

COMPTE RENDU CPS

Manipulation de la mémoire :

mem_init(void* mem, size_t taille) :

On appelle cette fonction à chaque fois que l'on veut initialiser une nouvelle zone mémoire, en rentrant comme argument une zone libre ainsi qu'une la taille maximale de la zone désirée.

***mem_alloc(size_t taille) :**

Cette fonction commence par prendre en paramètre la taille de l'allocation désirée, puis elle va aller chercher la première zone libre adéquate à l'allocation de la nouvelle zone mémoire. Si la zone libre choisie est assez grande, il y aura création d'une zone occupée à l'intérieur de cette dite zone et l'on gardera une partie de la zone libre, sinon toute la zone libre sera occupée.

mem_free(void* mem) :

Cette fonction va libérer une zone mémoire qui a été occupée au préalable. On passe en argument l'adresse d'une zone occupée, si cette zone occupée existe vraiment à cette adresse, la zone occupée est supprimée et la nouvelle zone libre est ajoutée à la zone libre la plus proche si il n'y a pas de zone occupée entre les deux.

mem_show(void (*print)(void *zone, size_t size, int free)):

La fonction mem_show sert à afficher toutes les zones de la mémoire, de ce fait les zones sont affichées selon leurs ordre d'apparition dans la zone mémoire.

mem_get_size(void *zone) :

Cette fonction doit normalement retourner la taille totale de la zone dont l'adresse est passée en paramètre. Malheureusement nous n'avons pas réussi à l'implémenter.

mem_fit_first(struct fb *list, size_t size) :

Cette fonction retourne la première zone libre de notre zone mémoire totale.

mem_fit_worst(struct fb *list, size_t size) (codé et fonctionnel):

Cette fonction nous sert à renvoyer la zone la plus grande qui a une taille adaptée à l'allocation que nous souhaitons faire.

mem_fit_best(struct fb *list, size_t size) (codé et fonctionnel):

Comme mem_fit_worst cette fonction nous renvoie la plus petite zone adaptée à l'allocation de mémoire que nous souhaitons faire.

Tests:

Tous nos tests sont disponibles en tapant 't' lors de l'exécution de memshell.

Nos vérifications sont donc :

- Vérifier que la mémoire est bien vide lors de l'initialisation.
- Vérifier si une allocation simple marche bien
- Voir comment notre code réagit si on décide d'allouer un espace mémoire sur un autre déjà alloué au préalable.
- Voir comment notre programme réagit lorsque l'on décide d'allouer une zone de taille supérieure à la zone totale initialement prise en compte.
- Voir comment notre programme réagit si l'on alloue une petite zone mémoire sur une mémoire totale déjà saturée.
- Voir si notre programme libère bien toutes les zones qui ont été allouées.