CSI3131 – Operating Systems
Tutorial 4 – CPU Scheduling

1. Define the difference between preemptive and cooperative scheduling.

2. What advantage is there in having different time-quantum sizes on different levels of a multilevel queuing system?

3. Many CPU-scheduling algorithms are parameterized. For example, the RR algorithm requires a parameter to indicate the time slice. Multilevel feedback queues require parameters to define the number of queues, the scheduling algorithms for each queue, the criteria used to move processes between queues, and so on. These algorithms are thus really sets of algorithms (for example, the set of RR algorithms for all time slices, and so on).One set of algorithms may include another (for example, the RR algorithm is the FCFS algorithm with an infinite time slice). What (if any) relation holds between the following pairs of sets of algorithms?
   - ⇒ Priority and SJF
   - ⇒ Multi feedback queues and FCFS
   - ⇒ Priority and FCFS
   - ⇒ RR and SJF

4. Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound processes and yet not permanently starve CPU-bound programs? Is it possible that the CPU-bound process be starved and under what conditions?

5. Assume an operating system maps user-level threads to the kernel using the many-to-many model where the mapping is done through the use of LWPs (light weight processes). Furthermore, the system allows program developers to create real-time threads. Is it necessary to bind a real-time thread to an LWP?

6. Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

   a) What is the average turnaround time for these processes with the FCFS scheduling algorithm?

   b) What is the average turnaround time for these processes with the SJF (no preemption) scheduling algorithm?

   c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process *P*1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first1 time unit and then SJF scheduling is used. Remember that processes *P*1 and *P*2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

   d) What is the average turnaround time for these processes with the preemptive SJF (SRTF – shortest remaining time first) scheduling algorithm?

7. Consider three processes P1, P2, and P3 with the following CPU burst times:

Burst times for P1: 14, 12, 17
Burst times for P2: 2, 2, 2, 3, 2, 2, 2, 3
Burst times for P3: 6, 3, 8, 2, 1, 3, 4, 1, 9, 7

All three arrive at time 0, in order P1, P2, P3. Each CPU burst is followed by an I/O operation taking 6 time units, except for the last burst after which the process terminates. Using the provided tables, simulate the execution of these processes using the following scheduling algorithms. Assume that overhead time for process switching and scheduling functions are negligible (assumed to be 0).

a) FCFS
b) SJF (no pre-emption)
c) Preemptive SJF
d) Round robin with a quantum of 5 time units.
e) Round robin with a quantum of 5 time units with priority: P2=P3 > P1.
f) Multi-level feedback queuing using three queues with the following parameters:
    i. Queue 0 – quantum of 2 time units (after which process is moved to Queue 1)
    ii. Queue 1 – quantum of 4 time units (after which process is moved to Queue 2)
    iii. Queue 2 – FCFS
    iv. All process that becomes ready is added to Queue 0.
    v. A process that arrives in Queue 0 preempts any executing process that belongs to either Queue 1 or Queue 2.
    vi. Processes in Queue 1 are allocated to the CPU only when Queue 0 is empty.
    vii. Processes in Queue 2 are allocated to the CPU only with both Queue 0 and Queue 1 are empty.

Table for algorithms a to d:

Algorithm:				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated

Multi-level Feedback Queue Scheduling						
Time	**CPU**	**Queue 0**	**Queue 1**	**Queue 2**	**Blocked**	**Terminated**