



TME : Garbage collector — semaine 4

1 mars 2018

Objectif(s)

- ★ Utilisation du garbage collector de Boehm.
- ★ Implémentation d'un garbage collector "stop-and-copy".

Exercice 1 – le GC de Boehm

1. Récupérer l'archive `td4.zip` sur le site du cours, elle contient les fichiers suivants :

```
alloc_man.c  crible_gc.c  crible_mm.c  pile.c
crible.c     crible_gen.c  gc_alloc.c  stat.c
```

ainsi que le GC de Boehm (une implémentation de GC en C, aussi disponible à l'adresse <http://www.hboehm.info/gc/>).

2. Écrivez un petit programme qui alloue de la mémoire avec `malloc` et examinez le avec `strace`. Quels sont les appels systèmes effectués par ce programme. Qu'en déduisez vous quant au `malloc` de la `glibc` ?
3. Faites de même en écrivant un programme simple allouant de la mémoire en utilisant le GC de Boehm.
4. Modifiez votre crible d'eratosthene pour utiliser le GC de Boehm.

Exercice 2 – Récupération automatique de `liste_entier`

1. Modifiez les fonctions d'allocation et de récupération pour implanter un GC `Stop & Copy` pour les listes d'entiers. Pour la copie, on inclura dans le champ `cdr` la nouvelle adresse de la cellule déplacée.
2. Que se passe t'il si un entier correspond à une adresse valide ? Comment peut on différencier un pointeur d'un entier ?