Simulation of a Scheduler

In this assignment, we will simulate a scheduler.

1. **Input files:**

   You MUST use exactly the formats described here, because we will be using an automated script for evaluation.

   Two files named PROCESS_SPEC and SCHEDULER_SPEC will be used.

   PROCESS_SPEC would contain specification of all processes that are to be run under the scheduler. The specification of each process is as follows:

   PROCESS
   (Process id,starting priority,admission)
   (#iterations,CPU burst,duration of I/O)
   (#iterations,CPU burst,duration of I/O)
   ...
   (#iterations,CPU burst,duration of I/O)
   End ...

   Where process id and starting priority are both integers, with a larger number implying a higher priority, and all times are in milli-seconds. A process contains several phases in its lifetime. Each phase is described by a triple giving the number of iterations and details of the CPU burst and duration of I/O operation in each iteration. A process may contain any number of phases, and any number of processes can be admitted in the system.

   SCHEDULER_SPEC would contain

   SCHEDULER
   # of scheduling levels
   (scheduling level,priority,time slice)
   ...
   (scheduling level,priority,time slice)
   END

   where scheduling levels are integers starting with 1, priority is an integer, and time slice is in milliseconds.

2. **Simulator schematic**

   The schematic is given in a separate file. You are supposed to analyse and understand the schematic and do the design accordingly.

   The basic idea is that there is a clock. The clock is advanced to the next significant event that occurs in the system. Various parts of the set-up decide which events will occur when and they put the information in the Event table. The event manager picks the next event. This is how the simulator runs.

   You can make changes in the schematic provided you can justify them.

3. **Scheduler report**

   When no more processes exist in the system, the scheduler should print an informative summary of its operation and stop. You can decide what information should appear in this report.

4. **Originality of design and implementation**

   Any similar designs or code will be taken to be instances of dishonesty.

5. **Schedulers you should implement:** You will implement the simulator for a variety of schedulers. To begin with, you should implement the following 3 schedulers:

   (a) A multiprogramming scheduler

   (b) A time-sharing scheduler using simple round robin scheduling: This scheduler will have only one scheduling level.

   (c) A **multi-level scheduler with promotions and demotions**: This scheduler will have many scheduling levels. It works as follows:

      i. Each scheduling level has a different priority and a different time slice.

      ii. A higher priority level has a smaller time slice and a lower priority level has a larger time slice.

      iii. Many processes may have the same priority. Such processes would exist at the same priority level and would be scheduled in a round-robin manner.

      iv. Processes at a scheduling level would be considered for scheduling only if no ready processes exist at scheduling levels with higher priorities.

v. To begin with, a process will have the priority given in its specification.

vi. **Demotion**: A process is demoted to the next lower priority level (if one exists) if, when scheduled, it uses up the time slice completely, i.e., it does not start an I/O operation before the time slice elapses.

vii. **Promotion**: A process is promoted to the next higher priority level (if one exists) if, when scheduled, it starts an I/O operation before the time slice elapses.

6. **Submission:**

   Submit a single tar file containing a README file and code files. You will submit only some specified schedulers at 5.00 pm and submit all of the specified schedulers at the specified deadline. Thereafter you can submit with 50% penalty until 11 am on Saturday 8 Feb 2014.

7. **Right to fix errors and/or make changes is reserved.**