

Midterm

- Put your name on all sheets.
- No books, notes or other type of written material are allowed.
- Mobile phones are not allowed.
- Show your work neatly for partial credit.
- You must show your work to receive any credit.
- Max points: 100 + 4 bonus.
- You can use the backside of the sheets.
- Do not get distracted by answering unnecessary questions.
- Read the statement below and sign your name.
- Use your laptop. You are not allowed to use anything else than your IDE except:
<http://man7.org/linux/man-pages/man2/mmap.2.html> and <https://linux.die.net/man/2/munmap>

I affirm that I neither will give nor receive unauthorized assistance on this examination. All the work that appears on the following pages is entirely my own.

Name:

Signature:

| Total points | Letter grade |
|--------------|--------------|
| | |

Points

Task 0: Read this

This is a sample midterm. It contains more work than the real one for practice purposes.

Points

Task 1: piping-overwrite stdin

Have a parent and his 4 children processes. The parent reads from the keyboard. All inputs should be printed on the screen.

Each children is in an infinity loop, waiting for some time between 1 and 10 seconds to change the color of the print (see below). Change the color, by signaling the parent to stop scanning, but instead print the color change from the child. For that, the child processes signal the parent to overwrite stdin and send the print color information per pipe. Leave the program with “q” and don’t forget to signal the childs to terminate/kill. Wait for the child!

Following code is just an example. Do not print “red” or “green”, just the color change. Stay in the same line when doing this and restore the read!

```
#define KNRM  "\x1B[0m"
#define KRED  "\x1B[31m"
#define KGRN  "\x1B[32m"
#define KYEL  "\x1B[33m"
#define KBLU  "\x1B[34m"
#define KMAG  "\x1B[35m"
#define KCYN  "\x1B[36m"
#define KWHT  "\x1B[37m"

int main()
{
    printf("%sred\n", KRED);
    printf("%sgreen\n", KGRN);
    printf("%syellow\n", KYEL);
    printf("%sblue\n", KBLU);
    printf("%smagenta\n", KMAG);
    printf("%scyan\n", KCYN);
    printf("%swhite\n", KWHT);
    printf("%snormal\n", KNRM);

    return 0;
}
```

Task 1-2: piping-overwrite stdin

Have a parent and two children processes. The parent reads from the keyboard. All inputs should be printed on the screen. The two children are in an infinite loop for 5 seconds and 11 seconds. Each child overwrites stdin and prints the actual time in a different color. Reset this color then and go back into reading from the parent process. For that, overwrite stdin and send the string. Leave the program with “q” and don’t forget to signal the childs to terminate/kill. Wait for the child!

Task 2: pipes

Have a parent and a child process. The parent reads from the keyboard. All inputs should be printed on the screen. Also, send all inputs per pipe to the child. All 5 seconds, the child should read the pipe and saves the content to a (one and the same) file. In the end, the file will contain all inputs of the parent. Leave the program with “q” and don’t forget to signal the child to 1. Close the file first and 2. Terminate/kill. Wait for the child!

Task 3: named shared memory

Write a program which generates (creates) a named shared memory “write”. Scan from the keyboard and write into it (strcpy and cat) and print as well on the terminal.

Write another program “listen” which opens this named shared memory. Every 2 seconds, check if there is content in the shared memory. If so, print it and reset the shared memory content (first element to null). Print a new line.

If a “q” is received, end the “listen” program as well as the “write” program. Prevent the “listen” program as good as you can to be terminated by other means.

There is a danger of race-conditions here which can be overcome with a mutex, but we will ignore that. So no mutex.

CSC 357 – Name: _____

Task 4: answer the 20 questions on PolyLearn

Points

The 20 questions will be from the pool of all lecture quizzes so far. Except the “nothing to lose” ones.