

Sections : LGLSI 1 – LIRS 1

## TD ORDONNANCEMENT

### Exercice 1 : Ordonnancement sur une machine multiprocesseurs avec unité d'E/S

On considère une machine informatique disposant de deux unités d'exécution (deux CPU notés CPU1 et CPU2) identiques et une unité d'E/S (banalisée). Le système d'exploitation adopte l'algorithme de **tourniquet** d'ordonnancement des processus (Round Robin) avec un quantum de trois unités de temps ( $Q = 3$ ). Tous les processus prêts sont dans une même file d'attente (de deux processeurs). La commutation de contexte est supposée de durée nulle. Une file d'attente FIFO est associée à l'unité d'E/S.

Si plusieurs événements surviennent en même temps, on adopte les priorités suivantes :

- Le CPU1 a la priorité d'accès à la file des processus prêts par rapport au CPU2.
- A la fin d'un quantum, le processus non terminé en cours n'est suspendu que si la file d'attente des processus prêts n'est pas vide.
- Le traitement réalisé à la fin d'un quantum est plus prioritaire que celui d'une fin d'E/S qui, à son tour, est plus prioritaire que l'arrivée de nouveaux processus dans le système.

On considère trois processus A, B et C décrits dans le tableau suivant :

Processus	Instant d'arrivée	Temps d'exécution (Burst Time)
A	0	4 unités de CPU, 2 unités d'E/S, 2 unités de CPU
B	2	3 unités de CPU, 4 unités d'E/S, 2 unités de CPU
C	3.5	5 unités de CPU

La première ligne signifie que le processus A arrive dans le système à l'instant  $t=0$ , son exécution nécessite dans l'ordre 4 unités de temps CPU, 2 unités de temps d'E/S et 2 unités de temps CPU. Au départ le processus A est élu par le processeur CPU1.

- a) Donnez les diagrammes de Gantt montrant l'allocation des deux processeurs, de l'unité d'E/S et l'évolution des états des files d'attente (celle des processus prêts et celle des processus en attente de l'unité d'E/S). Vous pouvez adopter la notation suivante : (Date début, Processus, Durée) ...
- b) Calculez le temps moyen de virement (temps moyen de séjour).

### Exercice 2 : Ordonnancement des processus multithreads

Soient deux processus utilisateur P1 et P2 qui s'exécutent sur un système monoprocesseur à un instant  $t (=0)$ . Le processus P1 est composé de deux threads. Le processus P2 est composé de trois threads. Les temps nécessaires à leurs exécutions sont résumés dans le tableau suivant :

Processus	Threads	Temps d'exécution
P1	T11	1 unité de CPU
	T12	3 unités de CPU
P2	T21	3 unités de CPU
	T22	2 unités de CPU
	T23	1 unité de CPU



- 1) On suppose que le processeur est initialement libre. Donnez les diagrammes de Gantt montrant l'allocation du processeur, pour chacun des cas suivants :
- a) Les threads sont **supportés par le noyau**. Le noyau ordonnance l'exécution des threads selon l'algorithme du tourniquet (Round Robin) avec un quantum de 2 unités. Dans ce cas, si un thread d'un processus ne consomme pas son quantum, le processeur peut être alloué à un thread d'un autre processus.  
La file d'attente des threads prêts, à l'instant t, est :  $\rightarrow T_{23} T_{12} T_{22} T_{21} T_{11} \rightarrow$  (T11 est en tête de file).
- b) Les threads sont implémentés entièrement **au niveau utilisateur**. Le noyau ordonnance l'exécution des processus selon l'algorithme du tourniquet avec un quantum de 2 unités. Les threads sont ordonnancés au niveau utilisateur selon l'algorithme du tourniquet avec un quantum de 1 unité. Si un thread d'un processus ne consomme pas son quantum, le processeur est alloué à un autre thread du même processus (s'il y en a).  
On suppose que le processeur est libre et que les processus sont passés à l'état prêt dans l'ordre suivant :
- P1 puis P2
  - Dans P1 : T11 puis T12
  - Dans P2 : T21 puis T22 puis T23
- Dans tous les cas, le temps de commutation de contexte est supposé toujours nul.
- 2) Calculez, pour chaque cas, le temps de virement (temps de séjour) de chaque processus. Comparez puis commentez les résultats.

### Extrait de la session de contrôle 2020-2021.

#### Ordonnanceur à priorité préemptive

On considère l'ensemble de processus avec l'heure d'arrivée (Arrival time), le temps d'exécution (Burst time) et la priorité (0 est la priorité la plus élevée) indiqué dans le tableau ci-dessous. Aucun des processus n'a de temps d'E/S.

Process	Arrival time	Burst time	Priority
P1	0	11	2
P2	5	28	0
P3	12	2	3
P4	2	10	1
P5	9	16	4

Sachant que l'OS utilise l'algorithme d'ordonnancement à priorité préemptive, quel est le temps d'attente du processus P1 ?

- ☐ 26    ☐ 29    ☐ 38    ☐ 48

#### Changement de contexte

On considère trois processus gourmands en CPU, qui nécessitent 10, 20, 30 unités et arrivent respectivement aux instants 0, 2, 6. Combien de changements de contexte sont nécessaires si la politique d'ordonnancement appliquée est SRTF (Short Remaining Time First) ? La commutation de contexte à  $t=0$  est incluse mais celle à la fin est ignorée.

- ☐ 1    ☐ 2    ☐ 3    ☐ 4

### Ordonnancement LRTF - Temps moyen d'exécution

On considère 03 processus (numérotés 0, 1 et 2) ayant respectivement les temps d'exécution (bursts time) 2, 4 et 8 unités. Tous les processus arrive à  $t=0$ . Le système d'exploitation applique une politique d'ordonnancement préemptive dite **Longest Remaining Time First (LRTF)**. Avec la politique LRTF, le processeur est attribué au processus ayant le plus grand temps d'exécution restant (l'inverse de SRTF). En cas d'égalité, la priorité est donnée au processus ayant le numéro le plus bas. Quel est le temps moyen d'exécution des 03 processus ?

- ☐ 13      ☐ 14      ☐ 15      ☐ 16

### Extrait de la session de principale 2020-2021.

On considère une machine informatique disposant de deux unités d'exécution (deux processeurs notés CPU 1 et CPU 2) identiques et une unité d'E/S (banalisée). Le système d'exploitation adopte l'algorithme d'ordonnancement **SRTF** des processus. Tous les processus prêts sont dans une même file d'attente (pour les deux processeurs). La commutation de contexte est supposée de durée nulle. Une file d'attente FIFO est associée à l'unité d'E/S.

Si plusieurs événements surviennent en même temps, le CPU 1 a la priorité d'accès à la file des processus prêts par rapport au CPU 2. En cas d'égalité de temps restant de traitement, le processus qui détient un CPU est plus prioritaire que celui d'une fin d'E/S qui est, à son tour, plus prioritaire que l'arrivée de nouveaux processus dans le système.

On considère trois processus A, B et C dont le temps d'arrivée, le temps d'exécution et celui des E/S sont décrits dans le tableau suivant :

Processus	Instant d'arrivée	Temps d'exécution et des E/S
A	0	4 unités de CPU, 2 unités d'E/S, 2 unités de CPU
B	2	3 unités de CPU, 4 unités d'E/S, 2 unités de CPU
C	3	1 unités de CPU

La première ligne signifie que le processus A arrive dans le système à l'instant  $t=0$ , son exécution nécessite dans l'ordre 4 unités de temps CPU, 2 unités de temps d'E/S et 2 unités de temps CPU.

- Donnez les diagrammes de Gantt montrant l'allocation des deux processeurs, de l'unité d'E/S et l'évolution des états des files d'attente (celle des processus prêts et celle des processus en attente de l'unité d'E/S). Vous pouvez adopter la notation suivante : (Date début, Processus, Date fin).
- Calculez le temps moyen de virement (temps moyen de séjour).

### Extrait de la session de contrôle 2022-2023 :Ordonnancement RMS

RMS est un algorithme d'ordonnancement temps-réel des processus périodiques. Un processus périodique ( $P_i$ ) est caractérisé par sa période d'activation ( $Pr_i$ ), son temps d'exécution ( $T_i$ ) et sa date d'arrivée dans le système ( $D_i$ ). Chaque processus possède une contrainte temporelle : il doit s'exécuter complètement avant que sa période arrive de nouveau. Le principe de l'algorithme RMS consiste à associer à chaque processus une **priorité fixe**, inversement proportionnelle à sa période. Lorsqu'il est appelé, l'ordonnanceur du système choisit le processus de plus forte priorité et le lance en exécution (ordonnancement à priorités). L'algorithme RMS a deux variantes : la première est préemptive et la deuxième est non non-préemptive.

Soit trois processus A, B et C ayant les caractéristiques suivantes :



Processus	Pr <sub>i</sub>	T <sub>i</sub>	D <sub>i</sub>
A	29	7	0
B	5	1	0
C	10	2	0

- 1) A votre avis, pourquoi la priorité d'un processus est inversement proportionnelle à sa période ?
- 2) Indiquez les priorités des processus A, B et C selon l'algorithme RMS.
- 3) Donnez le diagramme de Gantt pour les 30 premières unités de temps d'ordonnancement **RMS, préemptif**.
- 4) Donnez le diagramme de Gantt pour les 30 premières unités de temps d'ordonnancement **RMS, non préemptif**.
- 5) Est-ce que tous les processus respectent leurs contraintes temporelles sur les 30 premières ms dans le cas RMS préemptif ? justifiez votre réponse.
- 6) Est-ce que tous les processus respectent leurs contraintes temporelles sur les 30 premières ms dans le cas RMS non préemptif ? justifiez votre réponse.

### Extrait de la session principale 2022-2023 : Ordonnancement multiprocesseur avec File d'E/S

On considère une machine informatique disposant de deux unités d'exécution (deux processeurs notés CPU 1 et CPU 2) identiques et une unité d'E/S (banalisée). Le système d'exploitation adopte l'algorithme d'ordonnancement **SRTF** des processus. Tous les processus prêts sont dans une même file d'attente (pour les deux processeurs). La commutation de contexte est supposée de durée nulle. Une file d'attente FIFO est associée à l'unité d'E/S.

Si plusieurs événements surviennent en même temps, le CPU 1 a la priorité d'accès à la file des processus prêts par rapport au CPU 2. En cas d'égalité de temps restant de traitement, le processus qui détient un CPU est plus prioritaire que celui d'une fin d'E/S qui est, à son tour, plus prioritaire que l'arrivée de nouveaux processus dans le système.

On considère trois processus A, B et C dont le temps d'arrivée, le temps d'exécution et celui des E/S sont décrits dans le tableau suivant :

Processus	Instant d'arrivée	Temps d'exécution et des E/S
A	0	4 unités de CPU, 2 unités d'E/S, 2 unités de CPU
B	2	3 unités de CPU, 4 unités d'E/S, 2 unités de CPU
C	3	1 unités de CPU

La première ligne signifie que le processus A arrive dans le système à l'instant  $t=0$ , son exécution nécessite dans l'ordre 4 unités de temps CPU, 2 unités de temps d'E/S et 2 unités de temps CPU.

- a) Donnez les diagrammes de Gantt montrant l'allocation des deux processeurs, de l'unité d'E/S et l'évolution des états des files d'attente (celle des processus prêts et celle des processus en attente de l'unité d'E/S). Vous pouvez adopter la notation suivante : (Date début, Processus, Date fin).

- b) Calculez le temps moyen de virement (temps moyen de séjour).

