CS 4760	**Operating Systems**	Test 1
Name:	Fall 2019	Max Pts: 78

Important: This is an open book test. You can use any books, notes, or paper but no electronic device. *Do not log into the computer during the test, or 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] Multithreading and SMP complement each other and are used effectively together. What is the difference between them?

2. [6 pt] What is the distinction between spatial locality and temporal locality. Explain with an example for each.

3. [4+4+4 pt] Consider a hypothetical 32-bit microprocessor having 32-bit instructions composed of two fields. The first byte contains the opcode, and the remainder an immedidate operand or an operand address.

   (a) What is the maximum directly addressable memory capacity (in bytes)?

   (b) Discuss the impact on the system speed if the microprocessor bus has:

      i. a 32-bit local address bus and a 16-bit local data bus, or;

      ii. a 16-bit local address bus and a 16-bit local data bus

   (c) How many bits are needed for the program counter and the instruction register?

4. [6 pt] What is *availability* of a computer? How does it relate to reliability and downtime?

5. [12 pt] The following state transition table is a simplified model of process management, with the labels representing transitions between states of `READY`, `RUN`, `BLOCKED`, and `NONRESIDENT`.

	READY	RUN	BLOCKED	NONRESIDENT
READY	-	1	-	5
RUN	2	-	3	-
BLOCKED	4	-	-	6

Intrepret transition 2 as the fact that the process can change from `RUN` to `READY`. Give an example of an event that can cause each of the above transitions. Draw a diagram if that helps.

6. [6 pt] I wrote the following code to create a child to do something and return. The return value is to be caught by the parent. Can you see any problem with the code that I wrote? How will you fix it?

```
pid_t pid = fork();
if ( pid < 0 )
    exit ( 1 );
if ( pid == 0 )
    wait();
exit ( 0 );
```

7. [6 pt] Consider the three states a process can assume: running, ready, and blocked. A process is currently scheduled to run. There is a hardware interrupt raised. What will be the state transition for the process?

8. [6 pt] What is the distinction between *blocking* and *nonblocking* with respect to messages?

9. [6 pt] The bakery algorithm allows two processes to pick the same value for `turn`. Yet, it does not create a conflict for mutual exclusion. How is this situation resolved?