

Licence d'Informatique L2
Introduction aux Systèmes et Réseaux

TD n ° 11 : Ordonnancement des processus

1 Les algorithmes d'ordonnancement

Soient cinq processus décrits par la table ci-dessous. Déterminez l'ordre d'exécution de ces cinq processus pour chacun des algorithmes d'ordonnancement suivants :

- FIFO (First-In, First-Out : premier arrivé premier servi),
- SJF (Shortest Job First : le plus court d'abord),
- SRT (Shortest Remaining Time First : le temps restant le plus court),
- RR (Round-Robin : tourniquet avec un quanta fixé à 10 unités de temps, processus rangés en queue de file),
- RR new (Round-Robin : tourniquet avec un quanta fixé à 10 unités de temps, processus rangés en tête de file d'attente pour leur donner une sorte de priorité).

Pour chaque algorithme et chaque processus, calculez le temps d'attente afin de comparer les stratégies d'ordonnancement.

processus	date d'arrivée	durée
P1	0	10
P2	5	29
P3	35	3
P4	29	7
P5	24	12

2 Endormir un processus

Considérons un ordonnanceur simplifié qui applique la stratégie du tourniquet :

variables globales:

```
| # nombre de processus en cours
| MAXP: entier
|
| # état des processus
| état: liste de {prêt, non-prêt} de taille MAXP
|
| # numéro du processus courant
| pc: entier
```

procédure ordonnanceur(pc: entier):

```

| <sauver le MEP et le contexte du processus courant pc>
| pc = (pc + 1) % MAXP
| while état[pc] != prêt:
|     pc = (pc + 1) % MAXP
| <restaurer le contexte et le MEP de pc>

```

On veut désormais pouvoir *endormir* un processus pendant un certain temps (exprimé en secondes). Vous avez à votre disposition une fonction `temps()` qui renvoie le temps écoulé en secondes depuis le démarrage de la machine.

2.1

1. Définissez les structures de données supplémentaires de l'ordonnanceur et détaillez les états possibles d'un processus.
2. Écrivez ensuite la fonction `endormir(s)` qui endort le processus courant (celui qui appelle cette fonction) pendant au moins s secondes.

2.2

Comment l'ordonnanceur réveille-t-il les processus endormis (donner le nouvel algorithme de l'ordonnanceur) ?

2.3

Pour éviter que le test du réveil soit effectué à chaque appel de l'ordonnanceur (c-à-d plusieurs fois par seconde dans la pratique), une autre solution consiste à créer un nouveau processus système qui a la charge de réveiller les processus endormis. Écrire l'algorithme de ce processus système appelé `réveil()`, indépendant de l'ordonnanceur, et dont le rôle est de réveiller les processus endormis. Il n'utilise pas d'autres structures de données que celles utilisées précédemment.

2.4

1. Supposons qu'il n'y ait aucun processus endormi. Dans ce cas, le processus réveil n'a pas aucun intérêt à tourner et il devrait donc être passé à l'état endormi. Proposer un nouvel algorithme pour `réveil()`, qui utilise un compteur de processus endormis.
2. Dans ce dernier cas, il faut penser à réveiller le processus réveil si celui-ci est endormi quand on endort un nouveau processus. Donner la nouvelle version de la procédure `endormir()`.