

Assignment 2: CPU Scheduling Algorithm Simulation

Complete the following activities.

1. Write a Java program `YOURLASTNAME_YOURFIRSTNAME_CPU_Sched.java` that simulates the CPU scheduling algorithms (1) First Come First Served (fcfs), (2) Shortest Job Next Non-Preemptive (sjnnp), and (3) Priority Non-Preemptive (pnp). Your program will read an input file, `in.txt`, which will contain data about the CPU requests.
2. In addition to this document, you will find four files on the Moodle website.
 - The first file is `in.txt`, which is an example input file that your program is to read and process.
 - The second file is `fcfs_out.txt`, which is an example output file that was generated by a program simulating the First Come First Served algorithm using the input file. Your program must generate the same output for this algorithm.
 - The third file is `sjnnp_out.txt`, which is an example output file that was generated by a program simulating the Shortest Job Next Non-Preemptive algorithm using the input file. Your program must generate the same output for this algorithm.
 - The fourth file is `pnp_out.txt`, which is an example output file that was generated by a program simulating the Priority Non-Preemptive algorithm using the input file. Your program must generate the same output for this algorithm.
3. Your program will open (not a command line argument) and read/process the input file, `in.txt`, line by line.
 - The first line in the input file contains two tokens. The first token is a single character; either an 'a' or 'A', which specifies that the algorithm your program is to simulate. The second token on this line will defines the actual algorithm; `fcfs`, `sjnnp`, or `pnp`.
 - All other lines in the input file will begin with either 'p' or 'P', which specifies a CPU request by a process. Three additional tokens will follow on these lines for the `fcfs` and `sjnnp` algorithms. The first token defines the process id number (pid) of the process requesting the CPU. The second token defines the time stamp at which the CPU is requested by the process. The third, and last, token defines the CPU burst duration of the process.

- The pnp algorithm will define four tokens following a 'p' or 'P'. The first three tokens are the same as defined for the fcfs and sjnnp algorithms. The fourth token defines the priority of the process.
4. If the CPU scheduling algorithm specified in the input file is fcfs, then your program must process the input file, in.txt, using the First Come First Served CPU scheduling algorithm. If the CPU scheduling algorithm specified in the input file, in.txt, is sjnnp, then your program must process the input file using the Shortest Job Next Non-Preemptive CPU scheduling algorithm. If the CPU scheduling algorithm specified in the input file, in.txt, is pnp, then your program must process the input file using the Priority Non-Preemptive CPU scheduling algorithm.
 5. You may assume the following:
 - All process ids will be positive integers
 - All time stamps will be positive integers
 - All CPU bursts will be positive integers
 - All priorities will be positive integers
 - If no CPU requests exist at a certain clock interval, but there still exist pending CPU requests for later clock intervals, then your program should print the clock interval and move to the next clock interval
 - All output generated by your program should be directed to either the output file fcfs_out.txt, sjnnp_out.txt, or pnp_out.txt, depending on the algorithm specified by the 'a' or 'A' line in the input file.
 6. Study the sample input file in.txt, and the output files fcfs_out.txt, sjnnp_out.txt, and pnp_out.txt that were generated when the input file was processed by Dr. Wiedemeier's program. Your program must generate the same output.
 7. Your program must also handle the following scenarios: (1) in.txt does not exist, (2) in.txt cannot be opened for input, and (3) fcfs_out.txt, sjnnp_out.txt, or pnp_out.txt cannot be opened output.
 8. Submit your program, YOURLASTNAME_YOURFIRSTNAME_CPU_Sched.java, via Moodle before the due date and time.
 9. Five (5) additional points will be awarded if your program can correctly process CPU requests using the Round Robin CPU scheduling algorithm (i.e. the 'a' or 'A' line is rr or RR; the output file is rr_out.txt).

If you have questions concerning this homework assignment, please ask your instructor for help. You may discuss the homework assignment with your classmates. However, the homework assignment you complete and turn in, for a grade, **must be your own work**.