

Fiche 9 : Questions/Exercices Gestion Mémoire

1. Expliquez la fonction principale d'un éditeur de liens dans la génération de l'espace adresse d'un processus. Pourquoi doit-on relocaliser les adresses dans le fichier exécutable qui est généré par l'éditeur de liens ? Quels éléments du code doivent être relocalisés ? Pourquoi doit-on aussi relocaliser des adresses lors de l'exécution d'un programme ?
2. Expliquez l'une des raisons pour laquelle un processus peut être déalloué de la mémoire principale avant la fin de son exécution.
3. Comparez la fragmentation interne à la fragmentation externe pour la mémoire, et expliquez laquelle pénalise le plus la performance du système.
4. Identifiez les zones de mémoire où sont sauvegardés les divers éléments d'un programme en état d'exécution.
5. Donnez un avantage pour l'utilisation de partitions fixes de mémoire, et un avantage pour l'utilisation de partitions variables de mémoire.
6. Laquelle des stratégies d'allocations de partitions (first fit, best fit, worst fit) minimise le plus la fragmentation ?
7. Quel avantage procure au système d'exploitation l'utilisation de registres de relocation ?
8. Lequel des algorithmes d'allocation de mémoire nous fournirait le meilleur degré de multiprogrammation ? Expliquer.

Ex1.

Considérez un système dont l'espace mémoire utilisateur compte 1Mo. On décide d'effectuer une **partition fixe** de cet espace mémoire en trois partitions de tailles respectives 600 Ko, 300 Ko, 100 Ko. On suppose que les processus arrivent dans le système comme montré ci dessous :

Instant t	Événement
t = 0	A(200 Ko, 35) arrive
t = 10	B(400 Ko, 65) arrive
t = 30	C(400 Ko, 35) arrive
t = 40	D(80 Ko, 25) arrive
t = 50	E(200 Ko, 55) arrive
t = 70	F(300 Ko, 15) arrive

Fig – Besoins de mémoire et temps d'arrivée.

A(200 Ko, 35) veut dire que le processus A nécessite une partition de 200 Ko et que son temps de séjour en mémoire centrale est 35 unités de temps. Bien entendu, un processus qui ne peut pas être chargé en mémoire est placé dans la file des processus en attente de chargement en mémoire. Un processus chargé en mémoire y séjournera jusqu'à la fin de son exécution.

Donnez les états successifs d'occupation de la mémoire si :

- L'ordonnanceur de haut niveau fonctionne selon le schéma plus court process d'abord (SJF) et le mode d'allocation des trous utilise un algorithme du type best fit.
- L'ordonnanceur fonctionne selon PAPS (FCFS) et le mode d'allocation des trous utilise un algorithme du type *First Fit*.

Ex2.

Considérez un gestionnaire de mémoire utilisant la stratégie de partitions variables. Supposez que la liste de blocs libres contient des blocs de taille de 125, 50, 250, 256, 160, 500, et 75 octets. Le bloc de 125 octets est en tête de liste et le bloc de 75 octets est en fin de liste.

- Lequel des blocs sera fourni par un allocateur BestFit pour un processus nécessitant une taille de mémoire de 84 octets.
- Lequel des blocs sera fourni par un allocateur WorstFit pour un processus nécessitant une taille de mémoire de 221 octets.
- Lequel des blocs sera fourni par un allocateur FirstFit pour un processus nécessitant une taille de mémoire de 178 octets.

Ex3 :

Pour chacune des stratégies d'allocation de blocs de mémoire libre (meilleur, plus grand, ou premier), expliquez de quelle façon la liste de blocs libres devrait être organisés afin d'obtenir une performance optimale.

Ex4.

Considérez un système n'utilisant pas de registres de relocation. Par conséquent, c'est le chargeur de programme qui s'occupe de la relocation des instructions dans l'image mémoire à partir de l'exécutable.

Serait-il possible de décharger et de recharger l'image mémoire du programme durant le cycle d'exécution? Expliquez comment cela pourrait être réalisable ou pourquoi cela ne pourrait pas être réalisable ?