50.005 Quiz OS 3 (15 mins)

Name: _____ Student ID: _____

Note: This quiz is closed-book and closed-notes, except for one double-sided A4 cheat sheet allowed. You also can't go online or look at anything electronic, including your laptop, smartphone, etc.

1. **[1m]** Suppose a company runs a web server that is single threaded, on a uniprocessor machine. What is a main disadvantage of this approach compared with a multithreaded server architecture?

The company server is unresponsive to other new-coming clients when it is busy servicing the previously connected client. It cannot interleave the task between checking for a new client and servicing connected clients. (any answer along this line gain 1m)

2. **[2m]** A process P needs to transfer 3 Kbytes of data stored in a local buffer to another process Q. If message passing (socket) is used, what is the minimum needed (i) number of system calls, and (ii) total amount of data copied to successfully pass the message from P to Q?

(i) 2 system calls (one for send() and receive() each) **[1m]**, (ii) 6Kbytes. **[1m]**

2. State whether the following is true or false. No further explanation is needed.

(a) **[1m]** Two threads in the same process can pass data to each other directly (i.e., without using system calls). **True**

(b) **[1m]** A buggy thread in a process can corrupt / affect the stack of another thread in the same process. **True**
Explanation: (although not needed) threads share data, code, and files. A bug in one thread, can affect these shared code/data/files, which in turn affects the other thread's stack when this other thread access these corrupted things.

(c) **[1m]** You run a process to complete a task using 16 threads created by a thread library. For each process, the library supports only a many-to-one mapping from its threads to kernel threads. The process runs on a 16-core machine and the task is fully parallelizable. It is possible for the process to achieve a 16x speedup compared with a sequential execution of the same task.
False.

(d) **[1m]** Consider the following C function:

```
void test(){

    int x = 5;

    if(fork() == 0){
        x ++; //Child process
        printf("%d", x);
    }
    else{
        x --; //Parent process
        printf("%d", x);
    }
```

Date: 20/02/2019

50.005 Quiz OS 3 (15 mins)

The kernel schedules the child process to run first, so that the child increments x and prints its value as 6. Afterwards, the parent runs. The parent will decrement x and print the value 5. **False.**

4. **[2m]** Consider the following Java code:

```
4
5   private Stack<String> stack = new Stack<String>();
6   stack.push("HELLO");
7
8   public void doSomething() {
9       if (stack.isEmpty()) return; String s = stack.pop();
10      //do something with s...
11  }
12
```

The code manages a data structure that is a stack of strings. The doSomething() method checks if the stack is empty. If so, it returns immediately. Otherwise, it pops a string from the stack and does some work on the string. An attempt to pop a string from an empty stack will crash the program.

Assume that both the isEmpty() and pop() methods are atomic. Consider two threads S and T operating on the same (i.e., shared) stack concurrently through doSomething(). Initially, the stack has only one single string, i.e., “HELLO”. **Show an interleaved execution of S and T that will crash the program.**

1. S checks if stack.isEmpty(), *true* and S is interrupted **[0.5m]**
2. T runs and checks if stack isEmpty(), *true* and perform pop(), and do something with s. T exits. **[0.5m]**
3. S resumes, and attempts to pop() a now empty stack (due to T’s doing). **[0.5m]**
4. The program crashes. **[0.5m]**

Total marks: 9 marks