Université de M'sila
Faculté : MI Département : d'informatique
Année Universitaire : 2023 / 2024
2ème Année Licence (2L)

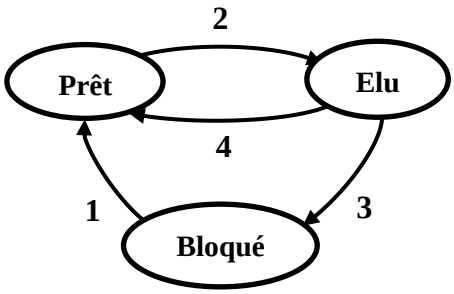
Nom : ……
Prénom :……...……
Groupe : ……..……...

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) 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).

Q3) 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.

Q4) 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 (Équite 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.

Q5) 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.

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 : **X**, **Y** et **Z**. (Si les processus arrivés en même temps ont file d'attente, le système prend l'ordre d'arrivée des processus **X**, **Y**, **Z**)

X	Y	Z	X	Y	X	inactif	Z

0 3 6 7 10 12 15 16 20

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

03 [UT]

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

06 [UT]

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

Le processus Y 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 **Z** ? Justifiez. **(0.75 pts)**

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

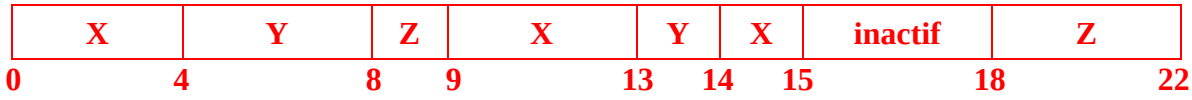
Q5) Que s'est-il passé entre les instants **t = 15** et **t = 16** ? Justifiez. **(1 pt)**

A l'instant t = 15, les processus X et Y sont terminés ; le seul processus restant est le processus Z, 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 Z à l'instant t = 16.

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

A l'instant t = 9, le processus Z 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)**



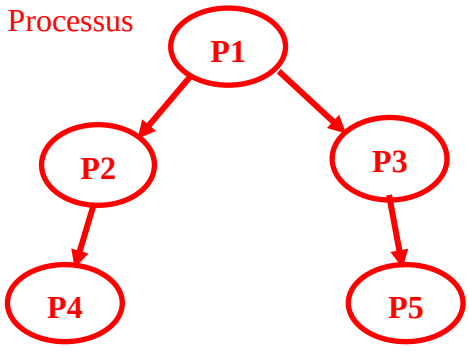
<u>Exercice 3 :</u> (La primitive fork : 3 pts) (20 minutes)

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** ⇒ PID(Fils1) = **n+1**, PID(Fils2) = **n+2**, ...etc.).
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 ;
}
```

Q1) Donnez le nombre ainsi que l'arborescence des processus créés par ce code. **(1.5 pts)**

5 Processus



Q2) Proposez pour chaque processus créé son affichage à l'écran. Nous rappelons que les **PIDs** sont affichés comme suit : *PID-P1=**501**, PID-P2=**502**, PID-P3=**503**, PID-P4=**504**, ...etc.*

Par exemple, le processus P8 (PID-P8=**508**) avec p1 (PID-P5) = **505** et p2=**0** aura pour affichage :

Je suis 508 : p1 = 505, p2 = 0. **(1.5 pts)**

Je suis 501, p1 = 502, p2 = 503.

Je suis 502, p1 = 0, p2 = 504.

Je suis 503, p1 = 502, p2 = 0.

Je suis 504, p1 = 0, p2 = 0.

Je suis 505, p1 = 502, 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 **640 Octets** et la taille de page est de **64 Octets**. Le programme fait référence, durant son exécution, aux adresses virtuelles suivantes :

184, 329, 114, 404, 344, 429, 514, 454, 174, 544, 344, 504, 564, 159

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

Nombre de pages MV = Taille MV / Taille de page = 640/64 = **10 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 :

2, 5, 1, 6, 5, 6, 8, 7, 2, 8, 5, 7, 8, 2

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 chaine 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 chaine provoquera ainsi un défaut de page.

Sachant que la taille de la mémoire physique est de **384 Octets.**

Q5) Donnez le nombre de cadres (**N**) ? **(0.25 pts)**

Nombre de cadres (N) = Taille MPhy / Taille de page =384/64 = **6 cadres**

Q6) En prenant **N=3**, déterminez le nombre de défauts de page génères 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	2	5	1	6	5	6	8	7	2	8	5	7	8	2
Cadre 1	2	2	2	6	6	6	6	6	2	2	2	2	2	2
Cadre 2		5	5	5	5	5	8	8	8	8	5	5	5	5
Cadre 3			1	1	1	1	1	7	7	7	7	7	8	8
Défaut de page	D	D	D	D			D	D	D		D		D	

LRU : le nombre de défauts de page = 10 (1 pt)														
Références	2	5	1	6	5	6	8	7	2	8	5	7	8	2
Cadre 1	2	2	2	6	6	6	6	6	2	2	2	7	7	7
Cadre 2		5	5	5	5	5	5	7	7	7	5	5	5	2
Cadre 3			1	1	1	1	8	8	8	8	8	8	8	8
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	2	5	1	6	5	6	8	7	2	8	5	7	8	2
Cadre 1	2_1^+	2_1^+	2_1^+	6_1	6_1	6_1	6_1^+	6_0	2_1	2_1	2_0	2_0^+	8_1	8_0
Cadre 2		5_1	5_1	5_0^+	5_1^+	5_1^+	5_0	7_1	7_1^+	7_1^+	5_1	5_1	5_1^+	2_1
Cadre 3			1_1	1_0	1_0	1_0	8_1	8_1^+	8_0	8_1	8_0^+	7_1	7_1	7_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 chaine de références pour l' algorithme **LRU**. **(1.5 pts)**

LRU	Références	2	5	1	6	5	6	8	7	2	8	5	7	8	2
	Défaut de page	D	D	D	D			D	D	D		D	D		D
	Temps de traitement	X	X	X	X + Y			X + Y	X + Y	X + Y		X + Y	X + Y		X + 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 :

$e_1 = 10 \quad ==> \quad e_2 = 0.5 \times 5 + 0.5 \times 10 = 7.5$

$e_2 = 7.5 \quad ==> \quad e_3 = 0.5 \times 8 + 0.5 \times 7.5 = 7.75$

$e_3 = 7.75 \quad ==> \quad e_4 = 0.5 \times 3 + 0.5 \times 7.75 = 5.375$

$e_4 = 5.375 \quad ==> \quad e_5 = 0.5 \times 5 + 0.5 \times 5.375 = 5.1875$

Bon courage