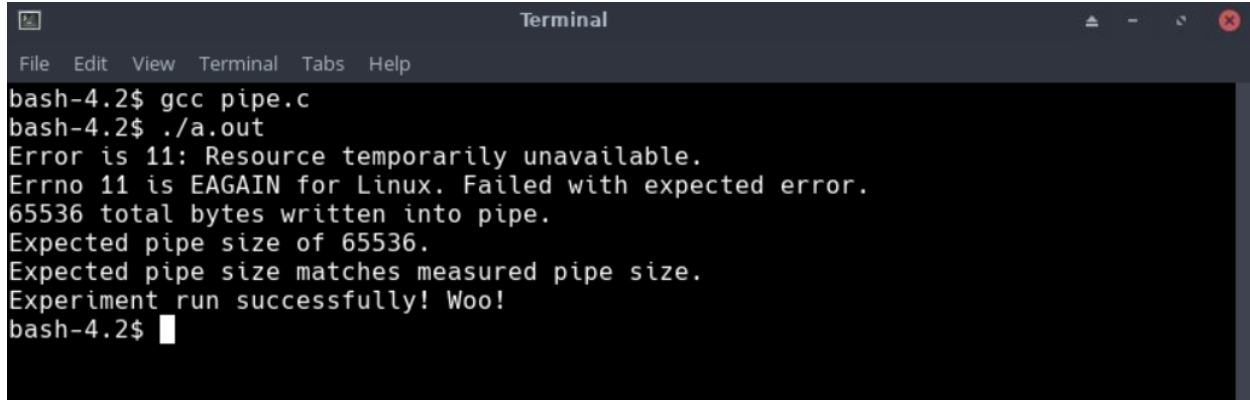


Joya Debi, Lucia Rhode  
ECE 357  
HW 3 Problems 3 & 4  
Sample Run  
11/12/21

#### Problem 3:

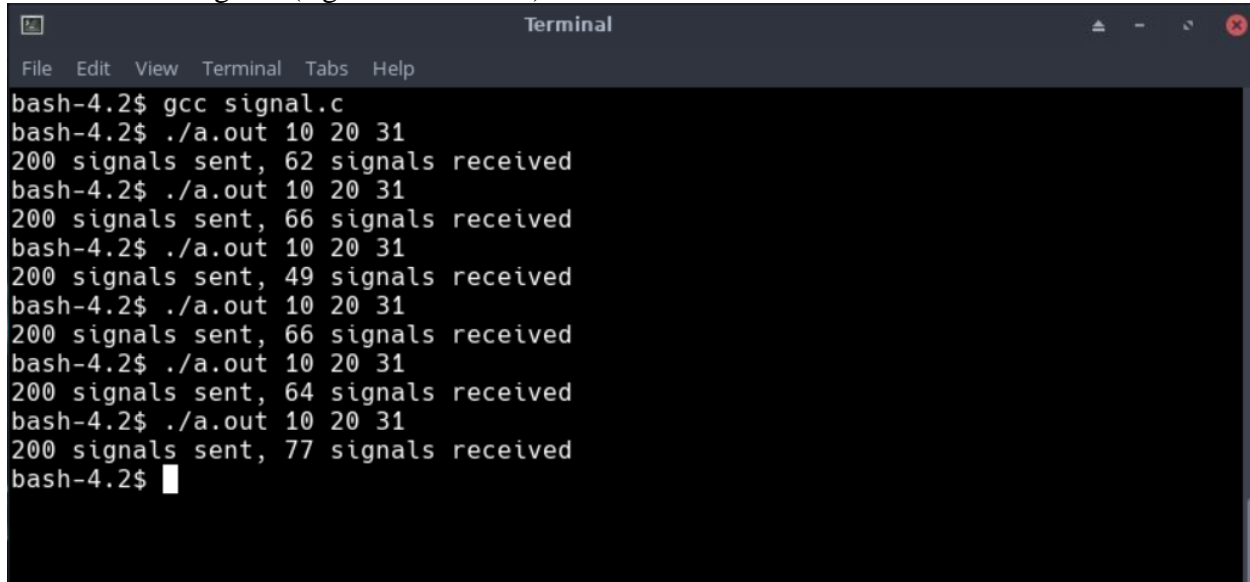
A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt is "bash-4.2\$". The user enters "gcc pipe.c" and then "./a.out". The program outputs: "Error is 11: Resource temporarily unavailable. Errno 11 is EAGAIN for Linux. Failed with expected error. 65536 total bytes written into pipe. Expected pipe size of 65536. Expected pipe size matches measured pipe size. Experiment run successfully! Woo!". The prompt returns to "bash-4.2\$".

```
bash-4.2$ gcc pipe.c
bash-4.2$ ./a.out
Error is 11: Resource temporarily unavailable.
Errno 11 is EAGAIN for Linux. Failed with expected error.
65536 total bytes written into pipe.
Expected pipe size of 65536.
Expected pipe size matches measured pipe size.
Experiment run successfully! Woo!
bash-4.2$
```

#### Problem 4:

The first number is the number of “killer” children, then the number of signals each child sent, and finally the signal number.

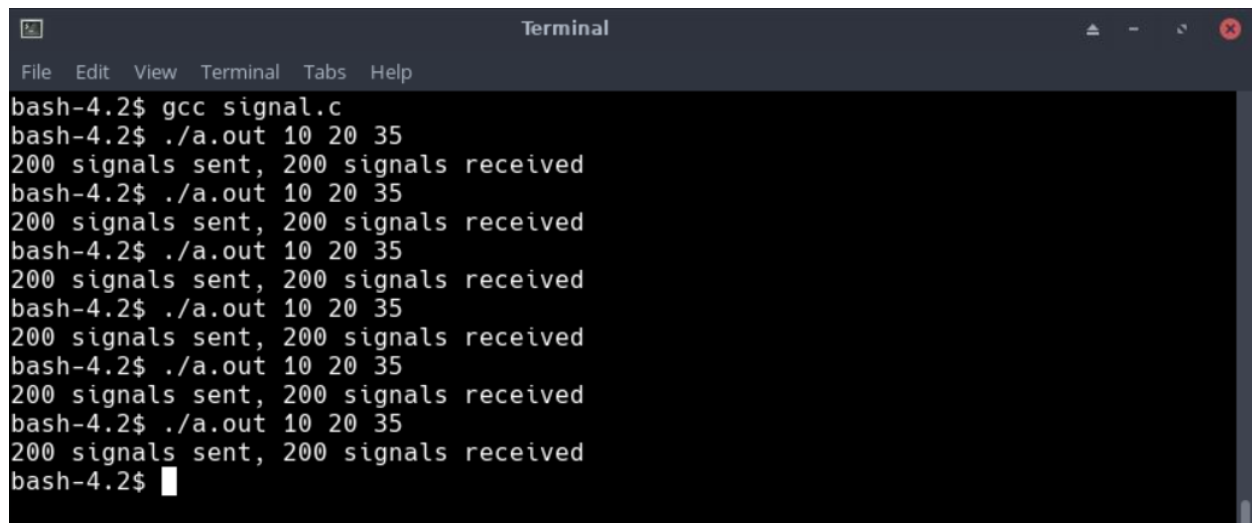
#### Non-real time signals (signal number<32)

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt is "bash-4.2\$". The user enters "gcc signal.c" and then runs the program multiple times with arguments "10 20 31". Each run outputs: "200 signals sent, [number] signals received". The received counts are 62, 66, 49, 66, 64, and 77. The prompt returns to "bash-4.2\$".

```
bash-4.2$ gcc signal.c
bash-4.2$ ./a.out 10 20 31
200 signals sent, 62 signals received
bash-4.2$ ./a.out 10 20 31
200 signals sent, 66 signals received
bash-4.2$ ./a.out 10 20 31
200 signals sent, 49 signals received
bash-4.2$ ./a.out 10 20 31
200 signals sent, 66 signals received
bash-4.2$ ./a.out 10 20 31
200 signals sent, 64 signals received
bash-4.2$ ./a.out 10 20 31
200 signals sent, 77 signals received
bash-4.2$
```

We see that because of job scheduling, each time while 200 signals are sent, only a random number of signals less than 200 are received.

#### Real time signals (signal number>=32)

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Tabs, Help) and standard window controls. The terminal shows a series of commands and their outputs. The first command is `gcc signal.c`. Then, there is a loop of executing `./a.out 10 20 35`, each time followed by the output `200 signals sent, 200 signals received`. The loop ends with a prompt `bash-4.2$` and a cursor.

```
bash-4.2$ gcc signal.c
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$ ./a.out 10 20 35
200 signals sent, 200 signals received
bash-4.2$
```

We see that for a real time signal, every signal that is sent is received.