

Cours
systèmes d'exploitation II

Enseigné par:
Imen SFAIHI

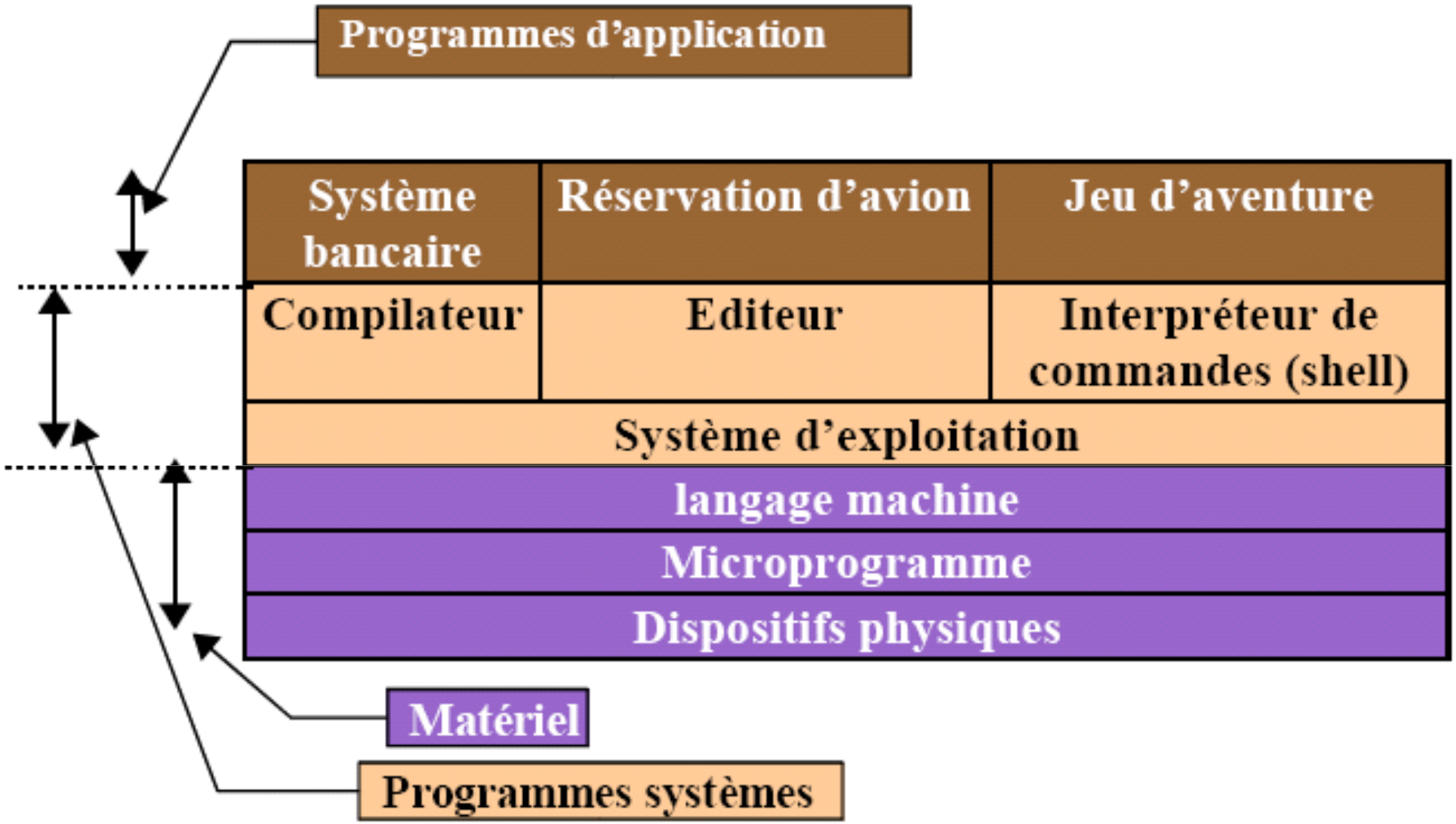
☞ L'objectif de ce cours est de :

Présenter les concepts fondamentaux des systèmes d'exploitation (gestion de fichiers et E/S, processus, concurrence, communication, gestion mémoire,...) ;

Un ordinateur contient :

- **Un ou plusieurs processeurs,**
- **une mémoire principale,**
- **des horloges,**
- **des terminaux,**
- **des disques,**
- **des interfaces de connexion à des réseaux et**
- **des périphériques d'entrées/sorties.**

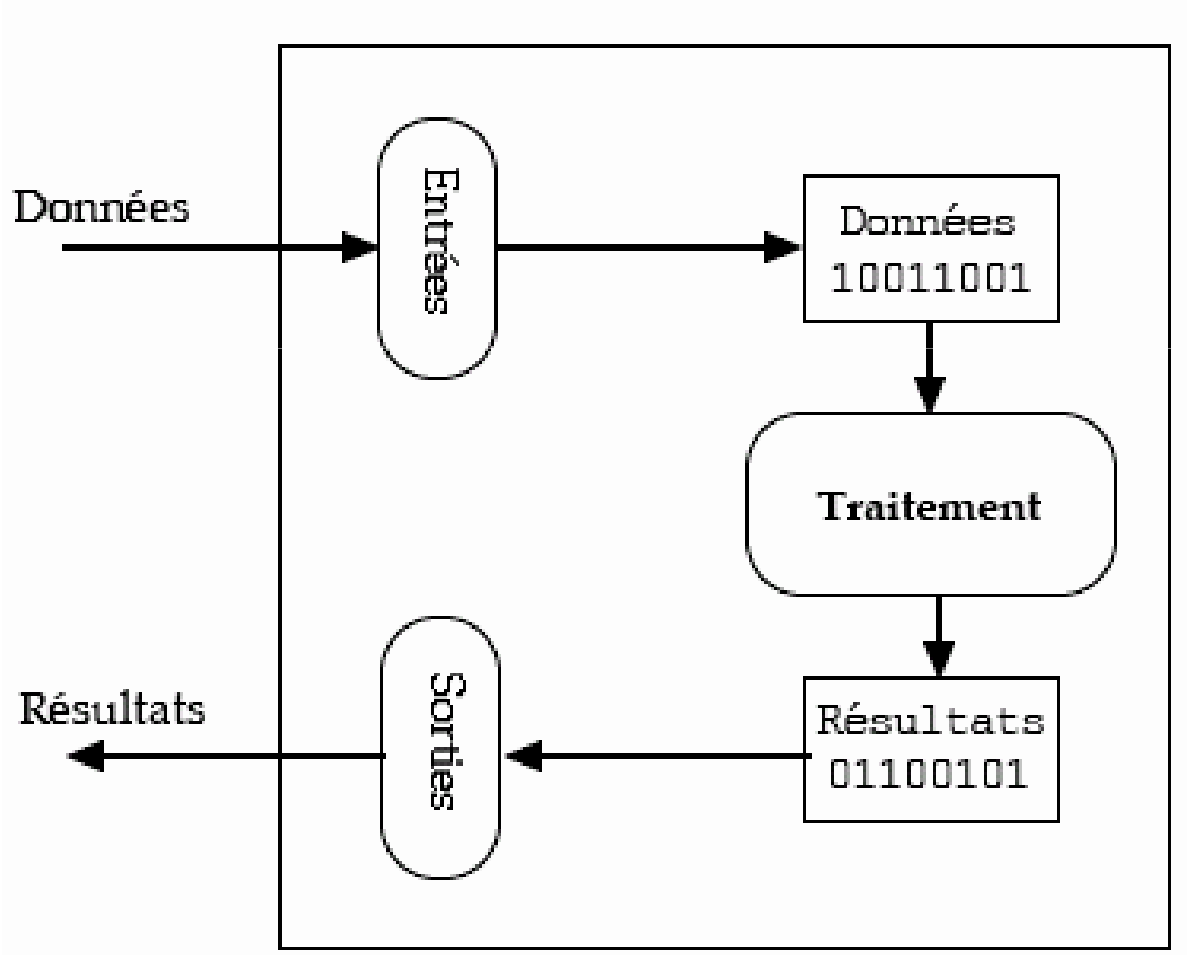
La complexité évidente du matériel implique la réalisation d'une **machine virtuelle** qui gère le matériel : **c'est le système d'exploitation.**


Programmes d'application
Système bancaire
Réservation d'avion
Jeu d'aventure
Compilateur
Editeur
Interpréteur de commandes (shell)
Système d'exploitation
langage machine
Microprogramme
Dispositifs physiques
Matériel
Programmes systèmes

Introduction

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

Exploitation d'un *système informatique*



Fonctions

Saisie et restitution de l'information (gestion des Entrées/Sorties)

Stockage de l'information (gestion de la mémoire)

Traitement de l'information (exécution de programmes)

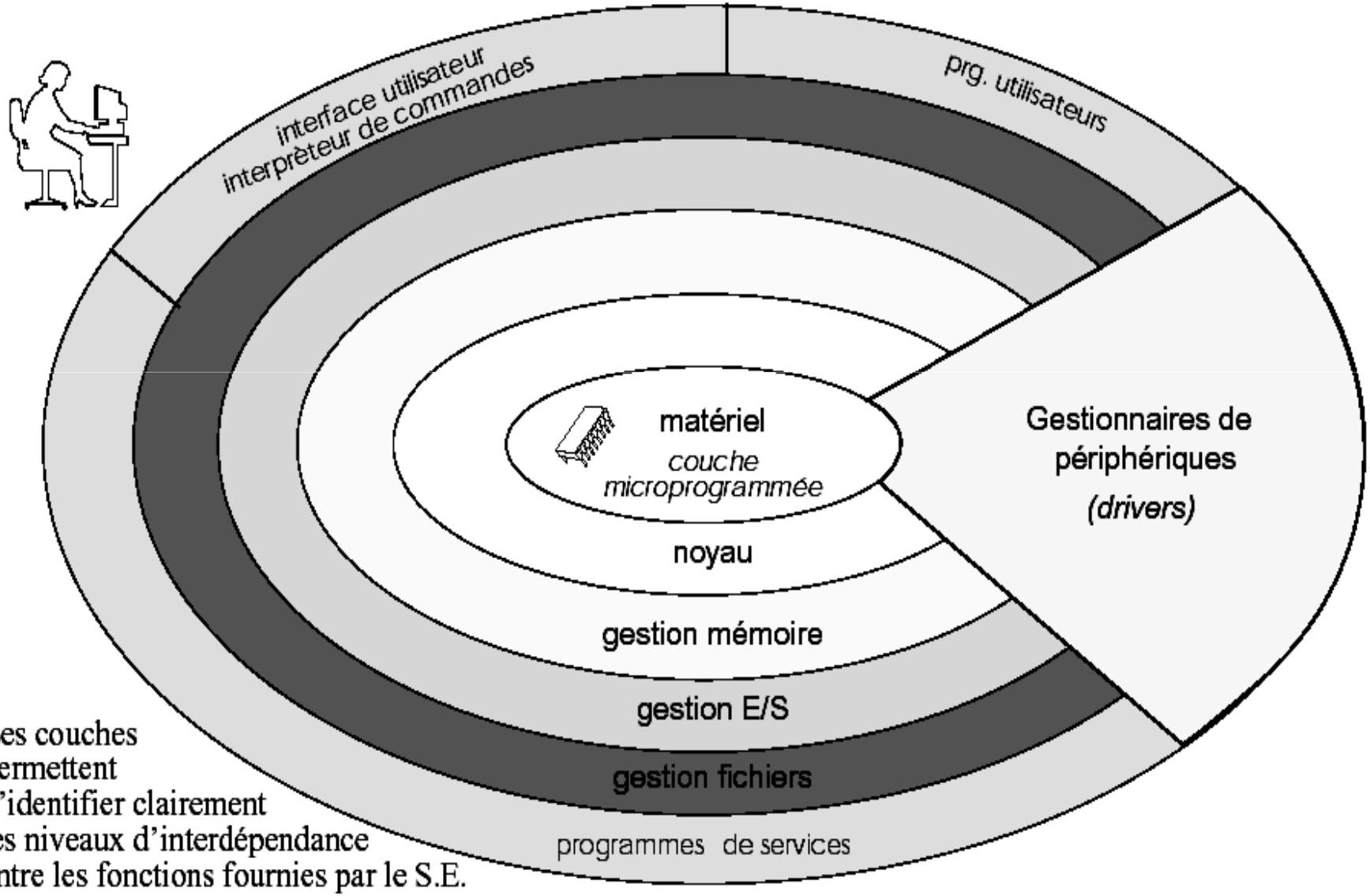
Préparation et mise au point des programmes

Définition

Le système d'exploitation est un ensemble logiciel qui pilote la partie matérielle d'un ordinateur. Un système multitâches alloue les ressources physiques ou logiques aux tâches concourantes (concurrentes!).
Les principales ressources gérés par un système d'exploitation sont :

- les fichiers
- les processus
- la mémoire virtuelle
- les périphériques (stockage, communications).

Structure en couche d'un SE moderne



Les couches permettent d'identifier clairement les niveaux d'interdépendance entre les fonctions fournies par le S.E.

Pourquoi étudier les systèmes d'exploitation ?

Tout programme est concerné : il est important d'appréhender la façon dont fonctionne un système d'exploitation pour améliorer l'efficacité de ses propres programmes ;

Tout programmeur est susceptible de rencontrer les mêmes problèmes de mise en œuvre dans son propre domaine : pas la peine de réinventer la roue ;

Problématique

Pour que les programmes puissent s'exécuter de façon portable et efficace, il faut Pouvoir gérer simultanément:

La multiplicité des différentes ressources;

La complexité des composants de chacune d'elles, qui requiert la prise en compte de nombreux détails embêtants, sources de bogues

Fonctionnalités d'un système d'exploitation

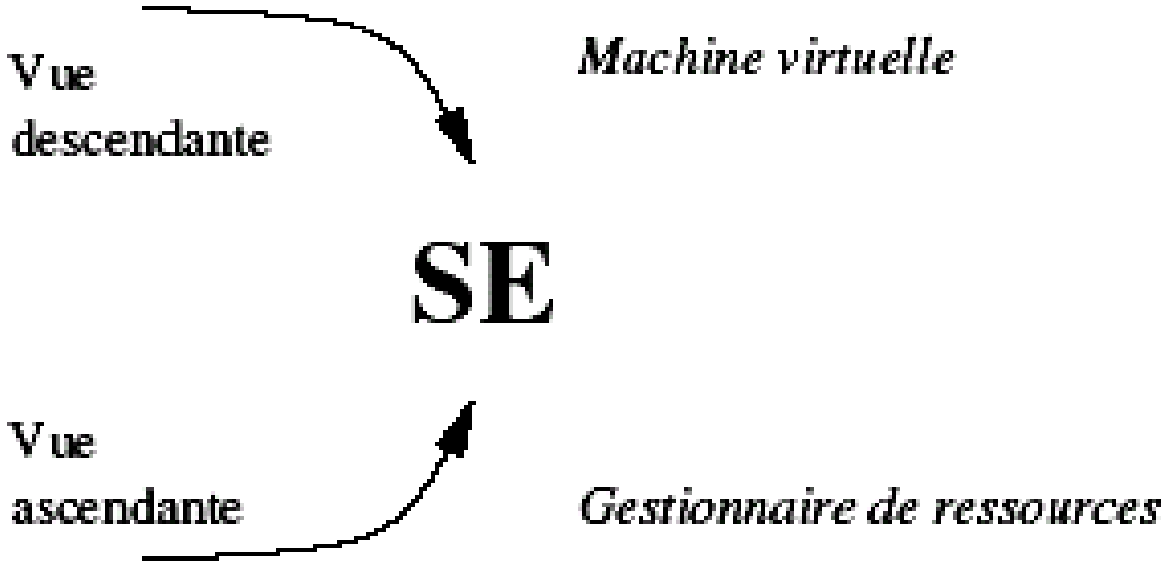
Un système d'exploitation a pour but:

• De décharger le programmeur d'une tâche de programmation énorme et fastidieuse, et de lui permettre de se concentrer sur l'écriture de son application

• De protéger le système et ses usagers de fausses manipulations

• D'offrir une vue simple, uniforme et cohérente de la machine et de ses ressources

On peut considérer un système d'exploitation de deux points de vue:



Machine étendue ou encore machine virtuelle.

La machine virtuelle fournit à l'utilisateur:

Rôle => Masquer des éléments fastidieux liés au matériel, comme les interruptions, les horloges, la gestion de la mémoire, la gestion des périphériques (déplacement du bras du lecteur de disquette) ...

-Une vue uniforme des E/S

-Une mémoire virtuelle et partageable

-La gestion de fichiers et répertoires

-La gestion de droits d'accès, sécurité et du traitement des erreurs

-La gestion de processus

-La gestion des communications inter-processus

En tant que gestionnaire de ressources, le système d'exploitation doit permettre:

Rôle => Ordonnancer et contrôler l'allocation des ressources (processeurs, mémoires et périphériques d'E/S) entre les différents programmes qui y font appel.

-d'assurer la bon fonctionnement des ressources et le respect des délais

-l'identification de l'usager d'une ressource

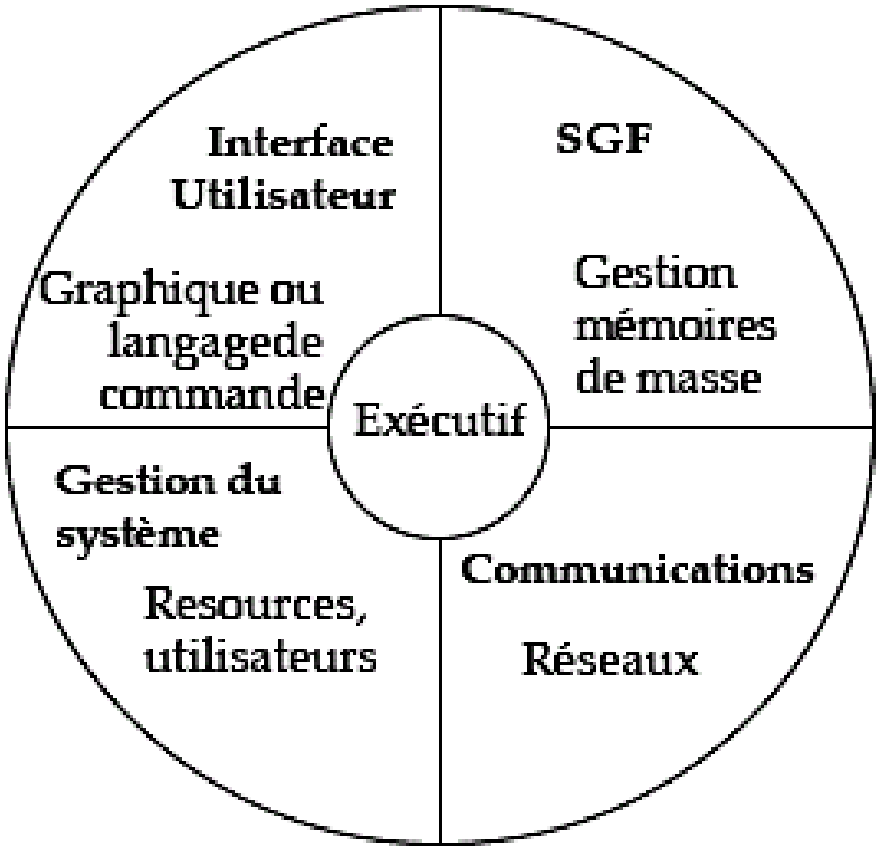
-le contrôle des accès aux ressources

-l'interruption d'une utilisation de ressource

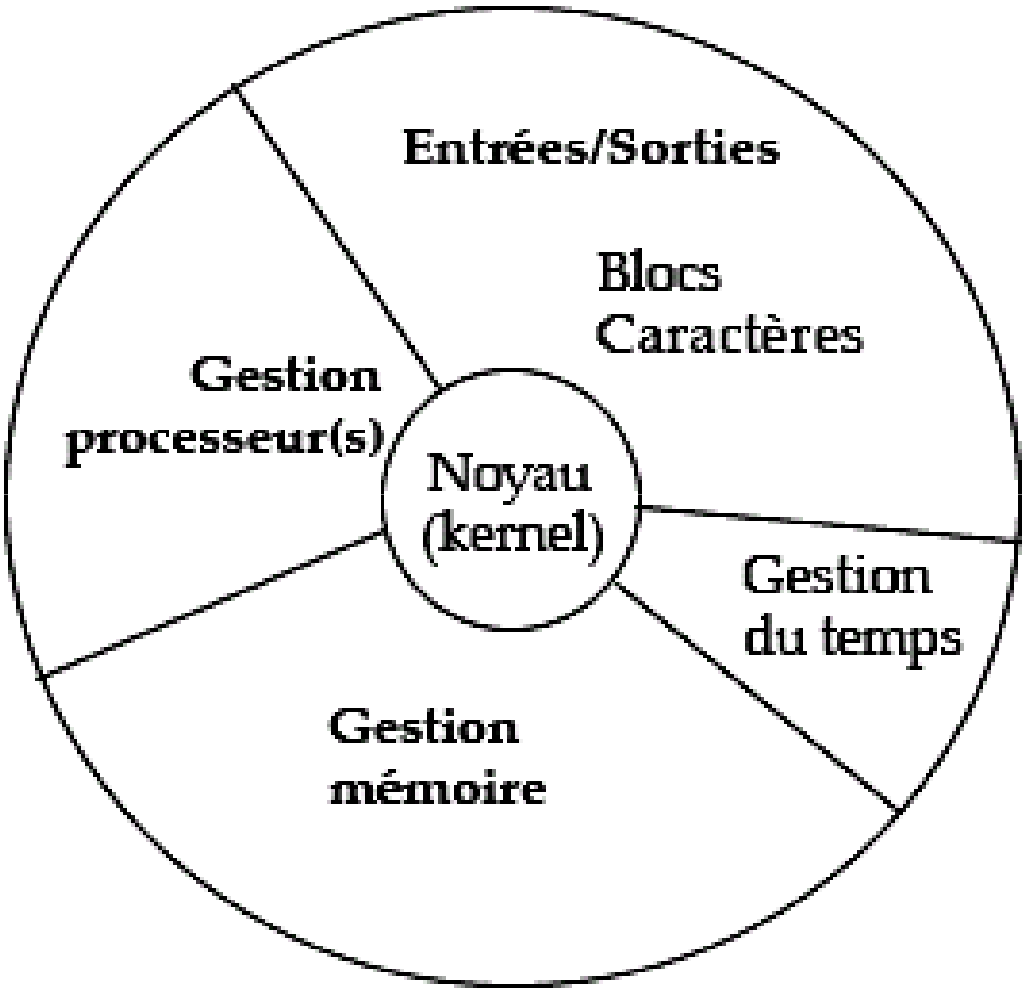
-la gestion des erreurs

-l'évitement des conflits

Niveau 1 : le système d'exploitation

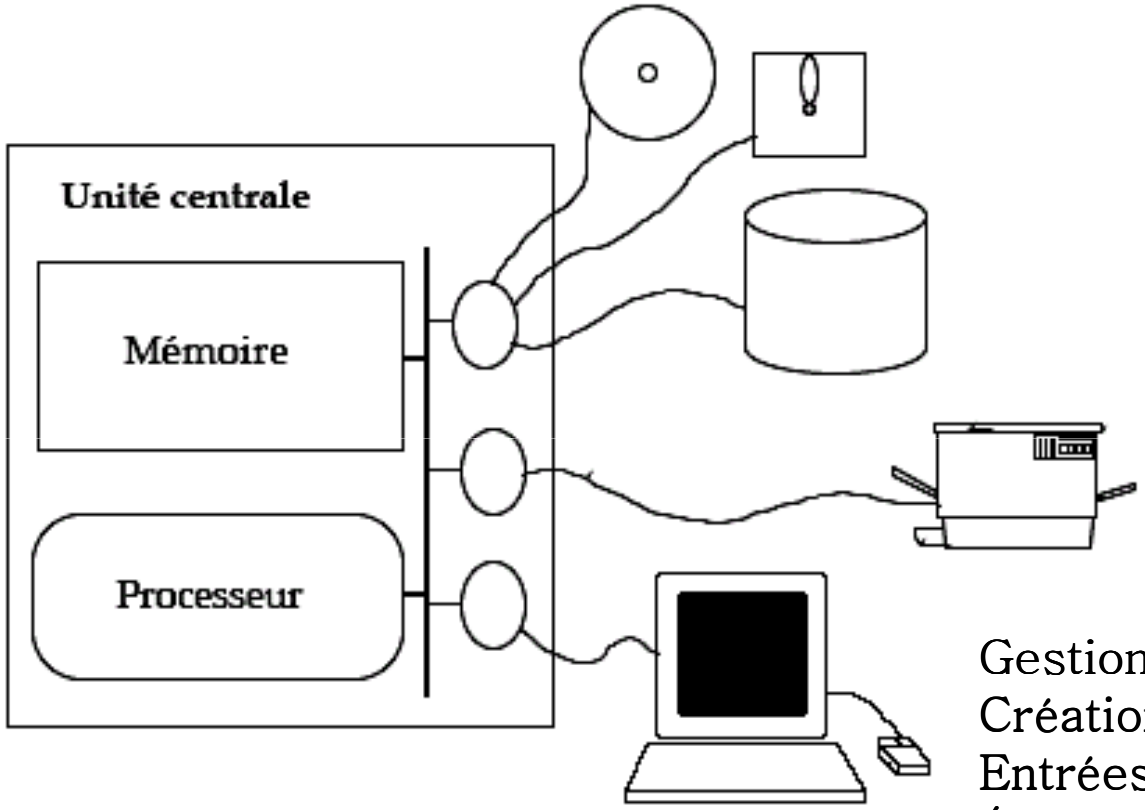


Niveau 0 : exécutif



Exemples (1) : ordinateur personnel

type MS-DOS, Windows, MacOS, …

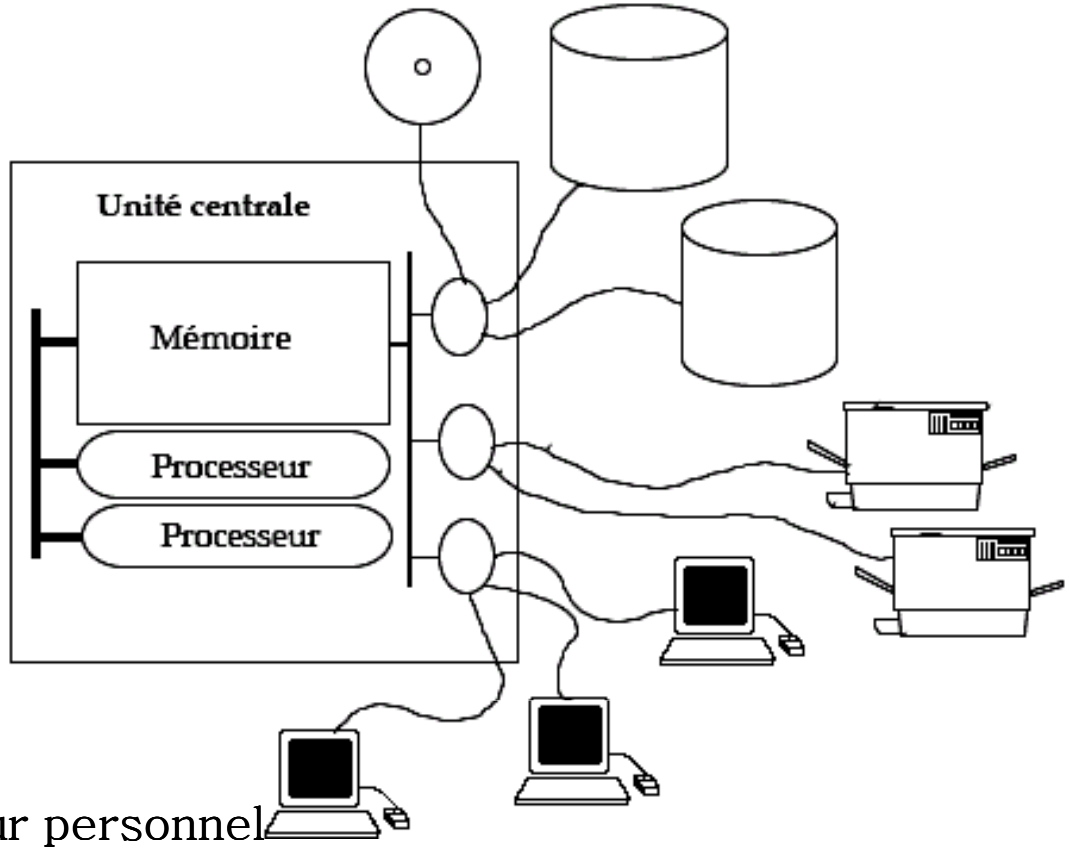


Gestion des fichiers
Création, Modification, Suppression
Entrées/Sorties
Écran, Imprimantes
Exécution de programmes
Interface utilisateur
langage de commande ou graphique

Exemples (2) :

système à temps partagé

type Unix, VM-CMS, …



Fonctions d'un système d'ordinateur personnel

+ Gestion des utilisateurs

Espace de travail, droits d'accès, ...

+ Partage des ressources communes

Imprimantes, Disques, …

+ Gestion des communications

Informations partagée, communications entre utilisateurs

Historique

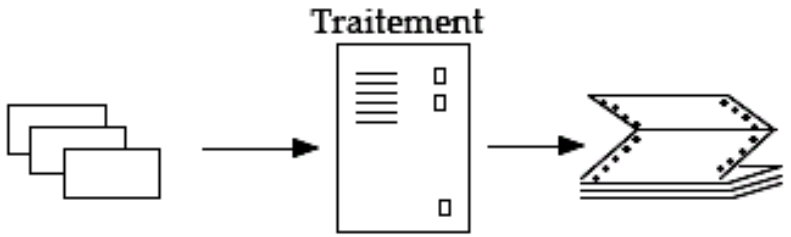
De la porte ouverte… (fin 50)

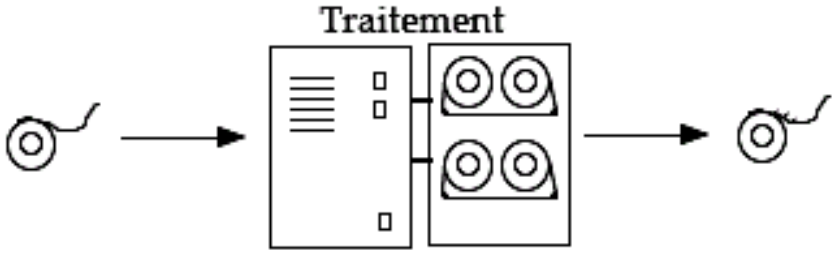
Réservation de l'ordinateur

Programmation aux clés, puis par téléty

Enchaînement par paquets de cartes

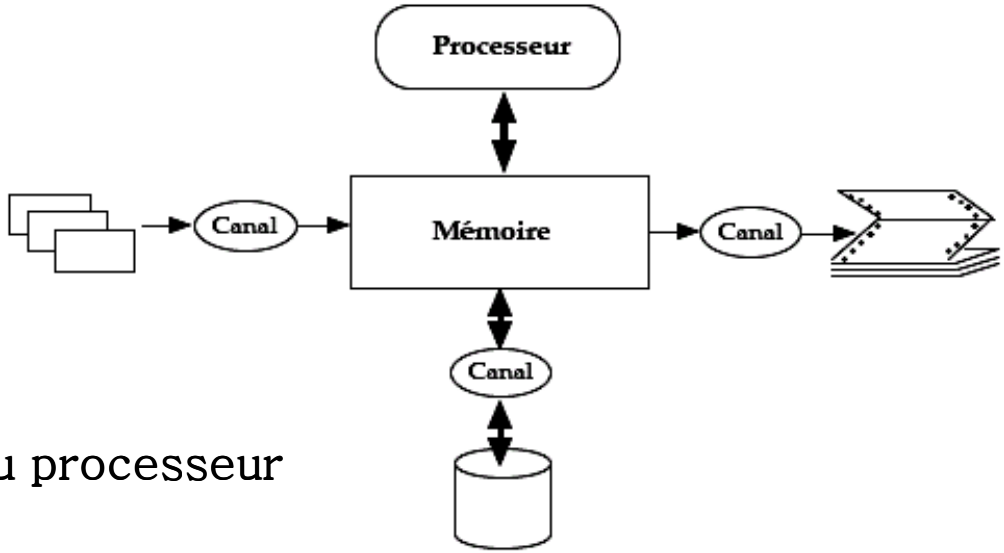
…au traitement par lots (début 60)





Entrées/Sorties tamponnées (65-70

Multi-programmation…

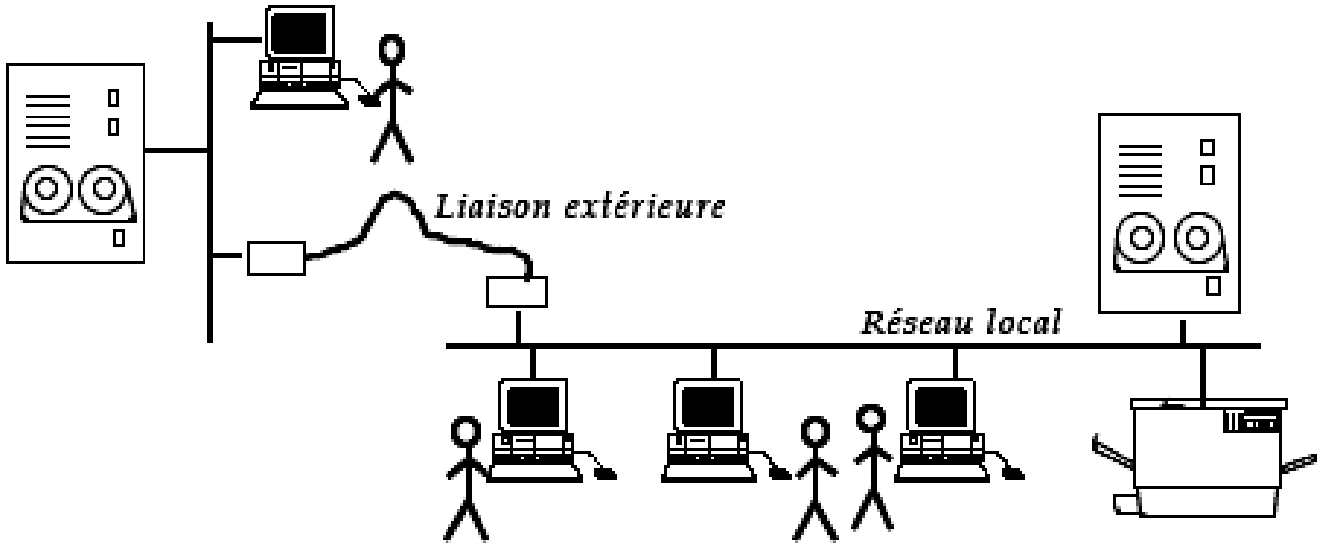


…et temps partagé (70-)

amélioration du taux d'occupation du processeur

multi-utilisateurs

Réseaux, machines individuelles (75-80)

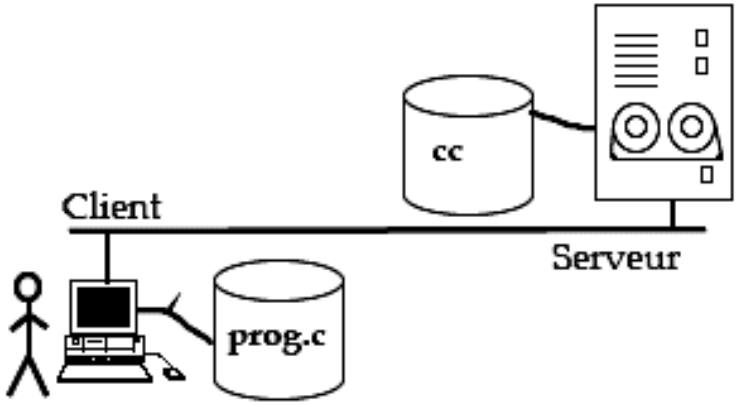


Systèmes répartis et client/serveur (85-)

Accès transparent

Partage de ressources coûteuses

Maintenance logicielle centralisée



Services offerts par un SE

- Abstraction Machine

  - Programme voient une machine idéale: Cache particularités du matériel et donne l'illusion de matériel non partagé (Processus).

  - Librairies Services simplifient programmes.

- Gestions de Ressources

  - Processeur(s).

  - Mémoire.

  - Périphériques (Entrées/Sorties)

  - Système de Fichiers.

- Interface Utilisateur

  - Permet à l'utilisateur de communique avec le système.

  - Gestion Utilisateurs.

Processus

- Abstraction Principale du SE.
- Plusieurs Processus → Multitâche.
- Tâche qui s'exécute dans le système

  - "Croit" qu'il est seul à accéder à la machine
    - Mémoire
    - CPU
    - Périphériques
  - Le SE gère le "multiplexage" de toutes ces ressources.

Gestion des ressources

- Ressource sont partagées entre processus

- Problèmes:
  - Isoler des erreur (un processus isolé des autres)
  - Partage équitable
  - Sécurité
  - Interblocage

- Géré par le cœur du SE: noyau

Abstractions utiles (Services)

- Communication entre processus.
- Gestion des erreurs.
- Gestion de fichier.
- Gestion des utilisateurs.
- Gestion du réseau.
- Gestion de l'affichage graphique.
- Etc..

Les différentes classes de systèmes d'exploitation

Selon les services rendus

- ***mono/multi tâches*** :

Multi−tâches : capacité du système à pouvoir exécuter plusieurs processus simultanément ; par exemple effectuer une compilation et consulter le fichier source du programme correspondant.

C'est le cas d'UNIX, d'OS/2 d'IBM et de Windows 95.

- ***mono/multi−utilisateurs*** :

Multi−utilisateurs : capacité à pouvoir gérer un panel d'utilisateurs utilisant simultanément les mêmes ressources matérielles.

C'est le cas d'UNIX, de MVS, de Gecos ...

Selon leur architecture

- ***Systèmes centralisés*** :

  •L’ensemble du système est entièrement présent sur la machine

  •considérée.

  •Les machines éventuellement reliées sont vues comme des entités

  •étrangères disposant elles aussi d’un système centralisé.

  •Le système ne gère que les ressources de la machine sur laquelle il est présent.

C’est le cas d’UNIX, même si les applications réseaux (X11, FTP, Mail ...) se sont développées.

- ***Systèmes répartis* (*distributed systems)*** :

•Les différentes abstractions du système sont réparties sur un ensemble (domaine) de machines (site).

•Le système d'exploitation réparti apparaît aux yeux de ses utilisateurs comme une machine virtuelle monoprocesseur même lorsque cela n'est pas le cas.

•Avec un système réparti, l'utilisateur n'a pas à se soucier de la localisation des ressources. Quand il lance un programme, il n'a pas à connaître le nom de la machine qui l'exécutera.

•Ils exploitent au mieux les capacités de parallélisme d'un domaine.

•Ils offrent des solutions aux problèmes de la résistance aux pannes.

Selon leur capacité à évoluer

- ***Systèmes fermés* (ou propriétaires)** :

Extensibilité réduite : Quand on veut rajouter des fonctionnalités à un système fermé, il faut remettre en cause sa conception et refaire une archive (système complet).
C'est le cas d'UNIX, MS−DOS ...
Il n'y a aucun ou peu d'échange possible avec d'autres systèmes de type différent, voir même avec des types identiques.

- ***Systèmes ouverts*** :

Extensibilité accrue : Il est possible de rajouter des fonctionnalités et des abstractions sans avoir à repenser le système et même sans avoir à l'arrêter sur une machine.
Cela implique souvent une conception modulaire basée sur le modèle « **client−serveur** ».
Cela implique aussi une communication entre systèmes, nécessitant des modules spécialisés.

Selon l'architecture matérielle qui les supporte

- ***Architecture* MONOPROCESSEUR *(temps partagé ou multi−programmation)*** :

Ressource processeur unique : Il a fallu développer un mécanisme de gestion des processus pour offrir un (pseudo) parallélisme à l'utilisateur : c'est la multi−programmation ; il s'agit en fait d'une commutation rapide entre les différents processus pour donner l'illusion d'un parallélisme.

- ***Architectures* MULTIPROCESSEURS *(parallélisme)*** :

On trouve une grande variété d'architectures multiprocesseurs :

SIMD (Single Instruction Multiple Data) : Tous les processeurs exécutent les mêmes instructions mais sur des données différentes.

MIMD (Multiple Instructions Multiple Data) :

Chaque processeur et complètement indépendant des autres et exécute des instructions sur des données différentes.

Pipeline : Les différentes unités d'exécution sont mises en chaîne et font chacune partie du traitement à effectuer.

On parle aussi d’architecture *faiblement* ou *fortement couplée*.

- ***Architecture fortement couplée*** : Ce sont principalement des architectures à mémoire commune.
- ***Architecture faiblement couplée*** : Ce sont des architectures où chaque processeur possède sa propre mémoire locale ; c’est le cas d’un réseau de stations.
- ***Architecture mixte*** : Ce sont des architectures à différents niveaux de mémoire (commune et privée).

Un ordinateur principal (mainframe) du milieu des annnées ‘60

disques

rubans

lecteur de cartes



UCT
(mémoire probabl.
autour de 500K)

console opérateur



Musée de l’histoire de l’informatique http://www.computerhistory.org/

Source: **http://www.tietokonemuseo.saunalahti.fi/eng/kuva_32_eng.htm**

Finnish Data Processing Museum Association

Opérateur lisant un paquet de cartes perforées

