Jonathan Carreiro

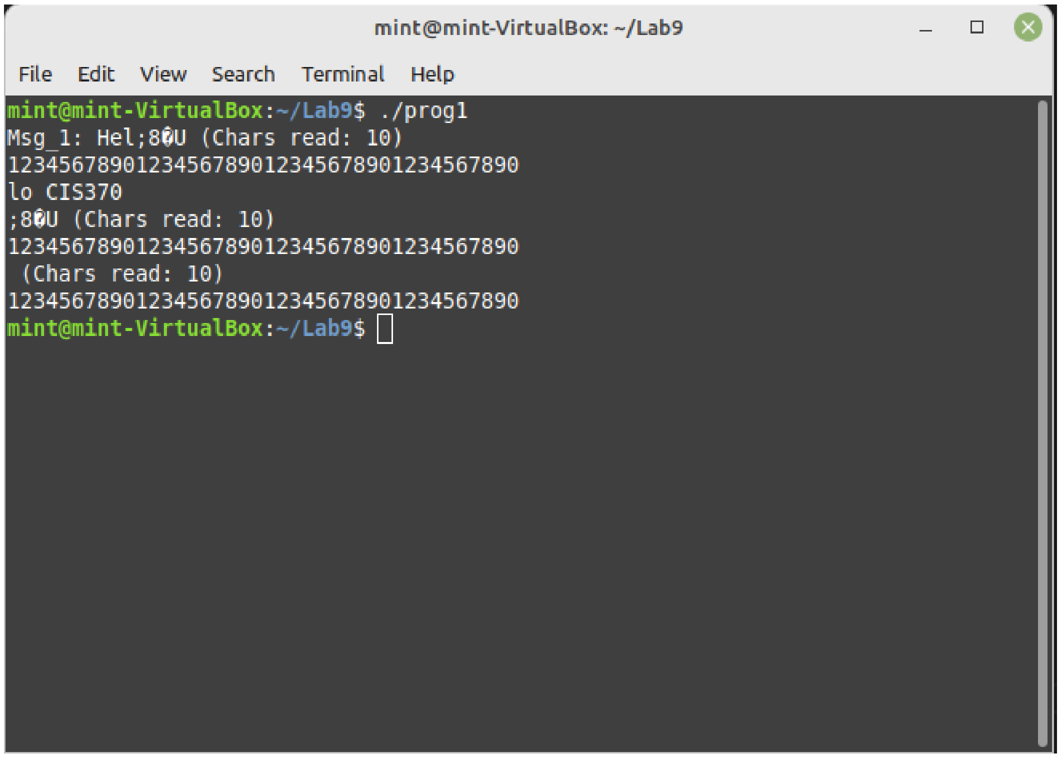
Lab 9

Part 1:

a. 32 bytes are written to the upstream end of the pipe

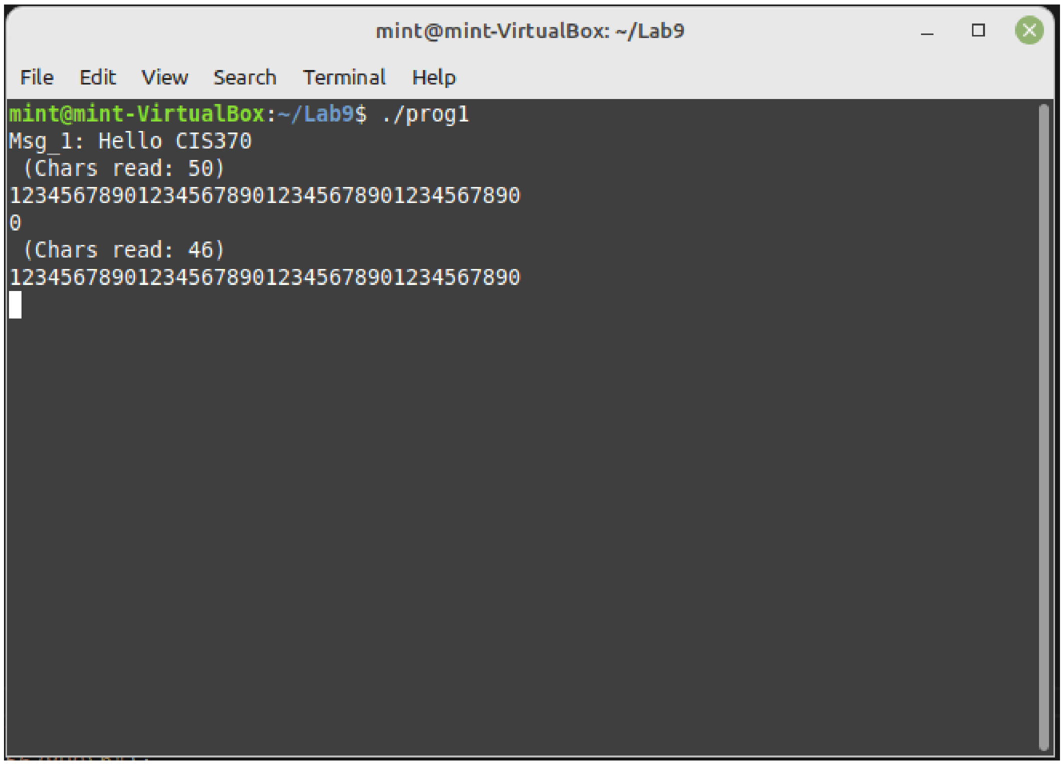
b. 32 bytes are read from the downstream end of the pipe

c. Below is the output I got after changing the number of bytes to read from 32 to 10. It looks like its reading the first 10 bytes of the first message followed by some junk characters (not sure why those are there, maybe because one end of the pipe is never closed?), then the next 10 followed by more junk characters, then it reads nothing because the message is empty.


mint@mint-VirtualBox: ~/Lab9
File Edit View Search Terminal Help

```
mint@mint-VirtualBox:~/Lab9$ ./prog1
Msg_1: Hel;8�U (Chars read: 10)
1234567890123456789012345678901234567890
lo CIS370
;8�U (Chars read: 10)
1234567890123456789012345678901234567890
 (Chars read: 10)
1234567890123456789012345678901234567890
mint@mint-VirtualBox:~/Lab9$ 
```

d. Below is the output I got from changing the number of bytes read from 32 to 50. The program never exits. I assume this is because the pipe is empty, causing the read to get blocked until a message is entered into it.


mint@mint-VirtualBox: ~/Lab9
File Edit View Search Terminal Help

```
mint@mint-VirtualBox:~/Lab9$ ./prog1
Msg_1: Hello CIS370
 (Chars read: 50)
12345678901234567890123456789012345678900
0
 (Chars read: 46)
1234567890123456789012345678901234567890
```

Part 2:

a. When I comment out line 30, the process never terminates. If I had to guess, I would say that this is because the while loop on line 19 needs to read from the pipe to know if charCount is 0. Since the upstream end of the pipe is never closed, if the pipe is empty then the read gets blocked and cannot return 0 as charCount.

b. After moving line 31 before(?) line 27, the program prints nothing and gets trapped. I think this is for a similar reason as the previous question. The parent process starts waiting for the child before writing to the pipe. This causes the read to get blocked, and since the parent is waiting, the read will stay blocked forever.