**EEL358 : Operating Systems**
**MINOR-II**

*Note:* Give brief and precise answers; Max. Marks : 15 ; Time: 1 hr

1. Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. Use non-preemptive scheduling with zero context-switch overhead and base all decisions on the information that you have at the time the decision must be made.

| Process | Arrival Time | Bust Time |
|---|---|---|
| P1 | 0.0 | 6 |
| P2 | 0.3 | 3 |
| P3 | 1.0 | 1 |

(a) In the above problem assume a three level MLFQ; a process can execute in the first level for 1 time unit, in the second level for 2 time units and in the third level for 4 time units. Calculate the average turn around time and waiting times for each of the processes.

(b) What is the average turn around time and waiting time for these processes with RR scheduling algorithm for a quantum of 1 time unit? Ignore arrival times. (5)

2. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 10 | 3 |
| P2 | 1 | 1 |
| P3 | 2 | 3 |
| P4 | 1 | 4 |
| P5 | 5 | 2 |

The processes are assumed to have arrived in the order *P*1, *P*2, *P*3, *P*4, *P*5, all at time 0.
(a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. (b) What is the turnaround time of each process for each of the scheduling algorithms in part (a) ? (c ) What is the waiting time of each process for each of the scheduling algorithms in part (a) ? (d) Which of the schedules in part (a) results in the minimal average waiting time (over all processes)? (4)

3. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other? What resources are used when a thread is created? How do they differ from those used when a process is created? (3)

4. Show that, if the wait() and signal() semaphore operations are not executed atomically, then mutual exclusion may be violated. Explain why spinlocks are not appropriate for single-processor systems yet are often used in multiprocessor systems. (3)