**PRE-LAB QUESTIONS**

1. What scheduling policy will you use for each of the following cases? Explain your reasons for choosing them.
   1. The processes arrive at large time intervals:
      1. **FCFS. Since the process are arriving at large time intervals so instead of switching or waiting for any other process it is best to execute those processes which have arrived first.**
   2. The system’s efficiency is measured by the percentage of jobs completed.
      1. **SJF. In SJF, jobs having smallest burst time will be executed first and thus SJF will provide better results.**
   3. All the processes take almost equal amounts of time to complete.
      1. **Round Robin. Since all processes are taking almost equal amount of time, it is a better choice of algorithm for scheduling. FCFS can be used when time interval is almost same for all processes.**
2. What are the factors that need to be considered to determine the degree of multiprogramming in a system?
   1. **The overheads in context switching may become excessive.**
   2. **The response time may become unacceptable with excessive multiprogramming.**
3. What happens if the time allocated in a Round Robin Scheduling is very large? And what happens if the time allocated is very low?
   1. **If the time allocated is very large, the scheduling algorithm becomes similar to FCFS.**
   2. **If the time allocated is very low, context switching of processes increases and as a consequence, throughput (or) efficiency decreases.**
4. What is the difference between the idle and blocked state of a process?

|  |  |
| --- | --- |
| **IDLE STATE OF A PROCESS** | **BLOCKED STATE OF A PROCESS** |
| The process is waiting for the processor to become free so that it can execute. | The process has been put out from running state by the processor due to some interrupts, such as I/O. |

1. When a process requests for I/O, how many process switches take place?
   1. **Two switches.**
   2. **First: Process to be switched is taken out and the scheduler starts executing.**
   3. **Second: Next process is brought to execution.**

**POST-LAB QUESTIONS**

1. Suppose a new process in a system arrives at an average of six processes per minute and each such process requires an average of 8 seconds of service time. Estimate the fraction of time the CPU is busy in a system with a single processor.
2. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

|  |  |
| --- | --- |
| Process ID | Burst Time |
| 0 | 80 |
| 1 | 20 |
| 2 | 10 |
| 3 | 20 |
| 4 | 50 |

* 1. Suppose a system uses FCFS scheduling .Create a Gantt chart illustrating the execution of these processes?
  2. What is the turnaround time for process p3?
  3. What is the average wait time for the processes?

1. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

|  |  |  |
| --- | --- | --- |
| Process ID | T(pi) | Arrival Time |
| 0 | 80 | 0 |
| 1 | 20 | 10 |
| 2 | 10 | 10 |
| 3 | 20 | 80 |
| 4 | 50 | 85 |

* 1. Suppose a system uses RR scheduling with a quantum of 15. Create a Gantt chart illustrating the execution of these processes?
  2. What is the turnaround time for process p3?
  3. What is the average wait time for the processes?

1. A CPU scheduling algorithm determines an order for the execution of its scheduled processes. Given n processes to be scheduled on one processor, how many possible different schedules are there? Give a formula in terms of n.
2. Consider the following preemptive priority-scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue but not running), its priority changes at a rate when it is running, its priority changes at a rate Y. All processes are given a priority of 0 when they enter the ready queue. The parameters X and Y can be set to give many different scheduling algorithms. What is the algorithm that results from Y>X>0?
   1. LIFO
   2. FCFS
   3. Round Robin
   4. None of the above