

# TD 3 - ASR7 Programmation Concurrente

## Ordonnancement

Matthieu Moy, Fabien Rico, Adil Khalfa

Printemps 2018

### I Fin du TD2

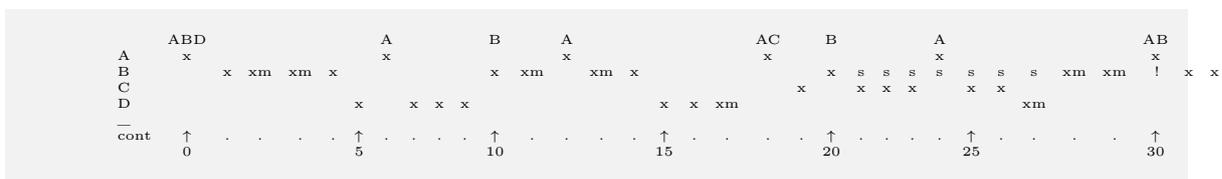
Fin du pont de Miralonde ou Producteur-consommateur.

### II Ordonnancement avec des mutexes

Nous utilisons un ordonnancement préemptif avec priorité<sup>1</sup>. Nous allons utiliser un jeu de tâches qui mélange des tâches périodiques et des tâches ponctuelles. De plus, deux tâches partagent un mutex.

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	0, 6, 12, 18, 24, 30	10	1	Périodique
B	0, 10, 20, 30	8	4	Périodique, à chaque itération, la tâche doit acquérir le mutex à la fin du temps 1 et la libérer à la fin du temps 3.
C	18	5	6	Ponctuelle
D	0	1	8	Ponctuelle, à la fin du temps 6, la tâche acquiert le mutex et le conserve jusqu'à la fin du temps 8.

Q.II.1) - Faire l'ordonnancement de ces tâches sur 32 unités de temps.



Q.II.2) - Quel est le temps de réponse de chaque tâche ?

- A : 1
- B : 12 (le pire est la seconde fois). Une échéance est loupée.
- C : 9 (arrivée en 18 fini en 27)
- D : 28 (arrivée en 0 terminée en 28)

1. Plus la valeur de priorité est importante plus la tâche est prioritaire

**Q.II.3)** - Que remarque-t-on aux temps 21 à 27 ?

Une inversion de priorité, B est bloquée par D qui est bloquée par C. Donc C qui est moins prioritaire que B et ne partage pas de mutex avec B passe avant B.

### III Ordonancement

Nous allons utiliser l'implantation POSIX 1003b sur Linux. Il existe 100 niveaux de priorité :

- Le niveau 0 est réservé à SCHED\_OTHER et les niveaux de priorité 1 à 99 aux politiques SCHED\_FIFO et SCHED\_RR.
- Les tâches de priorité 99 sont les tâches de plus forte priorité.
- SCHED\_OTHER est dédié à l'ordonnanceur temps partagé.
- Le quantum utilisé par la politique SCHED\_RR est de 2 unités de temps.
- Pour SCHED\_FIFO et SCHED\_RR, lorsqu'une nouvelle tâche arrive, elle est placée en queue (i.e. les tâches sont exécutées dans l'ordre d'arrivée).
- L'ordonancement est préemptif.

Soit le jeu de tâches a périodiques suivant :

Tâche	Date d'arrivée	Priorité	Durée	Politique
o1	0	0	5	SCHED_OTHER
rr1	7	5	2	SCHED_RR
rr2	10	10	6	SCHED_RR
rr3	6	10	7	SCHED_RR
fifo1	1	10	4	SCHED_FIFO
fifo2	3	10	2	SCHED_FIFO

**Q.III.1)** - On suppose qu'une fois arrivées, les tâches sont toujours prêtes. Dessinez de l'instant 0 à l'instant 26, l'ordonancement généré par l'ordonnanceur.



**Q.III.2)** - Donner le temps réponse de chaque tâche.

**o1** :  $25 - 0 + 1 = 26$   
**rr1** :  $21 - 7 + 1 = 15$   
**rr2** :  $19 - 10 + 1 = 10$   
**rr3** :  $17 - 6 + 1 = 12$   
**fifo1** :  $4 - 1 + 1 = 4$   
**fifo2** :  $6 - 3 + 1 = 4$