CS 4760 | **Operating Systems** | **Test 2**

Name: Fall 2019 Max Pts: 68

Important: This is an open book test. You can use any books, notes, or paper. *Do not use any electronic or communications device. Change your cell phones to silent mode.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. Please write legibly; if I cannot read what you wrote, I'll give you a zero. You will lose five points for not writing your name.

1. [6 pt] What is a kernel primitive? Give an example of a primitive in the Linux kernel.

2. [6 pt] Consider a variant of the round-robin scheduling algorithm where the entries in the ready queue are pointers to the PCBs. What would be the effect of putting two pointers to the same process in the ready queue? Specify advantages and any special conditions you need to handle.

3. [6 pt] Why are spin locks of no use in a uniprocessor environment?

4. [6 pt] In Linux, the PIDs are normally not reused immediately. Linux uses the bit-vector to find the next PID to be assigned to a new process, using the last-used PID that it saves. Why doesn't it immediately reuse the PIDs that have become available?

5. [6 pt] Briefly describe the COW approach.

6. [6 pt] What is an arbitration rule? Give an example of an arbitration rule in a scheduling algorithm of your choice.

7. [6 pt] Give two examples each of reusable and consumable resources.

8. [10 pt] Consider a system with the following set of processes and states:

$$P = \{p_0, p_1, p_2\},\ S = \{s_0, s_1, s_2, s_3\}$$

State changes due to processes are:

$$\begin{array}{lll}
p_0(s_0) = \{s_2\} & p_1(s_0) = \{s_0, s_1\} & p_2(s_0) = \emptyset \\
p_0(s_1) = \{s_1, s_2\} & p_1(s_1) = \emptyset & p_2(s_1) = \emptyset \\
p_0(s_2) = \{s_0, s_1, s_3\} & p_1(s_2) = \{s_0\} & p_2(s_2) = \{s_0\} \\
p_0(s_3) = \{s_0\} & p_1(s_3) = \{s_0, s_1, s_2\} & p_2(s_3) = \{s_1, s_2\}
\end{array}$$

Draw the corresponding state change diagram. Is the system safe? Is it deadlocked? Is there a knot in the system?

9. [6 pt] What is the difference between deadlock prevention and deadlock avoidance? Which of the two makes better use of resources?

10. [10 pt] Consider a system with a total of 150 units of memory allocated to three processes as shown:

Process	Max	Current Alloc
p_0	70	45
p_1	60	40
p_2	60	15

Apply the Banker's algorithm to determine whether it would be safe to grant each of the following requests. If yes, indicate a sequence of terminations that could be guaranteed possible. If no, show the reduction of the resulting allocation table.

(a) A fourth process arrives, with a maximum memory need of 60 and an initial need of 25 units.

(b) A fourth process arrives, with a maximum memory need of 60 and an initial need of 35 units.