

LSINF1252  
Rapport de projet 1 :  
Implémentation de malloc, calloc et free

Aigret Julien (8343-13-00)      Vanvyve Nicolas (6590-13-00)

16 mars 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.

### 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. Dans le cas où il n'y a pas

## 2 Tests unitaires

Nous avons implémenté les tests unitaires décrits ci dessous via CUnit.

- Myfree desalloc : nous vérifions que `alloc` de `block_header` vaut bien 0.
- Mymalloc alloc : idem que ci dessus mais `alloc` de `block_header` doit valoir 1
- Mycalloc alloc : idem que Mymalloc alloc
- Mymalloc size : nous vérifions que l'espace alloué correspond bien au multiple de 4 supérieur ou égal(en byte) de celui demandé.
- Mymalloc NULL : nous vérifions que si 0 byte sont demandé `mymalloc` renvoie NULL
- Mymalloc 2 alloc : nous vérifions que deux allocation successive via `mymalloc` ne pointe pas vers la meme adresse
- Mycalloc initialisation : nous vérifions que après un appel à `mycalloc` la mémoire allouée est bien initialisée à 0

Les test qui s'applique à `mymalloc` aurait pu etre transformés pour `mycalloc`. Mais étant donné que nous avons utilisé `mymalloc` pour implémenter le second cela ne servirait à rien.

### 3 Tests comparatifs