

Programming Assignment 3

Assigned 11/09 DUE DATE 11/25 Midnight---Start Immediately [There is opportunity for Extra Credit]

Programming Assignment 3: Program For Memory Management-Processes and Holes

In this program you will be modeling a Memory Manager in it's task of assigning Processes to free spaces in the RAM and reclaiming space from programs that have completed.

The layout of the memory is to be maintained in a doubly linked list which will represent the RAM. (Refer to chapter 4 of the text and page 11 of Notes 4a.) Each node (or cell) in the list records information about a Process or a Hole. It will contain the following information,

a). Representation Of Memory Layout

For either a Hole or Process Node there are 3 fields in each cell of the linked list

ID, size, and Base Address

ID-For a Process the *ID* is its unique identifier. It is an integer greater than or equal to 0.

For a Hole the *ID* is -1

For either a Process or Hole:

size is the amount of RAM that a Process or Hole uses. It is an integer greater than 1.

Base Address is the location in the RAM where a process or hole starts. It is an integer greater than or equal to 0.

In addition, while any number of Processes can be next to each other in the list, there can only be one Hole in between two Processes. What this means is that if you had the list:

Process#1<-->Hole<-->Process#2<-->Process#3

and were told to delete process #2, the list should look like this after the deletion:

Process#1<-->Hole<-->Process#3

so holes must be resized to maintain this requirement.

b). Description Of Inputs Of The Memory Management Program

Input for the program will come in the form of a data file which will provide the model for the starting map of the RAM, as well as commands to, **load, insert, remove** and **print** out the contents of the list. The general format for the data file is as follows:

The input will consist of :

Initial Layout or L statements which describe the names (*IDs*) and sizes of processes and holes to be added to the RAM given in increasing address order. These have the format

L 1 ID size (ex. L 1 25)

L -1 ID size (ex. L -1 16)

The first L command should be a hole. **No two successive L commands represent Holes.**

After the initializing statements which describe the initial configuration of Holes and Processes there is to be a line to indicate the end of the initial layout.

B

Then the remaining lines will consist of two kinds of statements: statements for **inserting** and statements **removing** Processes:

Statements which request the **insertion** of a Process (identified by its *ID*) These have the format:10

I ID (ex. I 10 4)

The process is placed in the position in which it fits according to the algorithm used.

-----CONTINUED----->

Statements which request the **removal** of a process (identified by its *ID* these have the format *It leaves a hole then of the same size of the deleted process (which must be consolidated with any surrounding holes).* It has the format command option,

R ID (ex. R 17)

Statement which cause a **printout**, of the **entire list** with format

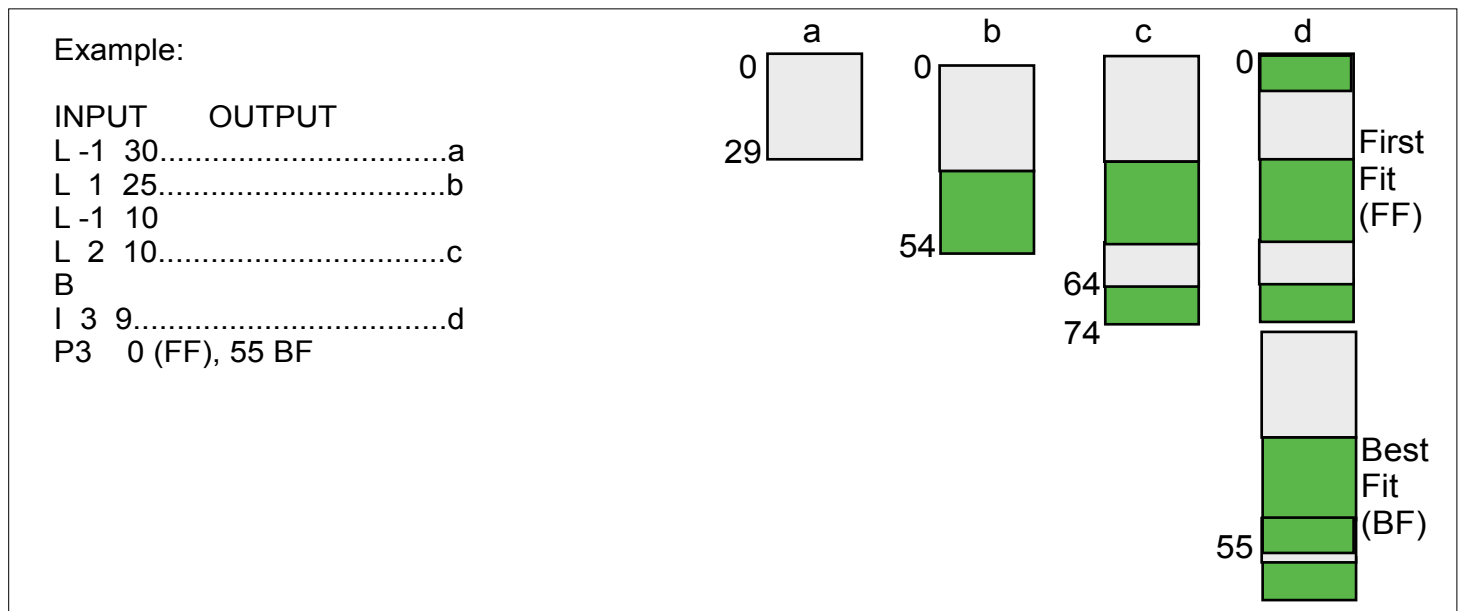
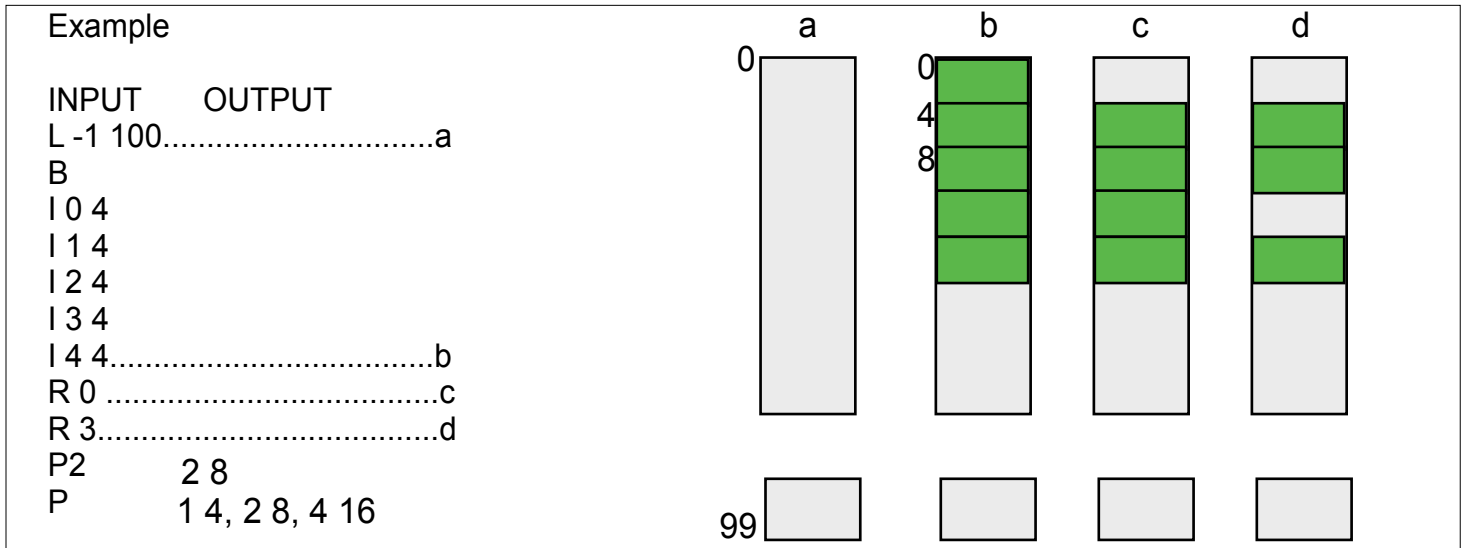
P

The print out gives the *ID* and location of each process, *ID loc*, (ex: 2 8, 4 20...) or for the **printout** of a single node with format

P ID (ex. P 4).

The print out gives the *ID* and **location** of the process, *ID location* (ex: 2 8)

Sample data file are shown below with the corresponding MM layout at various stages.They can be used for testing (however you are encouraged to create your own datafile to fully test your code.



The Assignment Algorithm

- You may use 1) First Fit only or
- 2) Best Fit only
- 3) or both.

Extra Credit for 2 and 3.

(Once they're built you can experiment with how well they do for different inputs.)