Systèmes d'exploitations : généralités

I - Introduction

Le système d'exploitation (Operating System) est la couche logicielle qui est entre le matériel et le programme utilisateur. C'est lui qui traduit un ensemble d'instructions et de programmes de base en langage directement compréhensible par la machine. Dans une perception en couches plus détaillée, on place le système d'exploitation comme ci-dessous :

- Applications
- Éditeur / compilateur / interpréteur de commandes
- Système d'exploitation
- Langage machine
- Micro programmation
- Dispositif physique

Les plus connus sont Windows, MacOS, UNIX, Linux ...
Un système d'exploitation peut être envisagé selon deux points de vue :

- Utilisateur
- Machine

Le point de vue de l'utilisateur correspond à une vision descendante : de l'utilisateur vers le matériel. Le système d'exploitation est alors une sorte de machine virtuelle, extension de la machine réelle, plus simple à exploiter. Il offre une base pour le développement et l'exécution d'applications. Le système d'exploitation permet par exemple :

- D'effectuer plusieurs processus en même temps
- Une gestion directe de la mémoire
- Un accès simplifié aux différents périphériques
- Une gestion facilité de l'arborescence des répertoires et des fichiers
- Une gestion de la protection (ex : droit des différents utilisateurs)

Le point de vue machine correspond à une vision ascendante : du matériel vers l'utilisateur. Le système d'exploitation est vu comme un gestionnaire des ressources physiques de l'ordinateur. Il gère notamment :

- Le processeur
- La mémoire
- Les périphériques (ex : les priorités, les files d'attente)
- Le systèmes de fichiers (emplacement physiques, table d'allocation)
- La protection
- Les erreurs (ex: traitement du signal de l'imprimante : plus de papier)

II – Historique

L'apparition et l'évolution des systèmes d'exploitations est lié à l'évolution des ordinateurs.

1) Première génération

Dans les années 40, les premiers « ordinateur » apparaissent, les programmes et les données sont codés directement en binaire sur des tableaux d'interrupteurs, puis plus tard (1950) sur des cartes perforées (la présence ou l'absence de trous correspondent à 0 ou 1), ils sont chargés en mémoire, exécutés et mis au point à partir d'un pupitre de commande. C'est la même équipe qui conçoit, construit, programme, administre et maintient la machine.

L'arrivée de l'assembleur facilite l'écriture des programmes, mais l'utilisation des machines est toujours la même. L'ordinateur ne traite qu'un programme à la fois. Il faut pour chaque travail (job), en amont insérer les cartes dans un lecteur de cartes, en aval imprimer les résultats, ce sont des opérations très lentes par rapport au travail du CPU. Par la suite, c'est un opérateur qui se charge de l'exploitation de la machine mais le principe reste le même.

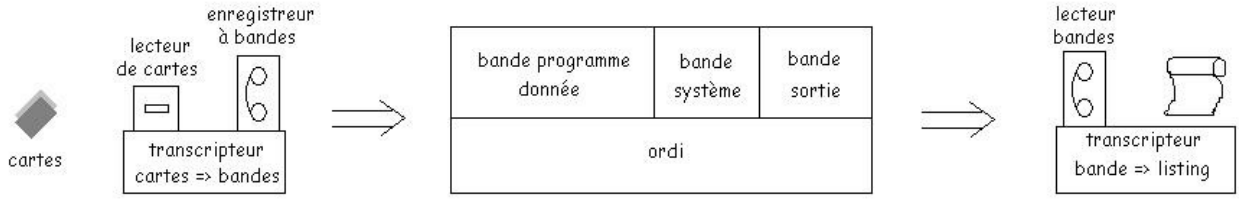
2) Deuxième génération

Avec l'apparition du transistor (1955 environ), les ordinateurs sont beaucoup plus fiables, mais coûtent très cher (seules de grandes entreprises privées ou publiques en possèdent). Ce sont maintenant des équipes différentes qui conçoivent, construisent, programment, administre et maintiennent la machine.

Pour chaque travail (job) à réaliser, le programmeur transmet son paquet de cartes perforés (programmes en assembleur ou en FORTRAN) à l'opérateur qui les soumet au compilateur puis à l'ordinateur. Une fois l'exécution terminée, l'opérateur récupère le résultat sur l'imprimante. Comme les ordinateurs sont très coûteux, on cherche à réduire les temps d'attente en automatisant les opérations manuelles.

Les gros ordinateurs disposent alors de trois dérouleurs de bande : un pour le système d'exploitation, un pour le

programme et ses données, un pour les données en sortie. En annexe, il existe des machines pour transcrire les cartes perforées (apportés par les programmeurs) sur bandes et des machines pour imprimer les données en sortie (contenues sur la bande) sur papier. On parle d'impression off-line (l'imprimante n'est pas directement connectée à l'ordinateur). Mais le calculateur principal ne fonctionne pas tout le temps : beaucoup de temps est encore perdu lors des déplacements des opérateurs qui sont chargés d'alimenter les machines en cartes, papier, bandes.



On procède au traitement par lots (batch) : plusieurs travaux sont transcrits sur une même bande d'entrée. Le calculateur principal lit le 1er job, puis lorsque celui-ci est terminé, lit le 2nd, etc... jusqu'à la fin de la bande. Un interprète de commande permet le chargement du programme et des données puis l'exécution du programme, le moniteur est le programme charger du séquencement des travaux des utilisateurs et de la continuité des opérations. Le moniteur des années 60 est le précurseur du système d'exploitation.

3) Troisième génération

Au milieu des années 60, l'apparition des circuits intégrés permet une grande avancée sur la performance et le coût des ordinateurs. Des familles de machines partageant le même langage machine, le même système d'exploitation se mettent en place. Les travaux sont entièrement traités par l'ordinateur sans passer par des machines annexes, les jobs sont stockés sur des fichiers disques. Comme l'accès sur un disque est aléatoire (dans le sens de non-séquentiel) le moniteur peut choisir l'ordre des travaux.

Dans le moniteur, le planificateur (scheduler) assure cette tâche. Mais les moniteurs batch exécutent toujours un seul job à la fois, à tout instant un seul programme se trouve en mémoire et peut seul exploiter le CPU.

On arrive à la multiprogrammation. La mémoire est partagée entre différents programmes en attente d'une donnée en entrée par exemple peut être suspendu temporairement au profit d'une tâche. Le but étant d'exploiter au maximum le CPU. On appelle cette technique le spoule (de SPOOL : Simultaneous Peripheral Operation On Line). Dans le système d'exploitation, l'allocateur (dispatcher) s'occupe de la gestion instantanée du CPU en tenant compte de la planification établie par le scheduler. Il a fallut mettre en place des systèmes de contrôle des accès mémoires, de protection des données.

On voit apparaître également le temps-partagé (time-sharing) pour des ordinateur multi-utilisateurs (un ordinateur central connecté à plusieurs terminaux).

4) Quatrième génération

Depuis les années 80, on assiste au développement des ordinateurs personnels, à l'amélioration constante du taux d'intégration des circuits (LSI : Large Scale Integration, VLSI, Very LSI), à la baisse des coûts et au développement des réseaux de communications, au développement des réseaux locaux, à l'explosion d'internet.
Les interfaces Homme/Machine sont toujours améliorées.

III – Les grands principes

1) Les processus

Un programme est une suite statique d'instructions.
Un processus est une instance d'un programme en train de s'exécuter. Il est représenté par son code, ses données, sa pile d'exécution, les valeurs courantes des registres du processeurs, son état (suspendu, en cours) , sa liste des fichiers ouverts...
On peut décomposer le déroulement d'un calcul par exemple en un processus d'entrée de données, un processus pour l'opération à effecteur, un processus pour enregistrer la données en sortie, etc...
L'activité d'un système d'exploitation peut être vue comme un ensemble de processus concurrent. On distingue les processus lourds possédant leurs propres données et les tâches légères (threads) qui interviennent dans les processus lourds.

2) Les appels système, le mode noyau

Lorsqu'un programme d'application fait appel à une procédure spéciale fournie par le système d'exploitation, on dit qu'il fait un appel système : system calls (l'ensemble de ces procédures est aussi appelé API, Application Programming Interface). Par exemple : Besoin d'une donné sur le disque dur

Les appels système se font au moyen d'instruction spéciales : les traps. Le principes des traps est de dérouter l'exécution d'un programme d'application pour exécuter le code du système d'exploitation. Les programmes d'applications s'exécutent en mode utilisateur (mode non privilégié). Lorsqu'il font appel au système au moyen des traps, celles-ci s'exécutent en mode superviseur, appelé aussi mode noyau (kernel), qui est un mode privilégié. Au niveau du processeur, le mode noyau se distingue du mode utilisateur par le fait qu'il n'y a qu'en mode noyau que l'on peut :

- Accéder aux codes et aux données que le système utilise
- Utiliser les instructions de modifications de la table des segments mémoire
- Utiliser les instructions de lecture et d'écriture sur les ports d'entrées/sorties.

NB : la différence mode noyau/mode utilisateur est gérée au niveau du processeur (ne pas confondre avec la notion de super-utilisateur gérer au niveau logiciel du système d'exploitation)

On appelle interruption un événement qui modifie le flux de commande d'un programme. Cela peut être une interruption :

- Matériel (reprise en compte d'une requête système)
- Logiciel (par ex : division par zéro)

Il existe plusieurs mécanismes d'interruption
Méthode simple :

- Le programme en cours est arrêté
- le système de gestion des interruptions prend le contrôle
- Une procédure spécifique aux types de l'interruption est lancé
- Lorsque la procédure est terminé, le programme interrompu reprend son exécution

L'adresse de l'instruction interrompue doit être sauvegardée. Lors de la reprise, la machine doit se retrouver exactement dans l'état où elle était au moment de l'interruption (restauration de certains registres, notamment de l'ancien PSW : Program Status Word = registre d'état du programme...).

3) Les fichiers

Le système de fichiers, organisé en arborescence, permet de classer les informations persistantes (durée de vie supérieure à celle des processus). Le système d'exploitation permet les opérations classiques sur les fichiers : création, suppression, ouverture, fermeture, positionnement, lecture, écriture. Il doit veiller à la sauvegarde et à la cohérence des données. Par exemple : Il ne doit pas permettre l'accès à un même fichier en écriture pour deux processus concurrent.
ex :

- processus p1 : additionner A et B et met le résultat dans C
- processus p2 : modifie A
- processus p1 : affiche contenue de A + contenue de B = contenue de C

4) Structure générale

On peut modéliser les systèmes d'exploitation par une représentation en couches fonctionnelles, les couches les plus basses étant les plus proches du matériel, les plus hautes les plus proches de l'utilisateur.

Utilisateurs

interface utilisateur/interpréteur de commande
allocation des ressources/planifications du travail
gestion des fichiers
organisations des E/S (Entrées/sorties)
gestion de la mémoire
noyau (gestion des processus des interruptions / allocation CPU)

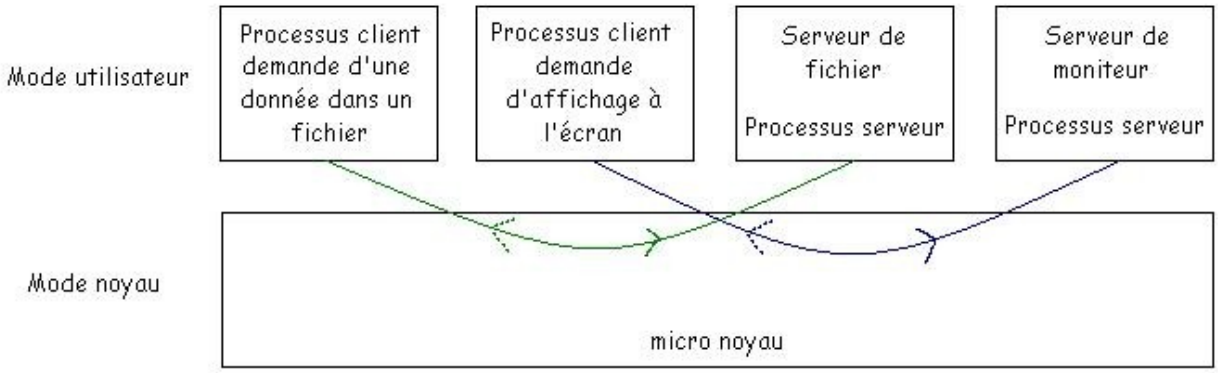
Hardware

NB : les compilateurs, les éditeurs de lien, les éditeurs de textes, ne sont pas des fonctions de base du système d'exploitation, ils sont considérés comme des programmes d'applications particuliers.

NB :

- une partie du système d'exploitation est stockée dans la ROM (BIOS, gestion interruptions des périphériques, contrôles carte mère (température...), POST, bootstrap (adresse sur tel ou tel disque pour charger le noyau),...)
- une autre partie, le noyau, stockée sur une mémoire genre disque dur par exemple, est chargée dans la RAM lors du démarrage.

Il y a plusieurs approches possibles pour l'implémentation d'un système d'exploitation. L'une d'elle est appelée le modèle Client-Serveur. L'idée est de découper une demande (un appel système) en processus client qui envoie une requête à un processus serveur, celui-ci effectue le service et renvoie le résultat au client, le rôle du noyau se réduit alors a la gestion des communications clients-serveur. Cela permet entre autre de mettre un maximum de code dans les couches les plus hautes (mode utilisateur) et de minimiser la partie fonctionnant en mode noyau (on parle de micro-noyau). Le système d'exploitation est alors constitué de nombreuses parties indépendantes les unes des autres, donc plus simples et plus faciles à maintenir.



IV – Les différents systèmes

Il n'y a pas de système d'exploitation idéal, cela dépend du contexte d'utilisation.

Les systèmes généraux sont multi-utilisateurs et multi-tâches, ils doivent couvrir un grand nombre de domaines d'applications.

Les systèmes mono-utilisateur n'autorisent qu'un seul utilisateur, l'utilisation des périphériques est plus simple, ils peuvent être multi-tâches, mais il n'y a pas de notion de protection (des données d'un utilisateur par rapport à un autre).

Les système de contrôle de processus sont utilisés en milieu industriel pour les machines-outils, ils doivent réagir rapidement à des données issue de capteur, ils sont orientés temps-réel et fiabilité.

Les système de serveurs de fichiers gèrent de grands ensembles d'informations, interrogeables à distance, il faut des temps de réponse courts.

Les systèmes transactionnels contrôlent de grandes bases de données et doivent assurer leurs cohérences, par exemple dans le cas de commande d'un certains produit par un client, il faut mettre à jour les stocks et ne pas permettre une autre commande simultanée du même produit si le stock n'est pas suffisant.

Pour ces deux derniers types de système, la tendance est de les construire en surcouche de système généraux.