

LSINF1252

Rapport de projet 1 :

Implémentation de malloc, calloc et free

Aigret Julien Vanvyve Nicolas

16.03.2016

1 Gestion de la mémoire

1.1 Initialisation

Nous avons fait le choix d'initialiser notre heap à 1MB, taille qui semble raisonnable pour notre utilisation actuelle, mais qui eut être modifiée si besoin est. Cette limite n'est pas modifiée tout au long de l'exécution de notre code.

1.2 Fragmentation

Dans notre programme, nous avons fait le choix d'allouer immédiatement une zone de mémoire (ou une combinaison de zones de mémoire) si elle est de la taille demandée par l'utilisateur. Dans le cas où il n'y a pas de zone de mémoire de taille précise, deux cas de figure se présentent : soit une ou des zones plus grandes ont été trouvées, ou aucune de taille suffisante n'a été trouvée. Dans le cas où une plus grande zone a été trouvée, nous gardons la plus grande de toutes et y installons notre zone de mémoire, créant un nouveau blockheader à la fin de celle-ci pointant sur la fin de la "zone libre".

2 Tests comparatifs

Nous constatons, comme illustré sur les graphes (fig!! Ref needed) que nos fonctions sont plus rapides pour les faibles allocations de mémoires, mais plus la mémoire devient grande et plus elles prennent du temps. Cela est certainement dû aux algorithmes de recherches utilisés par les fonctions built-in qui doivent être optimisés pour les grandes zones de mémoire. Notre initialisation de la zone mémoire par calloc est aussi très rudimentaire, et prend de plus en plus de temps au fur et à mesure que la taille de la zone à initialiser augmente.