Systèmes d’exploitation

A quoi sert un OS :

* Gérer les ressources hardware, software
* Cache tous les détails que l’on ne peut pas voir
* Offre une machine virtuelle avec plus de possibilités

Ressources Hardware :

* + Processeur
  + Mémoire
  + GPU

Ressources Software

* + Pilotes
  + Fichiers
  + Interfaces

Les abstractions :

* Fichier
* Mémoire
* Interface graphique
* Chaque programme pense qu’il est tout seul dans la machine

Ou trouve-t-on des OS ?

* Ordinateurs
* Serveurs
* Lecteurs
* Routers

Des OS :

* OS ENIAC
  + Pas de distance Hardware-Software
* OS Time sharing
  + Plusieurs utilisateurs
  + Partage des ressources
* OS Personnal Computer
  + Retour en Arrière
  + 1 seul utilisateur
  + 1 seul programme
* OS Temps réel
  + Importance du respect des délais pour les tâches
  + Hard deadline
  + Soft deadline

OS et matériel :

* Composants hardware
  + Mémoire  
    Accès mémoire : le CPU communique avec des BUS  
    Data  
    Address  
    Control
  + Accès périphériques  
    On passe par un contrôleur de périphérique

Comment accède-t-on au hardware ?

Le contrôleur offre la possibilité de piloter un périphérique physique au moyen d‘instructions exécutées par le CPU

* Assure la transformation des signaux digitaux en signaux analogiques
* Accéder via des registres de contrôleurs
  + Accès via des ports
  + Mappés en mémoire

Comment empêcher cela ?

* Certaines instructions privilégiées (permettant l’accès au HW) ne sont accessibles qu’en mode OS.
* On distingue l'OS par un flag de mode superviseur (ou noyau) situé dans le registre d'état ou Program Status Word (PSW)

Les processus

Un OS exécute plusieurs processus en même temps (Pseudo-Parallélisme), en effet chaque programme pense qu’il est tout seul sur la machine. Il a son propre processeur, sa propre mémoire ainsi que ses périphériques.

Pour nous => Processus = Abstraction de la machine hardware pour permettre à programme de « croire qu'il est seul ».

Pour l’OS => Processus = Code à exécuter + ensemble des ressources utilisées + état de la mémoire et du processeur (+droits+propriétaire+...).

On a donc dans l’OS une table qui contient tous les processus, une entrée par processus (Process control block PCB)

Une image contenant table

Description générée automatiquement

Création de processus :

* Quand : lors du démarrage de la machine, par un autre processus via un appel système, par un utilisateur via le shell,

Fin du processus :

* Quand : Arret anormal, Arret sur erreur, Arret par un autre processus, Arret via Shell ou task manager
* Parent meurt, tous les processus enfant aussi.

Etats d’un processus :

* Running, blocked, ready : Processus arreté pour attente d’une entrée -> Choix d’un autre processus -> Choix du processus -> Entrée devient accessible
* Blocage :
  + Attente ressources
  + Mis en attente par l’OS
* Déblocage :
  + Ressources disponibles
  + OS le choisi