

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

Ref : *ING1-SE-TP-ORDONNANCEUR*

A l'intention de : Etudiants d'ING1

Créé le : 13/11/2013

## 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, le nombre total d'instructions du programme à exécuter.

Nous présumerons que le processus à exécuter se trouve à la position 0. L'ordonnanceur devra alors élire un processus, le placer en début de tableau, affiche l'ensemble des processus à traiter, choisir un temps d'exécution (un nombre d'instructions à exécuter), puis boucler jusqu'à ce qu'il n'y ait plus de processus à exécuter.

### 2 Ordonnanceur basique

1. Complétez la fonction de création un processus avec l'ID du processus passé en paramètre.
2. Ecrire une fonction qui permet la création de n processus avec n choisi générer aléatoirement.
3. Ecrire une fonction qui affiche l'état d'un processus.
4. Ecrire une fonction qui choisit le prochain processus à exécuter, dans un premier temps, vous ne prenez pas en compte la priorité.

### 3 Ordonnanceur à priorité

1. Changer la fonction d'élection pour prendre en compte la priorité.
2. Comment traiter le fait que les processus peuvent avoir des entrées/sorties ?