CS 4760	**Operating Systems**	**Test 2**
Name:	Fall 2020	Max Pts: 68

Important: This is an open book test. You can use any books, notes, or paper. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem.

1. [6 pt] Linux always picks up the process from the highest priority queue that is not empty. How does it prevent starvation of processes in the lower priority queues?

2. [6 pt] What system call does Linux use to create a lightweight process?

3. [6 pt] m processes, labeled p_0 to p_{m-1}, each of length n seconds, are scheduled to execute under round robin. The quantum is q seconds. If they start at time 0, what time will process p_0 finish execution? How about process p_{m-1}?

4. [12 pt] Assume you have the following jobs to execute with one processor:

Process	Burst time	Arrival time
p_0	8	0
p_1	8	2
p_2	6	10
p_3	9	12
p_4	9	17

   Calculate and show the average wait time for this set of processes using the following algorithms. Specify the arbitration rule used for each algorithm, if needed. Assume no time taken by the scheduler except where context switch cost is mentioned. Show your work.

   (a) First in first out

   (b) Shortest job next

   (c) Shortest remaining time next (pre-emptive)

   (d) Round robin, with a quantum of 4

5. [6 pt] Give an example of a deadlock in a system with a few processes and a single resource class. Show a situation that leads to deadlock in such a system.

6. [6 pt] Is a process in a blocked state deadlocked? Is a process in a deadlocked state blocked?

7. [6 pt] What is the *most favored execution* of an unblocked process in deadlock detection by process-resource graphs?

8. [10 pt] In the code below, three processes are competing for six resources labeled `A` through `F`. There are two instances of each resource class.

```
void p0()
{
  while ( 1 )
  {
    get ( B );
    get ( D );
```

```
void p1()
{
  while ( 1 )
  {
    get ( D );
    get ( C );
```

```
void p2()
{
  while ( 1 )
  {
    get ( D );
    get ( D );
```

```
    get ( A );                  get ( B );                  get ( C );
    crit_sec ( A, B, C );       crit_sec ( B, D, E );       crit_sec ( C, D, F );
    release ( C );              release ( D );              release ( C );
    release ( A );              release ( B );              release ( D );
    release ( B );              release ( C );              release ( D );
  }                           }                           }
}                           }                           }
```

Is there a deadlock in the system? Under what conditions may it occur?

9. [6 pt] What is the difference between static and dynamic linking?

10. [6 pt] What is the relationship between swap time and the quantum for which the process is expected to have control of CPU?