Assignment #4

Due Date: December 7, 2017

Total: 100 marks

1. (60 marks) Write a CPU scheduler using the Shortest Job First Scheduling algorithm (non-preemptive). Your scheduler gets its process data from a text file in a standard format (see below). You can assume that your system has a single CPU and only one I/O device (and that device operates sequentially, serving only one process at a time).

   Process Data File:

   The process data file consists of a number of text lines of the following format:

   ```
   <process id> <arrival time> <cpu burst> <I/O burst>...<CPU burst>...-99
   ```

   For example:

   ```
   1   0   100  2  200  3  25  -99
   2   30  15  2  15  2  15  2  15  2  15  2  15  2  15  2  15  2  15  2  15  2  15   2  15  2  15  2  15  -99
   ```

   Notes:

   - The first line means we have
     - PID =1, arrival time = 0, CPU burst= 100, I/O burst = 2 CPU burst=200 I/O burst=3 CPU burst = 25.
   - Each line ends with -99.
   - A process can contain any number of CPU or I/O bursts.
   - It should always start and end with a CPU burst.
   - You can also assume that arrival times are in non-decreasing order.

   Output:

   - For each process, compute and display the turnaround time and the waiting time.
   - For the entire system, compute and display the average turnaround time, average waiting time, and throughput.

   For the input above, we have:

   - Total CPU burst is : 100+200+25=325
   - Turn around time for 1st process is: (100+2+200+3+25)=330
   - Throughput is the total number of jobs completed/total time required for their execution, and for the example above, considering just first line, if only one job is done, throughput is 1 in 330.

- Adding the second line (and others), some processes must be put in a waiting queue, thus, waiting time will increase from 0 to a number that you'll have to compute using your program.

2. (60 marks) Repeat the exercise for a Round Robin Scheduling algorithm (non-preemptive).

Marking scheme:

1. receiving tasks and creating the CPU queue 20 marks.
2. creating the I/O queue 20 marks.
3. activating/suspending tasks at the right time(10 marks)
4. computing final results and producing the output (10 marks).
5. a demonstration of your programs (10 marks).