

CS 347 QUIZ 2 (July 28, 2016)

Name: _____ **Roll No.** _____

Please write your answer in the space next to the question.

1. Consider a parent process that has forked a child in the code snippet below.

```
int count = 0;
ret = fork();
if(ret == 0) {
    printf("count in child=%d\n", count);
}
else {
    count = 1;
}
```

The parent executes the statement "count = 1" before the child executes for the first time. Now, what is the value of count printed by the code above? Assume that the OS implements a regular fork (not a copy-on-write fork).

2. Repeat the previous question for a copy-on-write fork implementation in the OS.

3. Consider the wait family of system calls (wait, waitpid etc.) provided by Linux. A parent process uses some variant of the wait system call to wait for a child that it has forked. Which of the following statements is always true when the parent invokes the system call?

- A. The parent will always block.
- B. The parent will never block.
- C. The parent will always block if the child is still running.
- D. Whether the parent will block or not will depend on the system call variant and the options with which it is invoked.

4. Consider a simple linux shell implementing the command 'sleep 100'. Which of the following is an accurate ordered list of system calls invoked by the shell from the time the user enters this command to the time the shell comes back and asks the user for the next input?

- A. wait-exec-fork
- B. exec-wait-fork
- C. fork-exec-wait
- D. fork-wait-exec

5. Consider a process that has requested to read some data from the disk and blocks. Subsequently, the data from the disk arrives and the interrupt is serviced. However, the process doesn't start running immediately. What is the state of this process at this stage?

6. A process performs a read from a socket. Which of the following statements is true regarding the read system call?
- A. Always blocks
 - B. Never blocks
 - C. Always blocks if the data to read is not ready
 - D. It may block or not depending on the options set on the socket
7. Under what conditions is every user-level thread created by a process scheduled as a separate entity by the OS?
8. Which of the following statements is a correct description of what the pipe system call (to create anonymous pipes) does?
- A. It connects the STDIN and STDOUT descriptors of a process.
 - B. It connects two new file descriptors of a process via a buffer.
 - C. It connects a file descriptor in a parent and another in a child via a buffer.
 - D. It connects two new file descriptors of two separate processes via a buffer.
9. Consider an OS that maps N user-level threads in a process to one kernel thread. Can concurrent updates of data between these N user threads still cause race conditions? (Yes/No)
10. Repeat the above question when there are N user level threads mapped to N kernel threads, and executing on an N -core CPU.
11. Consider an OS that maps N user-level threads to one kernel thread. Now, when this kernel thread is scheduled by the OS scheduler, who makes the decision of which of the N user level threads should execute?
12. Consider N threads in a process that share a global variable in the program. If one thread makes a change to the variable, is this change visible to other threads? (Yes/No)
13. Consider N threads in a process. If one thread passes certain arguments to a function in the program, are these arguments visible to the other threads? (Yes/No)
14. Do multiple user-level threads in a process always share a kernel stack? (Yes/No)
15. Consider a process P_1 that forks P_2 , P_2 forks P_3 , and P_3 forks P_4 . P_1 and P_2 continue to execute while P_3 terminates. Now, when P_4 terminates, which process must wait for and reap P_4 ?