

Complexité de machines et des OS

Dans toute l'histoire de l'informatique, les systèmes d'exploitation ont évolué. Les progrès de ces logiciels ont entraîné de nouveaux usages qui ont à leur tour créé de nouveaux besoins. Des concepts nouveaux sont apparus qui ont présidé à l'élaboration de la nouvelle génération de systèmes d'exploitation, et ainsi de suite. On peut redessiner l'évolution des systèmes d'exploitation en quelques étapes, de la première machine, l'ENIAC à nos jours, sur la base de trois critères principaux :

- Le nombre de processeurs de la machine (mono,multi)
- Le nombre d'utilisateurs qui peuvent s'en servir (mono,multi)
- Le nombre de "choses simultanées" (tâches) qu'une machine peut faire (mono,multi)

La machine monotâche monoutilisateur : la calculette programmable

Première à voir le jour, le premier OS est monoprocesseur, monotâche et monoutilisateur. Il faut faire la queue pour utiliser le supercalculateur ENIAC né au beau milieu de la seconde guerre mondiale. On ne peut lui faire dérouler qu'un seul programme, écrit en binaire pur sur un ruban perforé.

Ce mode extrêmement simple induit une particularité essentielle dans l'interaction de l'homme avec la machine : il ne peut s'agir que d'un dialogue frontal : la machine s'arrête et attend une donnée, l'utilisateur fournit la donnée et la machine repart.

Lorsque la machine n'exécute plus un "programme applicatif", elle continue à exécuter un seul programme : la réception des commandes système de la part de l'utilisateur unique.

Pour être programmable, il faut que le code "programme" puisse être fourni à la machine et soit modifiable : Von Neumann a l'idée d'écrire le programme de la même façon que des données : comme des codes inscriptibles en mémoire.

-> Concepts clefs : Machine de Von Neumann, interpréteur/décodeur d'instructions, micro-code

La machine multitâche monoutilisateur : le début de l'ordinateur

Une première évolution est de passer en mode multitâche. Le processeur étant encore unique, l'aspect multitâche est une illusion : Le déroulement de plusieurs programmes est saucissonné pour être exécuté chacun son tour par le processeur. C'est le principe de l'ordonnancement qui détermine qui s'exécute à quel moment, dans quel ordre, et avec quelle priorité.

Par cet artifice, la machine, qui n'exécute à un instant toujours qu'une seule séquence d'instruction, fournit l'illusion de faire plusieurs choses à la fois. Avec la vitesse croissante des processeurs, cette illusion est suffisante pour nos besoins humains.

La machine reste monoutilisateur, et les "utilisateurs" doivent se succéder. On "réserve" sur un planning un créneau de calcul comme on réserve une salle de réunion.

Les programmes exécutés "en parallèle" restent monoutilisateur dans leur principe, mais plusieurs voies de dialogue simultanées peuvent être utilisées. Comme l'interface homme-machine (clavier, principalement) reste "monovoie", on ne peut "dialoguer" en

fait qu'avec un programme à la fois.

-> Concepts clefs : processus, ordonnanceur.

La machine multitâche multiutilisateur : l'ordinateur centralisé ou l'ordinateur d'équipe

La notion de multi-utilisateur ne suffit pas ici. En effet, des implications architecturales très différentes apparaissent suivant que les utilisateurs multiples se servent successivement ou simultanément de la machine.

L'ordinateur centralisé

Les "mainframes" ou centraux informatiques d'entreprise des années 60 investissent largement dans des OS multi-utilisateurs (MULTICS, IBM... DIGITAL). Cette architecture suppose l'attribution des processus multiples à des utilisateurs multiples. Le processeur ne partage plus son temps entre des processus, mais entre des processus attribués à des utilisateurs.

-> Concepts clefs : Gestion des ressources concurrentielle, verrous (sémaphores), droits d'accès.

Le paramétrage des applicatifs doit tenir compte de ce problème : plusieurs utilisateurs peuvent se servir du même programme, et dans le cas des "mainframes" (aujourd'hui appelés plutôt serveurs), simultanément. Il faut donc maintenir en mémoire des "contextes" ou "profils" qui permet à chacune des instances des processus applicatifs de se baser sur un jeu de paramètres propres à chacun des utilisateurs qui les exploitent.

L'ordinateur d'équipe

Nous sommes toujours dans une problématique multi-utilisateur, mais les utilisateurs n'utilisent pas simultanément la machine (principe de la station de travail individuelle). Cependant, plusieurs utilisateurs peuvent se succéder pour utiliser la même machine. A chaque nouvel utilisateur, l'ordinateur devra recharger un jeu de paramètres particulier. Il y a trois étapes d'évolution de cette prise en considération :

- Au début, il ne sait pas bien le faire, et il lui faut redémarrer.
- Plus tard, il sait recharger une configuration utilisateur sans s'arrêter, c'est-à-dire, sans avoir à recharger tout le système d'exploitation. (XP professionnel, Linux)
- Enfin, il peut "geler" une session en cours pour autoriser un autre utilisateur à utiliser la machine (Windows XP familial, Vista grand public). Ce type correspond bien à un usage grand public.

-> Concepts clefs : Session, profil.

L'intervention du multiprocesseur

Le multiprocesseur n'est pas une technologie nouvelle. Les Mainframe historiques (Bell, Ibm, Digital, Hp...) utilisent depuis longtemps ce principe de répartition des processus entre plusieurs cartes mères.

Le multiprocesseur n'apporte pas de nouveauté réelle par rapport au multitâche dans la façon de concevoir un processus et sa gestion. On ne fait que matérialiser physiquement ce que l'ordonnancement virtuel faisait depuis longtemps. Cependant, les OS doivent

apprendre à gérer des environnements physiques (CPU, mémoire et chipset) qui ne communiquent pas forcément entre eux, tout en contribuant à la même prestation.

Cependant, à part par certains aspects techniques de mode de répartition et de réglage technique des échanges entre les différents coeurs de traitement, vu de l'opérateur de l'OS, les grand principes de gestion (processus, utilisateurs, droits, système de fichiers) ne sont pas remis en cause.