

Vallon Arnaud
Isnel Maxime

Compte Rendu: allocateur mémoire

Principe et choix:

Nous avons décidé de stocker toute nos zones libres dans une liste chaînée dont la tête est stockée en global, chaque zones libres contiennent une structure fb qui contient l'adresse de la prochaine zone libre (NULL s'il n'y en a pas) et la taille de la zone libre.

Pour allouer, nous réduisons la taille de la zone libre (après avoir regardé si elle est suffisamment grande) et nous plaçons la taille (en size_t) de la zone qui vient d'être allouée juste avant cette zone ci.

Pour libérer, nous regardons juste avant le pointeur passé en argument, puis on va chercher entre quelles zones libres cette zone occupée se situe, quand on a la zone avant et la zone après (peut être NULL) on insère au milieu et s'il faut nous faisons une fusion de la nouvelle zone libre avec la zone suivante et/ou la zone précédentes si elles sont adjacente.

Test:

- test_basic: pour tester la fusion suivante puis à la fin la double fusion pour retomber sur la taille d'espace mémoire initial
- test_basic2: test de la fusion précédente
- test_basic3: test de la fusion suivante
- test_basic4: test de la fusion suivante jusqu'à libération totale de la mémoire allouée
- test_init: test donné par l'énoncé