

TRAVAUX PRATIQUES 2.

Évaluation : 20 points

Date de distribution : 17/11/2016

Date de remise : **8/12/2016**

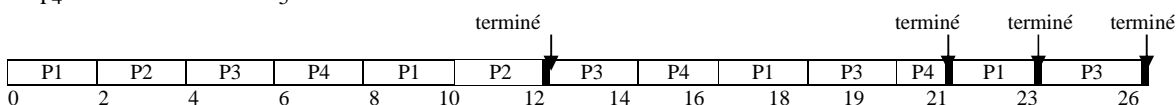
Remettre : le rapport et l'application bien commentée en C++ avec le jeu de tests.

Énoncés. Dans les systèmes d'exploitation modernes, plusieurs processus s'exécutent en même temps (multiprogrammation). Pour partager l'unité centrale (CPU) et les autres ressources entre ces plusieurs processus, le système d'exploitation organise et gère plusieurs files d'attente. Soit un système d'exploitation qui a une file d'attente des processus prêts pour l'exécution (ils n'attendent que le CPU) : **QueueReady**. L'ordonnancement des processus dans cette file d'attente est effectué par l'algorithme d'ordonnancement circulaire (tourniquet) (RR – Round Robin).

Dans la queue des processus prêts **QueueReady**, chaque processus possède son quantum de temps du processeur. Le premier processus est choisi et il s'exécute jusqu'à l'épuisement de son quantum, après il est placé à la fin de la file **QueueReady**. Si le processus en exécution termine son travail avant l'épuisement de son quantum, le processeur est immédiatement réalloué à un autre processus en attente. C'est un algorithme avec la réquisition du processeur, le plus ancien, le plus simple et le plus utilisé. Cet algorithme ne nécessite pas de l'estimation du temps d'exécution.

Exemple : Considérons les processus suivants (avec le quantum de temps du processeur, par exemple, 2 unités) :

Processus	Temps d'exécution
P1	8
P2	4
P3	9
P4	5



Travail à faire :

Écrire le programme qui simule la gestion de la file d'attente **QueueReady** d'un système d'exploitation. Les processus sont entrés à partir du clavier (le nombre de processus est précisé par l'utilisateur). Pour chaque processus il faut entrer l'information suivante : l'identité du processus (par exemple, p1, p2, p3, ...) et le temps d'exécution en unités (1, 2, 3, ...).

Quand tous les processus sont entrés, la simulation de la gestion de la file d'attente **QueueReady** par le système d'exploitation commence. À chaque **quantum de temps donné par l'utilisateur** (une itération dans une boucle), un processus est retiré de la file d'attente **QueueReady** et exécuté. Le processus pour l'exécution est pris de la tête de la file d'attente **QueueReady**. Le processus qui a été choisi est exécuté, son temps d'exécution est diminué du **quantum de temps** qu'il a utilisé le CPU. Si son temps d'exécution après la diminution est supérieur à 0, il est donc placé à la fin de la file d'attente **QueueReady**. Si son temps d'exécution après la décrémentation est inférieur ou égal à 0, le processus a terminé son cycle de vie et il meurt (sort de la file d'attente **QueueReady**). Quand la file d'attente **QueueReady** est vide, la simulation est terminée. *Pendant la simulation on ne tient pas compte de la libération anticipée du CPU avant la fin du quantum de temps (à cause de la terminaison ou du blocage du processus en exécution).*

Le programme doit afficher l'état initial de la file d'attente QueueReady d'un système d'exploitation et tous les états intermédiaires après chaque quantum de temps jusqu'à l'état final (la file QueueReady est vide). La file d'attente QueueReady doit être réalisée en utilisant les listes chaînées.

Exemple d'affichage:

État initial de la file d'attente des processus	- temps courant 0
P1 (8) P2 (4) P3 (9) P4 (5)	
État 1 après le 1-er quantum de temps de CPU (Q=2)	- temps courant 2
P2 (4) P3 (9) P4 (5) P1 (6)	
État 2 après le 2-e quantum de temps de CPU (Q=2)	- temps courant 4
P3 (9) P4 (5) P1 (6) P2 (2)	
État 3 après le 3-e quantum de temps de CPU (Q=2)	- temps courant 6
P4 (5) P1 (6) P2 (2) P3 (7)	
État 4 après le 4-e quantum de temps de CPU (Q=2)	- temps courant 8
P1 (6) P2 (2) P3 (7) P4 (3)	
État 5 après le 5-e quantum de temps de CPU (Q=2)	- temps courant 10
P2 (2) P3 (7) P4 (3) P1 (4)	
État 6 après le 6-e quantum de temps de CPU (Q=2)	- temps courant 12
P3 (7) P4 (3) P1 (4)	- processus P2 a terminé son exécution
État 7 après le 7-e quantum de temps de CPU (Q=2)	- temps courant 14
P4 (3) P1 (4) P3 (5)	
État 8 après le 8-e quantum de temps de CPU (Q=2)	- temps courant 16
P1 (4) P3 (5) P4 (1)	
État 9 après le 9-e quantum de temps de CPU (Q=2)	- temps courant 18
P3 (5) P4 (1) P1 (2)	
État 10 après le 10-e quantum de temps de CPU (Q=2)	- temps courant 20
P4 (1) P1 (2) P3 (3)	
État 11 après le 11-e quantum de temps de CPU (Q=2)	- temps courant 22
P1 (2) P3 (3)	- processus P4 a terminé son exécution
État 12 après le 12-e quantum de temps de CPU (Q=2)	- temps courant 24
P3 (3)	- processus P1 a terminé son exécution
État 13 après le 13-e quantum de temps de CPU (Q=2)	- temps courant 26
P3 (1)	
État 14 après le 14-e quantum de temps de CPU (Q=2)	- temps courant 28
File d'attente vide.	