Recolección de Basura en C

Edinson Sánchez
Kevin Filella
Adrian Aguilar

2 de febrero de 2014

Índice

1. Introducción

En lenguajes de programación, un recolector de basura tiene como objetivo el de gestionar la memoria de un programa informático. La memoria debe ser gestionada de tal forma que se puedan reservar espacios en memoria para su uso, se puedan liberar espacios en memoria anteriormente reservados, se puedan compactar espacios de memoria libres y consecutivos, y se pueda llevar cuenta de los espacios libres y utilizados en la memoria.

Como es de conocimiento para muchos, el lenguaje C no incorpora un método de gestión de memoria automático, como lo hacen lenguajes de programación como Java, C Sharp, entre otros. Esto brinda a los programadores un set de ventajas y desventajas a la hora de programar.

En este documento, discutiremos sobre varios métodos populares que han sido creados, en forma de librerías, para brindarle al lenguaje C esta muy importante característica. Asimismo, discutiremos sobe las ventajas y desventajas de tener un recolector de basura automático y gestionar la memoria manualmente.

2. Métodos

2.1. Boehm-Demers-Weiser Garbage Collector

El método de recolección de basura de la librería Boehm GC consiste en dos fases; primero, se realiza un escaneo de toda la memoria activa para marcar bloques en desuso; después, la fase de barrido se hace cargo de mover todos los bloques marcados (de la primera fase) a la lista de bloques libres. Estas dos fases pueden ser, y usualmente son, ejecutadas aparte para mejorar el tiempo de respuesta de la librería.

El algoritmo del Boehm GC es generacional, ya que se enfoca en buscar memoria libre en los bloques más nuevos. Esto se basa en la idea de que los bloques más viejos tienen mayor tiempo de uso; por consecuencia, los bloques más recientes tienen un tiempo de uso reducido y deben ser liberados con mayor frecuencia. Este algoritmo también es conservativo, ya que también se debe encargar de asumir cuales variables son punteros a datos dinámicos y cuales simplemente se ven de esa forma.

La librería Boehm GC viene en forma estática o dinámica, es fácil de instalar, y su uso es permitido en los lenguajes C y C++. Para utilizar la librería en programas en C que ya utilizan malloc(), calloc(), realloc(), o free(), simplemente se deben redefinir las funciones como se ve a continuación:

```
#define calloc(n,x) GC_malloc((n)*(x))
#define realloc(p,x) GC_realloc((p),(x))
#define free(x) (x) = NULL
```

2.2. Metodo 2

explicar metodo 2 aqui

2.3. Metodo 3

explicar metodo 3 aqui

3. Ventajas

3.1. Garbage Collector

ventajas de GC aqui

3.2. Gestión manual

ventajas de gestion manual aqui

4. Desventajas

4.1. Garbage Collector

desventajas de GC aqui

4.2. Gestión manual

desventajas de gestion manual aqui

5. Conclusiones

conclusiones aqui