

Chapitre 1: introduction aux systemes d'exploitation

Ouvrages de reference

- *Systemes d'exploitation*, Andrew Tanenbaum. (Très interessant)
- *Principes des Systemes d'Exploitation*, A. Silberschatz et P. B. Galvin.

Contents

1	Introduction	2
2	Qu'est-ce qu'un systeme d'exploitation ?	2
3	Pourquoi etudier les SE ?	2
4	Fonctions d'un SE	2
5	Organisation en couches d'un SE	3
6	Evolution des systemes informatiques	4
6.1	1ere generation (1945 - 1955): Les tubes a vide et les cartes perforées	4
6.2	2eme generation (1956 - 1965): Transistors et traitement par lots	4
6.3	3eme generation (1960 - 1980): Les circuits integrés et la multiprogrammation	5
6.4	4eme generation (1980 - aujourd'hui): Les ordinateurs personnels	7
7	Exemples de SE	7
8	Conclusion	7

1 Introduction

La théorie des SE a été développée surtout dans les années 1960. À cette époque, il y avait des machines très peu puissantes avec lesquelles on cherchait à faire des applications comparables à celles d'aujourd'hui. Ces machines devaient parfois desservir des dizaines d'utilisateurs. D'où le besoin de développer des principes pour optimiser l'utilisation d'un ordinateur. Principes qui sont encore utilisés.

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

Un système d'exploitation (SE), en anglais Operating System (OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications de l'utilisateur. C'est une sorte d'interface entre l'utilisateur et le matériel afin de cacher la spécificité et la complexité du matériel (FIGURE 1).

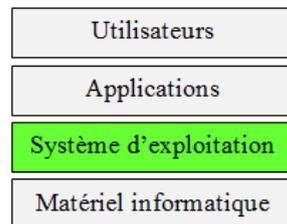


Figure 1: Le modèle en couches d'un ordinateur

Un SE joue aussi le rôle de gestionnaire de ressources (processeurs, mémoires, périphériques, etc).

3 Pourquoi étudier les SE ?

- 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 oeuvre dans son propre domaine: pas la peine de réinventer la roue;
- C'est un sujet intéressant, dont l'objectif est la recherche de l'efficacité, nécessitant une étude théorique approfondie mais dont l'objectif est la fourniture de solutions réalisables en pratique.

4 Fonctions d'un SE

Les fonctions couvertes par un système d'exploitation sont divers:

- **Gestion du processeur:** allocation du processeur aux différents programmes;
- **Gestion de la mémoire centrale:** segmentation et pagination;

- **Gestion des entrées/sorties:** accès aux périphériques;
- **Gestion de la concurrence:** synchronisation pour l'accès à des ressources partagées;
- **Gestion de la protection:** respect des droits d'accès aux ressources;
- etc.

Les algorithmes mis en oeuvre pour réaliser ces différentes fonctions varient beaucoup d'un système à l'autre selon les objectifs visés et ils peuvent être par exemple très différents selon que le concepteur cherche avant tout une utilisation optimum des ressources ou, au contraire, un temps de réponse minimum. D'une manière générale, l'implantation de toutes ces fonctions conduit à des logiciels très complexes qui doivent nécessairement être structurés pour pouvoir être maîtrisés.

5 Organisation en couches d'un SE

On peut organiser les couches d'un système d'exploitation selon la FIGURE 2.

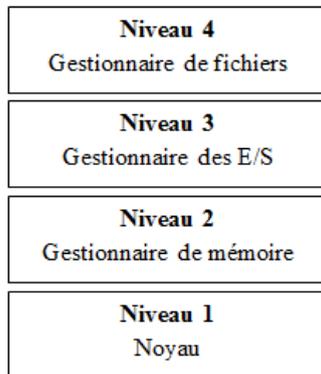


Figure 2: Organisation en couches d'un système d'exploitation

- **Le noyau (Kernel):** les systèmes d'exploitation sont conçus à partir d'un noyau (*kernel* en anglais) autour duquel gravitent toutes les autres couches. Il est en relation directe avec la machine physique. Il s'occupe de l'allocation du processeur aux programmes, ainsi que l'exécution des entrées/sorties physiques;
- **Gestionnaire de mémoire (Memory manager):** il se charge du partage (allocation et libération) de la mémoire centrale aux programmes;
- **Gestionnaire des Entrées/Sorties (Input/Output manager):** il se charge de gérer tous les périphériques (clavier, écran, disques, etc.);
- **Gestionnaire de fichiers (File manager):** il se charge de la gestion de l'espace disque, de la manipulation des fichiers (lecture/écriture) tout en assurant l'intégrité des données et la protection des fichiers.

6 Evolution des systemes informatiques

L'evolution des systemes d'exploitation est intimement liee a l'evolution de la technologie des ordinateurs sur lesquels ils etaient implémentés.

6.1 1ère génération (1945 - 1955): Les tubes à vide et les cartes perforées

À cette époque les systemes d'exploitation et les langages de programmation n'existaient pas encore. Toute programmation était faite en langage machine. Le mode d'utilisation de la machine consistait alors pour un programmeur à venir s'inscrire sur une feuille de réservation de la machine pour une heure donnée et pour un temps limité. Puis, à venir à l'heure prévue dans la salle machine insérer son tableau de connexion dans le calculateur et enfin à attendre, quelques heures, les résultats du calcul en espérant qu'aucune des 20000 tubes à vide ne grillerait pendant l'exécution du travail. Au début des années 1950, les cartes perforées (punched cards) pour l'introduction des programmes ont remplacé les tableaux de connexion. Néanmoins la procédure d'utilisation du calculateur était la même. (Voir FIGURE 3)

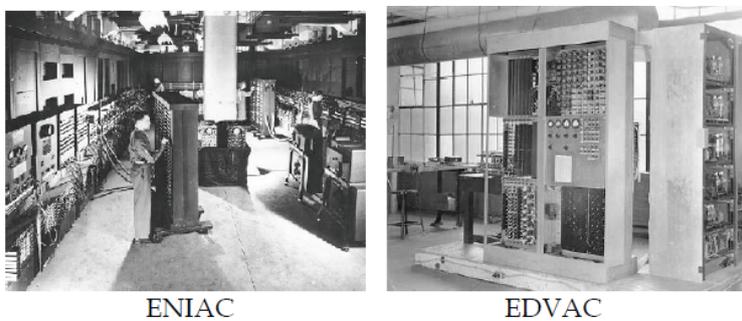


Figure 3: Exemples de machines à tubes

6.2 2ème génération (1956 - 1965): Transistors et traitement par lots

Les machines étaient construites de transistors et dotées d'unités de bandes magnétiques. Elles étaient plus fiables que celles de la génération précédente, mais toujours énormes (enfermées dans des salles climatisées). L'objectif était de réduire les pertes de temps occasionnées par l'oisiveté du processeur entre l'exécution de deux programmes.

L'idée était de collecter un ensemble de travaux, de les enregistrer sur bande magnétique en utilisant un ordinateur auxiliaire (Ex. IBM 1401), puis de faire lire la bande par l'ordinateur chargé des calculs (Ex. IBM 7094). Les résultats étaient récupérés sur une autre bande pour être imprimés par un autre ordinateur auxiliaire (Ex. IBM 1401). Chaque programme de la bande était lu, exécuté et ensuite l'ordinateur passait automatiquement au suivant. Cet automatisme était possible grâce à un programme résident en mémoire appelé **moniteur système** (l'ancêtre des SE d'aujourd'hui) qui permettait de passer le contrôle d'un programme à un autre. (Voir FIGURE 4)

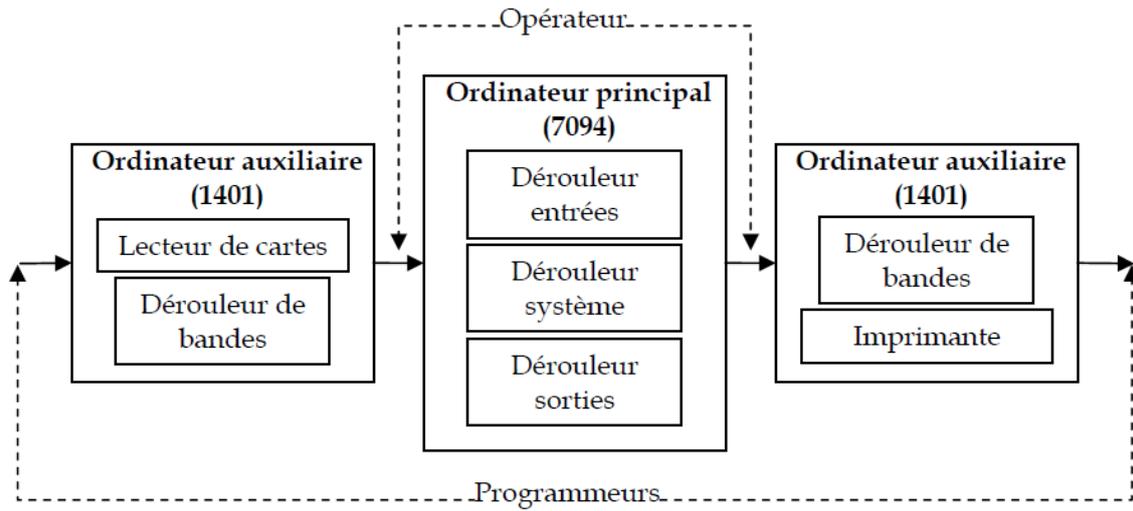


Figure 4: Système de traitement par lots

Le problème est que le processeur restait inutilisé pendant les opérations d'entrées/sorties, comme on le voit sur la FIGURE 5.

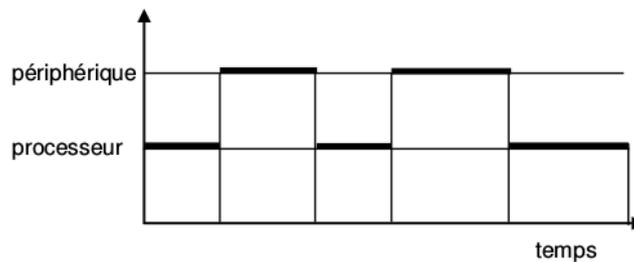


Figure 5: Utilisation du processeur et des périphériques

6.3 3ème génération (1960 - 1980): Les circuits intégrés et la multiprogrammation

L'introduction des circuits intégrés dans la construction des machines a permis d'offrir un meilleur rapport coût/performance. L'arrivée sur le marché des unités de disques, qui offrent l'accès aléatoire et des capacités de stockage importante a permis au système d'exploitation de conserver tous les travaux sur un disque dès leur arrivée dans la salle machine. L'idée était alors, pour pallier aux inconvénients du traitement par lots, de conserver en mémoire plusieurs travaux et gérer le partage du processeur entre ces différents travaux. Le processeur est alloué à un travail jusqu'à ce qu'il demande une E/S (premier arrivé, premier servi). Lorsqu'un travail demande une E/S, le processeur est alloué à un autre travail en mémoire. A la fin d'une E/S, une interruption se produit et le système d'exploitation reprend le contrôle pour traiter l'interruption et lancer ou poursuivre l'exécution d'un travail. Dès qu'un travail se termine, le système d'exploitation peut lancer le chargement, à partir du disque, d'un nouveau travail dans la partition qui vient de se libérer.

Exemple Considérons les travaux **A**, **B** et **C** en mémoire, qui sont montrés sur la FIGURE 6. L'exécution de **A** est entamée en premier puis lorsqu'il demande une Entrée/Sortie (E/S), le processeur commute sur **B**. À la fin de l'E/S demandée par **A**, le processeur suspend l'exécution de **B**

pour commuter sur **A**. On suppose que **A** est prioritaire. Après un certain temps de calcul, **A** demande de nouveau une E/S ; ce qui provoque la commutation du processeur sur **B**. Durant l'exécution de l'E/S de **A**, le travail **B** demande une E/S. Il se met donc en attente car le périphérique est occupé. Le processeur commute alors sur le travail **C**. Lorsque l'exécution de l'E/S demandée par **A** s'est terminée, le processeur commute sur **A** et le traitement de la demande d'E/S du travail **B** est entamé par le périphérique d'E/S.

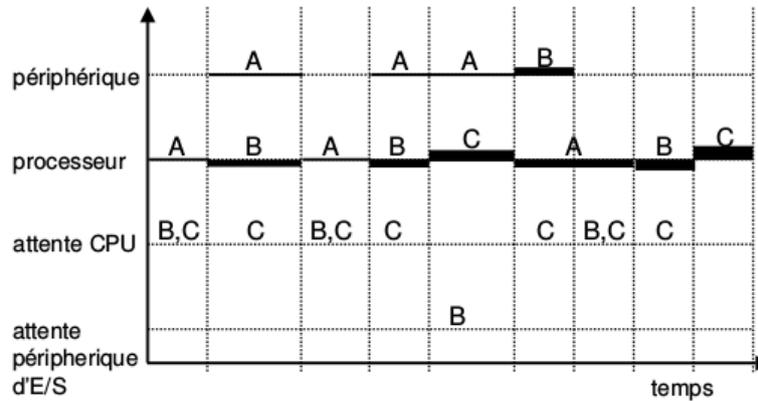


Figure 6: Commutation du processeur

Partage de temps: Le désir d'un temps de réponse plus rapide et d'interactivité de l'exploitation a introduit la technique de partage de temps (systèmes temps partagé ou multi-utilisateurs): plusieurs utilisateurs peuvent se connecter à la machine par l'intermédiaire de leurs terminaux et travailler en même temps. En effet, l'unité centrale est multiplexée entre les terminaux; chacun d'eux est servi pendant un **quantum de temps** d'où son nom : temps partagé (time sharing).

Le processeur est alloué, à tour de rôle, pendant un certain temps appelé à chacun des travaux en attente d'exécution. Au bout de ce temps, même si le travail en cours ne s'est pas terminé, son exécution est suspendue. Le processeur est alloué à un autre travail. Si plusieurs utilisateurs lancent à partir de leurs terminaux leurs programmes simultanément, ce mode d'exploitation donne l'impression que les programmes s'exécutent en parallèle. Le cas de trois travaux **A**, **B** et **C** est montré sur la FIGURE 7. Les temps de réponse pour chaque utilisateur sont acceptables.

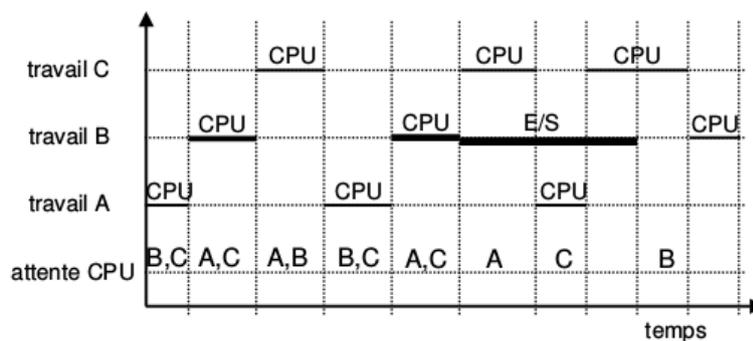


Figure 7: Temps partagé

6.4 4ème génération (1980 - aujourd'hui): Les ordinateurs personnels

Avec le développement des circuits LSI et VLSI a permis l'apparition de puces contenant des milliers de transistors sur un millimètre carré de silicium et ainsi le développement des microprocesseurs, l'âge des ordinateurs personnels était arrivé. En terme d'architecture, les ordinateurs personnels n'étaient pas si différent de ceux de la génération précédente.

7 Exemples de SE

Les systèmes d'exploitation les plus répandus sont:

- **Windows** pour les PC;
- **Mac OS** pour les ordinateurs d'Apple;
- **Linux** pour les PC et les serveurs;
- **Unix** pour les serveurs;
- **Android, iOS, Symbian et Windows Phone** pour téléphones portables et tablettes.

8 Conclusion

Les systèmes d'exploitation peuvent être envisagés sous deux angles: comme gestionnaire de ressources et comme machines étendues. Dans le premier cas, le rôle du système est de gérer efficacement ses différents éléments. Dans le second, il doit fournir aux utilisateurs une machine virtuelle plus facile à utiliser que la machine réelle.