.E. spécialisé, dédié à des applications spécifiques, en particulier des systèmes de contrôle, pourvus de capteurs. Ceux-ci captent de l'information qu'ils fournissent à l'ordinateur, puis celui-ci analyse ces informations, réalise les contrôles désirés et donne les résultats pour d'éventuelles interventions.

Un tel système est utilisé lorsqu'il y a des exigences de temps de réponse pour le fonctionnement d'un processeur. Un système d'exploitation temps réel possède des contraintes de temps fixes et bien définies. Le traitement doit être effectué dans les contraintes définies, sinon le système tombe en panne.

# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ Systèmes Temps Réels (Exemple)

- Contrôle d'un réacteur nucléaire : dans le cas de chauffage du réacteur, des procédures doivent être prises avant qu'il n'atteigne une température élevée.
- Chaîne industrielle robotisée.
- Systèmes d'imagerie médicale, ...etc.



## VIII. Taxonomie des systèmes d'exploitation

---

### ➤ Systèmes Embarqués (Embedded Systems)

Un système embarqué est un système prévu pour fonctionner sur : des appareils autonomes (ex. électroménager, robot, automobile), ou des appareils de petite taille (ex. PDA, téléphone, carte à puce). Ces systèmes sont caractérisés par une autonomie réduite, un besoin de gestion avancée de l'énergie, et une capacité de fonctionner avec des ressources limitées. Parmi ces systèmes, on trouve **Palm OS** et **Windows CE** pour PDA, **Java Card** pour cartes à puce, et **TinyOS** pour les réseaux de capteurs (**Sensor networks**), ...etc.

# VIII. Taxonomie des systèmes d'exploitation

---

## ➤ Systèmes Multi-cœurs (Multicore Systems)

Un ordinateur **multi-cœurs** (**Chip Multi-Processor**), combine deux ou plusieurs processeurs, appelés **cœurs** (**Cores**), sur une seule puce de silicone, appelée **die** (**Die**). L'apparition des architectures multi-cœurs nécessite de repenser la conception des systèmes d'exploitation afin de pouvoir en tirer profit du parallélisme offert par ces architectures. Le défi de conception d'un système d'exploitation pour une architecture multi-cœurs est de permettre une exploitation simple, sûre, et efficace des processeurs multi-cœurs.

Même si les systèmes d'exploitation actuels ne supportent pas totalement les architectures multi cœurs, des améliorations ont été apportées à leur gestion de ressources (notamment la gestion de l'ordonnancement et la gestion des caches) pour prendre en compte ces architectures.



## **IX. Classification des systèmes d'exploitation**

---

- **Classement des systèmes selon les contraintes d'utilisation**
- **Classement des systèmes selon les services**
- **Classement des systèmes selon les architectures**

# IX. Classification des systèmes d'exploitation

---

## ➤ Selon les contraintes d'utilisation

- Les systèmes mono-utilisateur/mono-tâche où un seul utilisateur utilise le système à la fois et une seule tâche peut être exécutée à la fois. C'est le cas de MS-DOS à titre d'exemple.
- Les systèmes mono-utilisateur/multitâches où un seul utilisateur à la fois peut exécuter plusieurs tâches simultanément comme il est le cas de Windows.
- Les systèmes multi-utilisateurs/multitâches où plusieurs utilisateurs à la fois peuvent exécuter chacun plusieurs tâches simultanément et partager les mêmes ressources matérielles comme c'est le cas d'Unix.



# IX. Classification des systèmes d'exploitation

---

## ➤ Selon les services

- ❑ Les systèmes temps réel qui sont utilisés dans des domaines spécifiques (procédés, robotique, centrales nucléaires...). Ces systèmes assurent un temps de réponse des tâches critiques court et sont fiables et tolérants aux pannes.
- ❑ Les systèmes transactionnels qui sont dédiés à la gestion des bases de données énormes (systèmes de réservation, systèmes bancaires...) et qui permettent de garantir des mises à jour sans incohérence.

# IX. Classification des systèmes d'exploitation

---

## ➤ Selon les architectures

- ❑ Les systèmes mono-processeur qui réfèrent aux systèmes multiprogrammés et à temps partagé. Ces systèmes n'ont besoin que d'un seul processeur qui peut faire des calculs pseudo-parallèles pour faire progresser les tâches à la fois.
- ❑ Les systèmes parallèles (multiprocesseurs) comme SunOS 4, SunOS 5, Solaris 2 et Linux qui sont capables d'effectuer un traitement parallèle par plusieurs processeurs. Ils sont fiables et caractérisés par une grande capacité de traitement et un temps de réponse court.



# X. Mode d'exécution et appels systèmes

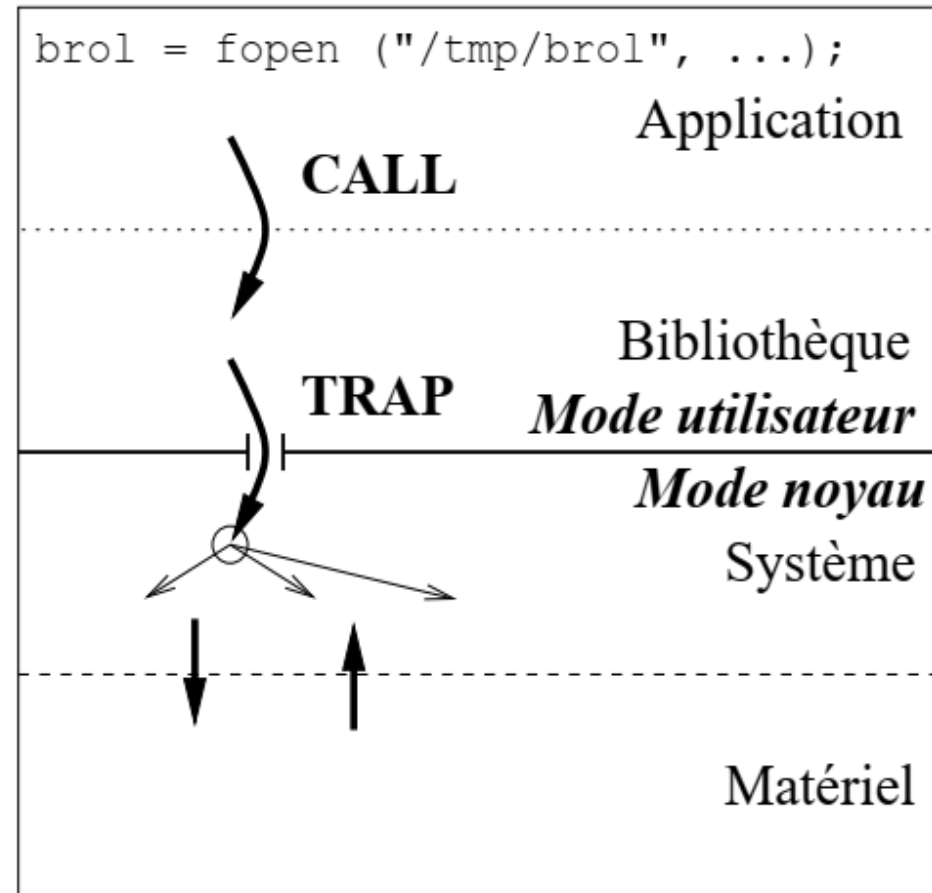
---

Dans les systèmes d'exploitation, l'exécution est effectuée selon deux modes :

- **Le mode utilisateur** : c'est un mode non protégé réservé à l'exécution des programmes des utilisateurs. Il permet à l'utilisateur de modifier des données de son programme.
- **Le mode noyau** : ce mode protégé est réservé au système d'exploitation. Il permet l'accès au code et données utilisées par ce dernier ainsi que la lecture et l'écriture sur les ports d'entrées/sorties. Aussi, prend-il la charge de protéger les données sensibles.

# X. Mode d'exécution et appels systèmes

Un appel système est une interface entre les programmes utilisateurs et le système d'exploitation. Lors de l'appel d'une fonction système telle que `fopen`, une interruption logicielle Trap survient et le système bascule du mode utilisateur où le programme appelant s'exécutait au mode noyau où l'appel système `fopen` sera exécutée (voir figure 1.7).





## **XI. Exemples de systèmes d'exploitation**

---

- **MS-DOS: Microsoft Disk Operating system**
- **Le système Windows**
- **Le système UNIX : Uniplexed Information and Computer Service**

## XI. Exemples de systèmes d'exploitation

---

### ➤ **MS-DOS: Microsoft Disk Operating system**

C'est un système mono-utilisateur, il est connu au début des années 80, il est destiné au microordinateur (les premiers IBM-PC 8088), conçu par Bill Gates, associé aujourd'hui avec Windows.



# XI. Exemples de systèmes d'exploitation

---

## ➤ Le système Windows

- ❑ Système conçu par Microsoft avec une interface graphique, son nom désigne l'utilisation d'une fenêtre pour chaque exécution. Il est destiné à plusieurs architectures (PC, Station de travail, Portable, Réseau client/serveur, ...).
- ❑ Il est passé par plusieurs versions comme pour PC : 1.0, 2.0, 3.10, 3.11, 9x, 2000, Me, XP, Vista, Seven, 10, aussi : Windows Server, NT pour l'architecture multi-utilisateur et client/serveur.

## XI. Exemples de systèmes d'exploitation

---

### ➤ **Le système UNIX : Uniplexed Information and Computer Service**

Une famille de systèmes en temps partagé proposée pour la plupart des architectures, utilisée pour certains milieux d'industrie et de recherche.



# XI. Exemples de systèmes d'exploitation

---

## ➤ Le système UNIX : Uniplexed Information and Computer Service (*Historique*)

- ❑ 1969 : Naissance d'UNICS aux laboratoires de Bell par Ken Thompson → une altération en UNIX.
- ❑ 1974 : Publication d'UNIX en langage C par Thompson et Ritchie.
- ❑ 1979 : Apparition des versions BSD (Berkeley Software Distribution) de l'université de Berkeley.
- ❑ Jusqu'à 1980 : UNIX est offert aux maisons d'enseignements et aux entreprises.
- ❑ Aux années 80 : La société AT & T commercialisa les versions System III, System IV et System V
- ❑ 1981-1994 : Apparition de Linux.

# XI. Exemples de systèmes d'exploitation

---

## ➤ Le système UNIX : Uniplexed Information and Computer Service (*Les raisons de succès d'UNIX*)

- Ecrit à l'aide d'un langage de haut niveau par rapport aux autres systèmes à l'époque.
- Interface utilisateur adaptée.
- Open source → possibilité d'amélioration dans le code source par n'importe le qui ce qui résulte l'apparition de plusieurs variantes par plusieurs sociétés comme : System V de AT & T, XENIX par Microsoft, ULTRIX par DEC, pour Linux → Redhat, Coldera, ...
- Disponibilité de versions gratuites.



# XI. Exemples de systèmes d'exploitation

---

## ➤ Le système UNIX : Uniplexed Information and Computer Service (*Les services offerts par le noyau UNIX*)

- La gestion des événements matériels.
- La gestion des ressources.
- La gestion des structures de données.
- La gestion de la mémoire et le processeur.
- Fournit des services élémentaires.

**C'est un système multitâches, multi-utilisateurs, multissessions, et multipostes.**

# XI. Exemples de systèmes d'exploitation

## ➤ Les types de systèmes d'exploitation

Systeme	Codage	Mono-utilisateur	Multi-utilisateur	Mono-tâche	Multitâche
DOS	16 bits	X		X	
Windows3.1	16/32 bits	X			non préemptif
Windows95/98/ Me	32 bits	X			coopératif
WindowsNT/20 00	32 bits		X		préemptif
WindowsXP	32/64 bits		X		préemptif
Windows7	32/64 bits		X		préemptif
Unix / Linux	32/64 bits		X		préemptif
MAC/OS X	32 bits		X		préemptif
VMS	32 bits		X		préemptif



## XII. La procédure d'amorçage d'un PC

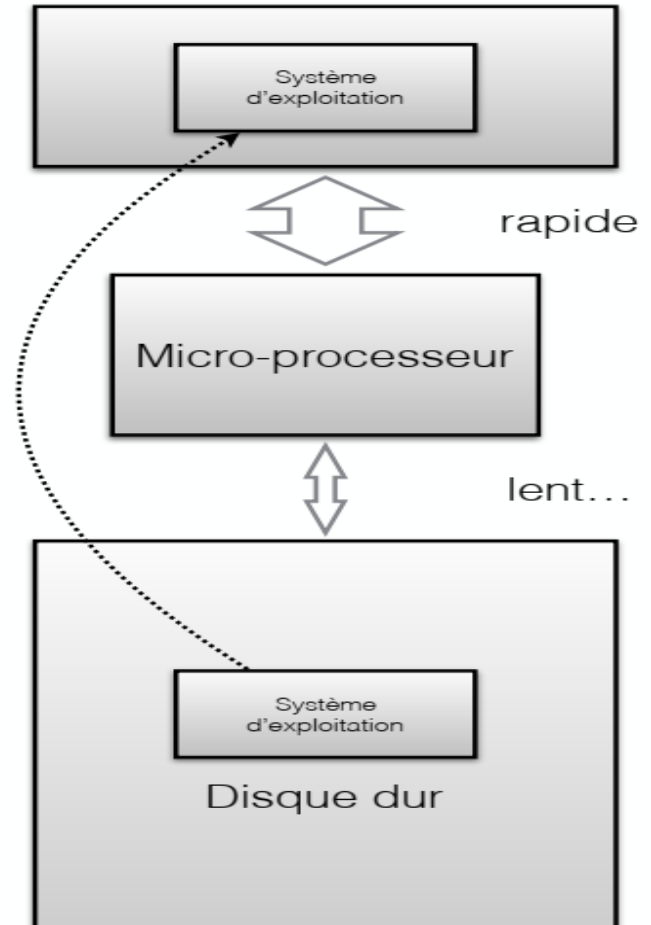
### ➤ Démarrage: objectifs

L'objectif de la séquence de démarrage est de démarrer le système d'exploitation (SE)

### ➤ Où est-il situé? Sur le disque dur

➤ **Peut-on l'exécuter s'il est sur le disque dur?** Non, le disque dur est un périphérique de stockage lent

➤ **Donc, il nous faut le transférer dans la mémoire principale (RAM)**



## XII. La procédure d'amorçage d'un PC

---

### ➤ Démarrage: défis

- La RAM n'est pas initialisée au démarrage (elle perd son contenu lorsqu'on coupe l'alimentation), que faire?
- Comment faire pour savoir à quel endroit le SE est-il situé sur le disque dur?
- Que faire s'il y a plusieurs SE sur le disque dur?



## XII. La procédure d'amorçage d'un PC

---

### ➤ Démarrage: défis & solutions

- La RAM n'est pas initialisée au démarrage (elle perd son contenu lorsqu'on coupe l'alimentation), que faire?
  - ✓ Utilise une **ROM** qui contient un petit programme qui sera exécuté au démarrage
- Comment faire pour savoir où le SE est-il situé sur le disque dur?
  - ✓ Un programme «spécial» (nommé le «**bootloader**») est toujours situé au même endroit sur le disque
- Que faire s'il y a plusieurs SE sur le disque dur?
  - ✓ C'est le «bootloader» qui demandera à l'utilisateur de faire un choix

## XII. La procédure d'amorçage d'un PC

---

### ➤ Le processus de démarrage

est une série d'étapes **séquentielles exécutées** par l'ordinateur, si l'une des étapes n'est pas exécutée ou génère une erreur, l'ordinateur ne peut pas démarrer.

**Avant** de procéder au démarrage d'un pc, il faut bien se rassurer de la disponibilité des choses suivantes

- Alimentation suffisante en énergie électrique ;
- La bonne connexion de tous les périphériques absolument nécessaires au démarrage ;
- La présence des applications permettant d'arriver au bureau.



## XII. La procédure d'amorçage d'un PC

---

- Le processus d'amorçage (ou démarrage) d'un ordinateur
  - Alimentation de l'ordinateur
  - Test POST (Power-On Self Test)
  - Sélection du périphérique de démarrage
  - Chargement du chargeur d'amorçage (boot loader)
  - Chargement du système d'exploitation
  - Initialisation du système d'exploitation
  - Affichage de l'interface utilisateur

## XII. La procédure d'amorçage d'un PC

---

### ➤ Alimentation de l'ordinateur ;

- Lorsque vous appuyez sur le **bouton d'alimentation** de l'ordinateur, l'alimentation électrique est fournie à tous les composants du système. Cela permet de mettre en marche l'ensemble du matériel de l'ordinateur.
- Celui envoie un signal **Power-On** au processeur.



## XII. La procédure d'amorçage d'un PC

---

### ➤ **Test POST (Power-On Self Test) :**

- Une fois que l'alimentation est fournie, le BIOS (Basic Input/Output System) ou l'UEFI (Unified Extensible Firmware Interface) est chargé depuis une mémoire permanente (ROM ou flash) sur la carte mère.
- Le BIOS est un logiciel intégré qui fournit des instructions de base pour le fonctionnement de l'ordinateur.

## XII. La procédure d'amorçage d'un PC

---

### ➤ **Test POST (Power-On Self Test) :**

- Le BIOS (Basic Input/Output System) ou l'UEFI (Unified Extensible Firmware Interface) effectue un test de démarrage automatique appelé POST. Ce test vérifie l'état et la fonctionnalité des composants matériels tels que le processeur, la mémoire, les disques durs, les cartes graphiques, etc.
- Si des erreurs sont détectées lors du POST, un message d'erreur peut s'afficher à l'écran.



## XII. La procédure d'amorçage d'un PC

---

### ➤ Sélection du périphérique de démarrage

- Le BIOS / l'UEFI recherche le périphérique de démarrage prédéfini dans l'ordre de démarrage. Le périphérique de démarrage peut être le disque dur principal, une clé USB, un DVD, etc.
- Le BIOS/UEFI localise le MBR (Master Boot Record) sur le périphérique de démarrage spécifié.

## XII. La procédure d'amorçage d'un PC

---

### ➤ Sélection du périphérique de démarrage

- Le MBR est situé dans le premier secteur du périphérique de démarrage, généralement le disque dur. Il contient le code de démarrage et la table de partition.
- Le code de démarrage du MBR est un petit programme situé dans les premiers 446 octets du MBR.



## XII. La procédure d'amorçage d'un PC

---

### ➤ Sélection du périphérique de démarrage

- Lorsque le BIOS/UEFI le trouve, il est chargé en mémoire et exécuté. Le code de démarrage du MBR a pour rôle de rechercher et de lancer le **chargeur d'amorçage (boot loader)**.
- Le code de démarrage du MBR utilise la table de partition du MBR pour identifier la **partition active**. La partition active est celle qui contient le **chargeur d'amorçage (boot loader)**.

## XII. La procédure d'amorçage d'un PC

---

### ➤ Chargement du chargeur d'amorçage (boot loader) :

- ❑ Une fois que le périphérique de démarrage est identifié, le BIOS / l'UEFI charge le chargeur d'amorçage depuis ce périphérique. Le chargeur d'amorçage est un petit programme situé dans le secteur d'amorçage du périphérique de démarrage. Son rôle est de localiser et de charger le système d'exploitation.



## XII. La procédure d'amorçage d'un PC

---

### ➤ Chargement du chargeur d'amorçage (boot loader) :

- ❑ Il peut s'appeler GRUB (GNU GRand Unified Bootloader) pour les systèmes **Linux**, ou NTLDR (NT Loader) pour les anciennes versions de **Windows**, par exemple.

## XII. La procédure d'amorçage d'un PC

---

### ➤ Chargement du système d'exploitation :

- ❑ Le chargeur d'amorçage transfère le contrôle au système d'exploitation. Le système d'exploitation est chargé depuis le disque dur ou un autre support de stockage, et le noyau du système d'exploitation est chargé en mémoire.



## XII. La procédure d'amorçage d'un PC

---

### ➤ Initialisation du système d'exploitation :

- ❑ Une fois que le noyau du système d'exploitation est chargé, il procède à l'initialisation du système. Cela comprend la configuration des pilotes de périphériques, le montage des systèmes de fichiers, la détection et l'initialisation des périphériques matériels, l'établissement des connexions réseau, etc.

## XII. La procédure d'amorçage d'un PC

---

### ➤ Affichage de l'interface utilisateur

- ❑ Une fois que le système d'exploitation est initialisé, il affiche l'interface utilisateur, que ce soit un environnement de bureau graphique, une ligne de commande ou autre. L'utilisateur peut alors interagir avec l'ordinateur et utiliser les fonctionnalités du système d'exploitation.



## XII. La procédure d'amorçage d'un PC

---

### ➤ La RAM CMOS (Complementary Metal-Oxide Semiconductor )

- très petite zone de mémoire (64 octets initialement) maintenue alimentée par une pile lorsque l'ordinateur est éteint.
- contient plusieurs informations relatives au matériel d'un PC telles que les types de disque dur, de clavier, d'écran et de chipset.
- contient aussi la date et le temps du système qui est mis à jour par un circuit qui compte le temps, alimenté aussi par la pile.
- L'information contenue dans la RAM CMOS est utilisée par le BIOS.**

## XII. La procédure d'amorçage d'un PC

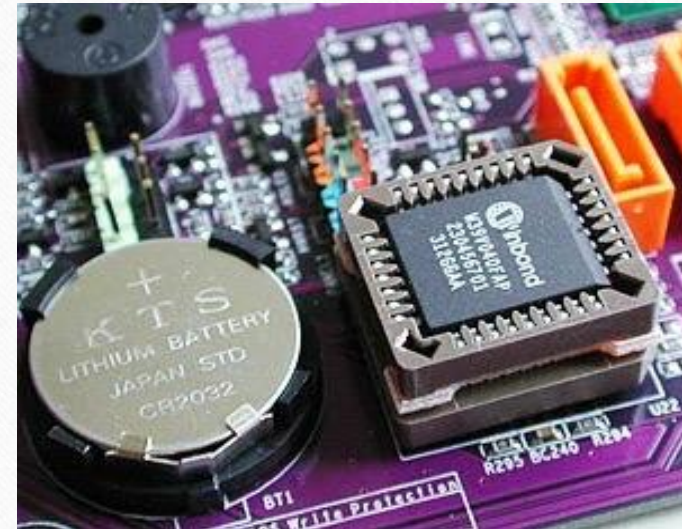
### ➤ Le BIOS (Basic Input Output System)

#### ❑ Programme

- ✓ Stocké dans une petite mémoire ROM, soudée sur la carte mère

#### ❑ Paramètres

- ✓ Stockés dans une petite mémoire “RAM CMOS” non volatile
- ✓ Mémoire alimentée par batterie (2–10 ans de longévité)





# XII. La procédure d'amorçage d'un PC

---

## ➤ Les rôles du BIOS

- Test de l'ordinateur (POST)
- Lancer le système d'exploitation
- Acquérir et maintenir l'information de base sur les périphériques du système.
- Fournir une librairie de fonctions au système d'exploitation afin de contrôler les périphériques (sous la forme d'interruptions).
  - ✓ **Beaucoup moins utilisées avec les systèmes d'exploitation récents.**
- Dans les ordinateurs modernes, le BIOS a acquis plusieurs autres tâches secondaires, souvent reliées aux périphériques: gestion de la puissance, gestion de la température, support pour le plug N play, etc.

## XII. La procédure d'amorçage d'un PC

---

### ➤ **Le POST: Power On Self Test**

Les tests suivants sont effectués dans l'ordre sur les PC compatibles:

- Test du microprocesseur lui-même (test des registres en écriture et en lecture)
- Test de la ROM contenant le programme de démarrage et le BIOS
- Initialisation du contrôleur de mémoire
- Test des 16 premiers K de mémoire RAM
- Initialisation du contrôleur d'interruption et des interruptions



## XII. La procédure d'amorçage d'un PC

---

### ➤ **Le POST: Power On Self Test**

- Test du temporisateur servant à compter le temps
- Test de l'adaptateur d'écran et affichage du curseur
- Test de l'ensemble de la mémoire RAM
- Test du clavier
- Vérification de la présence d'un lecteur de disquette ou d'un disque dur
- Test de l'imprimante et des ports de communication
- Comparaison des résultats obtenus lors des tests avec la configuration sauvegardée en mémoire CMOS
- «Beep» du haut-parleur

## XII. La procédure d'amorçage d'un PC

---

### ➤ **MBR (Master Boot Record)**

- ❑ Le MBR (Master Boot Record) est une petite partie d'un disque dur ou d'un périphérique de stockage qui contient des informations essentielles pour le processus de démarrage d'un ordinateur.



## XII. La procédure d'amorçage d'un PC

---

### ➤ MBR (Master Boot Record)

La structure du MBR est identique pour tous les SE :

- ❑ Les **446 premiers octets** contiennent un petit programme dont le but est de rechercher la partition active.
- ❑ Les **64 octets** contiennent la **table de partition** à quatre entrées de 16 octets chacune.
- ❑ Le MBR se termine par **une signature de 2 octets**, appelée signature du MBR. Cette signature, généralement **0x55AA** en hexadécimal, indique au BIOS/UEFI que le MBR est valide et peut être utilisé pour le processus de démarrage.

# Questions ?

---





## Exercice 1.1 (Notion de base)

---

1. Quel est le rôle d'un système d'exploitation ? Les interpréteurs de commandes et les compilateurs font-ils parties du système d'exploitation ?

Il gère et contrôle le matériel et offre aux utilisateurs une machine virtuelle plus simple d'emploi que la machine réelle (appels systèmes).

Non, les interpréteurs et les compilateurs ne font pas parties du système d'exploitation.

# Exercice 1.1 (Notion de base)

---

2. Qu'est-ce qu'un système multiprogrammé ? Un système de traitement par lots ? Un système en temps partagé ?

Un système multiprogrammé gère le partage des ressources (mémoire, processeur, périphériques...) de l'ordinateur entre plusieurs programmes chargés en mémoire.

Dans un système de traitement par lots, les processus sont exécutés l'un à la suite de l'autre selon l'ordre d'arrivée.

Dans un système en temps partagé, le processeur est alloué à chaque processus pendant au plus un quantum de temps. Au bout de ce quantum, le processeur est alloué à un autre processus.



# Exercice 1.1 (Notion de base)

---

## 3. Quel est les fonctions du système d'exploitation ?

Les rôles du système d'exploitation sont multiples :

- ✓ Gestion du processeur ;
- ✓ Gestion de la mémoire vive ;
- ✓ Gestion des entrées/sorties ;
- ✓ Gestion de l'exécution des applications ;
- ✓ Gestion des droits ;
- ✓ Gestion des fichiers ;
- ✓ Gestion des informations: le système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine.

## Exercice 1.1 (Notion de base)

---

4. Quelle est la différence entre un système mono-programmé (mono-tâche) et un système multiprogrammé ? quels sont les avantages et les défis imposés par la multitâche ?

En monoprogrammation (mono programmé, mono tâche), il y a un seul programme utilisateur dans la MC. (Il y a un seul processus à la fois en mémoire). Lorsqu'une tâche est soumise et que le processeur est disponible, on la charge en mémoire puis on exécute le processus associé jusqu'à ce qu'il soit terminé. On passe alors à la tâche suivante.



# Exercice 1.1 (Notion de base)

---

4. Quelle est la différence entre un système mono-programmé (mono-tâche) et un système multiprogrammé ? quels sont les avantages et les défis imposés par la multitâche ?

En multiprogrammation, il peut y avoir plusieurs programmes (processus) à la fois en mémoire. (Une tâche soumise est chargée en mémoire s'il y a de la place et donne naissance à un processus). Un programme est prêt s'il n'est pas en attente d'une entrée-sortie. Les processus prêts s'exécutent à tour de rôle, on parle de commutation de processus.

La multiprogrammation peut être utilisée en mode non préemptif ou préemptif (temps partagé). En multiprogrammation non préemptive, il n'y a pas de commutation sauf si le processus (programme) actif doit effectuer une entrée sortie.

# Exercice 1.1 (Notion de base)

---

4. Quelle est la différence entre un système mono-programmé (mono-tâche) et un système multiprogrammé ? quels sont les avantages et les défis imposés par la multitâche ?

Les avantages et les défis de multitâche :

- Augmentation de taux d'occupation du CPU.
- Augmentation de rendement de l'UC + Temps de réponse d'un processus.
- Nécessite d'un matériel spécialisé plus coûteux (système d'exploitation plus complexe) + gestion des ressources.



## Exercice 1.1 (Notion de base)

---

5. Que fait l'UC quand il n'y a aucun programme à exécuter ?

Il y a toujours un programme à exécuter (dès lors que l'ordinateur est sous-tension). Le cycle de recherche et d'exécution ne s'arrête jamais. Lorsqu'il n'y a aucun programme utilisateur à exécuter. Le système d'exploitation s'exécute dans une boucle qui ne fait rien (boucle d'attente active ou boucle inactive) jusqu'à ce que survienne une interruption.

## Exercice 1.1 (Notion de base)

---

6. Qu'est-ce que L'amorce (Bootstrap) ?

**L'amorce (Bootstrap)** est un petit programme qui est chargé en mémoire en une seule opération d'entrée lors de la mise en route de l'ordinateur, et dont l'exécution permet le chargement et le lancement d'un programme plus important.



# Exercice 1.1 (Notion de base)

---

7. Pourquoi le partage de données pose des problèmes dans un système multiprogrammé en temps partagé ? Le système UNIX permet-il de contrôler les accès aux données partagées ? Qu'est-ce qu'une section critique ?

Un autre processus peut accéder aux ressources partagées avant qu'un processus n'ait fini de les utiliser (modifier).

Oui, par exemple les sémaphores, les tubes et les files d'attente de messages.

Une suite d'instructions qui accèdent à des objets partagés avec d'autres processus.

## Exercice 1.1 (Notion de base)

---

8. Dans le système UNIX, les véritables appels système sont effectués à partir

- d'un programme utilisateur
- d'une commande shell
- d'une procédure de la bibliothèque standard

Sont-ils exécutés en mode superviseur ou en mode utilisateur ?

A partir de la bibliothèque standard des appels système (instruction TRAP). Ils sont exécutés en mode superviseur (Leurs codes constituent le système d'exploitation).



## Exercice 1.1 (Notion de base)

---

9. Comment sont organisés les fichiers dans le système UNIX ?  
Un utilisateur peut-il accéder à un fichier d'un autre utilisateur ?  
Si oui, comment ? (Question TP)

Les fichiers sont organisés dans des répertoires. Chaque répertoire peut contenir des fichiers ou des répertoires (une structure arborescente). Pour contrôler les accès aux fichiers, chaque fichier a son propre code d'accès sur bits. Un utilisateur peut accéder à un fichier d'un autre utilisateur si le code d'accès du fichier le permet. Le chemin d'accès est absolu.