

CS252 Systems Programming

Midterm Exam Preparation

Please write by hand the answers and turn them in the day of the exam. The solutions will be posted the day before the exam. I suggest you try to solve the questions before looking at the solution.

Answer True/False (T/F) (1 point each)

- _____ ELF stands for Executable Link Format.
- _____ A process has a stack for every thread it contains.
- _____ A process ID may be reused
- _____ Race conditons are more likely to happen in multiprocessor machines.
- _____ In POSIX threads, the thread calling fork will create a child process that is a copy of the parent including all the threads in the process.

Part 2. Short questions.

2. Write down the fields of an i-node

3. Explain what is deadlock and what is starvation and give an example of each.

4. What are the advantages and disadvantages of using threads vs. using processes?

5. Mention three important files that are stored in /etc in UNIX.

<p>T1:</p> <p>a) mutex_lock(&m3)</p> <p>b) mutex_lock(&m5);</p> <p>c) mutex_lock(&m1);</p>	<p>T2:</p> <p>d) mutex_lock(&m2)</p> <p>e)mutex_lock(&m5);</p> <p>f)mutex_lock(&m4);</p>	<p>T3:</p> <p>g)mutex_lock(&m4);</p> <p>h)mutex_lock(&m3);</p> <p>i)mutex_lock(&m1);</p>
--	--	--

b) Draw the graph representation of the deadlock.

T1:	T2:	T3:
-----	-----	-----

7. Assume that the following code for adding items into an array. `addToArray` will return the index in the array where the value was added or -1 if the buffer is full. a) What problems do you see can happen if multiple threads try to run it? Rewrite the code to solve the problem.

```
#define MAXCOUNT;
int count = 0;
int array[MAXCOUNT];

int addToArray(int value)
{
    if (count == MAXCOUNT) {
        return -1;
    }

    array[count]=value;

    count = count + 1;

    return count-1;
}
```

New Code

8. A program calls the system call `write(file_descriptor, buffer, nbytes)`. Explain step by step the sequence of events, the checks, and the interrupts in user mode and kernel mode that happen from the time `write()` is called until it returns.

Part 3. Programming questions.

9. Write a shell script that will run forever and it will check every minute if a file has been modified and it will send e-mail to the user running the script when it has been modified with the changes made using "diff".

10. Write a program "grepsort arg1 arg2 arg3" that implements the command "grep arg1 | sort < arg2 >> arg3". The program should not return until the command finishes. "arg1", "arg2", and "arg3" are passed as arguments to the program. Example of the usage is "**grepsort hello infile outfile**". This command will print the entries in file **infile** that contain the string **hello** and will append the output sorted to file **outfile**. Do error checking. Notice that the output is appended to arg3.

```
int main( int argc, char ** argv)
{
```

```
}
```



11. **Using C++ and Semaphores** write a class SynchronizedStackSemaphores of int values where pop() will block if the stack is empty and push will block if the stack is full. Write the member variables that you think are necessary. Implement the stack with an array of int's and allocate it dynamically in the constructor. Hint: Use the "Bounded Buffer Problem" with semaphores as an example in your implementation.

```
class SynchronizedStackSemaphores {
    // Add your member variables here
    int top;
    int * stack;

public:
    SynchronizedStackSemaphores(int maxStackSize);
    void push(int val);
    int pop();
};

SynchronizedStackSemaphores SynchronizedStackSemaphores(int maxStackSize) {

}

void SynchronizedStackSemaphores::push(int val) {

}

int SynchronizedStackSemaphores::pop() {
```



}