BALADI AYOUB

COMPTE RENDU: TP MEMOIRE

1- Résumé

Les fonctions {mem_init, mem_alloc, mem_free, mem_get_size, mem_show, mem_first_fit, mem_best_fit, mem_worst_fit} sont implémentées, fonctionnelles et testés (tests fournis, tests complémentaires).

2- Principes d'implémentation

Pour gérer l'espace mémoire j'ai utilisé 2 structures de données (simple linked list), une pour les zones occupée et la 2ème pour les zones libre. Ce choix a été pour valoriser la performance, faire moins de calculs et faciliter la gestion alors j'ai sacrifié un peu d'espace mémoire pour cela.

Cas limits mem_alloc:

- l'allocateur autorise l'allocation de 0. Il crée un noeud avec 0 affecté à l'attribut size de ce noeud et renvoit une adresse correcte.

- Quand l'allocateur alloue un nouveau bloc, s'il reste un epsace < taille da la structure freeBlock (pour la marquer comme libre) entre le bloc alloué et celui de dvant ou la fin de la mémoire. Le reste est rajouté au nouveau bloc alloué.

- Si la taille souhaité est négative ou très grande, l'allocateur renvoit null.

Cas limits mem_free:

- Si la zone en paramètre n'existe pas dans la liste chainée des bloc allouée l'allocateur notifie le système que la zone n'est pas valide.

- Après chaque libération l'allocateur vérifie s'il y a une fusion à faire, soit à gauche ou à droite du bloc récement libéré.

3- Structure de code

- La structure de donnée utilisée est implémenté dans le fichier mem_os.h.

- La spécification/signature des fonctions auxilliaires sont dans le fichier mem_os.h. Ces fonction sont implémenter dans l'ordre avant tout dans mem.c.

4- Test effectués

Tests fournis on tous passé plus des tests complémentaire pour tester le fonctionnement des fonctions implémentés (allocations des 0, allocation des tailles négatives des tailles très grandes, free avant alloc)

5- Compliation et exécution

Pour exécuter le programme:

1. Se placer dans le dossier tp-mem-alloc/src
2. Tapez la commande make pour génerer l'exécutable
3. ./mem_shell pour démarrer le programme.