TD3 : Ordonnancement de la CPU

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 (premier arrivé premier servi), SJF (le plus court d'abord), SRT (le temps restant le plus court) et RR (tourniquet avec un quanta fixé à 10 unités de temps). Pour chaque exécution 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 :
  |  /* état des processus */
  |  état : tableau [0 .. max-1] de {prêt, non-prêt}
  |
  |  /* numéro du processus courant */
  |  pc   : entier

procédure ordonnanceur
  |  ⟨sauver le MEP et le contexte du processus courant pc⟩
  |  répéter
  |     pc := (pc + 1) modulo max;
  |  jusqu'à (état[pc] = prêt);
  |  ⟨restaurer le contexte et le MEP de pc⟩
```

On veut *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.

a) Définissez les structures de données supplémentaires de l'ordonnanceur et détaillez les états possibles d'un processus. écrivez ensuite la fonction `endormir_processus(`p`,`s`)` qui endort le processus p pendant **au moins** s secondes.

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