

Java Programming assignment #2: Resource Map and Memory Allocator

due by April 6th, 2019

Introduction

Memory is precious hardware resource in computing system, so memory usages are carefully observed and managed by special software, memory manager. A user requests specific amount of memory (size) to the memory manager. (a.k.a malloc() call) Then, the manager finds the empty memory region (free space), and returns the starting address to the user. The memory manager keeps track of free memory and used memory region, so that the used memory region cannot overlap with the currently used region. The memory manager gives the free memory and marks it with the owner so that it cannot be used by another user or program, which is called as memory allocation.

There are several memory allocation schemes, and one of simple schemes is resource map. The resource map consists of multiple <starting address, end address> pairs. When a user requests memory, the resource map finds either first, or smallest memory region that is at least larger than the requested size. Initially, the entire memory region is free, so that the free map is the same as <starting address, entire memory size>. If we assume 1MB memory, beginning from the address 0, then the initial map would be <0, 0x100000>.

Upon a request to allocate memory region for 4KB, the memory manager would allocate memory region 0~0xFFF (4KB). Then, the free map is changed to <0x1000, 0x100000>. Next 4KB allocation makes the free map as <0x2000, 0x100000>.

A user can reclaim the memory after use. The free memory region can be divided into multiple regions, as the allocation status. For example, if the first user who uses memory region (0x0000~0xFFF) frees the memory, then the free map would be <0x0, 0xFFFF>, <0x2000, 0x100000>. Note that there are two regions in the free map.

Additionally, free map can be merged according to the allocation status. For example, if the second user frees the memory region (0x1000~0x1FFF), then the free map would be <0x0, 0x100000>.

The homework is to make a Java application that mimics memory manager using resource map.

To begin with, you can get input from the standard input or input file. The first line contains the total number of requests/reclaims. (number of lines)

Each line consists of two strings. The first string could be '+' or '-'. '+' stands for allocation, and '-' stands for free request. the second string is either 1) the requested memory size for the allocation request or 2) the freeing entry in the allocation line (allocation line number). Sample input could be as follows:

```
4
+ 0x1000
+ 0x1000
- 1
- 2
```

You can think of using List, Array, or any kinds of data structure that you've learned.

Happy hacking with Java Programming!

Seehwan Yoo