Système d’exploitation – Théorie –

## Introduction :

Un système d’exploitation peut être :

* **Une extension du matériel** 🡪 Fournit aux programmes utilisateurs des interfaces vers le matériel
* **Un gestionnaire de ressources** 🡪 Responsable de l’attribution et de la gestion des différentes ressources

**Le processeur** est :

* Associé à des registres (zones mémoires)
* 2 modes de fonctionnement :
  + Kernel 🡪 Mode ayant la possibilité de tout faire, réservé au SE (accès à l’ensemble du matériel)
  + Utilisateur 🡪 Possibilité limitées (exemple : Mode de programme utilisateur)

Appels système permettent aux programmes utilisateurs de demander des services au SE.

Le SE peut :

* Allouer les ressources
* Gérer les périphériques

**Multithreading** 🡪 Permet de « faire travailler » plusieurs morceaux de programme en même temps.

**La mémoire** :

* Toutes les données d’un programme doivent être en mémoire pour qu’il s’exécute
* Plusieurs niveaux (registres, RAM, …)
* Volatile
* Partagée

**Le disque dur** :

* Non-volatile
* Performance inférieure
* Technologie variée

**Les entrées-sorties** :

* SE dialogue avec le contrôleur sur lequel le périph est connecté
* SE doit utiliser un driver adapté à ce matériel (le driver s’exécute en mode kernel et permet l’échange entre contrôler et SE)

**La virtualisation** :

* Permet de simuler un environnement matériel différent
* Permet d’exécuter plusieurs SE sur la même machine physique
  + Permet de réduire les coûts
  + Allocation de mémoire et de capacité centrale en fonction des besoins

## Chapitre 1 : Gestion des processus

## Introduction :

**Processus** 🡪 Programme en cours d’exécution. Il comporte :

* Code du programme
* Program couter et registres
  + Program Counter : Registre qui indique la prochaine instruction à exécuter
* Pile et données du programme

**Etats du processus :**

Voir vidéo 1 SWILABUS, 45ème minute.

## Introduction (4) :

* On garde en mémoire le processus
* Le scheduler choisit les processus à exécuter
  + Lorsqu’un processus doit attendre, un autre est sélectionné pour qu’il s’exécute
  + But : maximiser l’utilisation du CPU

**Le scheduler :**

* Sélectionne, parmi les processus prêts, celui qui va obtenir le CPU
* Type de processus
  + Orienté CPU (font fonctionner le CPU)
  + Orienté E/S (font fonctionner les entrées/sorties)

**Le changement de contexte :**

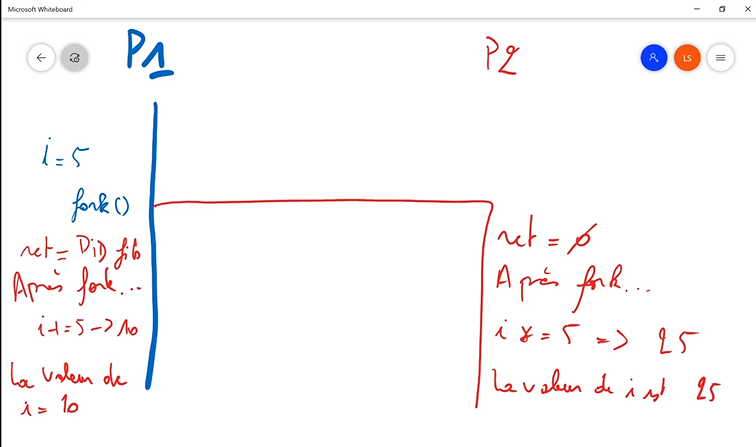
* Processus devient actif 🡪 Changement de contexte
  + Sauvegarde info du processus courant
  + Scheduler sélectionne autre processus et charge son contexte
  + Le processus sélectionné démarre

## Création d’un processus :

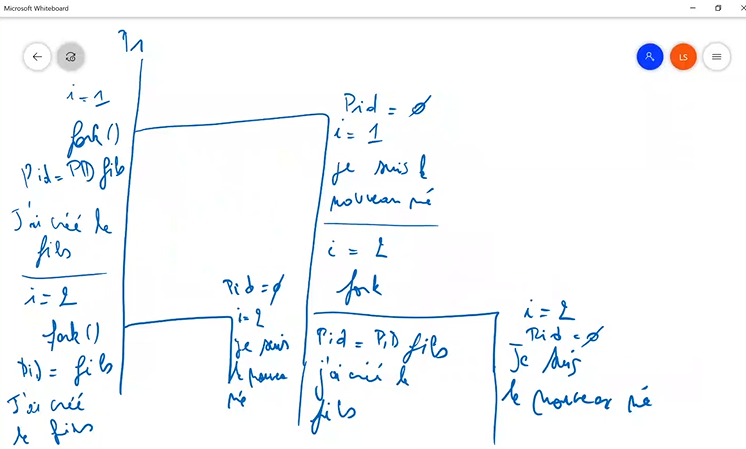
Utilisation de fork 🡪 Clone le processus père pour créer le processus fils

* **pid\_t fork(void)**
  + - 1 cas d’erreur
  + 0 cas du processus FILS
  + Default (pid fils) cas du processus PÈRE

Explication diapo 37 :



Explication diapo 38 :



**Fin de processus :**

* Plus d’instruction à exécuter
* Appel système exit(int) 🡪Renvoie une valeur entière au processus père
* Une fois tous les paramètres mentionnés, il faut terminer la liste par le pointeur NULL

## Création d’un processus (5) :

* **pid\_t getpid(void)** 🡪 Permet d’obtenir son PID
* **pid\_t getppid(void)** 🡪 Permet de connaitre son PPID (PID de son parent)

**Attendre mort du fils :**

* Wait() ou waitpid()
* Récupère la valeur renvoyée
* **pid\_t wait(int\* status)** 🡪Attendre la mort d’un processus fils (peu importe lequel)
* **pid\_t waitpid(pid\_t wpid, int\* status, int options)** 🡪 Attendre la mort d’un processus déterminé

**Execl :**

* **int execl(const char\* path, const char\* arg, …) 🡪** Permet de remplacer le code du processus par le code d’un autre

## Section critique (6) :

* Open : Permet d’ouvrir un fichier
* Unlink : permet de détruire un fichier

### **Protection par fichier**

* Pas fiable car nombre d’essais limité
* Pas le système le plus rapide 🡪 Pas efficace

### **Protection par sémaphore**

* La solution est fiable et performante
* En utilisant le sémaphore à 1.
* SETVAL 🡪 Commande qui permet d’initialiser le sémaphore
* Alloue une zone de mémoire pour 2 entiers ligne 6. Voir photo téléphone