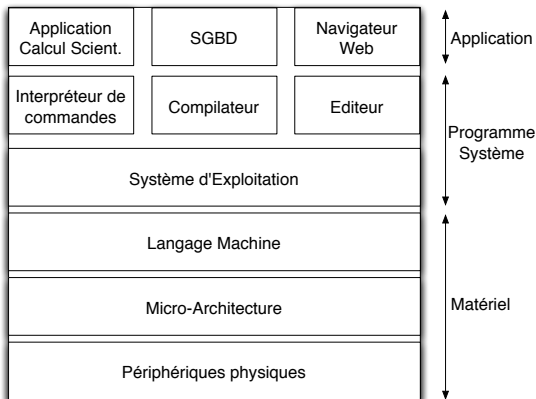


Introduction aux Systèmes d'exploitation

Définition d'un système d'exploitation

- Gestion des périphériques
- Interface programmeur-matériel



Fonctionnement

- portion de code qui fonctionne en mode noyau (ou superviseur).
- il est protégé des modifications apportées par l'utilisateur.

Rôle

- Gestion des ressources (périphériques, mémoire, processeurs)
- Extension de la machine
- Exemple : Lecteur de disquettes
 - Vision bas niveau : Surface d'enregistrement, tête de lecture, moteur ...
 - Vision haut niveau : Système de fichiers

Gestion des ressources

- Partager l'accès (exemple fichier imprimante et mémoire)
- Solution dépendante de la ressource
- Équité
- Connaissance des accès concurrents sur une ressource

Multiplexage

- Gestion des ressources sur 2 dimensions
 - le temps
 - l'espace
- Exemple du CPU et de la Mémoire

Historique

- Machine de Charles Babbage : non fonctionnel, mécanique pas assez précise (1792-1871)
- 1945-55 Tubes à vide et tableaux d'interrupteurs. Programme codé dans un langage absolu : par des tableaux d'interrupteurs (année 50 : introduction de la carte perforée)
- 1955-65 Transistors et systèmes par lots (Mainframe ou ordinateurs centraux)
 - Accroissement de la fiabilité
 - Soumission en 4 étapes (Ecriture du programme, soumission à l'opérateur, introduction des cartes perforées, compilation fortran)
 - Perte de temps à cause des manipulations humaines
 - Batch ou traitement par lots (FMS)

Historique

- 1965-80 Troisième génération
 - 2 architectures incompatibles (Unité de données : Mots VS Caractères)
 - IBM introduit son system/360 compatibilité au niveau logiciel (OS/360)
 - Multiprogrammation (Occupation CPU proche de 100%)
 - Spoolage
 - Temps partagé
 - Multitâche (d'abord simple puis préemptif)
 - Début d'Unix (System V et BSD)

Historique

- 1980-Aujourd'hui Ordinateurs personnels
- Intel 8080
- MSDOS (Bill Gates Tim Paterson)
- Lisa, Macintosh (Steve Jobs introduit les IHM)
- Windows (d'abord comme surcouche à MSDOS, puis comme Système d'exploitation à part entière)
- Unix
- Système d'exploitation distribué

Types

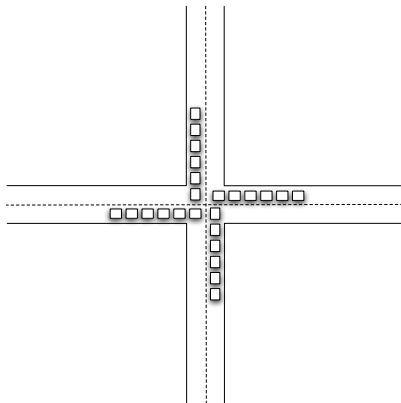
- Systèmes pour mainframe
- Systèmes pour serveurs
- Systèmes pour multi-processeur
- Systèmes personnels
- Systèmes temps réel
- Systèmes embarqués
- Systèmes SmartCard

Les processus

- Programme en cours d'exécution
- Espace d'adressage propre
 - Code du programme exécutable
 - Donnée
 - pile
 - ensemble de registres
- Table de processus
- Arborescence

Les interblocages

- Situation où plusieurs processus sont bloqués l'un à cause de l'autre.
- Exemple



Gestion de la mémoire

- Problème de pagination
 - Pas de solution idéale
 - Solutions acceptables (FIFO, LRU, FWF, ...)

Entrée Sortie

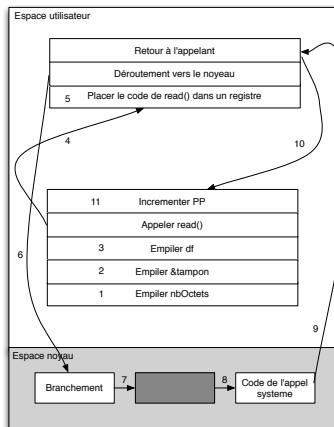
- Gestion pour l'utilisateur (programmeur)
- Pilotes

Les fichiers

- Existence d'une racine
- Arborescence
- Notion de chemin (relatif et absolu)

Concept de base

- Interface entre le système d'exploitation et les programmes utilisateurs
- Exemple de fonctionnement : appel de $cpt = read(df, \& tampon, nbOctets)$



Quelques appels système

- Gestion des processus (*fork()*, *waitpid()*, *execve()*,...)
- Gestion des fichiers (*open()*, *close()*, *read()*, *write()*, *stat()*,...)
- Gestion des répertoires (*mkdir()*, *rmdir()*,...)
- Gestion du système de fichiers
(*link()*, *unlink()*, *mount()*, *unmount()*,...)
- Autres (*kill()*, *time()*,...)

Les différentes structures

- Les systèmes monolithiques
- Les systèmes en couches
- Les machines virtuelles
- Les exonoys
- Le modèle client-serveur