**OPERATING SYSTEM (PRACTICALS) – FALL 2012**

**EXPERIMENT 6 – PROCESS SCHEDULING**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **Students Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | |
| **Marks Obtained: \_\_\_\_\_** | | **COURSE: BESE 16 \_\_\_\_\_** | |
| **Deadline: 1400 hrs 19th Nov 2012** | | **Instructor: Engr. Umar Mahmud** | |
|  | **Instructions**   * This lab is to be performed by a syndicate of at most **TWO** students. Write your remarks next to the space provided. * Plagiarism is strictly forbidden. * Submit hard copy of the report before deadline. Marks will be deducted for late submissions. | |  |
| 1. | **Objectives:**   1. Learning process scheduling 2. Comparing among FCFS and SJF scheduling algorithms through a simulation | |  |
| 2. | **Time Required:** 3 hrs | |  |
| 3. | **Software Required:**   1. Java/C/C++/C# 2. Windows/Ubuntu | |  |
| 4. | **Process Schedulers:** Process schedulers are responsible for selecting which process to execute so that overall performance is increased. | |  |
| 5. | **Performance Metrics:** Metrics include throughput, CPU overhead, response time, average wait time etc. | |  |
| 6. | **First Come First Served (FCFS) Scheduler:** This scheduler implements a FIFO Queue and schedules all process on their order of arrival.  <http://cs.uttyler.edu/Faculty/Rainwater/COSC3355/Animations/fcfs.htm> | |  |
| 7. | **Shortest Job First (SJF) Scheduler:** This scheduler priorities processes on the basis of their CPU usage time.  <http://sharingmythreeyears.blogspot.com/> | |  |
| 8. | **Comparison of Schedulers:** Implement a queue of length 5 to hold processes for FCFS and SJF both and for different algorithms observe the throughput at each second. To observe throughput you have to mark at each instant how many processes have been executed for each scheduler. | |  |
| 9. | Assume that all processes arrive at initial time and burst time is available for each process. | |  |
| 10. | |  |  | | --- | --- | | **Process ID** | **Burst Time (Seconds)** | | P1 | 5 | | P2 | 8 | | P3 | 11 | | P4 | 4 | | P5 | 2 | | |  |
| 11. | |  |  | | --- | --- | | **Process ID** | **Burst Time (Seconds)** | | P1 | 5 | | P2 | 10 | | P3 | 15 | | P4 | 20 | | P5 | 25 | | |  |
| 12. | |  |  | | --- | --- | | **Process ID** | **Burst Time (Seconds)** | | P1 | 25 | | P2 | 20 | | P3 | 15 | | P4 | 10 | | P5 | 5 | | |  |
| 13. | Given the states in Points 10, 11 and 12 schedule using both techniques and plot throughput on a graph. The horizontal specifies the time and the vertical marks the total number of processes completed so far.  **10th:**   * FIFO: P1, P2, P3, P4, P5 * SJF: P5, P4, P1, P2, P3   **11th:**   * FIFO: P1, P2, P3, P4, P5 * SJF: P1, P2, P3, P4, P5   **12th:**   * FIFO: P1, P2, P3, P4, P5 * SJF: P5, P4, P3, P2, P1   Throughput = # of process / total time | | (5) |
| 14. | Calculate average wait time for each algorithm and for all three states on Points 10, 11 and 12. Average wait time is the sum of waiting times by each process divided by the total number of processes   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Processes and Time** | **Point # 10** | | **Point # 11** | | **Point # 12** | | | **FIFO** | **SJF** | **FIFO** | **SJF** | **FIFO** | **SJF** | | **P1** | 0 | 6 | 0 | 0 | 0 | 50 | | **P2** | 5 | 11 | 5 | 5 | 25 | 30 | | **P3** | 13 | 19 | 15 | 15 | 45 | 15 | | **P4** | 24 | 2 | 30 | 30 | 60 | 5 | | **P5** | 28 | 0 | 50 | 50 | 70 | 0 | | **Total Waiting Time** | 70 | 38 | 100 | 100 | 200 | 100 | | **Average Waiting Time** | 14 | 7.6 | 20 | 20 | 40 | 20 | | | (2) |
|  |  | |  |
| 15. | What is your conclusion based on the throughput and average wait time as well as the type of input. | | (3) |

**No overhead in FCFS .. overhead in SJF is present … from overhead point of view, FIFO (fCFS) is better, from throughput point of view SJF is better.**