CS 4760	Operating Systems	Test 2
Name:	Fall 2022	Max Pts: 64

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] Out of interactive, batch, and real-time processes, which type of processes are typically handled at the highest priority? Why?

2. [6 pt] Why does Linux use spin locks instead of semaphores to handle race conditions inside kernel? Isn't spin lock a form of busy-wait that wastes CPU cycles to perform synchronization?

3. [6 pt] Does the following code solve the producer-consumer problem? Explain your answer.

```
semaphore mutex;          // For exclusive access to buffers; initialized to 1
semaphore empty;          // Number of available buffers; initialized to n
semaphore full;           // Number of buffers with content; initialized to 0

void producer()
{
    while ( 1 )
    {
        produce ( item );
        mutex.P();
        empty.P();
        put ( item );
        full.V();
        mutex.V();
    }
}
```

```
void consumer()
{
    while ( 1 )
    {
        full.P();
        mutex.P();
        remove ( item );
        empty.V();
        mutex.V();
        consume ( item );
    }
}
```

4. [6 pt] n processes are time-sharing the CPU using round-robin scheduling on a single CPU machine. Each of them requires T ms of CPU time to complete. What is the average turnaround time for the processes if they all arrived at the same time? Assume that the context switching overhead is zero.

5. [6 pt] How does Linux ensure that real-time processes are always executed ahead of other processes?

6. [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.

7. [6 pt] What is the difference between a *blocked state* and a *deadlock state*? Is a process in a blocked state deadlocked? Is a process in a deadlocked state blocked?

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 )
```

```
void p1()
{
  while ( 1 )
```

```
void p2()
{
  while ( 1 )
```

```
  {                          {                          {
    get ( B );                 get ( D );                 get ( D );
    get ( D );                 get ( C );                 get ( D );
    get ( A );                 get ( B );                 get ( C );
    crit_sec ( A, B, D );      crit_sec ( B, C, D );      crit_sec ( C, D, D );
    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] Why do we need the capability to relocate processes?

10. [6 pt] You can perform linking after compilation with either the chain method or by indirect addressing. Which of these methods results in the symbol table being eliminated from the final binary executable?