TD n°1 : Gestion de la mémoire centrale

Exercice 1. On considère la suite de demandes d'allocation (+) et de libération (-) suivantes, dans un espace mémoire de 1000 blocs, utilisant l'allocation contiguë :

+300, +200, +260, -200, +100, -300, +90, +400, -260, +150, +120, +500, -100, -120, +200, -150, -90, +100, -400, -100, -200

Indiquer comment, à partir d'une mémoire initialement libre, le système d'exploitation réalise l'allocation avec les stratégies

1. du meilleur ajustement, avec une gestion de la liste chaînée des zones libres triée par taille croissante,
2. de la première zone libre avec une gestion de la liste chaînée par adresse croissante et
3. du plus grand résidu avec une gestion de la liste chaînée des zones libres triée par taille de zone décroissante.

Montrer l'évolution de ces listes au cours des demandes considérées.

Exercice 2. Un buffet est organisé sur une table comprenant 23 places contiguës. Les invités arrivent soit seuls, soit par couple, mangent puis repartent. La maîtresse de maison veut imposer une stratégie, fixant une limite au nombre de personnes simultanément attablées, de telle sorte que tout couple qui arrive et qui peut s'installer sans que la limite ne soit dépassée, trouve deux places libres contiguës. La limite est fixée à 16 et en même temps, les places 2, 5, 8, 11, 14, 17, 20 sont interdites aux personnes seules.

Montrer que cette stratégie donne le résultat escompté.

Exercice 3. Un mini-ordinateur a une gestion de la mémoire fondée sur l'allocation par subdivisions binaires. Il dispose initialement d'un seul bloc de 256 Ko situé à l'adresse 0. Les demandes d'allocation à satisfaire correspondent à une suite de blocs de 5 K, 25 K, 35 K puis 20 K.

- Combien de blocs libres reste-t-il et quelles sont leurs tailles et leurs adresses ? Faire un schéma des différentes allocations.
- Même question avec une gestion de la mémoire se basant sur la suite de Fibonacci.

Exercice 4. Soit un système à partitions variables de mémoire avec allocation contiguë (mais blocs de partitions non contigus).

À un instant donné, les partitions libres sont 100K, 500K, 200K, 300K et 600K (par ordre croissant des adresses). On considère une liste d'arrivée des processus qui demandent 212K, 417K, 112K et 426K.

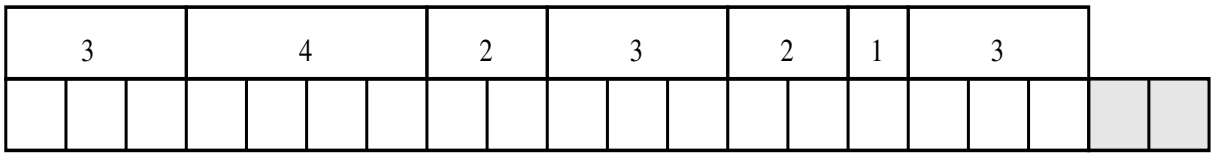
- Si aucun espace mémoire n'est suffisant pour contenir le bloc à allouer, la mémoire est compactée.
- Si après compactage, l'allocation n'est toujours pas possible, alors l'allocation est refusée.

Donner le comportement des algorithmes selon les stratégies de la première zone libre (First Fit), du meilleur ajustement (Best Fit) et du plus grand résidu (Worst Fit).

Quel est le meilleur algorithme dans ce cas ? Justifier.

Exercice 5. La figure suivante représente un espace mémoire de 20 blocs après les allocations successives à partir de l'état initial, de zones de tailles respectives comptées en nombre de blocs

3, 4, 2, 3, 2, 1 et 3.



1. Donner le nombre de blocs libres après la libération de la $2^{\text{ème}}$, la $4^{\text{ème}}$ et la $5^{\text{ème}}$ zone, avant et après compactage de la mémoire.
2. Après la libération de ces blocs et le compactage de la mémoire, et pour satisfaire une demande de 1 bloc, préciser les zones libres choisies dans le cas où on applique :
   a) la stratégie du First-Fit,
   b) la stratégie du Best-Fit,
   c) la stratégie du Worst-Fit.

Exercice 6. Écrire les algorithmes d'allocation et de libération de mémoire avec une gestion des zones libres implantée par blocs chaînés, en mettant en œuvre les trois politiques : première zone libre, meilleur ajustement et plus grand résidu. L'allocation est refusée si aucun espace mémoire n'est suffisant pour le contenir (on ne fait pas de compactage).

On demande de préciser les structures de données nécessaires pour implanter ces algorithmes.

Exercice 7. Écrire l'algorithme d'allocation de mémoire pour le buddy system binaire. On satisfait une demande par allocation du bloc libre de taille minimale supérieure à celle de la demande.