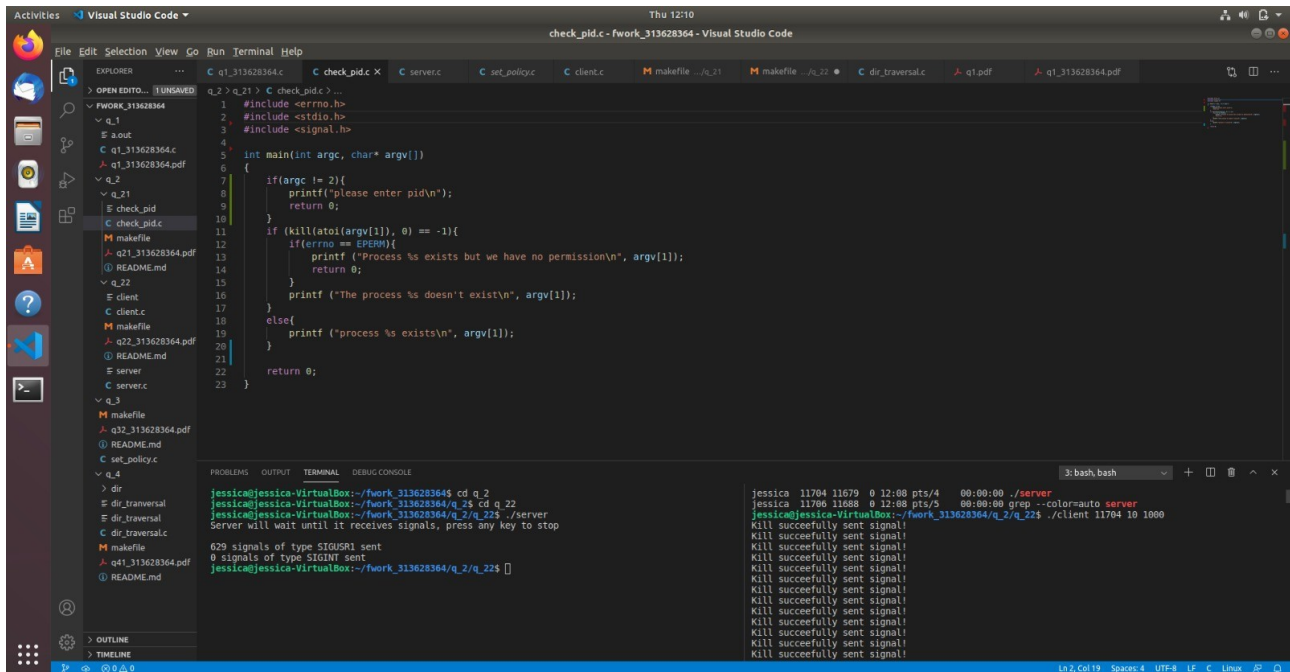


Question 2.2:

1 – This picture shows an example of running the client and server programs together and sending 1000 signals to the server from the client. After sending these you can see that the server did not receive all 1000 signals.



The screenshot shows a Visual Studio Code editor with a C program named `check_pid.c` open. The program is designed to handle signals. It includes `<errno.h>` and `<signal.h>`. The `main` function takes an argument `argc` and a character array `argv`. It checks if `argc` is greater than 2. If so, it prints "please enter pid\n" and returns 0. If `kill` fails (returns 0), it prints "Process %s exists but we have no permission\n", `argv[1]` and returns 0. If `kill` succeeds (returns non-zero), it prints "The process %s doesn't exist\n", `argv[1]`. Otherwise, it prints "process %s exists\n", `argv[1]`. The program then returns 0.

The terminal output shows the execution of the program. It starts with a prompt `jessica@jessica-VirtualBox:~/fwork_313628364$ cd q_2`. Then, it runs `cd q_2` and `./server`. The server output shows "Server will wait until it receives signals, press any key to stop". The client output shows "629 signals of type SIGUSR1 sent" and "0 signals of type SIGINT sent". The client then runs `./client`, which sends 1000 signals. The server output shows "Kill successfully sent signal!" repeated 1000 times.

2 – Real time signals vs. Standard signals

In standard signals some of the signals get lost out while in real time signals no signals get thrown out, they are saved in a queue in real time signals. In standard signals their numbers are known ahead of time (a number for each signal) and on the other hand real time signals, their numbers are not always known ahead of time and to find them we need to check in the program for them.