

## **OS Lab (CS236) Spring 2025 Lab Quiz 1 Grading Rubric**

### **General instructions to TAs:**

Download student submission from submission server. Extract the submitted code. Grade each question as per the rubric given below. Enter the final total marks on Moodle under the lab quiz 1 assignment. Provide a question-wise breakup of the marks in the feedback.

### **Q1 Linux commands**

Given below are suggested solutions for the various parts. But please note that alternate answers may exist, e.g., using grep to extract fields from ps output. So please run the student solution on the terminal and give marks for correct output, even if the command differs from what is shown below. (It is okay to excuse minor changes in output format but check with instructor before doing so.)

***[1 mark for each of the 5 subparts]***

- (a) `ps -eo pid, state, rss`
- (b) `pstree -A 123`
- (c) `ps -e`
- (d) `ps -ejH`
- (e) `Ps -N -C init -o pid, comm, time`

### **Q2 Simple shell**

Compile and run the student code using gcc

```
gcc my-shell-q2.c  
./a.out
```

If compilation is successful and the shell starts to run, check the following on the student's shell and assign marks as described below.

***[1 mark]*** Run any two simple commands on the student's shell with command line arguments (echo hi, ls -l, sleep 2, ps aux, and so on) and make sure the output looks fine (ignore exit status for now)

***[1 mark]*** Use the cd command to change directory (you can create some directory first or change to parent directory via "cd ..") and check that it works

***[1 mark]*** Check that the command prompt is displaying the current working directory properly in the previous commands

**[1 mark]** Check that previous commands are printing an exit status of 0. Give an incorrect command (some garbage command) and check that exit status of 1 is printed

**[1 mark]** Run a long command like “sleep 100”. The command should finish after 10 seconds and print a message like “killed child after 10 seconds”

**[1 mark]** Run any command like “sleep 10”. Open another terminal and check that the child process is reaped after the command completes. That is, after 10 seconds, if you run “ps aux | grep sleep” in another terminal, you should not see the “sleep 10” process

Hit Ctrl+C to exit the student code

### **Q3 Shell with sequential execution**

Compile and run code as follows.

```
gcc my-shell-q3.c  
./a.out
```

If compilation is successful and the shell starts to run, check the following on the student’s shell and assign marks as described below.

**[0.5 mark]** Run the following command:

```
sleep 5 ## echo hi ## ps
```

The shell should first pause for 5 seconds, then print “hi”, then output of “ps”

**[0.5 mark]** Run the following command:

```
sleep 5 ;; echo hi ;; ps
```

The shell should first pause for 5 seconds, then print “hi”, then output of “ps”

**[1 mark]** Run the following command

```
echo hi ## sleep 100 ## echo hello
```

Now, after “the shell prints hi”, press Ctrl+C while the long sleep is running. The shell should come back to the command prompt and “hello” is NOT printed.

**[1 mark]** Run the following command

```
echo hi ;; sleep 100 ;; echo hello
```

Now, after “the shell prints hi”, press Ctrl+C while the long sleep is running. The shell should print “hello” and then come back to the command prompt.

**[1 mark]** Run the following command to check precedence of ## over ;;

```
sleep 100 ## echo hi ;; echo hello ## sleep 100 ## echo world ;; echo bye
```

During the first long sleep, press Ctrl+C, the shell should NOT print “hi” but print “hello”

Then there will be another long sleep. Press Ctrl+C again. The shell should NOT print “world” but print “bye”

#### **Q4 Parent-child communication with signals**

Compile and run each of the two programs as shown below. Enter a string of 0s and 1s of length 8 and you should see the same string printed back by the child. The output will be printed slower in part (a) and faster in part (b).

```
$ gcc bitstring-send.c
$ ./a.out
Please input a 8-bit bitstring: 10111111
[Parent] Input bitstring is    10111111
[Child] Received bitstring is  10111111
$ gcc bitstring-send-2.c
$ ./a.out
Please input a 8-bit bitstring: 10111011
[Parent] Input bitstring is    10111011
[Child] Received bitstring is  10111011
```

Assign marks **[3 marks]** for part (a) and **[2 marks]** for part (b) if the above test passes AND the code has some usage of signals via kill etc. That is, please make sure the students have not simply printed the output without using signals.