1. [0.25 point] Which of the following is NOT true about the multi-level queue scheduler?
   
   a) It has multiple queues with different priorities.
   b) If two jobs have the same priority, the scheduler runs them in the round robin way.
   c) The scheduler runs I/O-intensive jobs on a queue with the lowest priority.
   d) When a job enters a system, it is placed on a queue with the highest priority.

2. [0.5 point] What is the advantage of having different time slices at different levels of the multi-level queue scheduler?
   
   Interactive (I/O) jobs run on queues with shorter time slices and have shorter response times. CPU-intensive jobs run on queues with longer time slices and have shorter turnaround times.

3. [0.5 point] The lottery scheduling assigns tickets to jobs. Tickets are used to allocate CPU time to jobs. At the start of a time slice, the scheduler chooses a lottery ticket randomly. The job that holds that ticket is scheduled on CPU. Explain how the scheduler ensures that the high-priority jobs receive more CPU time compared to the low-priority jobs.
   
   By assigning more tickets to high-priority jobs

4. [0.25 point] True/False
   
   A thread is composed of a set of privately owned registers including program counter, and privately owned heap.
   
   False

5. [1 point] Explain what parallelism means. You may justify your answer with an example.
   
   Running a program on multiple CPUs to speedup computations

6. [2 points] Consider the following Code snippet:

   ```c
   pid_t pid;
   pid = fork();
   if(pid == 0) { // child process
     fork();
     thread_create(...);
   }
   ```
How many **threads** are created?

Three threads (a program that runs the above code snippet is the main thread)

7. [2 points] Explain what **critical section** means. What is the name of the problem that occurs when multiple threads enter a critical section at roughly the same time?

   - Critical section contains a shared resource.
   - Race condition