King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC227 Project

Instructions:

Introduction:

It is required in this assignment, to write a program in Java that simulates the behavior of the multiprogramming operating system. At the end of the simulation, you are expected to output some statistics regarding the behavior of the system. In the following sections, we will introduce the hardware specification, the multiprogramming OS features and the jobs requirements.

Hardware:

The computer hardware is assumed to have:
1. A hard disk of size of 10 GB where 1/5 of this size is used to store the user programs.
2. A RAM of size 192MB, where 32MB is used to store the OS.
3. A CPU that executes one instruction each unit of time.
4. An IO device for input and output operations.
5. An internal clock that ticks every unit of time.

Operating System:

The operating system is the multiprogramming OS. We would be interested in only 2 features in this simulation: The Job and CPU scheduling.
1. Job Scheduling: The program with the smallest size is first selected to be loaded in the main memory. We call this technique by SSPF.
2. CPU scheduling: The CPU is allocated to the program with the smallest expected running time. We call this technique by SETF.

Program specifications:

Each program has 2 main requirements: A **program size** in KB and an expected execution time. Of course, the **expected execution time** is greater or equal to the exact execution time. In addition, each program should have an **Id** and **state**. The student should specify any extra information that is required by the simulation.
1. The memory sizes are uniformly distributed between 16KB and 16384KB
2. The expected execution time is also distributed between 16ut and 512ut.

Initialization phase:

You should perform the following steps before running the simulation:
1. Generate enough programs with random memory size and random expected execution time so to fill the hard disk. (Σ Program sizes $\leq$ 2 GB)

2. Load the RAM with the maximum number of user programs.
3. Start the simulation which consists of a simulation of the Machine Execution Cycle.

The Machine Execution Cycle:
The following algorithm simulates the machine Execution Cycle:

  MEC algorithm:
  While true do {
    Increments the simulated clock by one unit of time
    (* This assumes that one instruction is executed *)
    If there are interrupts
      Then Interrupts the current program and calls the ISRi
    endif
  }

Interrupts are also randomly generated:
1. The possibility that there are interrupts is 10%
2. The possibility that there is an IO request is 20%
3. The possibility that the busy IO device will terminate is 20%
4. The possibility that the program terminates normally is 5%
5. The possibility that the program terminates abnormally is 1%

The main simulator program is like this:

    Initialize the simulation
    While there are jobs in the H-Disk do {
      Run the Machine Execution Cycle
    }
    Print the required statistics

Output from the simulation:
At the end of the simulation, you should print the following results:
1. The number of initially generated jobs stored on the H-disk.
2. The average program size of all jobs.
3. The average number of jobs that have completed their execution normally.
4. The average number of jobs that have completed their execution abnormally.
5. The number of CPU bound jobs.