TP : Allocateur mémoire

El Yandouzi Elias[11700363] Salmon Amad[11912852]

Polytech Grenoble - INFO3 - API

Objectif Nous allons dans ce TP chercher à réaliser un allocateur mémoire. Il faudra faire attention à l'arithmétique des pointeurs

1 Choix d'implémentation

1.1 Représentation de la mémoire

Pour représenter la mémoire en architecture de blocs, nous avons eu recours à 3 structures différentes. En fait, chaque bloc de mémoire contient lui même un bloc de méta-données contenant un ou deux champs d'informations.

Une première structure first_bloc qui représente un bloc . Une seconde structure free_bloc qui représente un bloc libre et contenant 2 informations : la taille de ce bloc libre et un pointeur vers le bloc libre suivant. Une dernière structure used_bloc qui représente un bloc occupé et ne contenant qu'une seule information : la taille de ce bloc occupé.

Les trois tailles sont contenues en size_t.

1.2 Allocation de mémoire

Pour l'allocation de mémoire, le seul paramètre demandé à l'utilisateur est la taille de la mémoire à allouée. Une fois cette variable recupérée, le programme cherche une zone libre assez grande pour contenir un bloc de la taille demandée. Un bloc libre est défini comme convenant selon mem_fit. Si aucun bloc libre n'est assez grand pour y mettre un bloc de la taille demandée, le programme retourne null. Si un espace trop grand est trouvé, la

1.3 Libération de mémoire

Le processus de libération de mémoire demande à l'utilisateur l'adresse de la zone mémoire à libérer. Le programme parcourt ensuite la mémoire de bloc libre en bloc libre jusqu'à arriver au bloc libre dont l'adresse et la taille additionnées donnent l'adresse donnée par l'utilisateur. Avant d'effacer les informations contenues dans ce bloc occupé, il faut regarder les blocs situés avant et après pour voir s'ils sont libres ou non afin de voir si une ou plusieurs fusions sont nécessairees. Il faut ensuite selon les cas réajuster les métadonnées du bloc libre précédent le bloc courant afin que le champs taille soit agrandi de la taille du bloc courant et que le pointage vers le bloc libre suivant soit actualisé.

1.4 Affichage de mémoire