

Systemes d'Exploitation 1

Série TD N°4

Gestion de la mémoire : Allocation de mémoire contiguë

Exercice 1:

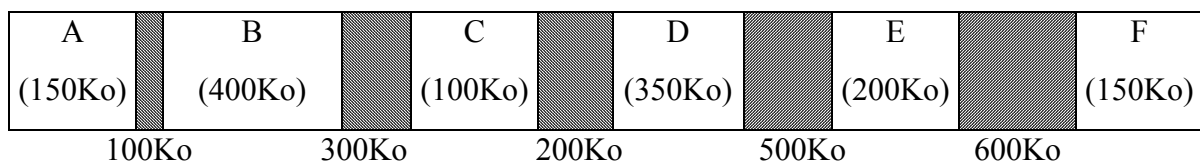
Soit une mémoire centrale utilisant la technique d'allocation contiguë par partitionnement fixe. Cette mémoire est composée de 5 partitions P1, P2, P3, P4 et P5. Ces partitions ont pour tailles respectives 100, 500, 200, 300 et 600 KO. Soient 4 processus A, B, C et D de tailles respectives 212, 417, 112 et 426 KO.

Donner les différents états de la mémoire centrale pour charger les 4 processus A, B, C et D (dans l'ordre) en utilisant les algorithmes d'allocation suivants :

- a- FIRST-FIT
- b- BEST-FIT
- c- WORST-FIT
- d- NEXT-FIT

Exercice 2:

Soit un système d'exploitation, qui utilise l'allocation contiguë par partition variable. On considère à l'instant T l'état suivant de la mémoire centrale (les zones hachurées sont libres):



A) Représenter l'évolution de la mémoire centrale, en fonction de l'arrivée des processus K, L, M et O, selon cet ordre: K (200 Ko), L (450 Ko), M (250 Ko) et O (50 Ko) :

- 1. en utilisant la stratégie d'allocation First Fit
- 2. en utilisant la stratégie d'allocation Best Fit

B) Nous désirons choisir un moyen de mémorisation des parties libres et parties occupées de la mémoire, ainsi qu'elle était représentée avant l'arrivée des nouveaux processus.

- 1. Donner la table de bits correspondant à la répartition précédente, en supposant que l'unité d'allocation est le 50Ko.
- 2. Soit une représentation des parties libres/occupées sous la forme d'une liste chaînée, dont chaque nœud contient un bit d'état (L/O), la taille de la partition, et un pointeur sur la prochaine partition. Représenter la liste chaînée correspondant à la répartition précédente (unité d'allocation : 50Ko).
- 3. Nous désirons charger le processus P (400Ko) dans la mémoire selon la stratégie d'allocation First Fit.

- a. Représenter l'état de la mémoire suite au chargement de P.
- b. Décrire les étapes nécessaires pour localiser l'emplacement adéquat du nouveau processus dans la mémoire en utilisant :
 - i. La table de bits
 - ii. La liste chaînée
- c. Comment proposez-vous de modifier la structure de la liste chaînée de manière à rendre la mise à jour de la mémoire encore plus rapide ?

Exercice 3:

On considère un système disposant de **16 Ko** de mémoire centrale dont **4 Ko** sont occupés par la partie résidente du système d'exploitation.

Nous supposons que :

- La mémoire centrale utilise un partitionnement *variable*,
- La mémoire centrale utilise une stratégie de placement se basant sur l'algorithme premier ajustement ou **First Fit**,
- L'ordonnancement sur la *mémoire* se fait selon **FIFO**,
- L'ordonnancement sur la **CPU** se fait selon l'algorithme du tourniquet (**Round Robin**) avec un quantum $q = 3ms$,
- Chaque processus effectue un calcul suivi d'une **E/S** en utilisant son propre périphérique,
- Les processus arrivent conformément au tableau suivant :

Processus	Instant (t)	Taille (Ko)	Temps CPU (ms)	Durée E/S (ms)
A	0	3	9	2
B	4	5	6	9
C	6	5	4	4
D	8	4	2	6
E	10	1	4	3
F	12	1	5	1
G	16	1	3	2
H	18	3	3	8

Donnez le diagramme de GANT ainsi que les états d'occupation de la mémoire aux différentes étapes de traitement de ces processus.