

Rapport TP Allocateur mémoire

Théo Teyssier, Hugo Prat-Capilla

Objectifs :

Notre objectif est d'implémenter un allocateur mémoire simple utilisant un table en tant que mémoire.

Notre implémentation est basée sur 2 structures relativement similaires :

- **Structure fb** comprenant un `size_t` `size` représentant la taille d'une zone mémoire libre + la taille de la structure et un pointeur vers la prochaine zone mémoire libre.
- **Structure variableGlobal** comprenant un `size_t` `size` représentant la taille de la mémoire total + la taille de la structure et un pointeur sur la première zone libre dans la mémoire.

Fonctions implémentées :

Mem_Init :

Réserve la mémoire globale

Créer une structure `variableGlobal` au tout début de la mémoire et lui assigne les valeurs correspondantes.

Le pointeur de `variableGlobal` `start` pointe sur une structure `fb` correspondant à toute la mémoire restante.

Initialise la fonction de recherche de zone libre ici `mem_fit_worst`.

Mem_Alloc :

Cas général :

On utilise la fonction de recherche pour trouver une zone mémoire correspondant à la taille demandé. Ensuite on assigne la place nécessaire à la fin de la zone libre renvoyé par la fonction de recherche et on change la taille de cette zone libre. Quand à la zone occupé créé place un `size_t` représentant la taille de cette zone au début de celle-ci.

Si aucune zone ne correspond la fonction renvoie `null`.

Cas particuliers :

- Si la place demandé est supérieure à la place disponible mais inférieure à la place disponible + la taille de la structure : on alloue toute la zone mémoire en supprimant la structure `fb` la régissant.
- Si la place demandé n'est pas un multiple de 8 on assigne un peu plus de mémoire pour revenir à un multiple de 8.

Mem_Show :

Pour parcourir la mémoire on utilise 2 pointeur l'un que l'on vas déplacer sur toutes les zones mémoires et l'autre qui se déplace seulement sur les zones libres. S'il sont égaux

c'est que nous sommes dans une zone libre sinon le pointeur qui se déplace dans la mémoire a atteint une zone occupée. On affiche en fonction de cette comparaison.

Mem_Free :

Cas général :

On récupère la zone occupée et on en fait une zone mémoire libre en mettant une structure fb au début. On parcourt toutes les zones libres pour trouver la zone libre la précédant la plus proche puis l'ajoute à la chaîne des zones libres.

Cas particuliers :

- Si la zone libre précédente est contiguë à la zone libre que l'on vient de créer on fusionne les deux.
- Si la zone libre suivante est contiguë à la zone libre que l'on vient de créer on fusionne aussi les deux.

En conclusion, nous avons beaucoup appris sur la manipulation de pointeur et sur l'organisation et la manipulation d'une mémoire.