## Operating System Labs Jan-May 2018

## Scheduling and Synchronization

1. The Multi Level Feedback Queue (MLFQ) has multiple levels of queues and uses feedback to determine the priority of a given job. In our treatment, the MLFQ has a number of distinct queues, each assigned a different priority level. At any given time, a job is on a single queue only. MLFQ uses priorities to decide which job should run at a given time: a process with higher priority (i.e., a process on a  higher queue) is chosen to run. Of course, more than one process may be in a given queue, and thus, have the same priority. In this case, we will just use round-robin scheduling among those processes.

   **Write a program to implement multi-level feedback queue by following the rules:**

   1. There are 3 queues with RR time quantum as 4, 8 and 16 time unit respectively.
   2. When a process enters the system, it is placed at the highest priority (the topmost queue).
   3. Once a process runs for one RR time quantum at a given queue, and if it does not finish, it will be moved to the tail of the queue one level down.
   4. Processes on the lower priority queue will not execute until all the higher priority queues become empty.
   5. If there are more than one process available on the single queue then, Round Robin scheduling will be used among those processes.
   6. Once a process will reach to the lowest level of the queue, it will remain there till the time it finishes.

   Print the order of process completion.

   Example: Assume there are five processes that arrives at the same time. Following are the burst time unit of processes

   Input: P1: 64, P2: 12, P3: 7, P4: 2, P5: 41

   Output: P4, P2, P3, P5, P1

2. Write a program in which a process creates two threads. Both the threads run one after another collaboratively and intermittently and print numbers till 100. One thread will print even numbers and another thread will print odd numbers.

   0 1 2 3 4 5 6....

   Remember they are accessing the same number and changing it.