

Introduction aux systèmes d'exploitation

Version 1



Y. CHALLAL, H. BETTAHAR, M. VAYSSADE

Table des matières

Objectifs	5
I - Introduction aux systèmes d'exploitation	7
A. Définition d'un système d'exploitation.....	7
1. Définition d'un système d'exploitation.....	7
B. Situation d'un système d'exploitation.....	8
C. Évolution de l'informatique.....	9
1. Première génération (1945-1955) : tubes à vide et tableaux d'interrupteurs	9
2. Deuxième génération (1955-1965) : transistors et systèmes par lots.....	11
3. Troisième génération (1965-1980) : compatibilité, circuits intégrés.....	12
4. Quatrième génération.....	15
D. Types de systèmes d'exploitation.....	17
1. Les systèmes pour mainframes.....	17
2. Les systèmes serveurs.....	17
3. Les systèmes multiprocesseurs.....	17
4. Les systèmes personnels.....	17
5. Les systèmes temps réel.....	18
6. Les systèmes embarqués.....	18
7. Les systèmes pour smart cards.....	18

Objectifs



- Analyser l'évolution des systèmes informatiques
- Analyser l'architecture d'un système d'exploitation
- Analyser les différents types de systèmes d'exploitation

Introduction aux systèmes d'exploitation

Définition d'un système d'exploitation	7
Situation d'un système d'exploitation	8
Évolution de l'informatique	9
Types de systèmes d'exploitation	17

A. Définition d'un système d'exploitation

Complexité des systèmes informatiques

Un système informatique moderne consiste en un ou plusieurs processeurs, de la mémoire principale, des disques, des imprimantes, un clavier, un écran, des interfaces réseau et autres périphériques d'entrées / sorties. Tout bien considéré, c'est un système complexe. Écrire des programmes qui prennent en compte tous ces composants, et les utilisant correctement, de surcroît de façon optimale, est une tâche extrêmement difficile. Pour cette raison, les ordinateurs sont équipés d'une couche logicielle appelée système d'exploitation, dont le rôle est de gérer tous les périphériques et de fournir aux programmes utilisateur une interface simplifiée avec le matériel [tanenbaum03].

1. Définition d'un système d'exploitation



Définition

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

1. Présenter une machine virtuelle simplifiée à l'utilisateur ;
2. Gérer les ressources de la machine.

Le système d'exploitation comme machine virtuelle simplifiée

L'architecture de la plupart des ordinateurs, au niveau du langage machine, est primitive et assez fastidieuse à programmer, particulièrement en ce qui concerne les entrées/sorties. Le programme qui cache le détail de fonctionnement du matériel et présente au programmeur une interface simple à base de fichiers à lire et/ou écrire est bien entendu le système d'exploitation [tanenbaum03].

Le système d'exploitation comme gestionnaire de ressources

Le système d'exploitation doit gérer l'ensemble des éléments d'un système fort complexe. Les ordinateurs modernes sont constitués d'un ou plusieurs processeurs, d'une mémoire, de timers, de disques, d'une souris, d'une ou plusieurs interfaces

réseau et imprimantes, et le cas échéant de divers autres périphériques. Le système d'exploitation doit gérer de manière équitable et optimale l'allocation des processeurs, de la mémoire et des périphériques d'E/S aux différents programmes concurrents qui les sollicitent [tanenbaum03].

B. Situation d'un système d'exploitation

Le système d'exploitation dans un ordinateur

La situation du système d'exploitation est montrée à la figure (cf. 'Situation d'un système d'exploitation' p 8). Tout en bas, on trouve le matériel qui est lui-même structuré en couche :

- le plus bas niveau contient les périphériques, circuits intégrés, les câbles, les alimentations et autres.
- Vient ensuite la niveau micro-architecture. En général, ce niveau comprend des registres internes à la CPU (Central Processing Unit) et un chemin de données contenant une ALU (Arithmetic and Logic Unit). Le rôle du chemin de données est d'exécuter un ensemble d'instructions. Ces instructions peuvent utiliser des registres ou d'autres ressources.
- Le matériel et les instructions accessibles au programmeur en assembleur constituent le niveau ISA (Instruction Set Architecture). Ce niveau est souvent appelé le langage machine [tanenbaum03].

Système bancaire	Réservation aérienne	Navigateur Web	Applicatifs
Editeurs	Compilateurs	Interpréteur de commande	Programmes Système
Système d'exploitation			Matériel
Langage Machine			
Micro-Architecture			
Périphériques physiques			

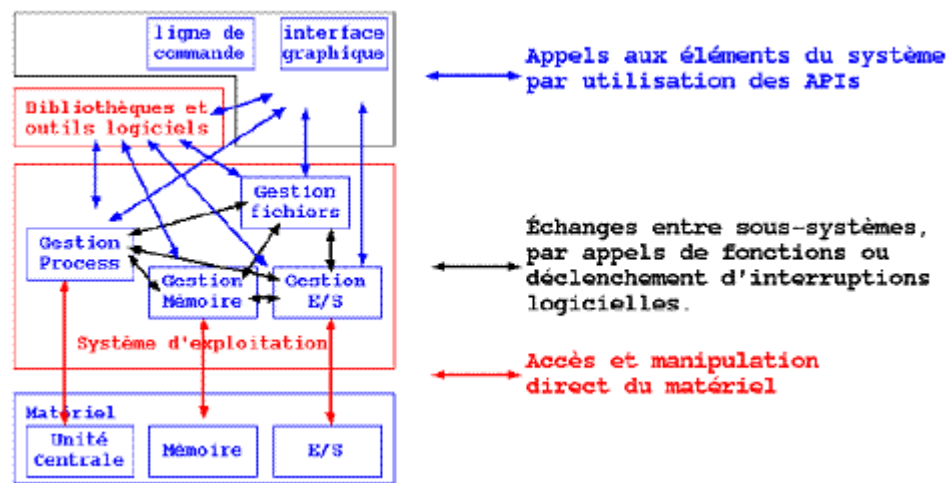
Situation d'un système d'exploitation

- Pour masquer cette complexité du bas niveau, on a recours à un système d'exploitation. Il consiste en une couche logicielle qui cache partiellement le matériel et fournit au développeur un jeu d'instructions plus pratique pour son travail.
- Au dessus du système d'exploitation, on trouve le reste du logiciel système. Il comprend l'interpréteur de commandes, le système de fenêtrage, les compilateurs, et d'autres programmes indépendants des applications.
- Enfin, on trouve au-dessus des programmes systèmes des programmes applicatifs. Ces programmes sont achetés ou conçus par les utilisateurs pour leurs besoins particuliers, comme le calcul scientifique, le tableur, le traitement de texte, ou la gestion de bases de données.

Sous systèmes d'un système d'exploitation

Le système sert d'intermédiaire entre l'utilisateur et le matériel. Sa complexité a conduit à l'organiser en sous systèmes. Les différents sous-systèmes (liés

étroitement à l'organisation du matériel) interagissent entre eux et avec l'utilisateur (voir figure (cf. 'Sous-systèmes et interactions' p 9)).



Sous-systèmes et interactions

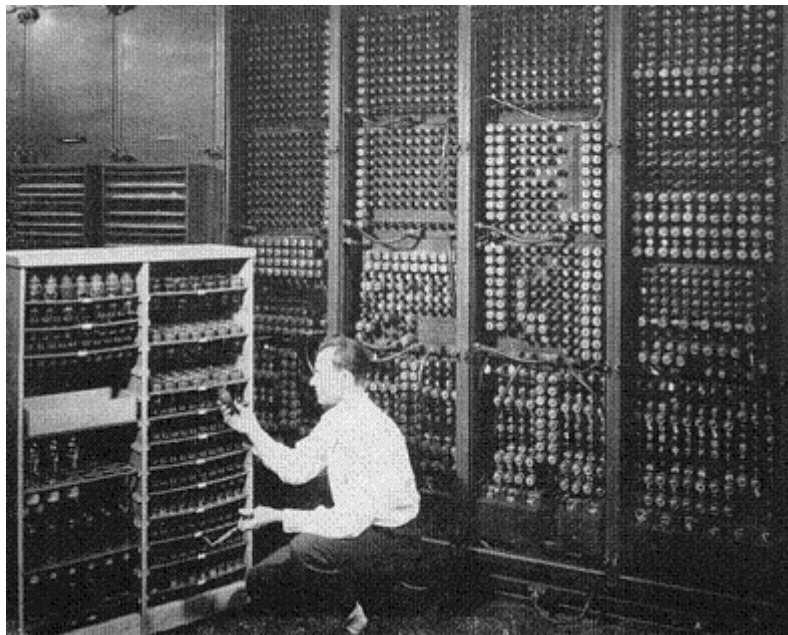
C. Évolution de l'informatique

Les systèmes d'exploitation ont été historiquement liés à l'architecture des ordinateurs sur lesquels ils étaient implantés. Nous décrivons les générations successives des ordinateurs et observons à quoi ressemblait leur système d'exploitation.

1. Première génération (1945-1955) : tubes à vide et tableaux d'interrupteurs

Tubes à vide

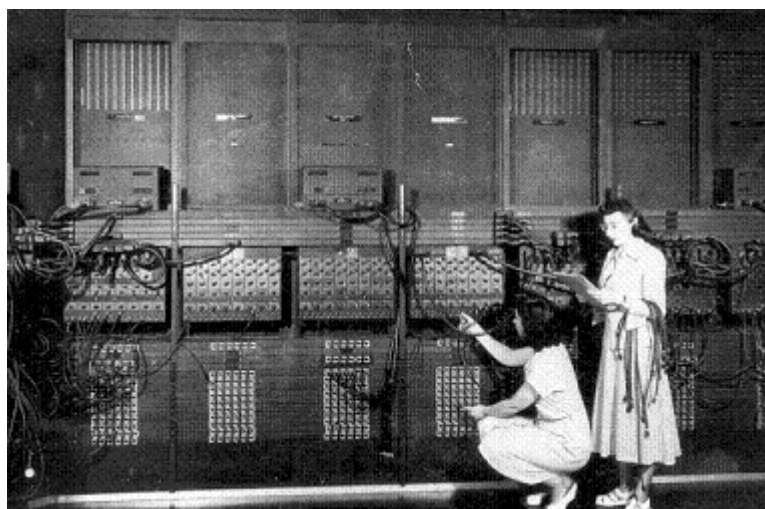
Les premiers calculateurs de cette époque étaient basés sur des tubes à vide, et remplissaient des salles entières avec les dizaines de milliers de tubes à vide (Figure (cf. 'Un technicien réparant des tubes à vide' p 10)).



Un technicien réparant des tubes à vide

Pas de langage de programmation

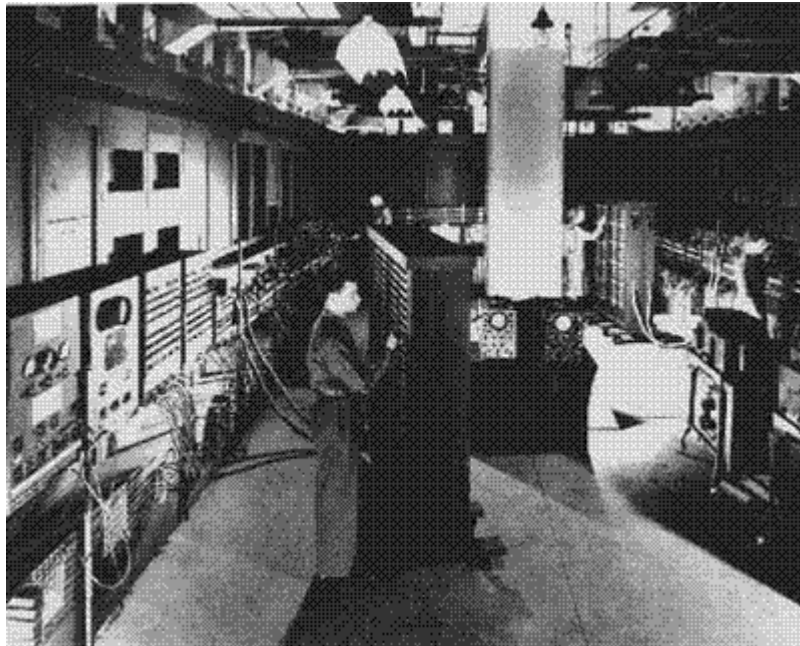
Les systèmes d'exploitation étaient encore inconnus et les langages de programmation étaient inexistant. Le programmeur devait contrôler également les fonctions de base du calculateur en utilisant un tableau d'interrupteur (Figure (cf. 'Deux dames en train de programmer en utilisant un tableau d'interrupteurs' p 10)).



Deux dames en train de programmer en utilisant un tableau d'interrupteurs

ENIAC

En 1945, John Presper Ecker et John Mauchly, construisent l'ENIAC (Figure (cf. 'ENIAC' p 11)). Il pesait 30 tonnes et consommait 200 kilo watts quand il était en marche. Il comprenait 18.000 tubes à vide.

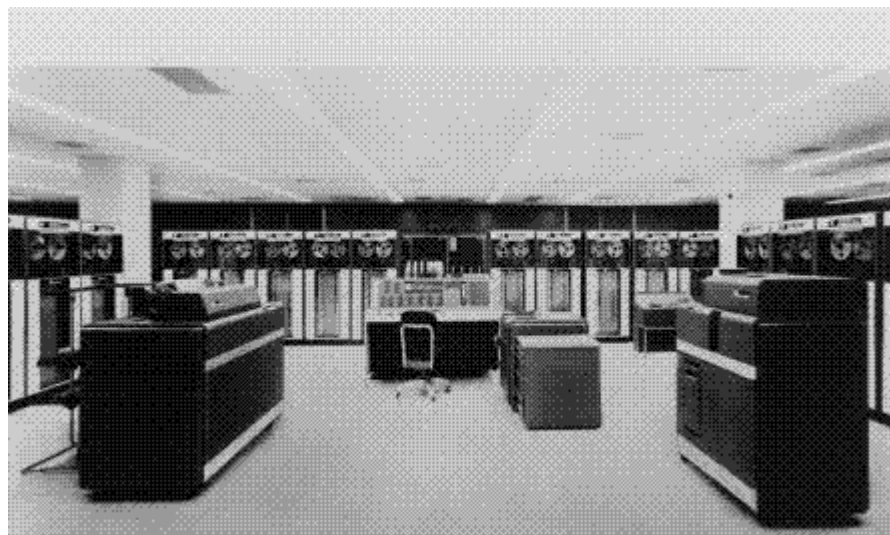


ENIAC

2. Deuxième génération (1955-1965) : transistors et systèmes par lots

Le transistor

L'apparition du transistor au milieu des années 1950 a radicalement changé la donne. Les ordinateurs devenaient suffisamment fiables pour être produits et vendus dans l'idée qu'ils fonctionneraient suffisamment longtemps pour produire un retour sur investissements. Ces machines, appelées de nos jours mainframes (ordinateurs centraux) (Figure (cf. 'IBM 7090' p 11)), étaient enfermées dans des locaux spéciaux climatisés. Seules les grandes firmes ou établissements publics pouvaient investir les millions de dollars nécessaires à leur achat.

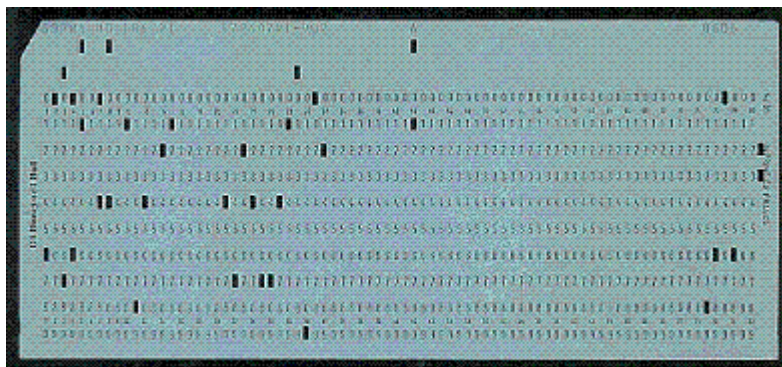


IBM 7090

Carte perforée

Pour soumettre un job (un programme ou un ensemble de programmes), le programmeur commençait par écrire son programme sur papier (en FORTRAN ou

en assembleur), puis il le codait sur des cartes perforées (Figure (cf. 'Carte Perforée' p 12)). Le mécanisme de lecture de ces cartes perforées était très particulier. Une aiguille passait en revue les lignes et colonnes de la carte. De l'autre côté se trouvait un baquet rempli de mercure. Si l'aiguille touchait le mercure, le courant transmis dans l'aiguille passait et fermait le circuit, ce qui signifie qu'il y avait présence d'un trou. Il apportait ensuite son paquet de cartes dans la salle de soumission des jobs et le confiait à l'opérateur présent. Il pouvait ensuite aller boire un café en attendant le résultat.



Carte Perforée

Quand l'ordinateur finissait un job, un opérateur récupérait sur une imprimante la trace d'exécution correspondante et la stockait dans la salle des résultats afin que le programmeur la récupère. Il prenait ensuite l'un des paquets de cartes de la salle de soumission et le faisait lire par la machine.

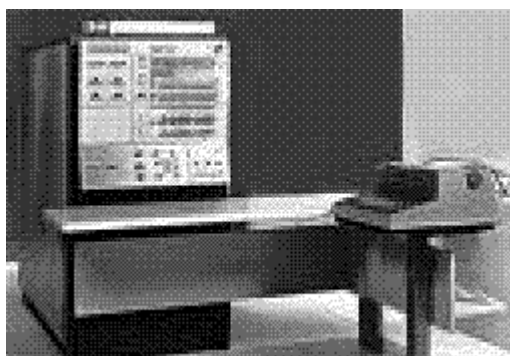
Traitement par lots (batch)

Une grande quantité de temps était gaspillée pendant ces diverses manipulations humaines. La solution la plus fréquemment adoptée est le système de batch (ou traitement par lots). L'idée était d'utiliser une machine peu onéreuse (comme l'IBM 1401, qui était plus performant pour ce type d'entrées/sorties que pour les calculs numériques) pour lire en une fois, à partir d'une bande magnétique, un ensemble de jobs soumis. On donnait alors au vrai calculateur un ensemble de jobs en une seule fois, grâce à un programme spécial (l'ancêtre des systèmes d'exploitation actuels), qui lisait le premier job et l'exécuter. Le résultat des jobs était écrit sur une bande au lieu d'être imprimé. Quand l'ensemble des jobs était terminés l'opérateur reprenait les bandes, remplaçait une autre bande de soumission dans le calculateur et le relançait. Il allait ensuite imprimer les résultats en rechargeant la seconde bande sur l'ordinateur intermédiaire.

3. Troisième génération (1965-1980) : compatibilité, circuits intégrés

Compatibilité

C'est le 7 avril 1964 qu'IBM Corporation a annoncé la mise sur le marché des ordinateurs IBM architecture 360. L'IBM 360 (Figure (cf. 'IBM360' p 13)) constitue la première grande " famille " d'ordinateurs au monde utilisant les mêmes logiciels et périphériques dans toute la gamme, et quels que soient les usages (scientifiques, gestion, etc.). " Compatibilité " devient alors le maître mot et les utilisateurs découvrent une gamme de choix sans précédent et hautement performante.



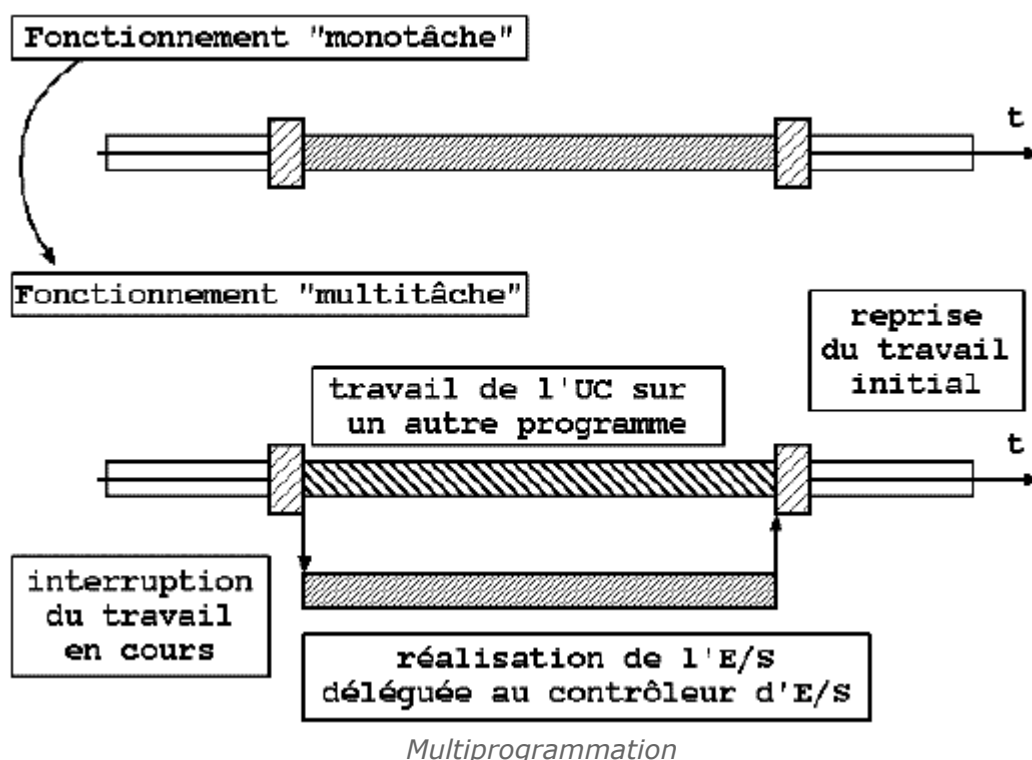
IBM360

Circuit intégré

Les 360 étaient les premières machines à utiliser des circuits intégrés, ce qui leur donnait un immense avantage au plan prix/performance sur les machines de seconde génération, entièrement conçues à base de transistors.

Multiprogrammation

Les systèmes d'exploitation de cette génération ont introduit des techniques fondamentales absentes dans la génération précédente. La plus importante d'entre elles est probablement la multiprogrammation. Elle consistait à charger plusieurs jobs en mémoire. Quand un job attendait la réalisation d'une opération d'entrée / sortie, un autre job pouvait s'emparer de la CPU (cf. 'Multiprogrammation' p 13). Si un nombre suffisant de jobs pouvait être chargé en mémoire simultanément, on pouvait arriver à un taux d'utilisation de la CPU voisin de 100% en permanence.



Spooling

Une autre caractéristique importante des systèmes d'exploitation de 3ème génération est leur capacité à stocker sur le disque les jobs écrits sur cartes au fur et à mesure de leur arrivée. Ainsi, dès qu'un job était terminé, le système

d'exploitation pouvait en charger un nouveau depuis le disque et l'exécuter. Cette technique est appelée spool (Simultaneous Peripheral Operation On Line) et est également utilisée en sortie. Bien que ces systèmes soient bien adaptés au calcul intensif et au traitement de grandes masses d'informations commerciales, ils n'étaient encore que des systèmes batch : la durée entre la soumission d'un job et la récupération du résultat se chiffrait en heures.

Temps partagé

Le désir de réponse rapide a conduit au temps partagé, une variante de la multiprogrammation dans laquelle chaque utilisateur dispose d'un terminal en ligne. Dans un tel système, si 20 utilisateurs sont en ligne et que 17 d'entre eux réfléchissent, discutent ou boivent un café, la CPU peut servir tout à tour les 3 jobs qui réclament effectivement des ressources. Les premiers systèmes réellement à temps partagé ont alors vu le jour :

- CTSS (Compatible Time Sharing System) développé au MIT sur un IBM 7094 modifié.
- MULTICS (MULTIplexed Information and Computing Service) développé par le MIT, Bell Labs et General Electric sur un GE-645.
- UNIX : System V d'AT&T, BSD (Berkeley Software Distribution). IEEE a développé un standard appelé POSIX, auquel la plupart des versions actuelles d'UNIX se conforment. POSIX définit une interface minimale d'appels système que tout système compatible doit implémenter.
- MINIX fut développé par Andrew Tanenbaum en 1987 un petit clone d'UNIX dans un but pédagogique.
- LINUX fut développé par Linus Torvalds un étudiant d'Andrew Tannenbaum. C'était une version de production par opposition à MINIX.

Mini-ordinateur

La percée des mini-ordinateurs est une autre caractéristique importante de la troisième génération. Elle a commencé avec le DEC PDP-1 en 1961. Il a été rapidement suivi de toute une famille (incompatibles entre eux), culminant avec le PDP-11 (Figure (cf. 'PDP11' p 14)).

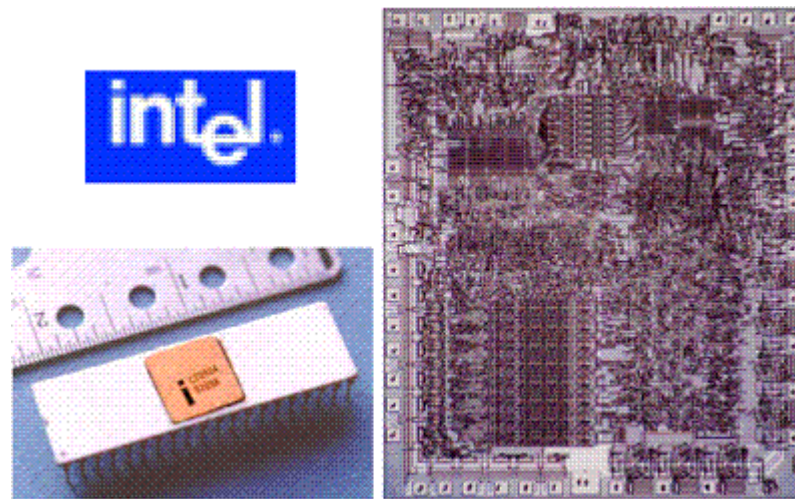


PDP11

4. Quatrième génération

Circuits intégrés à haute densité

Avec le développement des circuits intégrés à haute densité (LSI : Large Scale Integration), on a assisté à l'apparition de puces contenant des milliers de transistors sur un millimètre carré de silicium : l'âge des ordinateurs personnels est arrivé. En 1978, Intel a sorti le 8080, premier processeur généraliste à 8 bits (Figure (cf. 'Intel 8080' p 15)).



Intel 8080

CP/M OS pour Intel 8080

Gary Kildall (cf. 'Gary Kildall' p 15), un des consultants de Intel, a écrit un système d'exploitation pour le 8080 ainsi qu'un contrôleur de disquette. Ce système était appelé CP/M (Control Program for Microcomputers).



IBM-PC et MS-DOS

En 1980 IBM a lancé l'IBM PC (cf. 'IBM PC, Tim Paterson, Bill Gates' p 15). Elle demanda à Bill Gates s'il connaissait un système qui pouvait convenir à l'IBM PC. Gates racheta DOS (Disk Operating System) à \$50.000, et embaucha son auteur Tim Paterson, dans sa toute nouvelle entreprise Microsoft. Après modification, Microsoft lance son système MS-DOS.



IBM PC, Tim Paterson, Bill Gates

Système de fenêtrage et Macintosh

Doug Engelbart (cf. 'Doug Engelbart, Steve Jobs, Steve Wozniak' p 16), de l'institut de recherches de Stanford, a inventé le concept d'IHM graphique avec ses fenêtres, ses icônes, ses menus et sa souris. Ses idées ont été reprises par des chercheurs de Xerox PARC et incorporées aux machines qu'ils produisaient.



Doug Engelbart, Steve Jobs, Steve Wozniak

Un jour, Steve Jobs (co-inventeur d'Apple avec Steve Wozniak) a visité le PARC. Lorsqu'il a aperçu une IHM, il a immédiatement pris conscience de la plus-value potentielle qu'elle apportait. Jobs s'est mis à construire un Apple pourvu d'une IHM graphique. Ce projet a abouti à Lisa (cf. 'Apple Computer, Lisa, Macintosh' p 16), qui était trop chère et a été un échec commercial. Sa seconde tentative, le Macintosh d'Apple, a été un énorme succès, pour sa convivialité et sa facilité d'usage.



Apple Computer, Lisa, Macintosh



Complément : Systèmes distribués

Un développement intéressant a débuté vers le milieu des années 1980 : la croissance des réseaux d'ordinateurs personnels fonctionnant sous des systèmes d'exploitation en réseau ou des systèmes d'exploitation distribués. Un système d'exploitation distribué apparaît à l'utilisateur identique à un système classique à un processeur. L'utilisateur ne sait pas sur quelle machine s'exécute son programme, ni où se trouvent ses fichiers. Ces détails sont cachés et gérés efficacement par le système d'exploitation.

D. Types de systèmes d'exploitation

Une large palette de systèmes d'exploitation s'est développée au fil des ans. Dans cette section nous passons en revue les systèmes les plus célèbres.

1. Les systèmes pour mainframes

OS pour mainframes

Les mainframes (ces ordinateurs qui remplissent une pièce) se distinguent par leur capacité d'E/S. Ils arrivent à gérer des milliers de disques durs pour plusieurs téraoctets de mémoire. L'accent est mis dans leurs systèmes d'exploitation sur la capacité à gérer de manière optimale plusieurs jobs en même temps, chacun d'eux demandant de grandes ressources d'E/S. Ils offrent typiquement trois types de services : le batch, le transactionnel et le temps partagé.



Exemple

OS/390, un descendant d'OS/360 d'IBM.

2. Les systèmes serveurs

OS serveurs

Ils fonctionnent sur des machines serveurs, qui sont soit de gros micro-ordinateurs, soit des stations de travail, voire des mainframes. Ils servent en parallèle de nombreux utilisateurs à travers le réseau et permettent à ces derniers de partager des ressources matérielles et logicielles. Les fournisseurs d'accès à Internet et hébergeurs de sites web utilisent nombre de ces machines pour servir leurs clients.



Exemple

UNIX, et Windows 2003.

3. Les systèmes multiprocesseurs

OS pour multi-processeurs

Le recours à plusieurs CPU sur une même plateforme pour augmenter la puissance de calcul est une technique de plus en plus courante. Ils nécessitent des systèmes spéciaux, le plus souvent des systèmes serveurs, améliorés au niveau de la connectivité et des communications.

4. Les systèmes personnels

OS personnels

Leur rôle est de fournir à l'utilisateur une interface conviviale. Ils sont principalement dédiés à la bureautique.



Exemple

Les systèmes les plus représentatifs sont Windows 98, Windows 2000, Mac OS et Linux.

5. Les systèmes temps réel

OS temps réel

Ces systèmes se caractérisent par le respect de contraintes temporelles. Un exemple typique est celui des systèmes d'ordinateurs qui contrôlent un processus de fabrication dans une usine (comme le montage de véhicule) : dans le cas d'un robot soudeur, une soudure faite trop tôt ou trop tard peut compromettre un véhicule.



Exemple

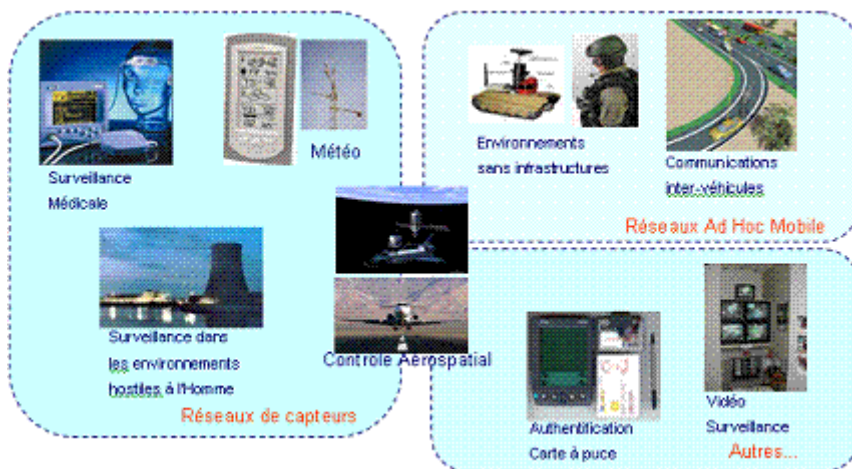
Les systèmes VxWorks et QNX en sont deux célèbres représentants.

6. Les systèmes embarqués

OS embarqués

Les systèmes embarqués tournent sur des ordinateurs qui pilotent des périphériques qui d'ordinaire ne dépendent pas d'un ordinateur, comme par

exemple une télévision, un four à micro-ondes, un PDA (Personal Digital Assistant), ou un téléphone portable (Figure (cf. 'Systèmes embarqués, temps réel, et smart cards' p 18) (cf. 'Systèmes embarqués, temps réel, et smart cards' p 18)). Ils ont souvent des caractéristiques temps réel avec des ressources limitées en terme de mémoire et d'énergie.



Systèmes embarqués, temps réel, et smart cards



Exemple

PalmOS et Windows CE (Consumer Electronics) sont deux représentants de cette famille.

7. Les systèmes pour smart cards

OS pour smart cards

Les plus petits systèmes d'exploitation se trouvent sur les smart cards, des périphériques de la taille d'une carte de crédit contenant une CPU. Ils sont sujet à de sévères contraintes de mémoire et de puissance de calcul. Certains ne savent remplir qu'une seule tâche comme le paiement électronique ou l'authentification. Certaines smart cards reposent sur JAVA, la ROM de la carte contient une machine virtuelle Java (JVM : Java Virtual Machine). Les applets Java sont téléchargées sur la carte et interprétées par la JVM.