

COMP 2560 Fall 2020

Lab 6

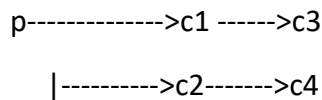
IMPORTANT CHANGE REGARDING VIDEO SUBMISSION

When video submission required, please submit a shareable link to your video file (use google drive, OneDrive, YouTube, etc.), do not submit the video file itself.

Part 1

This part explains a sample solution to one of the process control questions used in the midterm.

The question asked a parent process p to create two child processes c1 and c2. C1 creates its own child process c3 and c2 creates its own child process c4, as illustrated in the following diagram.



Please watch the video I prepared.

Part 2

(one of the questions below requires submission at the end of your lab session)

The sample solution code used in the video is also attached with the lab handout (lab6.c). Answer the following questions.

1. Try deleting the source code at line 31, i.e. “ else exit(11);”

What would happen to the output of the program? Could you explain the output? If you could, you have fully understood how fork works.

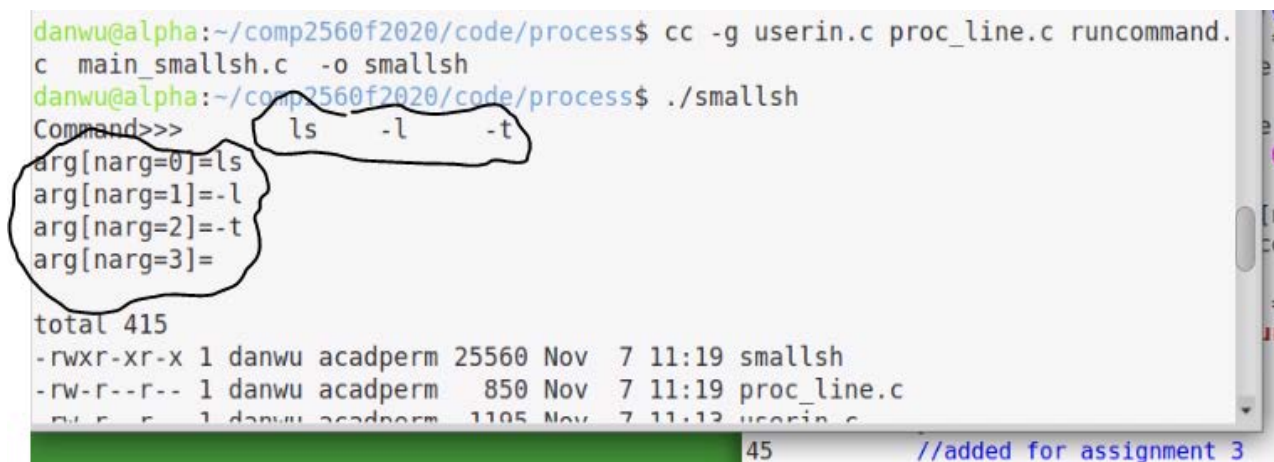
Record a video with audio showing you explaining the output as I did in my video using a simple diagram. **Submit a shareable link of your video. Please state at the beginning of the video which lab section you are in.**

2. Use the wait function, make changes to the lab6.c file so that each parent process waits for its child process to terminate before it terminates itself. For example, c1 waits for c3 to terminates before c1 terminates itself. **Submit your code with the script files as usual. Put in comments in your code to show where you use the wait function and who waits for whom. In your code, please put in your name and lab section no. as comments.**

3. Study the smallsh program we discussed in class (userin.c, proc_line.c, smallsh.h, runcommand.c, and main_smallsh.c) whose source is posted under week 8 in BB.

In order to fully understand how the user inputted command and its command line arguments are parsed as individual tokens (for example, the user typed " ls -l -t ", and the program will figure out the tokens in this input as "ls", "-l", and "-t").

Add some "printf (...)" statements at the appropriate places so that the output of the program looks as what the following screenshot shows.



```
danwu@alpha:~/comp2560f2020/code/process$ cc -g userin.c proc_line.c runcommand.c main_smallsh.c -o smallsh
danwu@alpha:~/comp2560f2020/code/process$ ./smallsh
Command>>> ls -l -t
total 415
-rwxr-xr-x 1 danwu acadperm 25560 Nov  7 11:19 smallsh
-rw-r--r-- 1 danwu acadperm  850 Nov  7 11:19 proc_line.c
-rw-r--r-- 1 danwu acadperm 1105 Nov  7 11:13 userin.c
45 //added for assignment 3
```

Where and how to add the printf statements demonstrate your understanding of this program.

Submit the following

- A. A text file explaining where you add the printf statements.
- B. The source code involved in which you added the printf statements.
- C. Script files showing your compiling and running the program which shows a similar result as the screenshot above.

In the text file and source code, please put in your name and lab section no.