The project is due to December 14 (11:59 pm). The program should be submitted ONLY via BlackBoard. No late submissions. The project is meant for individual work only.The program must compile and run on Linux Lab computers. If you submit more than 3 files please pack them into archive.

If your program requires compilation with some specific command please create a make-file or README.txt file with this specific compilation command in it. There is no need to send me the already-compiled executable files.

The project can be done in one of the following programming languages: C++ , Python , Java . Your main source-file should start with a comment with your name in it.

You should write a program that simulates some aspects of operating systems. There is no real system programming involved. The whole simulation is based on the text inputs that the program receives from user.

Scheduling: your program should implement preemptive priority CPU-scheduling. Higher numbers mean higher priority.

All I/O-queues are FCFS.

Memory: your program should simulate contiguous memory management with “best fit” approach.

Each process in your program should be represented by its own PCB (obviously!). We do not want to move around the whole PCBs. So we store all PCBs in a process table and we move around only the indexes of processes in that table. Most likely your process table is just a vector of PCBs but in general it’s up to you how to implement it efficiently.

At the start, your program asks the user three questions:

How much RAM memory is there on the simulated computer? Your program receives the number in bytes (no kilobytes or words). I can enter any number up to 4000000000 (4 billions). Make sure you use datatypes that will not be overflown by such value.

How many hard disks does the simulated computer have? The enumeration of the hard disks starts with 0.

After these questions are answered, the simulation begins. You program constantly listens for the user inputs. You should NOT ask for a confirmation to enter another input. The user inputs signal some system events. Your program simulates the corresponding system behavior. The possible inputs are:

A priority memory\_size ‘A’ input means that a new process has been created. This process has a priority priority and requires memory\_size bytes of memory.

When choosing a PID for a new process start from 1 and go up. Do NOT reuse PIDs of the terminated processes.

t The process that is currently using the CPU terminates. It leaves the system immediately. Make sure you release the memory used by this process in your simulation. Pay attention, once the process is gone from the system it is also gone from the page table. It can cause indexes of some processes in page table to change. You must apply this change of indexes to all affected processes in the CPU, ready-queue and I/O-queues.

d number file\_name The process that currently uses the CPU requests the hard disk #number. It wants to read or write the file file\_name.

D number The hard disk #number has finished the work for one process.

S r Shows a process currently using the CPU and processes waiting in the ready-queue. For each process I need to see its PID and priority. Make sure that the process using the CPU is obvious (mark it somehow).

S i Shows what processes are currently using the hard disks and what processes are waiting to use them. For each busy hard disk show the process that uses it and show its I/O-queue. Make sure to display the filenames (from the d command) for each process. The enumeration of hard disks starts from 0. Processes in I/O-queue should be displayed in the correct queue order. Remember that our I/O-queues are FCFS.

S m Shows the state of memory. Show the range of memory addresses used by each process in the system.

Good Luck!