

TD1 de Systèmes d'exploitation
Gestion de la mémoire

Exercice1 :

On considère un système disposant de 32 Mo de mémoire principale. La partie résidente du système est sur 8 Mo. Les processus arrivent dans le système dans l'ordre suivant :

Instant t (ms)	Processus	Taille partition	Durée de l'allocation
0	P0	10 Mo	6 ms
4	P1	6 Mo	10 ms
7	P2	2 Mo	9 ms
10	P3	10 Mo	6 ms
15	P4	14 Mo	5 ms
16	P5	10 Mo	5 ms

Donnez les états d'occupation de la mémoire aux différentes étapes de traitement de ces processus, en utilisant les algorithmes suivants : **First-Fit**, **Best-Fit**, (*Next-Fit et Worst-Fit*).

Remarques :

- Les requêtes sont satisfaites selon leur ordre d'arrivée.
- Si une requête n'est pas satisfaite, elle est mise en attente jusqu'à la libération de la mémoire.

N.B : Ne pas traiter Next-Fit et Worst-Fit en TD.

Exercice2 :

On considère un système disposant de 64Mo de mémoire principale. Les processus "arrivent" dans le système dans l'ordre suivant :

Instant t (ms)	Processus	Taille partition	Durée d'allocation
0	P0	15 Mo	6 ms
4	P1	17 Mo	10 ms
7	P2	850 Ko	9 ms
10	P3	6 Mo	6 ms
15	P4	23 Mo	5 ms

La plus petite partition que l'on peut allouer à un processus est de 1024 octets (ou 1ko).

L'allocation de la mémoire est réalisée avec l'algorithme Buddy System.

1. Donner le nombre d'entrées de la table des pointeurs de listes des partitions libres. Justifier votre réponse.
2. Donner les états d'occupation de la mémoire aux différentes étapes de traitement de ces processus (spécifier les adresses et les tailles des partitions sur les schémas et éventuellement les processus qui les occupent).

3. La taille de la mémoire est à égale à 2^z .

Quand une partition est libérée, le nombre de partitions libres dans le système peut :

- a) Augmenter de 1. Justifier votre réponse.
- b) Demeurer inchangé. Justifier votre réponse.
- c) Diminuer de n , avec $1 \leq n < z$. Justifier votre réponse.

4. Déterminer la valeur de n , si la taille de la mémoire est de 64Mo et la taille minimale d'une partition est de 1024 octets.

Exercice3 :

On considère un système sur une machine monoprocesseur disposant de 16Mo de mémoire physique, avec la partie résidente du système sur 4Mo. Les processus "arrivent" dans le système dans l'ordre suivant :

Processus	Arrivée	Taille	Temps processeur
P0	0	5 Mo	6 ms
P1	2	4 Mo	4 ms
P2	5	6 Mo	3 ms
P3	6	5 Mo	2 ms
P4	6	1 Mo	2 ms
P5	7	3 Mo	3 ms

Un processus dont la demande de mémoire n'est pas satisfaite est mis en attente.

1) L'ordonnancement des processus est réalisé avec l'algorithme SJF.

a) Donner les états d'occupation de la mémoire et le diagramme de Gantt aux différentes étapes de traitement de ces processus, sous les hypothèses suivantes :

- Partitions fixes de tailles : 6 Mo, 4 Mo, 2 Mo et 4 Mo (pour le système) ;
- L'allocateur de la mémoire utilise une file de processus par partition.

b) Calculer le temps d'attente de chacun des processus.

2) L'ordonnancement des processus est réalisé avec l'algorithme SJF (ou le plus court d'abord). Pour la gestion mémoire, on utilise la technique des partitions variables avec l'algorithme Buddy System ;

a) Donner les états d'occupation de la mémoire et le diagramme de Gantt aux différentes étapes de traitement de ces processus.

b) Calculer le temps d'attente de chacun des processus.

Remarque :

- Un processus n'est « réellement créé » que s'il dispose de toute la mémoire nécessaire à son exécution.
- A chaque libération de la mémoire, on commence d'abord par l'allocation de la mémoire au processus en attente, ensuite on alloue le processeur à un processus de la file « prêt ».

Exercice4 :

Soit un système qui utilise l'allocation par partitions variables. Taille de la mémoire principale est égale à 128ko. A l'instant t , l'état d'allocation de la mémoire centrale est donné dans le tableau suivant (Ordre d'arrivée: A, B, C, D, E et F) :

	A	B	C	D	E	F
Tailles demandées	6ko	20ko	5ko	30ko	5ko	15ko
Fin(libération)	-	$t+4$	-	$t+10$	$t+5$	$t+11$

En utilisant l'algorithme Buddy system :

A) Représenter l'état de la mémoire, à l'instant t , sous forme d'arbre.

B.1) Représenter l'état final de la mémoire centrale sous forme d'arbre, après les opérations suivantes :

1) à l'instant $t+2$, allocation d'une partition G de taille égale à 20 Ko;

2) à l'instant $t+6$, allocation d'une partition H de taille égale à 15 Ko;

3) à l'instant $t+8$, allocation d'une partition I de taille égale à 40 Ko.

Une requête(ou demande) non satisfaite est mise en attente. Les requêtes en attente sont satisfaites selon leur ordre d'arrivée.

B.2) Donner la date (le temps) d'allocation de la mémoire et l'adresse (puissance de 2) de chacune des partitions sous la forme ci-dessous :

	Temps d'arrivée	Tailles demandées	Dates d'allocation	Adresses
G	$t+2$	20ko	$t + \dots$	
H	$t+6$	15ko	$t + \dots$	
I	$t+8$	40ko	$t + \dots$	

A) Lorsqu'on libère une partition, le nombre de partitions libres peut diminuer de n . Déterminer la valeur de n , pour le cas de cet exercice (taille de la mémoire=128ko et Taille min d'une partition = 8ko).

TP : Ecrire en langage c les algorithmes first-fit, best-fit et buddy system.

=====