



Examen de Systèmes d'Exploitation 1

Date : 24/05/2023

Durée: 1h30 - Documentation non autorisée

Exercice 1 : (Questions de Compréhension : 7 pts) (20 minutes)

Partie A) QCM : Mettez une croix sur une seule réponse (1 pt).

Q1) Parmi les commandes ci-dessous, laquelle permet d'afficher les processus en cours d'exécution ?

- dir ps man ls

Q2) Quelle commande permet d'interrompre un processus dans un système d'exploitation de type UNIX ?

- stop interrupt end kill

Q3) Un processus peut passer par les états (Actif, Prêt, Bloqué) dans l'ordre suivant :

- Actif -> Bloqué -> Prêt Bloqué -> Actif -> Prêt Actif -> Prêt -> Bloqué

Q4) Lors de l'initialisation du système (boot) tous les processus sont créés par le mécanisme fork.

- Vrai. Faux.

Partie B) Qui suis-je ? (3 pts).

Q5) Je suis un appel système qui affiche le PID du processus père.
Q6) Je suis un appel système qui permet de créer un processus
Q7) Je suis un processus qui s'est terminé mais son père n'a pas encore lu son code de retour.
Q8) Je suis la différence entre le temps de la première exécution et le temps d'entrée dans le système.
Q9) Je suis la différence entre le temps de terminaison et le temps d'entrée dans le système.
Q10) Je suis le premier programme qui est lancé à la mise sous tension de l'ordinateur.

Partie C) Questions de Cours (3 pts).

Q11) Quelle est la différence entre un processeur et un processus ?

.....

.....

.....

.....

Q12) Décrire brièvement ce qu'est un ordonnanceur.

.....

Q13) Ecrire le microprogramme formel correspondant à l'exécution des instructions Assembleur suivantes (**Annexe** : Schéma d'un ordinateur sous le modèle de Von Neumann):

<p>SUB @ ; // Faire $ACC \leftarrow ACC - [@]$</p> <p><u>Réponse :</u></p> <p>..... </p>	<p>JMP @ ; // Effectue un saut à l'adresse @</p> <p><u>Réponse :</u></p> <p>..... </p>
--	---

Exercice 2 : (Edition de liens : 2 pts) (10 minutes)

La translation d'un module consiste à modifier son contenu pour qu'il puisse s'exécuter à un endroit différent de celui pour lequel il était prévu initialement.

Q1) Donnez le résultat de la translation du module suivant en assembleur 68000, Si l'éditeur de liens décide de mettre la section de code à l'adresse hexadécimale **032340** et la section des données à l'adresse hexadécimale **043320**, les adresses génères sont sur **4 octets**.

Section code :		
AJOUT10 : <i>move.w #10, D0</i>	0	: 30 3C 00 0A
<i>jmp C</i>	4	: 4E F9 00 00 00 0E
AJOUT16 : <i>move.w #16, D0</i>	A	: 30 3C 00 10
C : <i>add.w D1, D0</i>	E	: D1 01
<i>Move.w D0, MEMO</i>	10	: 33 C0 00 00 02
<i>rts</i>	16	: 2E 75
Section données :		
LOC : <i>ds.w 1</i>	0	: 00 00
MEMO : <i>ds.w 1</i>	2	: 00 00

Exercice 3 : (Ordonnancement : 4 pts) (20 minutes)

On considère un système possédant deux processeurs et une seule file d'attente pour les processus prêts. Soit le scénario d'arrivée des processus suivants : A, B, C, et D, ayant les caractéristiques suivantes (la priorité 1 correspond à la plus faible priorité). Le CPU1 a la priorité d'accès à la file des processus prêts par rapport au CPU2.

Processus	Priorité	Instant d'arrivée	Durée d'exécution
A	2	0	4
B	4	2	5
C	3	0	6
D	1	0	7

Q1) Complétez les diagrammes, pour chacun des algorithmes de scheduling suivants :

a) FCFS (First Come First Served),

CPU 1																										
CPU 2																										
File d'attente																										
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

b) Plus haute priorité préemptif (avec réquisition),

CPU 1																										
CPU 2																										
File d'attente																										
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

c) Round Robin (RR avec quantum = 2)

CPU 1																										
CPU 2																										
File d'attente																										
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Q2) Donnez les temps d'attente (TA) et de réponse (TR) des processus.

Processus	Algorithme FCFS		Algorithme Plus haute priorité		Algorithme RR avec Q = 2	
	Temps d'attente	Temps de Réponse	Temps d'attente	Temps de Réponse	Temps d'attente	Temps de Réponse
A
B
C
D

Exercice 4 : (Gestion de la mémoire : 4 pts) (20 minutes)

On considère un système utilisant la technique de pagination et ayant les caractéristiques suivantes :

- Une table de page ayant 2^{16} entrées
- Chaque entrée de la table de pages est codée sur **16 bits**. Une entrée contient un numéro de cadre de page et un bit de présence/absence.
- Le déplacement (offset) est codé sur **16 bits**
- Une adresse virtuelle indexe **2 octets**

Répondez aux questions suivantes en justifiant toujours votre réponse :

Q1) Quelle est la taille d'une page (un cadre)?

.....

Q2) Quelle est la taille de la mémoire physique ?

.....

Q3) Quelle est la taille de la mémoire virtuelle ?

.....

Q4) Quelle est la taille (en bit) du bus d'adresse de ce système ?

.....

Exercice 5 : (Algorithmes de remplacement de pages : 3 pts) (15 minutes)

Déterminez le nombre de défauts de page engendrés par les algorithmes FIFO, LRU et FIFO de la seconde chance sur la chaîne de références : 1, 2, 3, 1, 7, 4, 1, 2, 7, 4, 3, 1 avec 4 cadres de page :

FIFO : le nombre de défauts de page =												
La chaîne de références												
Cadre 1												
Cadre 2												
Cadre 3												
Cadre 4												
Défaut de page												

LRU : le nombre de défauts de page =												
La chaîne de références												
Cadre 1												
Cadre 2												
Cadre 3												
Cadre 4												
Défaut de page												

FIFO de la seconde chance : le nombre de défauts de page =												
La chaîne de références												
Cadre 1												
Cadre 2												
Cadre 3												
Cadre 4												
Défaut de page												

Exercice 6 : (Bonus : 2 pts) (5 minutes)

Avec une taille moyenne de processus **P**, une taille de page **S** et une taille d'une entrée de la table de pages **E**, Quelle taille de page minimise l'espaces gaspiller en raison de la fragmentation interne et de la fragmentation de tables ?

.....

.....

.....

.....

.....

Bon courage

ANNEXE

