

## Projet programmation système et réseau

### Gestion de la mémoire

- 
- *Le projet doit être réalisé en binôme*
  - *Le projet doit être réalisé en C*
  - *Le rapport à rendre doit décrire*
    - *l'organisation générale de l'application et son découpage éventuel en fichiers*
    - *le choix des structures de données utilisées*
    - *les algorithmes utilisés*
    - *le code source des programmes*
    - *un ensemble d'exemples*
- 

#### **1 – Description du projet**

Le but de ce mini-projet est de simuler la fonction de la gestion de la mémoire d'un système d'exploitation. Il s'agit de visualiser à l'écran l'évolution des allocations/désallocations de la mémoire selon trois stratégies et de les comparer.

Cette fonction a pour rôle de partager la mémoire entre les différents processus existants. Elle se décompose en deux parties : l'allocation d'une zone mémoire à un processus qui en fait la demande et la récupération d'une zone mémoire libérée par un processus.

Les zones mémoires libres sont décrites chacune par un bloc comprenant l'adresse de début et la longueur de la zone. Les blocs sont chaînés entre eux pour former une liste classée par adresses croissantes.

On considère trois algorithmes d'allocation : *FIRST-FIT*, *BEST-FIT* et *WORST-FIT*.

On donne ici une description de l'algorithme *FIRST-FIT* : pour une demande d'allocation mémoire de taille  $N$ , on recherche la première zone libre de la liste de taille supérieure ou égale à  $N$ . S'il n'en existe pas, la demande est rejetée ; s'il existe une zone libre de taille  $N$ , cette zone est ôtée de la liste des zones libres et son adresse est retournée ; s'il existe une zone libre de taille  $M$  strictement supérieure à  $N$ , on la scinde en deux zones, l'une de taille  $N$ , l'autre de taille  $M-N$ .

L'algorithme de libération consiste à remettre à sa place dans la liste des zones libres un bloc décrivant la zone libérée et de le fusionner avec ses voisins dans le cas où les zones mémoires seraient jointives.

## 2 – Réalisation

**a** - Développer un programme réalisant cette simulation.

Le programme principal présentera à l'utilisateur un menu proposant les choix suivants :

- 1 – Allocation d'une zone mémoire
- 2 – Libération d'une zone mémoire
- 3 – Visualisation de l'occupation de la mémoire

Choix 1 : l'utilisateur devra, à la demande du programme, entrer au clavier la taille de la zone mémoire demandée. En retour, on affichera à l'écran soit un message d'échec si la demande n'a pu être satisfaite, soit un message de succès et l'adresse de début de la zone allouée.

Choix 2 : l'utilisateur devra entrer au clavier l'adresse et la taille de la zone libérée. Les demandes incohérentes devront être rejetées et signalées par un message d'erreur.

Choix 3 : On décrira à l'écran, sous une forme libre, la taille et l'adresse des différentes zones ainsi que leur état (libre ou allouée).

La mémoire sera considérée entièrement libre au départ.

**b** - L'utilisation répétée des fonctions d'allocation *FIRST-FIT* et de libération de zones mémoire entraîne un morcellement de la mémoire libre en zones de petites tailles, interdisant toute demande ultérieure d'allocation mémoire de grande taille.

Réaliser une nouvelle fonction destinée à réorganiser la mémoire de façon à regrouper toutes les petites zones de mémoire libre.