

Questions de cours (4,5 points)

(0.5) 1- Pourquoi l'algorithme d'ordonnancement SJF n'est-il pas réellement applicable ?
car on ne peut pas prévoir la durée d'exécution d'un processus.

(1) 2- A. Définir la notion de PCB,

PCB : Process Control Block: structure de données associée à un processus et contenant toute l'information décrivant le contexte du processus => bloc de contrôle.

(1) B. Citer quatre attributs parmi ceux qui constituent le PCB.

PID, PPID, Etat, Priorité, Compteur Ordinal, Fichiers ouverts, pointeurs (seg de code, seg de données, seg de pile..), temps d'exécution...

(0.75) =
.5 diminu°
.25 cause
3- Quel est l'effet de la diminution du quantum sur les performances de l'algorithme RR (tourniquet)?

La diminution du quantum entraîne la dégradation des performances de l'algorithme RR, car le temps de commutation/changement de contexte augmente.

(0.5) 4- Les algorithmes d'ordonnancement basés sur des priorités peuvent engendrer la famine (non exécution) des processus à faible priorité. Comment peut-on éviter ce problème ?

On utilise un autre algorithme: tourniquet ou priorité dynamique.

5- Citer trois architectures des systèmes d'exploitation (sans détail).

(0.75) Monolithique, en couches, modulaire, client-serveur, machine virtuelle...

EXERCICE 1 : Gestion des processus (10 points)

On considère une architecture monoprocesseur dans laquelle on désire exécuter l'ensemble des processus suivants :

<i>Processus</i>	<i>Temps d'arrivée</i>	<i>Temps d'exécution total</i>
A	0	10
B	0	6
C	1	8
D	5	4

Tout au long de cet exercice, nous considérons les hypothèses suivantes :

- Nous disposons d'un seul canal pour gérer un disque.
- L'ordre des services des requêtes sur le disque se fait selon la politique FCFS.
- Une opération d'entrée-sortie commencée ne peut plus être préemptée.
- Arrivé à la moitié de son exécution, chaque processus doit faire 3 unités de temps d'entrée-sortie, puis reprendre son exécution.

A- On considère que l'algorithme utilisé pour ordonnancer ces processus est l'algorithme du tourniquet avec un quantum $q=2$.

A.1. Remplir les grilles annexes A en suivant les hypothèses suivantes :

- Si le système a le choix entre plusieurs processus, il choisit celui qui attend depuis le plus longtemps.
- Si le système a le choix entre plusieurs processus ayant le même temps d'attente, il adopte l'ordre suivant: A-B-C-D.

A	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S																			x	x	x																				
Attente																x	x	x	x																						
Prêt				x	x	x	x			x	x	x	x	x												x	x														
Actif	x	x						x	x						x											x	x			x	x	x									

B	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S											x	x	x																												
Attente																																									
Prêt	x	x				x	x	x	x					x	x			x	x																						
Actif				x	x					x						x	x			x																					

C	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S																	x	x	x																						
Attente																x	x																								
Prêt		x	x	x			x	x	x	x	x									x				x	x																
Actif						x	x						x	x								x	x			x	x														

D	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S														x	x	x																									
Attente													x																												
Prêt							x	x	x	x							x																								
Actif											x	x																													

A.2. Calculer le temps de rotation moyen TRM1 (appelé aussi temps de traitement moyen) de cet algorithme en donnant la formule détaillée.

$$TRM = [(28-0) + (19-0) + (25-1) + (18-5)]/4 = 21$$

B- On considère un algorithme d'ordonnancement à priorité préemptif, en supposant que la priorité d'un processus est inversement proportionnelle à son

temps d'exécution total restant. C'est-à-dire que le processus ayant le temps d'exécution total restant le plus court est le plus prioritaire.

B.1. S'agit-il d'un algorithme à priorité statique ou dynamique ? Justifier.

Il s'agit d'un algorithme à priorité dynamique car la valeur de la priorité change au cours de l'exécution.

B.2. Quel algorithme parmi ceux que vous connaissez produit un résultat équivalent à celui de cet algorithme? Justifier votre réponse.

SNRT ou SRT ou SJF avec préemption, car il favorise les processus les plus courts.

B.3. Remplir les grilles annexes B en suivant les hypothèses suivantes :

- Si le système a le choix entre plusieurs processus de même priorité, il favorise celui qui était entrain de s'exécuter.

A	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S																									x	x	x														
Attente																																									
Prêt	x	x	x	x	x	x	x	x	x	x	x	x	x	x				x	x	x	x																				
Actif																x	x	x						x	x				x	x	x	x	x								

B	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S					x	x	x																																		
Attente																																									
Prêt								x																																	
Actif	x	x	x						x	x	x																														

C	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S																x	x	x																							
Attente																																									
Prêt		x	x			x	x	x	x	x	x	x																													
Actif				x	x								x	x					x	x	x	x																			

D	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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
E/S								x	x	x																															
Attente																																									
Prêt																																									
Actif						x	x					x	x																												

B.4. Calculer le temps de rotation moyen TRM2 (appelé aussi temps de traitement moyen) de cet algorithme en donnant la formule détaillée.

$$TRM = [(31-0) + (10-0) + (21-1) + (12-5)]/4 = 17$$

C- Comparer entre TRM1 et TRM2 et analyser ces résultats.

TRM1 > TRM2 car SNRT fait en sorte de terminer les processus les plus courts avant les autres, ce qui fait que leur temps de séjour est le plus court possible.

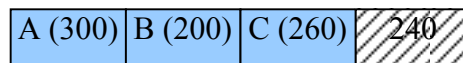
EXERCICE 2 : Gestion de la mémoire (5,5 points)

A. Allocation contiguë

On considère un espace mémoire de 1000 blocs, utilisant une allocation contiguë. On note par (+) une demande d'allocation et par (-) une demande de libération.

A.1. En utilisant l'algorithme d'allocation First Fit, donner les différents états de la mémoire centrale après chacune des étapes suivantes (les étapes sont successives, initialement la mémoire est vide) :

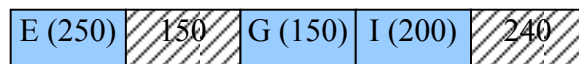
Etape 1 : A(+300), B(+200), C(+260),



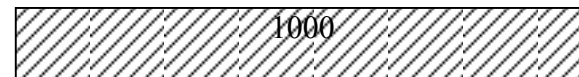
Etape 2 : B(-200), D(+100), A(-300), E(+250), C(-260),



Etape 3 : G(+150), H(+120), D(-100), H(-120), I(+200),



Etape 4 : G(-150), E(-250), J(+100), J(-100), I(-200).



A.2. Quel est l'avantage d'une stratégie d'allocation "First Fit" par rapport à une stratégie "Best Fit"?

First Fit est plus facile à implémenter et plus rapide à exécuter car elle permet au processus d'occuper la première case en mémoire disponible, alors que le best fit doit parcourir toutes les cases vides pour trouver la plus adéquate.

B. Pagination

B.1. Définir la pagination. Quels sont ses avantages par rapport aux techniques de la gestion contiguë de la mémoire, par partition fixe et dynamique?

La pagination est une technique de gestion de la mémoire non-contiguë qui partitionne la mémoire en petits morceaux de même taille: pages physiques, cadres, cases ou frames, et partitionne les processus en morceaux de même taille: pages (logiques). Les pages physiques correspondantes aux pages logiques sont

indiquées dans la table de pages.

B.2. Calculer l'adresse physique correspondante à l'adresse logique 36870, en supposant que :

(1)

- Bus d'adresses de 16 bits
- Taille d'une page 4Ko
- Table de page (ci contre)

@logique / taille page = num de la page, reste = offset

@logique = 36870

taille page = 4096

numéro de page = 9

offset = 6

=> @physique = 12292

= 0011 0000 0000 0110

15	000	0
14	000	0
13	010	1
12	000	0
11	100	1
10	111	1
09	011	1
08	000	0
07	110	1
06	000	0
05	000	0
04	101	1
03	000	0
02	000	1
01	000	0
00	001	1