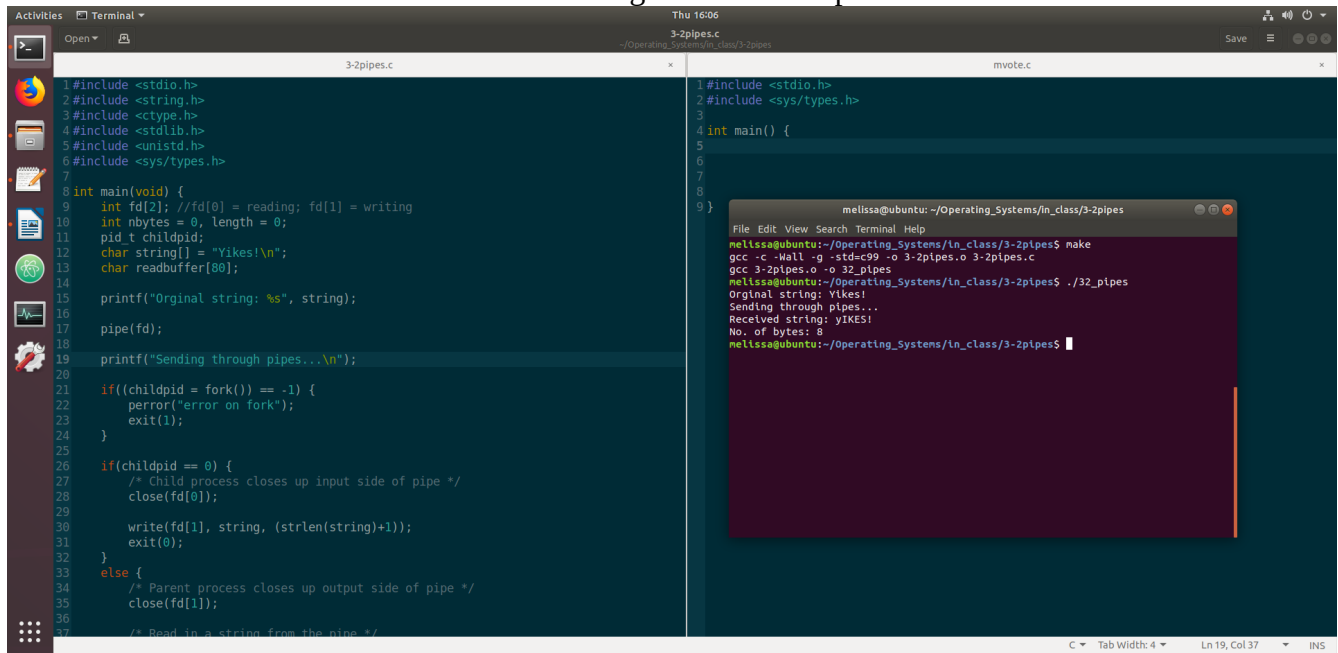


In-class Assignment – 3-2 Pipes



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 #include <stdlib.h>
5 #include <unistd.h>
6 #include <sys/types.h>
7
8 int main(void) {
9     int fd[2]; //fd[0] = reading; fd[1] = writing
10    int nbytes = 0, length = 0;
11    pid_t childpid;
12    char string[] = "Yikes!\n";
13    char readbuffer[80];
14
15    printf("Original string: %s", string);
16
17    pipe(fd);
18
19    printf("Sending through pipes...\n");
20
21    if((childpid = fork()) == -1) {
22        perror("error on fork");
23        exit(1);
24    }
25
26    if(childpid == 0) {
27        /* Child process closes up input side of pipe */
28        close(fd[0]);
29
30        write(fd[1], string, (strlen(string)+1));
31        exit(0);
32    }
33    else {
34        /* Parent process closes up output side of pipe */
35        close(fd[1]);
36
37        /* Read in a string from the pipe */
```

```
1 #include <stdio.h>
2 #include <sys/types.h>
3
4 int main() {
5
6
7
8
9 }
```

```
melissa@ubuntu: ~/Operating_Systems/in_class/3-2pipes$ make
gcc -c -Wall -g -std=c99 -o 3-2pipes.o 3-2pipes.c
gcc 3-2pipes.o -o 32_pipes
melissa@ubuntu: ~/Operating_Systems/in_class/3-2pipes$ ./32_pipes
Original string: Yikes!
Sending through pipes...
Received string: yIKES!
No. of bytes: 8
melissa@ubuntu: ~/Operating_Systems/in_class/3-2pipes$
```