Cours de
« système d'exploitation »

1ère année

IUT de Caen
Département d'Informatique
(François Bourdon)

Plan du cours

1ère ANNEE

I - Présentation générale des systèmes d'exploitation

II - Généralités sur UNIX / historique

III - Les bases du système

1. Système de fichiers (SGF) et commandes associées
2. Processus et commandes associées
3. Redirections d'entrées/sorties
4. Premier niveau de communication entre processus

IV - Les langages de commande (Shell)

V - Le langage « C » (avancé) et UNIX

VI – Le système de fichiers, représentation interne

VII – Les processus : concepts avancés

2ème ANNEE

I - Synchronisation de processus

II - La communication par signaux entre processus

III - La communication avancée entre processus : IPC

IV - La communication sur le réseau entre processus

V - La gestion de la mémoire

I. Présentation générale des systèmes d'exploitation

Plan

A - Introduction

B - Deux fonctions

- machine étendue/virtuelle
- gestionnaire de ressources

C - Historique des Systèmes d'Exploitation

- 1ère génération (1945 - 1955)
- 2ème génération (1955 - 1965)
- 3ème génération (1965 - 1980)
- 4ème génération (1980 - 1990)
- 5ème génération (1990 - ????)

D - Les différentes classes de Systèmes d'Exploitation

- selon les services rendus
- selon leur architecture
- selon leur capacité à évoluer
- selon l'architecture matérielle qui les supporte

A. Introduction

Deux catégories de logiciels :

Les programmes systèmes pour le fonctionnement des ordinateurs,

les programmes d'application qui résolvent les problèmes des utilisateurs.

> Le programme « système d'exploitation » est le programme fondamental des programmes systèmes. Il contrôle les ressources de l'ordinateur et fournit la base sur laquelle seront construits les programmes d'application.

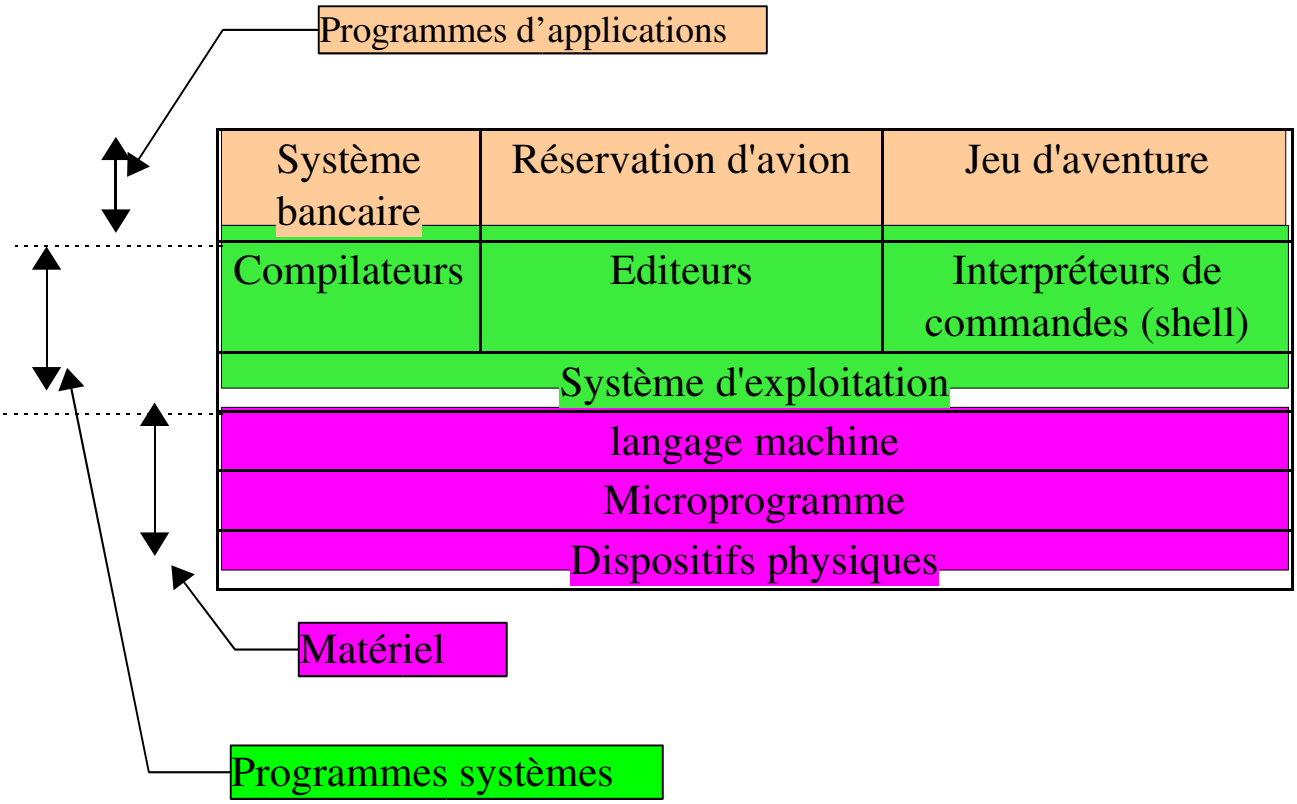
Deux modes de fonctionnement :

> Le mode **noyau** ou **superviseur** et
>
> le mode **utilisateur** (compilateur, éditeur, programmes utilisateurs ...).

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.



Dispositifs physique = Ils se composent de circuits intégrés, de fils électriques, de périphériques physiques ...

Microprogramme = C'est un logiciel de contrôle des périphériques (interprête).

Langage machine = C'est un ensemble (entre 50 et 300) d'instructions élémentaires (ADD, MOVE, JUMP) pour effectuer le déplacement des données, des calculs, ou la comparaison de valeurs.

Système d'exploitation = C'est un ensemble d'instructions plus simples, comme LIRE UN BLOC DU FICHIER.

B. DEUX FONCTIONS

MACHINE ETENDUE ou VIRTUELLE

GESTIONNAIRE de RESSOURCES

Machine étendue ou encore **machine virtuelle.**

Son rôle est de 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) ...

Ex. **READ** et **WRITE** = 13 paramètres sur 9 octets ; en retour le contrôleur renvoie 23 champs d'état et d'erreur regroupés sur 7 octets.

Gestionnaire de ressources.

Un ordinateur se compose de ressources (périphériques, mémoires, terminaux, disques ...).

> Le système d'exploitation permet l'ordonnancement et le contrôle de l'allocation des processeurs, des mémoires et des périphériques d'E/S entre les différents programmes qui y font appel.

Par exemple 3 programmes essaient d'imprimer simultanément leurs résultats sur une même imprimante :

=> recours à un fichier tampon sur disque.

Autre exemple, l'accès concurrent à une donnée ; lecture et écriture concurrentes (par deux processus) sur un même compteur.

Ce rôle de gestionnaire de ressources est crucial pour les systèmes d'exploitation manipulant plusieurs tâches en même temps (multi-tâches).

Plusieurs fonctionalités de gestion

du **processeur** : allocation du processeur aux différents programmes.

des **objets externes** : principalement les fichiers.

des **entrées-sorties** : accès aux périphériques, via les pilotes.

de la **mémoire** : segmentation et pagination.

de la **concurrence** : synchronisation pour l'accès à des ressources partagées.

de la **protection** : respect des droits d'accès aux ressources.

des **accès au réseau** : échange de données entre des machines distantes.

C. Historique des SYSTEMES d'EXPLOITATION

Tout système d'exploitation dépend étroitement de l'architecture de l'ordinateur sur lequel il fonctionne.

La 1ère génération (1945 - 1955) :
les tubes à vide et les cartes enfichables.

La 2ème génération (1955 - 1965) :
les transistors et le traitement par lots.

La 3ème génération (1965 - 1980) :
les circuits intégrés et la multi-programmation.

La 4ème génération (1980 - 1990) :
les ordinateurs personnels.

La 5ème génération (1990 - ????) :
les ordinateurs personnels portables et de poche.

La 1ère génération (1945 - 1955) : les tubes à vide et les cartes enfichables.

Il n'existait pas de système d'exploitation.

Les utilisateurs travaillaient chacun leur tour sur l'ordinateur qui remplissait une salle entière.

Ils étaient d'une très grande lenteur.

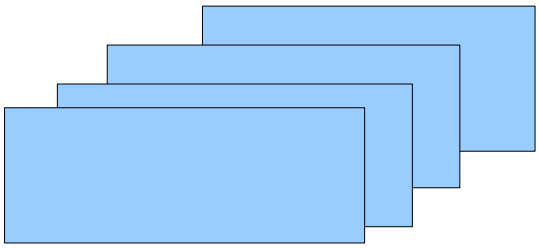
Ils étaient d'une très grande fragilité.

La 2ème génération (1955 - 1965) : les transistors et le traitement par lots.

Le passage aux **transistors** rendait les ordinateurs plus **fiables**.

Ils pouvaient être **vendus** à des utilisateurs (grandes compagnies, université ou administrations.

Mais devant les coûts d'équipement élevés on réduisit les temps grâce au **traitement par lots**.



Les utilisations principales étaient le **calcul scientifique** et **l'ingénierie** (équations différentielles).

Apparition des langages **FORTRAN** et **assembleur**.

Apparition des systèmes d'exploitation (SE) : FMS (Fortran Monitor System) et IBSYS (IBM 7094).

La 3ème génération (1965 - 1980) : les circuits intégrés et la multiprogrammation.

Amélioration des coûts et des performances (circuits intégrés).

Une **famille d'ordinateurs compatibles** entre eux.

Une **seule architecture** et un même jeu d'instructions.

Des **ordinateurs uniques** pour les calculs scientifiques et commerciaux.

Apparition du **spoule** (spool, Simultaneous Peripheral Operation On Line) pour le transfert des travaux des cartes vers le disque.

Apparition de la **multiprogrammation** (partitionnement de la mémoire pour des tâches différentes).

Mais, un système d'exploitation **énorme et très complexe** pour satisfaire tous les besoins (plusieurs millions de lignes d'assembleur).

Apparition du **partage de temps**, une variante de la multiprogrammation (chaque utilisateur possède un terminal en ligne) ; naissance du système MULTICS (MULTiplexed Information and Computing Service) pour ordinateur central.

Apparition des **mini-ordinateurs** (DEC PDP-1 en 1961, 4K mots de 18 bits, pour un prix de 120 000 $).

K. Thompson écrivit une version simplifiée (mono-utilisateur) de MULTICS ; B. Kernighan l'appela avec humour **UNICS** (Uniplexed Information and Computer Service) ; ce nom allait devenir **UNIX**[1].

D. Ritchie se joignit à K. Thompson pour **réécrire UNIX en langage C** ; ce système d'exploitation a été le plus porté sur toutes sortes de machine.

[1]UNIX est une marque déposée par les Laboratoires AT&T Bell.

La 4ème génération (1980 - 1990) : les ordinateurs personnels.

Ils sont dû au développement des **circuits LSI** (Large Scale Integration) contenant des centaines de transistors au cm2.

Ils ont la **même architecture que les mini-ordinateurs** mais leur prix est beaucoup moins élevé.

Il existe deux systèmes d'exploitation principaux : **MS-DOS** (Microsoft Inc.) et **UNIX**.

MS-DOS intègre petit à petit des concepts riches d'UNIX et de MULTICS.

Dans le milieu des années 80, on voit l'**apparition de réseaux d'ordinateurs individuels** qui fonctionnent sous des systèmes d'exploitation en réseau ou des systèmes d'exploitation distribués.

La 5ème génération (1990 - ????) : les ordinateurs personnels portables et de poche.

Apparition des PIC (Personal Intelligent Communicator de chez Sony) et des PDA (Personal Digital Assistant, comme le Newton de chez Apple), grâce à l'intégration des composants et l'arrivée des systèmes d'exploitation de type « micro-noyau ».

Ils sont utiles pour les « **nomades** » et les systèmes de gestion des informations (recherche, navigation, communication).

Ils utilisent la **reconnaissance de caractère** (OCR) et les modes de communication synchrone et asynchrone (mode messagerie).

Très bon marché, ils sont capables de **se connecter à des ordinateurs distants et performants**.

Les **systèmes d'exploitation de type « micro-noyau »** sont modulaires (un module par fonction) ; ils peuvent être réalisés avec plus ou moins de modules et donc adaptables à des très petites machines (PDA et PIC).

D. 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 (o

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.

C'est le cas entre UNIX BSD et SV.

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).

<u>Remarque</u> : Il n'y a pas de système universel pour cette multitude d'architectures. Les constructeurs de supercalculateurs ont toujours développés leurs propres systèmes. Aujourd'hui, compte tenu de la complexité croissante des systèmes d'exploitation et du coût inhérent, la tendance est à l'harmonisation notamment via le développement de systèmes polyvalents tels que les systèmes répartis.

Un cas particulier, les systèmes « temps-réel ».

Systèmes temps-réel :

> Ce sont des systèmes pour lesquels l'exécution des programmes est soumise à des contraintes temporelles. Les résultats de l'exéution d'un programme n'est plus valide au delà d'un certain temps connu et déterminé à l'avance.

Généralement, on trouve des systèmes « temps réel » dans les systèmes embarqués (satellites, sondes, avions, trains, téléphones portables, ...).

On distingue deux types de contraintes temporelles :

les contraintes strictes et les contraintes relatives.

Pour garantir ces contraintes, le système possède des mécanismes spécifiques dont le but est de réduire l'indéterminisme des durées d'exécution des programmes.

C'est le cas de **Linux-RT**.