

Rédigé par : équipe pédagogique du cours de Syst. D'expl.

Ref : *ING1-SE-TP-ORDONNANCEUR*

A l'intention de : Etudiants d'ING1

Créé le : 09/01/2012

## 1. Contexte du problème :

On désire simuler du traitement en temps partagé avec un seul micro processeur. Un processus est l'exécution d'un programme. Les OS actuels peuvent gérer plusieurs processus simultanément.

Le fonctionnement est entre autre assuré par un ordonnanceur. Il se sert d'une horloge interne. Au début de chaque cycle d'horloge, l'ordonnanceur attribue le microprocesseur à un processus. A la fin du cycle, il lui retire pour l'affecter à un autre processus et ainsi de suite. Les processus sont gérés par une file simulée par un tableau dans notre cas. Le processus en tête de file va être élu par l'ordonnanceur. Après avoir été servi, il repassera à la fin de la file. Entre temps de nouveaux processus peuvent être chargés en mémoire. Ils sont placés à la fin de la file.

## 2. Simulation d'un ordonnanceur en C

### 2.1 Spécifications

Pour simplifier, On a besoin d'un un tableau fini de processus. Un processus est une structure contenant le PID du processus, la priorité, durée cumulée d'exécution et le nombre total d'instructions du programme à exécuter.

Le programme pour démarrer :

1. génère un nombre aléatoire de processus, pour chacun des processus, on génère son ID, une priorité aléatoire entre 1 et 10, et un nombre total d'instructions comprise en 100 et 10000. on suppose que la durée d'exécution d'une instruction nécessite une unité de temps ou in incrément.
2. lance l'ordonnanceur,

L'ordonnanceur :

- a. affiche le nombre de processus disponibles dans le tableau des processus
  - b. prend le processus qui se trouve à la première position du tableau et décale les autre processus. Pour le processus pris, il affiche son PID, lui affecte un temps de traitement égal à un lapse de temps machine défini par l'utilisateur, multiplié par la priorité du processus. Le traitement est une boucle allant de 0 à lapse de temps \* priorité
  - c. on regarde si l'exécution du programme est terminé, le processus n'est plus remis dans le tableau, sinon on le remet à la dernière case du tableau.
3. Une fois ce traitement est terminé,
    - a- Pour simuler l'arrivée d'autres processus, on tire un nombre aléatoire compris entre 0 et 1, et s'il est inferieur à une valeur donnée paramétrable, on génère un autre processus et on l'ajoute à la fin du tableau, sinon on ne faire rien
    - b- On retourne à 2