

# Systeme d'Exploitation

## TD3

<jean-christophe.deneuille@unilim.fr>

## Ordonnancement

30 mars 2016

### Exercice 1 Tourniquons autour du FiFo

5 processus A, B, C, D et E sont soumis à un ordinateur dans cet ordre, mais quasi simultanément. Ces travaux ne font pas d'entrées-sorties. Leurs durées respectives sont 10, 6, 2, 4 et 8 secondes.

1. Déterminer les temps de réponse de chacun des processus, ainsi que le temps de réponse moyen, pour les disciplines FIFO (*First In First Out*).
2. Même question pour une discipline à priorité, avec Prio(A)=3, Prio(B)=5, Prio(C)=2, Prio(D)=1, Prio(E)=4 avec le plus petit chiffre égal à la priorité la plus forte.
3. Toujours la même avec la discipline Tourniquet, un quantum de 2s et un ordre initial des processus qui est celui de FiFo.

### Exercice 2 RMA

Considérant l'ordonnancement basé sur RMA :

1. Expliquer ce qu'on entend, lorsque l'on dit que la condition d'ordonnancement de Liu & Layland est une condition suffisante, mais pas nécessaire ?

Pour rappel, cette condition d'ordonnancement est définie par l'équation suivante :

$$\sum_{i=1}^N \left( \frac{C_i}{T_i} \right) \leq N(\sqrt[N]{2} - 1)$$

2. En utilisant la condition de Liu & Layland et l'ordonnancement graphique, que peut-on en dire des tâches de la table suivante ?

| Tâches | Période ( $T_i$ ) | Temps d'exécution ( $C_i$ ) |
|--------|-------------------|-----------------------------|
| T1     | 50                | 5                           |
| T2     | 40                | 10                          |
| T3     | 30                | 15                          |

### Exercice 3 EDF

Soit le tableau suivant :

| Tâches | Période ( $T_i$ ) | Temps d'exécution ( $C_i$ ) | Deadline ( $D_i$ ) |
|--------|-------------------|-----------------------------|--------------------|
| T1     | 10                | 5                           | 10                 |
| T2     | 40                | 20                          | 40                 |

1. Calculez l'utilisation du processeur sur lequel ces 2 tâches seraient ordonnancées.
2. Montrez qu'un ordonnancement est possible avec EDF (*Earliest Deadline First*).

### Exercice 4 Amélioration d'EDF

Soit l'algorithme d'ordonnancement dynamique LST (non présenté en classe) très proche du EDF.

L'idée derrière le LST (*Least Slack Time*) est de calculer, à chaque période d'ordonnancement, le *slack time* de chaque processus (tâche). Le *slack time* d'un processus est défini comme suit :

$$(d - t) - c'$$

ou  $d$  est la deadline,  $t$  est le temps réel depuis le début du cycle courant et  $c'$  est le temps d'exécution restant dans le cycle courant. Finalement, le processus avec le plus petit *slack time* se voit attribuer par l'ordonnanceur la plus grande priorité.

1. Vérifier si un ordonnancement utilisant l'algorithme du LST serait faisable pour l'ensemble des tâches de la table suivante :

| Tâches | Période ( $T_i$ ) | Temps d'exécution ( $C_i$ ) | Deadline ( $D_i$ ) |
|--------|-------------------|-----------------------------|--------------------|
| T1     | 4                 | 2                           | 4                  |
| T2     | 10                | 2                           | 10                 |

### Exercice 5 Un dernier pour la route

Vérifiez si les tâches suivantes peuvent être ordonnancées EDF :

| Tâches | Période ( $T_i$ ) | Temps d'exécution ( $C_i$ ) | Deadline ( $D_i$ ) |
|--------|-------------------|-----------------------------|--------------------|
| T1     | 20                | 3                           | 7                  |
| T2     | 5                 | 2                           | 4                  |
| T3     | 10                | 1                           | 8                  |