



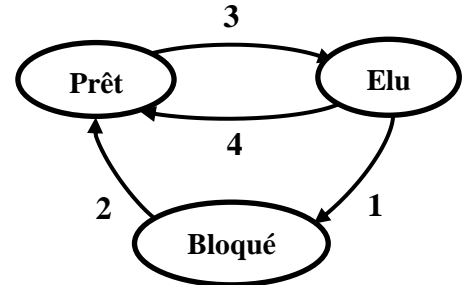
****Examen de Systèmes d'Exploitation 1****

Date : 21/05/2024

Durée: 1h30 - Documentation non autorisée

Exercice 1 : (Questions de Compréhension : 5 pts) (10 minutes)

Q1) Etant donné le diagramme d'états/transitions suivant, citer la/les transition(s) qui doivent être supprimée(s) si on utilise un algorithme d'ordonnancement sans réquisition (non préemptif). Justifier votre réponse. **(1 pt)**



La transition à supprimer est **la transition (4)** qui passe de l'état **Elu à l'état Prêt**, car cette transition a uniquement lieu lors d'une préemption.

Q2) Question : Qu'est-ce qu'un déroutement ? **(1 pt)**

Un déroutement est un type d'interruption interne (provoquée par le processus lui-même). Il nécessite l'intervention du système d'exploitation.

Q3) Quels sont les avantages et inconvénients du choix d'un quantum petit pour l'algorithme de scheduling Round Robin ? **(1 pt)**

Avantage : partage du processeur (Équité entre les processus) : Chacun des utilisateurs a l'impression de disposer de son propre processeur.

Inconvénients : surcharge du système du aux fréquentes commutations de contexte.

Q4) Décrivez ce qui se passe, du côté du système d'exploitation, lorsqu'une touche de clavier est pressée : **(1 pt)**

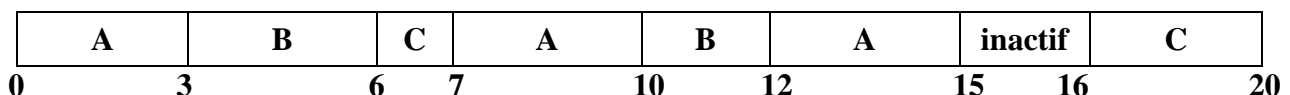
Après chaque touche pressée, une interruption (de type matérielle associée au clavier) est générée. Le processeur interrompt son traitement pour lancer la routine d'interruption associée.

Q5) Dans quel cas est-il intéressant de masquer une interruption ? **(1 pt)**

Par exemple, pour éviter que la routine d'interruption elle-même ne soit interrompue par une autre interruption (qu'il faut masquer).

Exercice 2 : (Ordonnancement : 5 pts) (25 minutes)

La figure suivante représente le diagramme de Gantt d'un scheduling du processeur utilisant l'algorithme « **Round Robin** » et trois processus : **A, B** et **C**. (Si les processus arrivés en même temps ont file d'attente, le système prend l'ordre d'arrivée des processus **A, B, C**)



Q1) Quelle est la durée du quantum ? **(0.25 pts)**

03 [UT]

Q2) Quel est le temps d'attente du processus A ? **(0.25 pts)**

06 [UT]

Q3) Quel est le temps de réponse du processus B ? Justifiez. **(0.75 pts)**

Le processus B peut être arrivé à l'instant 0, 1 ou 2. Le temps de réponse en dépend : il peut être 12, 11 ou 10 [UT].

Q4) Quel est le temps d'attente du processus C ? Justifiez. **(0.75 pts)**

Le processus C peut être arrivé à l'instant 0, 1 ou 2. Le temps d'attente en dépend : il peut être 6, 5 ou 4 [UT].

Q5) Que s'est-il passé entre les instants $t = 15$ et $t = 16$? Justifiez. **(1 pt)**

A l'instant $t = 15$, les processus A et B sont terminés ; le seul processus restant est le processus C, mais il est en attente d'une opération d'entrée/sortie ou d'un évènement. Le processeur est donc inactif jusqu'à la reprise de C à l'instant $t = 16$.

Q6) Quel est l'état du processus C à l'instant $t = 9$. **(0.5 pts)**

A l'instant $t = 9$, le processus C est en attente d'une opération d'entrée/sortie ou d'un évènement

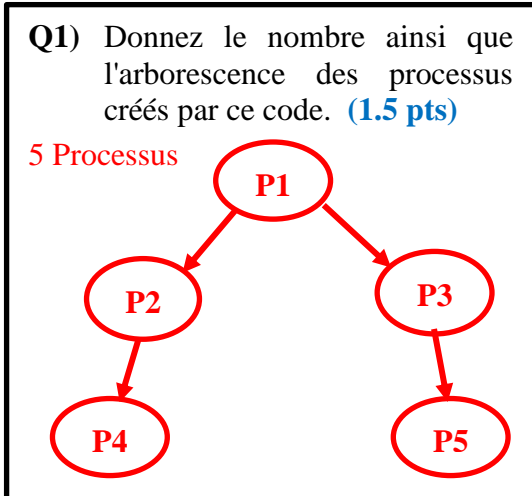
Q7) Dessinez le diagramme de Gantt du même problème, mais en considérant un quantum égal à 4. **(1.5 pts)**



Exercice 3 : (La primitive fork : 3 pts) (20 minutes)

Dans cet exercice on suppose que les numéros des **PIDs** attribués aux processus sont strictement croissants. Si un père a un **PID (n)**, son fils aura un **PID (m)** tel que $(m > n)$. Le premier processus fils sera créé aura un PID inférieur au seconde processus fils créé ($PID(Père) = n \Rightarrow PID(Fils1) = n+1, PID(Fils2) = n+2, \dots$). Soit le code suivant :

```
int main ()
{
    pid_t p1 , p2 ;
    p1 = fork () ;
    p2 = fork () ;
    if ( ( p1 - p2 ) > 0 )
        fork () ;
    printf ("Je suis %d : p1 = %d , p2 = %d \n",getpid(), p1, p2) ;
    return 0 ;
}
```



Q2) Proposez pour chaque processus créé son affichage à l'écran. Nous rappelons que les **PIDs** sont affichés comme suit : $PID-P1=301$, $PID-P2=302$, $PID-P3=303$, $PID-P4=304$, ...etc.

Par exemple, le processus P8 ($PID-P8=308$) avec $p1 (PID-P5) = 305$ et $p2=0$ aura pour affichage :

Je suis 308 : $p1 = 305$, $p2 = 0$. (1.5 pts)

Je suis 301, $p1 = 302$, $p2 = 303$.

Je suis 302, $p1 = 0$, $p2 = 304$.

Je suis 303, $p1 = 302$, $p2 = 0$.

Je suis 304, $p1 = 0$, $p2 = 0$.

Je suis 305, $p1 = 302$, $p2 = 0$.

Exercice 4 : (Algorithmes de Remplacement de Pages : 7 pts) (30 minutes)

On considère une mémoire paginée possédant **N** cadres de pages. Soit un programme possédant un espace virtuel de **512 Octets** et la taille de page est de **64 Octets**. Le programme fait référence, durant son exécution, aux adresses virtuelles suivantes :

120, 265, 50, 340, 280, 365, 450, 390, 110, 480, 280, 440, 500, 95

Q1) Donnez le nombre de pages de ce programme ? **(0.25 pts)**

Nombre de pages MV = Taille MV / Taille de page = $512/64 = 8$ pages.

Q2) Donnez la suite des numéros de pages référencés ? **(1 pt)**

Donc, il suffit de diviser l'adresse par 64, ce qui donne la suite des numéros de pages référencés suivante :

1, 4, 0, 5, 4, 5, 7, 6, 1, 7, 4, 6, 7, 1

Q3) Combien de défauts de pages peuvent se produire au minimum ? Justifiez. **(0.5 pts)**

Au minimum, on a 6 défauts de pages.

Justification : Chacune des 6 pages constituant la chaîne de références provoquera certainement un défaut de pages lors de la première utilisation.

Q4) Combien de défauts de pages peuvent se produire au maximum ? Justifiez. **(0.5 pts)**

Au maximum, on a 14 défauts de pages.

Justification : Cela arrivera lorsque $N=1$ (un seul cadre de page). Chaque page de la chaîne provoquera ainsi un défaut de page.

Sachant que la taille de la mémoire physique est de **320 Octets**. **(0.25 pts)**

Q5) Donnez le nombre de cadres (**N**) ?

Nombre de cadres (N) = Taille M phy / Taille de page = $320/64 = 5$ cadres

Q6) En prenant **N=3**, déterminez le nombre de défauts de page générés en applique les algorithmes de remplacement **FIFO**, **LRU**, et **FIFO de la seconde chance**. **(3 pts)**

FIFO : le nombre de défauts de page = 9 (1 pt)														
Références	1	4	0	5	4	5	7	6	1	7	4	6	7	1
Cadre 1	<u>1</u>	1	1	<u>5</u>	5	5	5	5	<u>1</u>	1	1	1	1	1
Cadre 2		<u>4</u>	4	4	4	4	<u>7</u>	7	7	7	<u>4</u>	4	4	4
Cadre 3			<u>0</u>	0	0	0	0	<u>6</u>	6	6	6	6	<u>7</u>	7
Défaut de page	D	D	D	D			D	D	D		D		D	

LRU : le nombre de défauts de page = 10 (1 pt)														
Références	1	4	0	5	4	5	7	6	1	7	4	6	7	1
Cadre 1	<u>1</u>	1	1	<u>5</u>	5	5	5	5	<u>1</u>	1	1	<u>6</u>	6	6
Cadre 2		<u>4</u>	4	4	4	4	4	<u>6</u>	6	6	<u>4</u>	4	4	<u>1</u>
Cadre 3			<u>0</u>	0	0	0	<u>7</u>	7	7	7	7	7	7	7
Défaut de page	D	D	D	D			D	D	D		D	D		D

FIFO de la seconde chance : le nombre de défauts de page = 11 (1 pt)														
Références	1	4	0	5	4	5	7	6	1	7	4	6	7	1
Cadre 1	<u>1</u> ₁ ⁺	<u>1</u> ₁ ⁺	<u>1</u> ₁ ⁺	<u>5</u> ₁	5 ₁	5 ₁	5 ₁ ⁺	5 ₀	<u>1</u> ₁	1 ₁	1 ₀	1 ₀ ⁺	<u>7</u> ₁	7 ₀
Cadre 2		<u>4</u> ₁	4 ₁	4 ₀ ⁺	4 ₁ ⁺	4 ₁ ⁺	4 ₀	<u>6</u> ₁	6 ₁ ⁺	6 ₁ ⁺	<u>4</u> ₁	4 ₁	4 ₁ ⁺	<u>1</u> ₁
Cadre 3			<u>0</u> ₁	0 ₀	0 ₀	0 ₀	<u>7</u> ₁	7 ₁ ⁺	7 ₀	7 ₁	7 ₀ ⁺	<u>6</u> ₁	6 ₁	6 ₀ ⁺
Défaut de page	D	D	D	D			D	D	D		D	D	D	D

Q7) On suppose que le temps de chargement d'une page du disque vers la mémoire est X et que le temps de sauvegarde d'une page de la mémoire vers le disque est Y. Calculer le temps total de traitement (en fonction de X et Y) de la chaîne de références pour l' algorithme LRU. (1.5 pts)

LRU	Références	1	4	0	5	4	5	7	6	1	7	4	6	7	1
	Défaut de page	D	D	D	D			D	D	D		D	D		D
	Temps de traitement	X	X	X	X			X	X	X		X	X		X
					+			+	+	+		+	+		+
					Y			Y	Y	Y		Y	Y		Y

Temps Total de Traitement = 10 X + 7 Y [U.T]

Exercice 5 : (Bonus : 2 pts) (5 minutes)

Si un système utiliser l'algorithme du travail le plus court d'abord pour l'ordonnancement à court terme et une moyenne exponentielle de $\alpha = 0.5$, quel est le prochaine cycle processeur estimé pour un processus aux cycles processeurs 5, 8, 3 et 5 et une valeur initiale de 10 pour e_1 . (2 pts)

Les différentes valeurs de e sont les suivants :

$$e_1 = 10 \implies e_2 = 0.5 \times 5 + 0.5 \times 10 = 7.5$$

$$e_2 = 7.5 \implies e_3 = 0.5 \times 8 + 0.5 \times 7.5 = 7.75$$

$$e_3 = 7.75 \implies e_4 = 0.5 \times 3 + 0.5 \times 7.75 = 5.375$$

$$e_4 = 5.375 \implies e_5 = 0.5 \times 5 + 0.5 \times 5.375 = 5.1875$$

Bon courage