GEBZE TECHNICAL UNIVERSITY
CSE344 – MIDTERM DOCUMENTATION

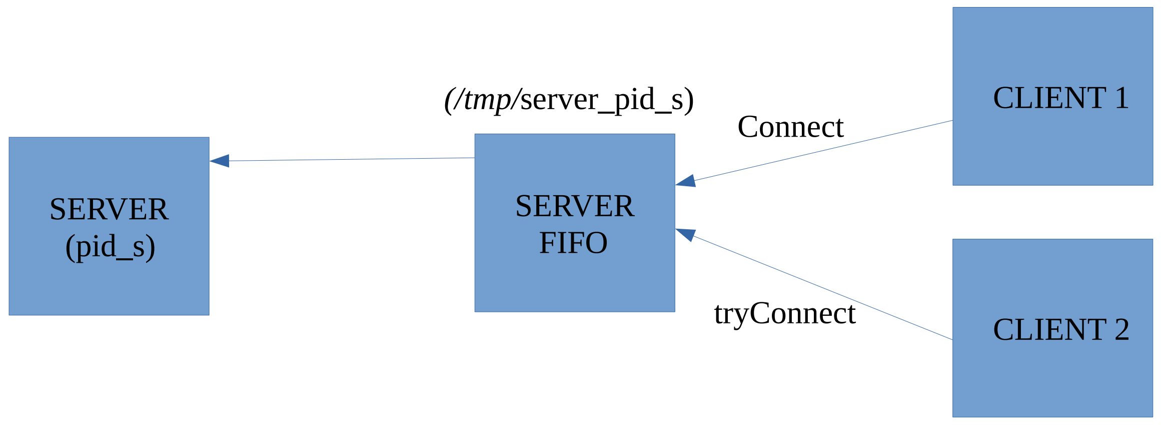
Student Name: Emre Oytun
Student Number: 200104004099

1) System Design and Decisions:

- In this system, all communication is done using FIFOs.
- Server's main work directory is always "/tmp" folder.

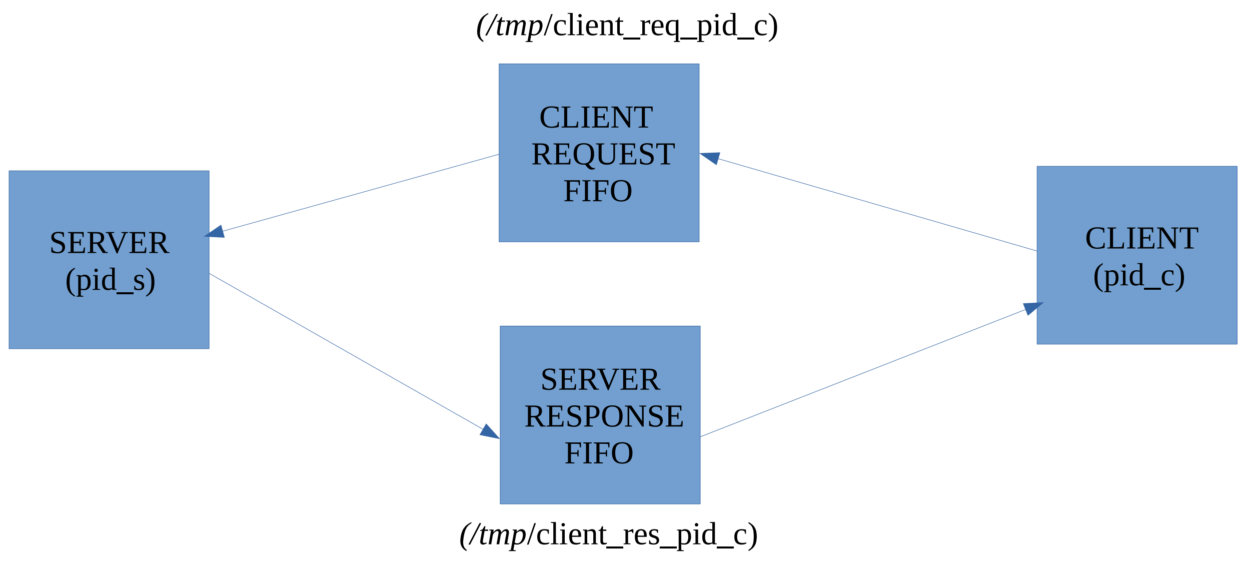
1.a) Server – Client First Connection Design:

- When server is started, it makes a FIFO under "/tmp" folder and prompts its PID.
- Clients can send connection request to server by sending their PID and connect type to this FIFO.
- Server then reads these requests one-by-one and evaluates if they can connect and waits if they need to wait.



1.b) Server – Client Bidirectional Communication Design:

- When client is connected to the server, there are 2 FIFOs for communication.
- One FIFO is for client requests, and another one is for server responses.
- Server waits request from client by trying to read the client request FIFO.
- Client then waits response from server by trying to read the server response FIFO.
- These FIFOs are also under "/tmp" folder.



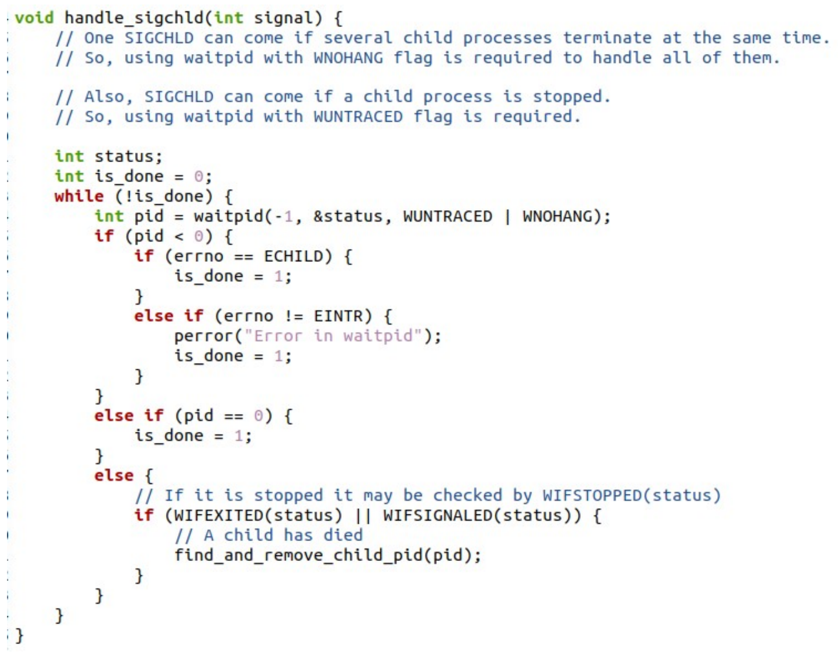
1.c) Synchronization for File Operations:

- Synchronization for file operations are mainly provided by using “fcntl write locks”.
- When a child process of the server wants to access to a file, it needs to get the lock first no matter if it’s read or write so that there is no race condition between operations.

1.d) Synchronization Between Server’s Parent and Child Processes:

- Server needs to know how many child processes are alive so that it can keep the counter and make check for new connection request.
- This is provided by handling SIGCHLD signal. When a child process terminates, the OS send SIGCHLD signal to the parent so the parent can decrement its counter.
- There is no race condition for the counter because we have only one thread in the server’s parent process and when SIGCHLD signal handler is working, the main thread has to wait. However, to prevent any miss read of counter, SIGCHLD signal is masked when a new connection request is read and unmask when connection check is done.

SIGCHLD Signal Handler:



```c
void handle_sigchld(int signal) {
    // One SIGCHLD can come if several child processes terminate at the same time.
    // So, using waitpid with WNOHANG flag is required to handle all of them.

    // Also, SIGCHLD can come if a child process is stopped.
    // So, using waitpid with WUNTRACED flag is required.

    int status;
    int is_done = 0;
    while (!is_done) {
        int pid = waitpid(-1, &status, WUNTRACED | WNOHANG);
        if (pid < 0) {
            if (errno == ECHILD) {
                is_done = 1;
            }
            else if (errno != EINTR) {
                perror("Error in waitpid");
                is_done = 1;
            }
        }
        else if (pid == 0) {
            is_done = 1;
        }
        else {
            // If it is stopped it may be checked by WIFSTOPPED(status)
            if (WIFEXITED(status) || WIFSIGNALED(status)) {
                // A child has died
                find_and_remove_child_pid(pid);
            }
        }
    }
}
```

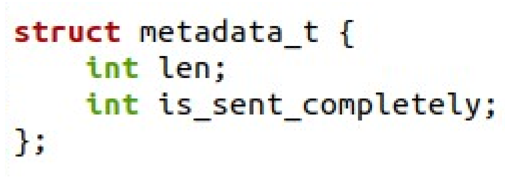
- When a SIGCHLD signal arrives, signal handler waits in a loop until there is no SIGCHLD signal remained because SIGCHLD signals can arrive at the same time. It does this by checking if there is no child remained, or waitpid result is 0.
- When a child pid is acquired, it finds this pid in the child pids array and removes. This array later is used to send kill signals to the child processes.

1.e) FIFO Long Message Sending/Reading Design:

- There is a need to design a system that can handle long messages for upload, download and archFile commands.
- For this purpose, I designed a message system that when server or client waits a message from another, they need to read the message in a loop.

- In this loop, they first get the metadata of the message which contains the message length and the status.
- Message length indicates that we need to read until this size, because the other part is writing that long message.
- Message status indicates whether message transmission is done or we need to continue reading.

Metadata Struct:



```c
struct metadata_t {
    int len;
    int is_sent_completely;
};
```

- This provides a way of tackling the exceeding FIFO/PIPE max length problem.

1.f) Signal Handling:

- There are some signal handlers set in server and client processes.

SIGCHLD:

- As I mentioned above, the server process has SIGCHLD handler for reaping the child PIDs.
- It provides a way of keeping the counter update and preventing zombie processes.

SIGINT – SIGTERM:

- Both server processes and client process have signal handlers for SIGINT and SIGTERM.
- When these signals are arrived:
* Opened files are closed.
* Opened FIFOs are closed.
* Temp files are closed and unlinked if any.
* SIGTERM signal is sent to child processes and child processes are waited for reaping up. So, child processes are closed if any.

1.g) Error Handling:

- Almost all system calls are wrapped up in a if condition to check if there is a problem with the system call.
- If the errno is set to EINTR or EAGAIN the system call is called again. This provides a way preventing interruptions in slow system calls like read/write.
- I/O operations are written in the “utility.c” by wrapping up them all to prevent interruptions. In this way, I prevented code duplication also.

2) Important Notes for Usage:

2.1) Input Constraints:

- To prevent exceeding the buffers and unintended results there are some restrictions on the inputs.

- Directory name can have at most 924 characters.
- Client commands can have at most 5095 characters.
- File names and strings in the client commands can have at most what it is remained from the max client command size above which is 5095.
- The max connection can be max 1024 since we need to keep an array of child pids to send kill signals to them afterwards.

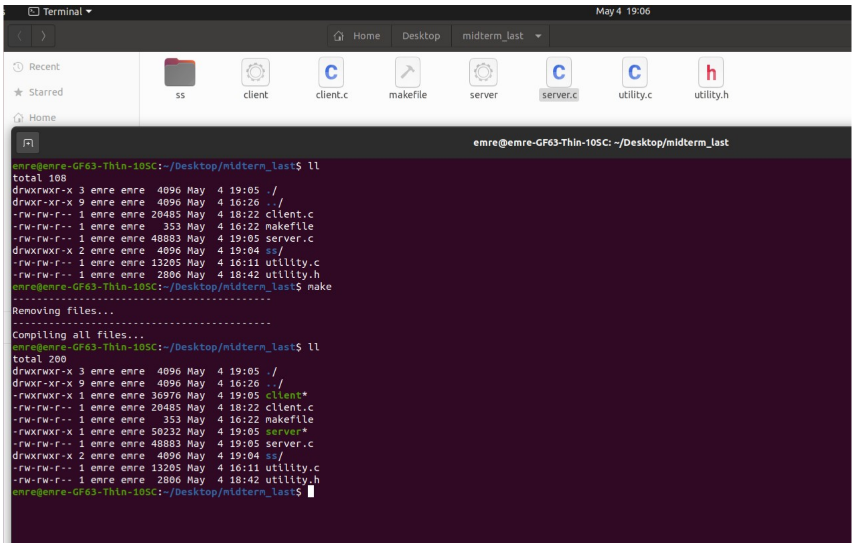
2.2) archServer Design:

- From the midterm documentation PDF, I concluded that we need to download the server side files into the client side since it says “downloading” after command in the PDF.
- So, I downloaded the files into the client side and tar them using fork + exec and tar utility as expected. The final archive file is in the client side for this reason. It is not written that we need to upload the files to the server side again in the midterm PDF.

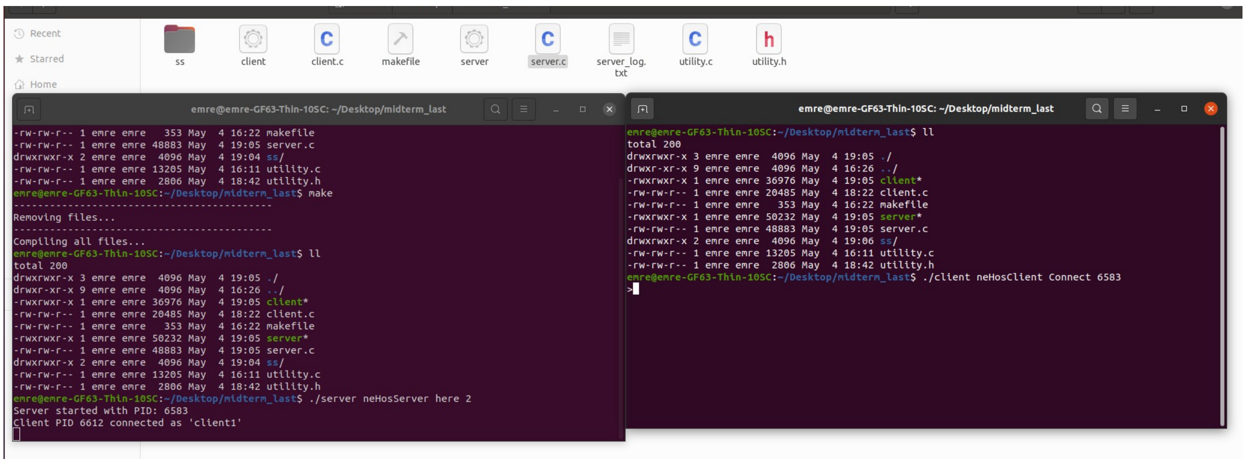
2.3) Compiling and Running the Server and Client:

Compiling:

- You can use make to compile the project as a whole as below. Two .out files will be generated. One of them is for server and one of them is for client. You can execute them by giving proper arguments.



```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 108
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```



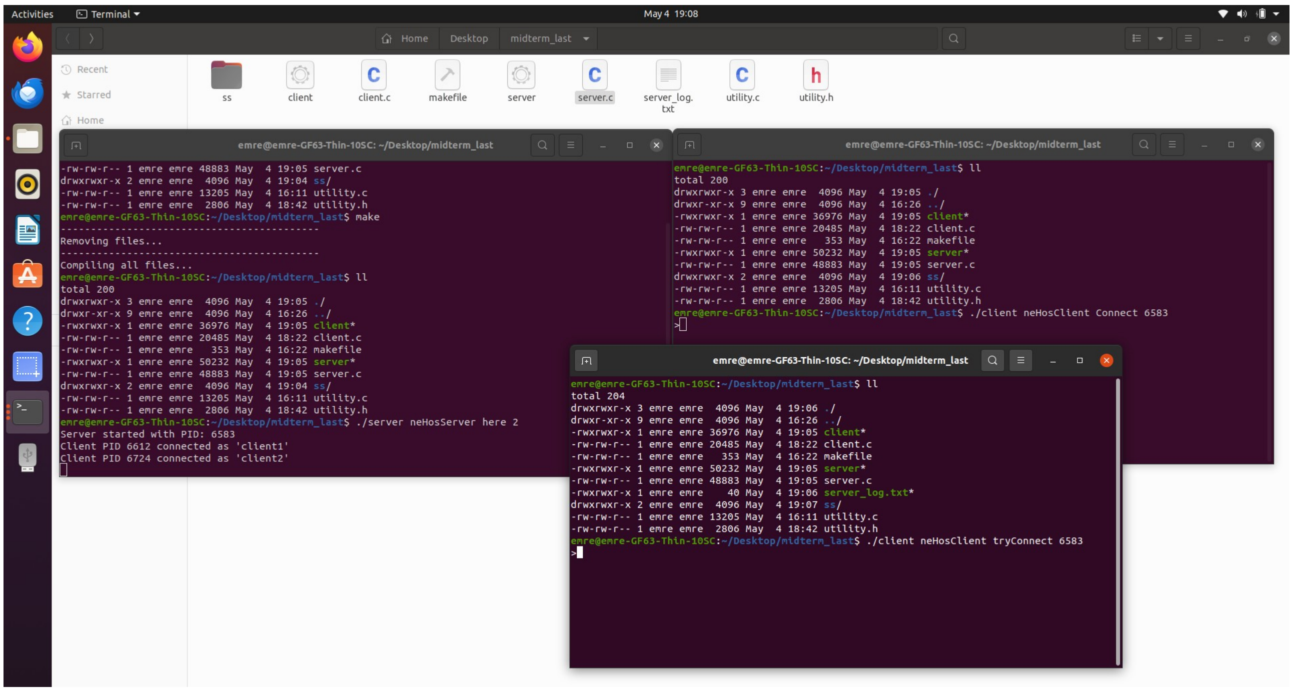
```
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

3) Test Cases and Results:

- You can find the test cases and their result screenshots and explanations below.

3.1) Connecting to server using “Connect” and “tryConnect” options:



```
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
------------------------------------------
Removing files...
------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
Client PID 6724 connected as 'client2'
```

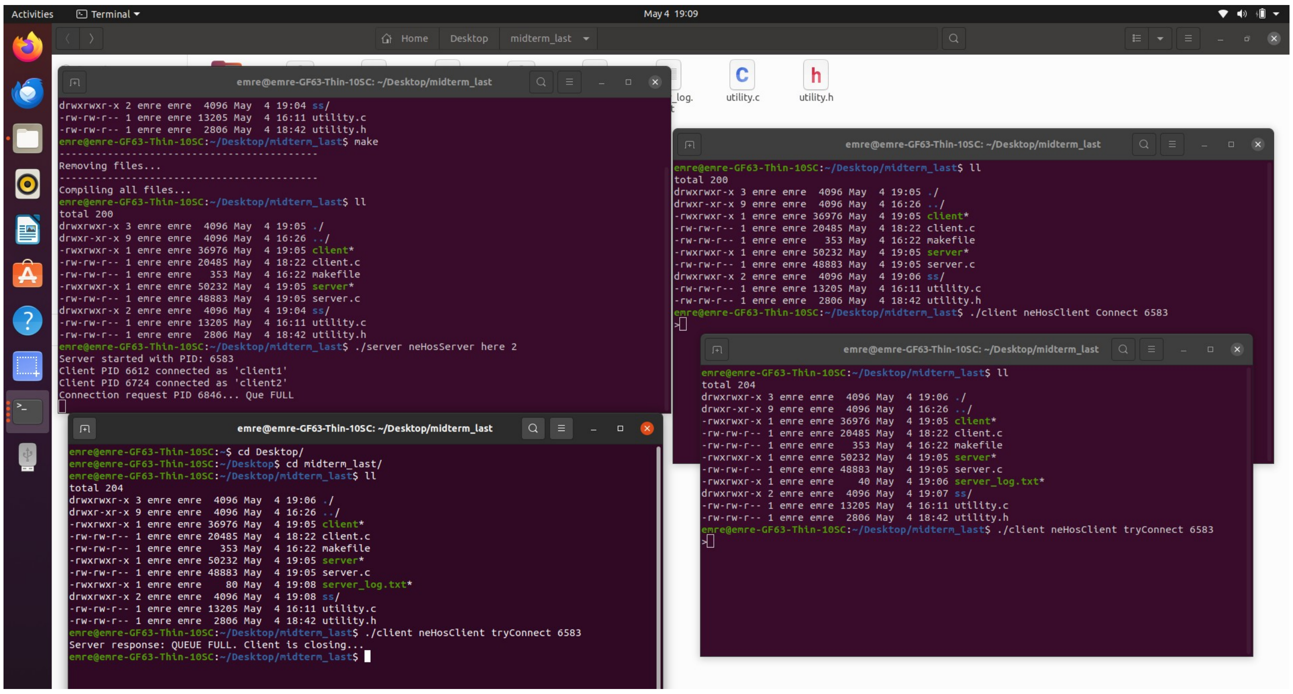
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    40 May  4 19:06 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:07 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
>
```

The leftmost one is server.
The right two ones are clients.

As it is seen from the screenshot, the server has 2 slots and 2 clients are connected to it now. One of them is connected using “Connect” and one of them is connected using “tryConnect”. Now, the client processes are waiting for input by prompting.

3.2) Trying to connect to server by “tryConnect” option when server is full:



```
emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
Client PID 6724 connected as 'client2'
Connection request PID 6846... Que FULL
```

```
emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last
emre@emre-GF63-Thin-10SC:~$ cd Desktop/
emre@emre-GF63-Thin-10SC:~/Desktop$ cd midterm_last/
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    80 May  4 19:08 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:08 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
Server response: QUEUE FULL. Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

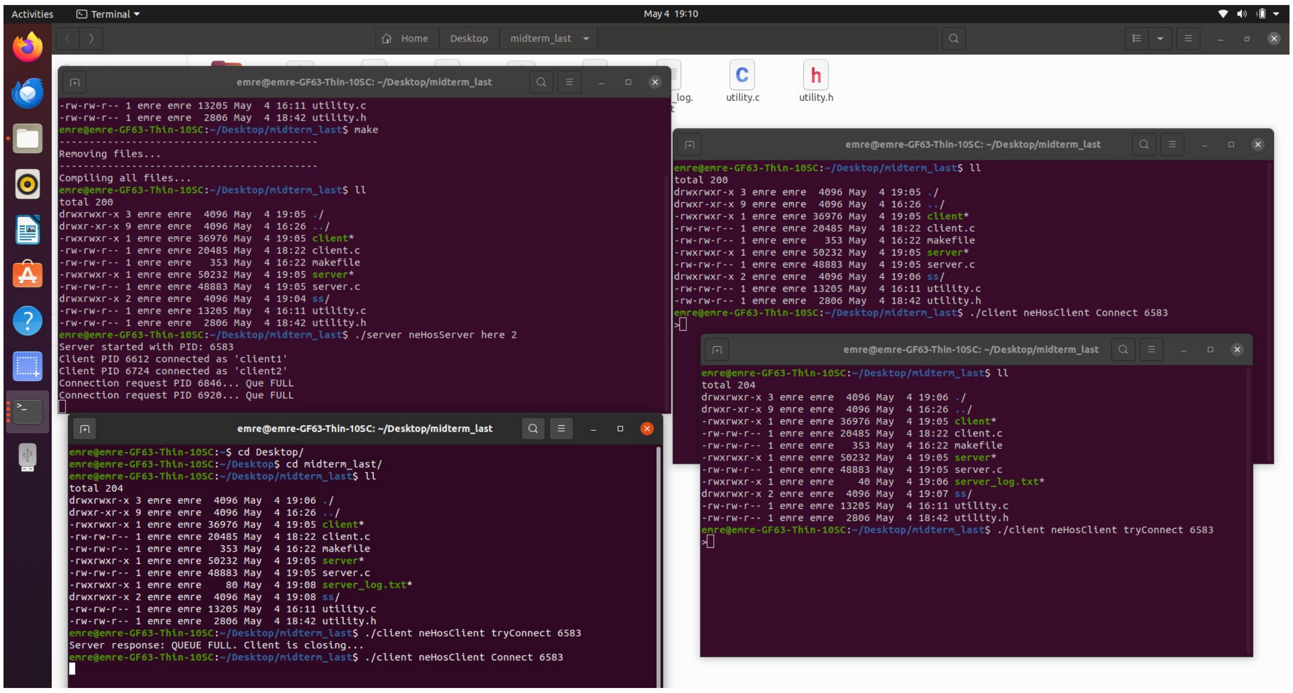
```
emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

```
emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    40 May  4 19:06 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:07 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
>
```

The top-left is server.
The right ones and the bottom-left are clients.

As it is seen from the screenshots, the bottom-left client is trying to connect to the server using “tryConnect”, but it can not connect because the server is full. It terminates the process immediately without waiting the queue.

3.3) Waiting to connect when the queue is full with “Connect” option:



```
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
Client PID 6724 connected as 'client2'
Connection request PID 6846... Que FULL
Connection request PID 6920... Que FULL
```

```
emre@emre-GF63-Thin-10SC:~$ cd Desktop/
emre@emre-GF63-Thin-10SC:~/Desktop$ cd midterm_last/
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    80 May  4 19:08 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:08 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
Server response: QUEUE FULL. Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
```

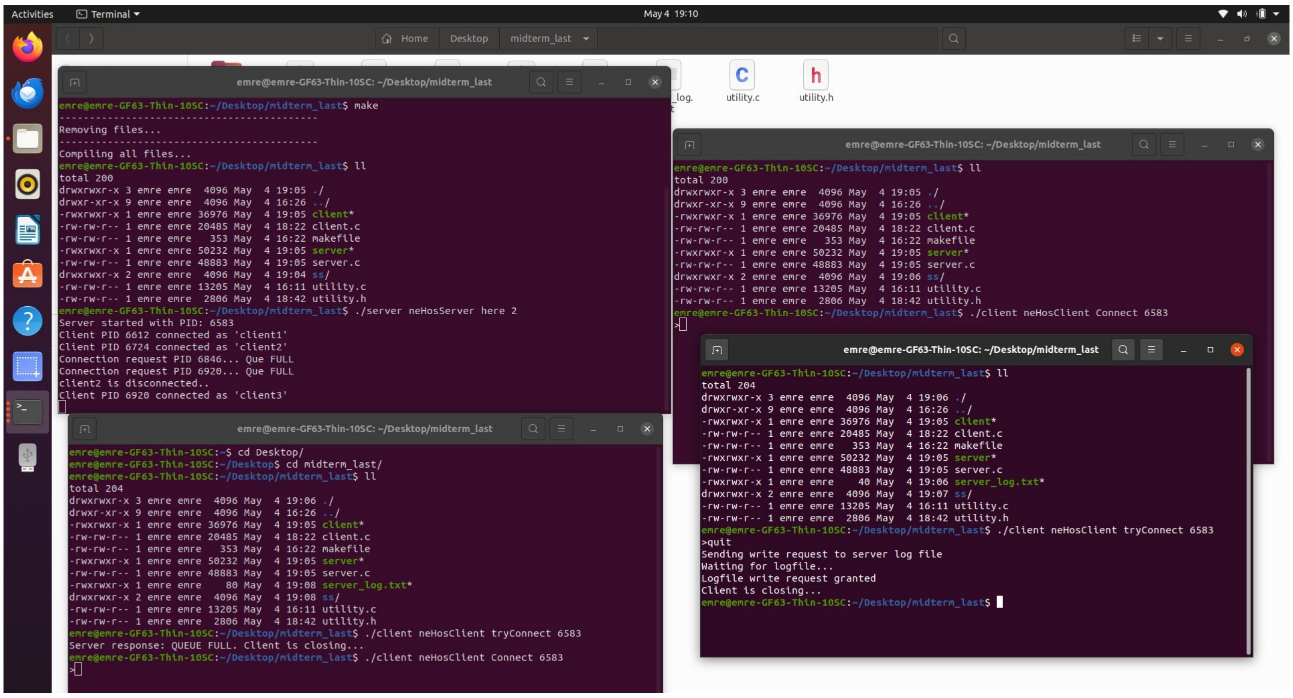
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    40 May  4 19:06 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:07 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
>
```

The top-left is server.
The right ones and the bottom-left are clients.

As it is seen from the screenshots, the bottom-left client is waiting for the queue since “Connect” option is used. In the next test case, we will see that it will connect to the server when one of the clients are disconnected so there is an available spot.

3.4) Connecting to the full server when one of the clients are disconnected by “Connect” option:



```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
Client PID 6724 connected as 'client2'
Connection request PID 6846... Que FULL
Connection request PID 6920... Que FULL
client2 is disconnected..
Client PID 6920 connected as 'client3'
```

```
emre@emre-GF63-Thin-10SC:~$ cd Desktop/
emre@emre-GF63-Thin-10SC:~/Desktop$ cd midterm_last/
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    80 May  4 19:08 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:08 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
Server response: QUEUE FULL. Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

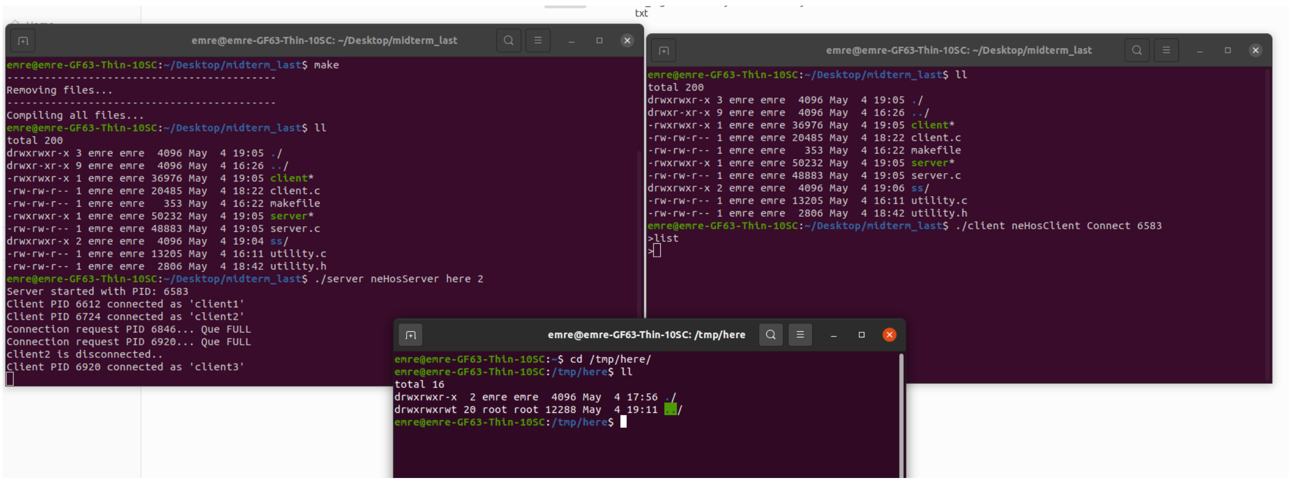
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 204
drwxrwxr-x 3 emre emre  4096 May  4 19:06 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre    40 May  4 19:06 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:07 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect 6583
>quit
Sending write request to server log file
Waiting for logfile...
Logfile write request granted
Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

The top-left is server.
The right ones and the bottom-left are clients.

As it is seen from the screenshots, the bottom-right client is disconnected and the waiting client in the bottom-left is connected to the server immediately.

3.5) List command when there is no file in the server side:



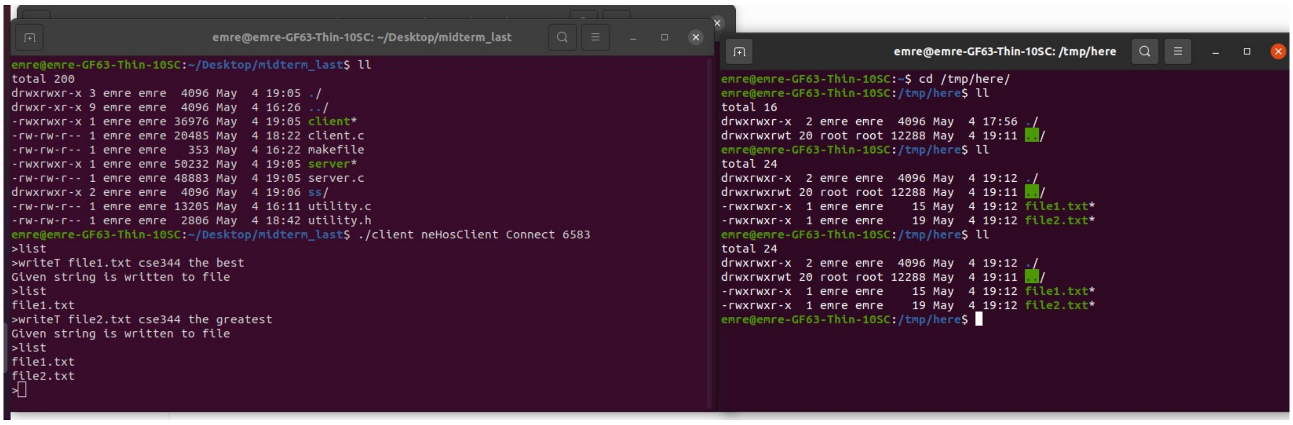
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ make
-------------------------------------------
Removing files...
-------------------------------------------
Compiling all files...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:04 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 6583
Client PID 6612 connected as 'client1'
Client PID 6724 connected as 'client2'
Connection request PID 6846... Que FULL
Connection request PID 6920... Que FULL
client2 is disconnected..
Client PID 6920 connected as 'client3'
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>list
>
```

```
emre@emre-GF63-Thin-10SC:~$ cd /tmp/here/
emre@emre-GF63-Thin-10SC:/tmp/here$ ll
total 16
drwxrwxr-x  2 emre emre  4096 May  4 17:56 ./
drwxrwxrwt 20 root root 12288 May  4 19:11 ../
emre@emre-GF63-Thin-10SC:/tmp/here$
```

When there is no file in the server side yet, the list command prints nothing.

3.6) “list” command when there are files in the server side and “write” command without line:



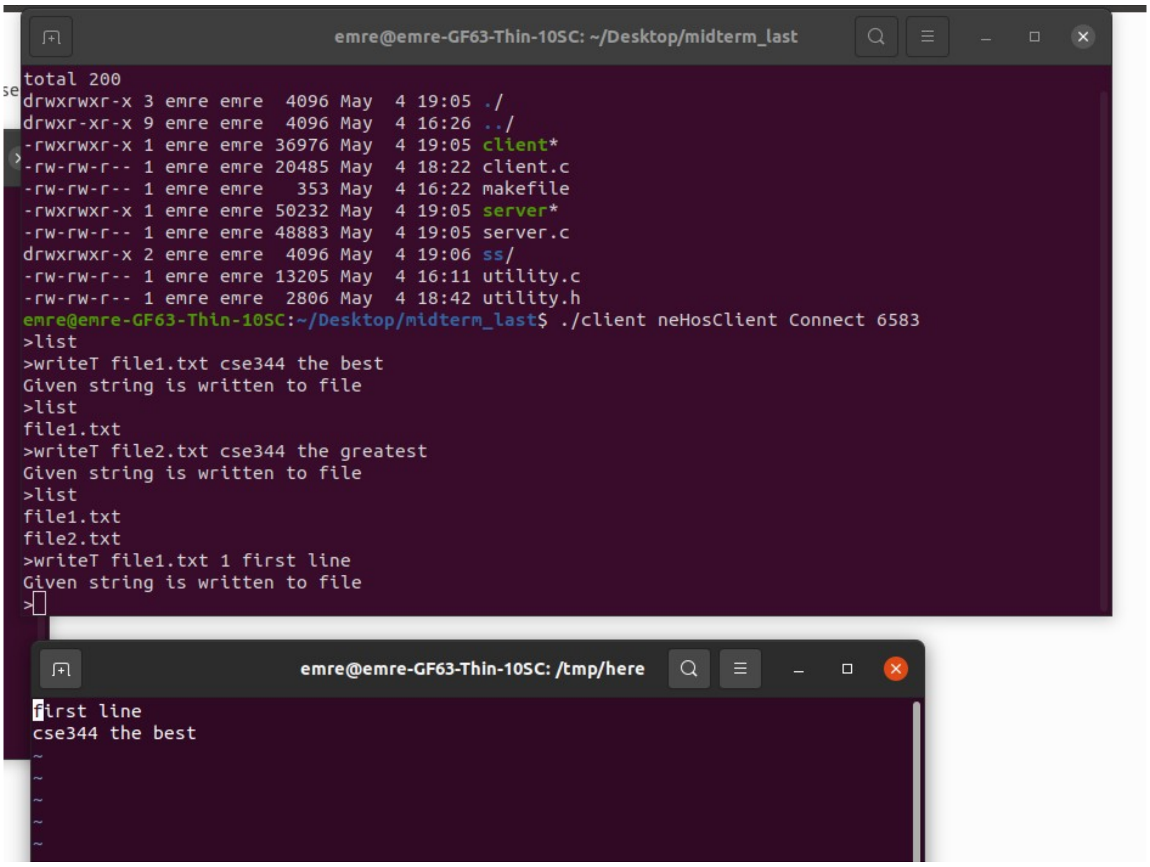
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>list
>writeT file1.txt cse344 the best
Given string is written to file
>list
file1.txt
>writeT file2.txt cse344 the greatest
Given string is written to file
>list
file1.txt
file2.txt
>
```

```
emre@emre-GF63-Thin-10SC:~$ cd /tmp/here/
emre@emre-GF63-Thin-10SC:/tmp/here$ ll
total 16
drwxrwxr-x  2 emre emre  4096 May  4 17:56 ./
drwxrwxrwt 20 root root 12288 May  4 19:11 ../
emre@emre-GF63-Thin-10SC:/tmp/here$ ll
total 24
drwxrwxr-x  2 emre emre  4096 May  4 19:12 ./
drwxrwxrwt 20 root root 12288 May  4 19:11 ../
-rwxrwxr-x  1 emre emre    15 May  4 19:12 file1.txt*
-rwxrwxr-x  1 emre emre    19 May  4 19:12 file2.txt*
emre@emre-GF63-Thin-10SC:/tmp/here$ ll
total 24
drwxrwxr-x  2 emre emre  4096 May  4 19:12 ./
drwxrwxrwt 20 root root 12288 May  4 19:11 ../
-rwxrwxr-x  1 emre emre    15 May  4 19:12 file1.txt*
-rwxrwxr-x  1 emre emre    19 May  4 19:12 file2.txt*
emre@emre-GF63-Thin-10SC:/tmp/here$
```

When we write some files to the server using writeT command and after enter list command, we can see the list of files in the server side.

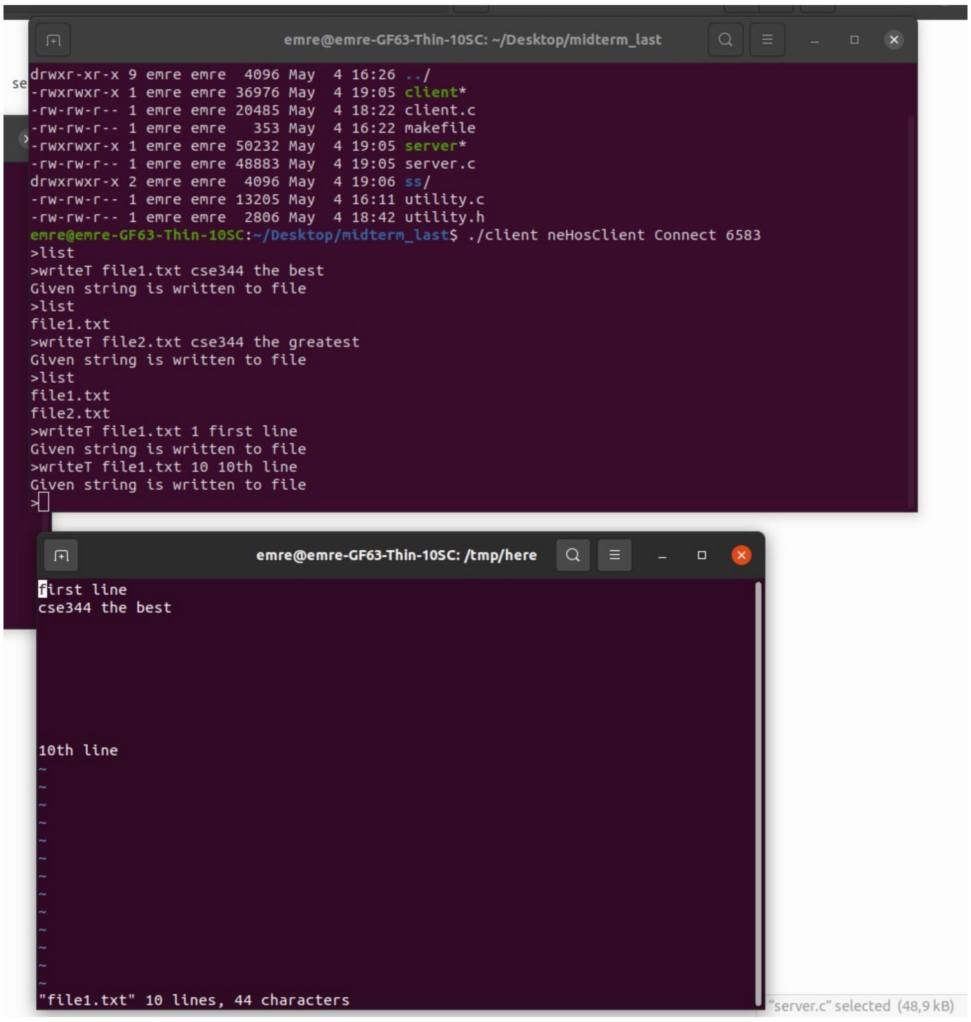
When there is no line specified in the command, writeT command writes the given string to the file by creating the file.

3.7) “writeT” command with line given:



```
total 200
drwxrwxr-x 3 emre emre  4096 May  4 19:05 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>list
>writeT file1.txt cse344 the best
Given string is written to file
>list
file1.txt
>writeT file2.txt cse344 the greatest
Given string is written to file
>list
file1.txt
file2.txt
>writeT file1.txt 1 first line
Given string is written to file
>
```

```
first line
cse344 the best
~
~
~
~
~
```



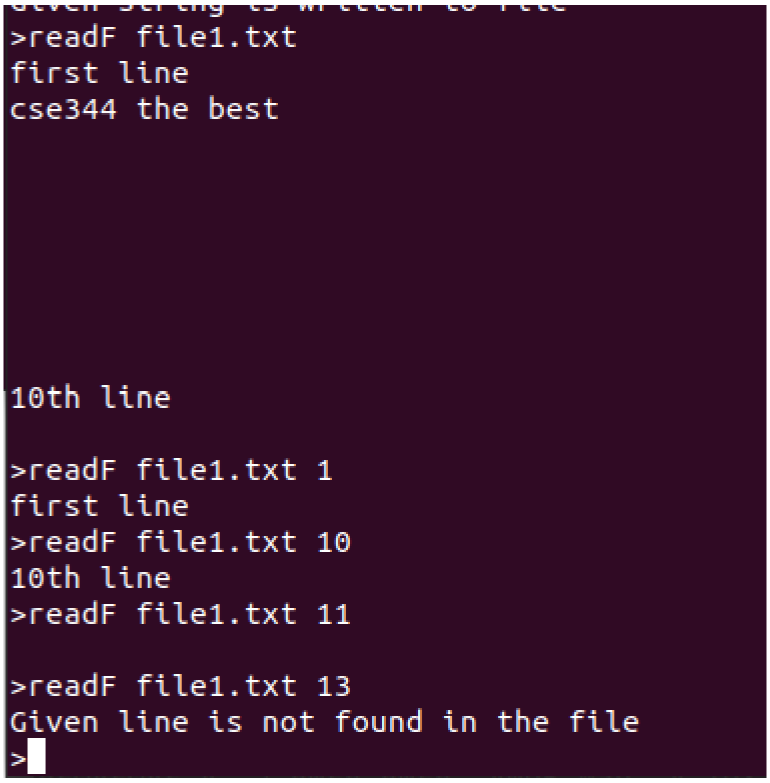
```
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
drwxrwxr-x 2 emre emre  4096 May  4 19:06 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 6583
>list
>writeT file1.txt cse344 the best
Given string is written to file
>list
file1.txt
>writeT file2.txt cse344 the greatest
Given string is written to file
>list
file1.txt
file2.txt
>writeT file1.txt 1 first line
Given string is written to file
>writeT file1.txt 10 10th line
Given string is written to file
>
```

```
first line
cse344 the best







10th line
~
~
~
~
~
~
~
~
~
~
~
~
~
"file1.txt" 10 lines, 44 characters
```

“server.c” selected (48,9 kB)

As we can see, we entered the command writeT to file1.txt with line given as 1 two times. The first one is written to the first line, and then the second command is entered so this also wrote to the first line. As a result, we can confirm that writeT with line works by checking the file1.txt's content.

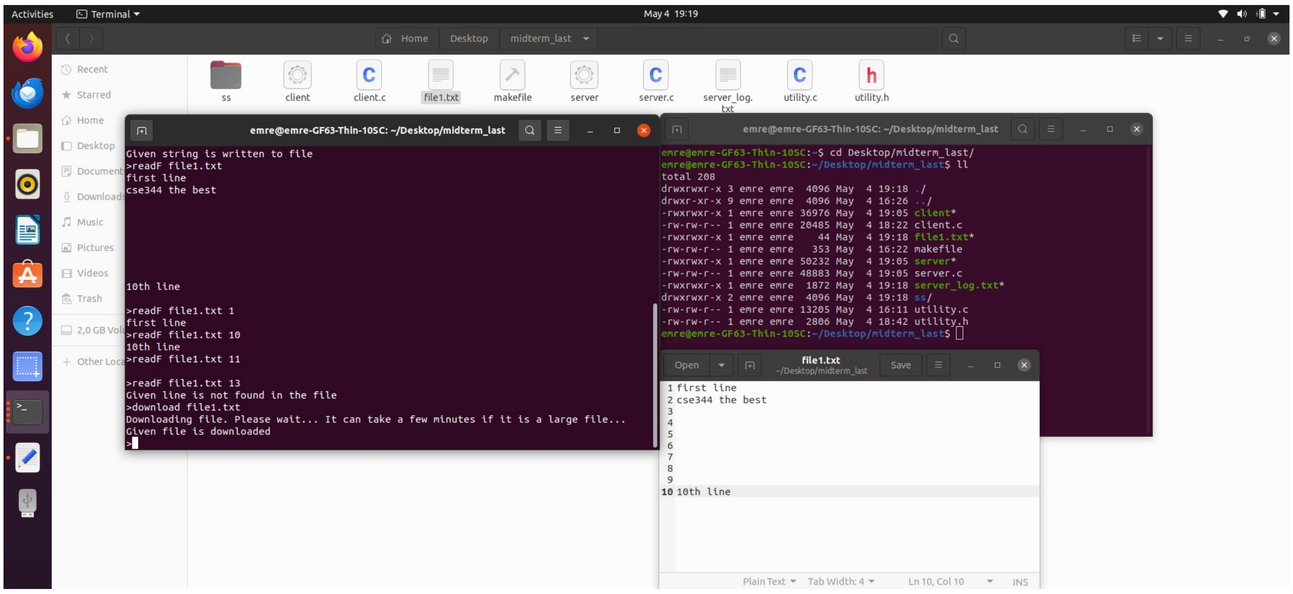
3.8) "readF" command with/out line given:



```
Given string is written to file
>readF file1.txt
first line
cse344 the best







10th line

>readF file1.txt 1
first line
>readF file1.txt 10
10th line
>readF file1.txt 11

>readF file1.txt 13
Given line is not found in the file
>
```

As we saw the content of the file1.txt file above in test 3.7, we can confirm that readF is working with and without line by checking the results.

3.9) “download” command working result:



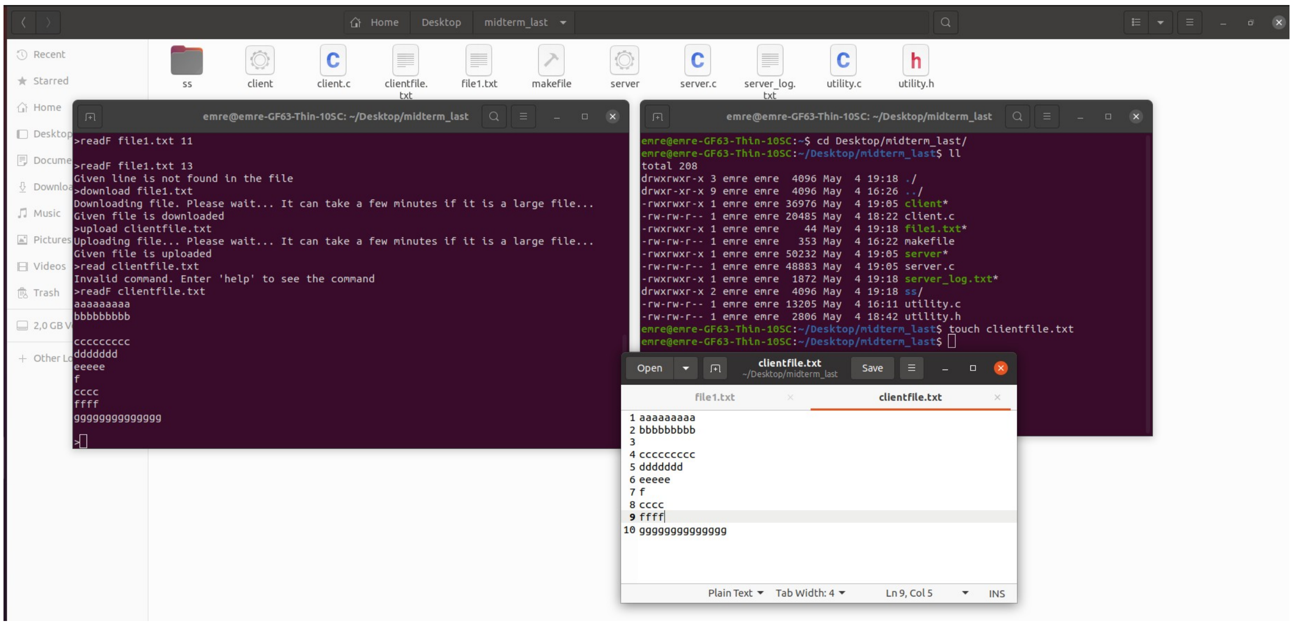
```
Given string is written to file
>readF file1.txt
first line
cse344 the best






10th line

>readF file1.txt 1
first line
>readF file1.txt 10
10th line
>readF file1.txt 11

>readF file1.txt 13
Given line is not found in the file
>download file1.txt
Downloading file. Please wait... It can take a few minutes if it is a large file...
Given file is downloaded
>
```

```
emre@emre-GF63-Thin-10SC:~$ cd Desktop/midterm_last/
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 208
drwxrwxr-x 3 emre emre  4096 May  4 19:18 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rwxrwxr-x 1 emre emre    44 May  4 19:18 file1.txt*
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre  1872 May  4 19:18 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:18 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

```
first line
cse344 the best







10th line
```

As we see, the file1.txt is downloaded to the client side properly. We can confirm that by checking if the file is really in the client side and the content of the file. As you see, I show these by using ll command and showing the client side directory. There is nothing related to client and server are running in the same folder, the server side files are kept under “/*tmp*/given_dir_name” directory. You can confirm that in the upload test’s screenshot as I’m showing the contents of this directory there.

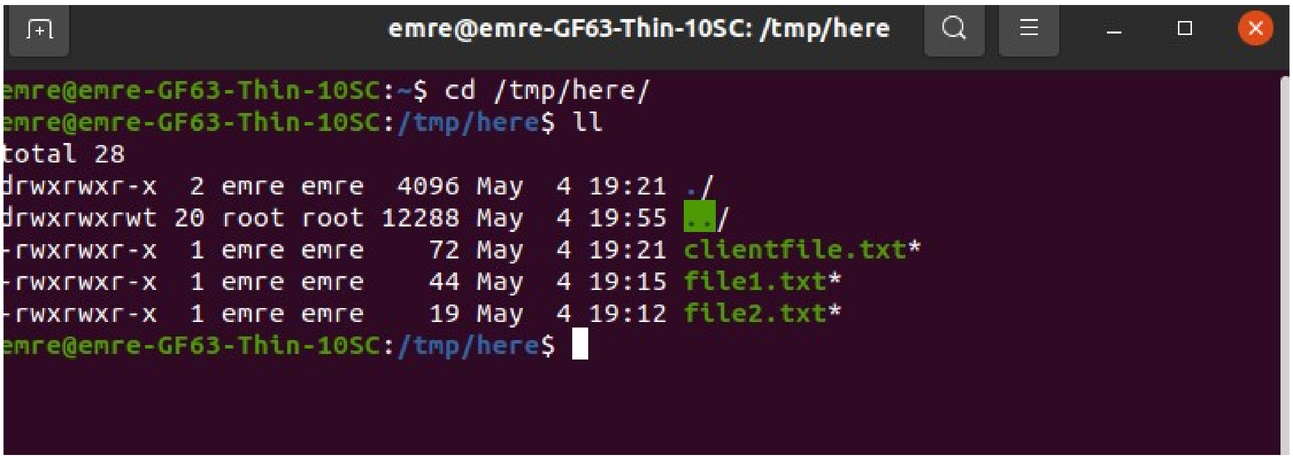
3.10) “upload” command working result:



```
>readF file1.txt 11

>readF file1.txt 13
Given line is not found in the file
>download file1.txt
Downloading file. Please wait... It can take a few minutes if it is a large file...
Given file is downloaded
>upload clientfile.txt
Uploading file... Please wait... It can take a few minutes if it is a large file...
Given file is uploaded
>read clientfile.txt
Invalid command. Enter 'help' to see the command
>readF clientfile.txt
aaaaaaaaa
bbbbbbbbb

ccccccccc
ddddddd
eeeee
f
cccc
ffff
ggggggggggggg

>
```

```
emre@emre-GF63-Thin-10SC:~$ cd Desktop/midterm_last/
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ll
total 208
drwxrwxr-x 3 emre emre  4096 May  4 19:18 ./
drwxr-xr-x 9 emre emre  4096 May  4 16:26 ../
-rwxrwxr-x 1 emre emre 36976 May  4 19:05 client*
-rw-rw-r-- 1 emre emre 20485 May  4 18:22 client.c
-rwxrwxr-x 1 emre emre    44 May  4 19:18 file1.txt*
-rw-rw-r-- 1 emre emre   353 May  4 16:22 makefile
-rwxrwxr-x 1 emre emre 50232 May  4 19:05 server*
-rw-rw-r-- 1 emre emre 48883 May  4 19:05 server.c
-rwxrwxr-x 1 emre emre  1872 May  4 19:18 server_log.txt*
drwxrwxr-x 2 emre emre  4096 May  4 19:18 ss/
-rw-rw-r-- 1 emre emre 13205 May  4 16:11 utility.c
-rw-rw-r-- 1 emre emre  2806 May  4 18:42 utility.h
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ touch clientfile.txt
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

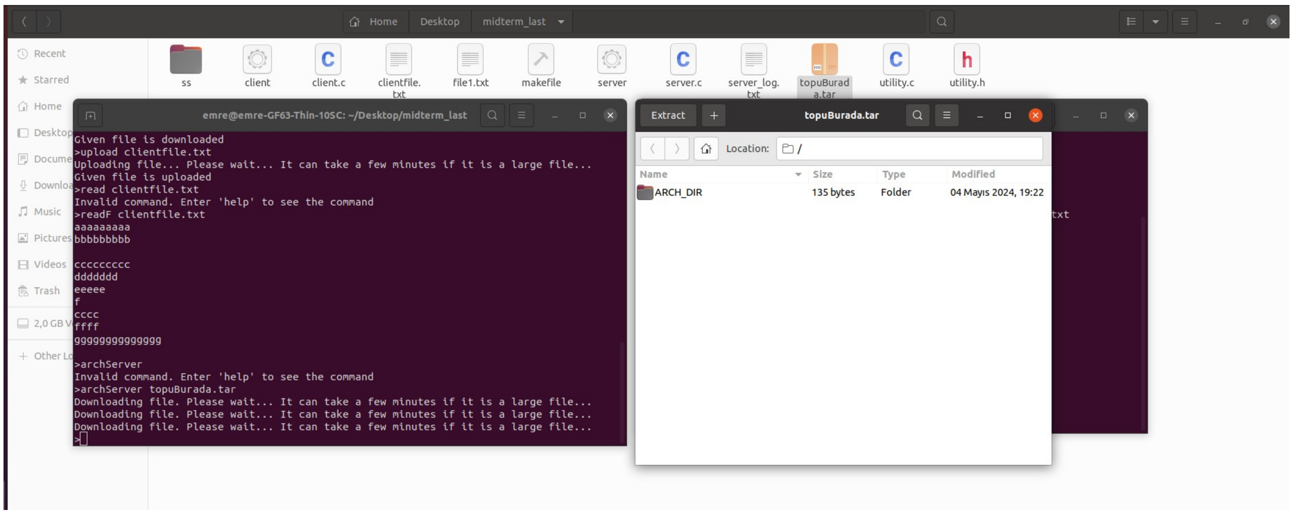
```
aaaaaaaaa
bbbbbbbbb

ccccccccc
ddddddd
eeeee
f
cccc
ffff
ggggggggggggg
```



```
emre@emre-GF63-Thin-10SC:~$ cd /tmp/here/
emre@emre-GF63-Thin-10SC:/tmp/here$ ll
total 28
drwxrwxr-x  2 emre emre  4096 May  4 19:21 ./
drwxrwxrwt 20 root root 12288 May  4 19:55 ../
-rwxrwxr-x  1 emre emre    72 May  4 19:21 clientfile.txt*
-rwxrwxr-x  1 emre emre    44 May  4 19:15 file1.txt*
-rwxrwxr-x  1 emre emre    19 May  4 19:12 file2.txt*
emre@emre-GF63-Thin-10SC:/tmp/here$ 
```

As you can see, client.txt is uploaded to the server side properly.

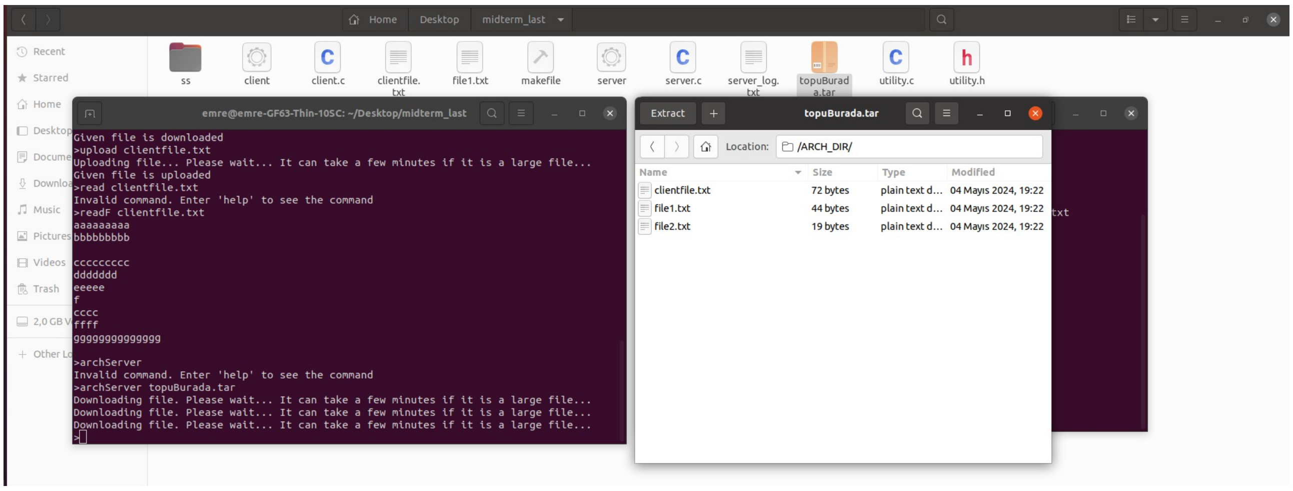
3.11) “archServer” command working result:



```
Given file is downloaded
>upload clientfile.txt
Uploading file... Please wait... It can take a few minutes if it is a large file...
Given file is uploaded
>read clientfile.txt
Invalid command. Enter 'help' to see the command
>readF clientfile.txt
aaaaaaaaa
bbbbbbbbb

ccccccccc
ddddddd
eeeee
f
cccc
ffff
ggggggggggggg

>archServer
Invalid command. Enter 'help' to see the command
>archServer topuBurada.tar
Downloading file. Please wait... It can take a few minutes if it is a large file...
Downloading file. Please wait... It can take a few minutes if it is a large file...
Downloading file. Please wait... It can take a few minutes if it is a large file...
>
```

topuBurada.tar — Location: /

Name	Size	Type	Modified
ARCH_DIR	135 bytes	Folder	04 Mayıs 2024, 19:22



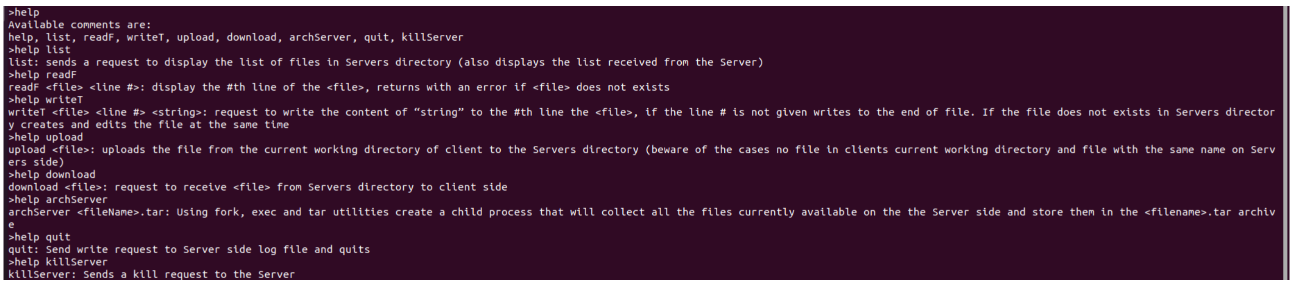
```
Given file is downloaded
>upload clientfile.txt
Uploading file... Please wait... It can take a few minutes if it is a large file...
Given file is uploaded
>read clientfile.txt
Invalid command. Enter 'help' to see the command
>readF clientfile.txt
aaaaaaaaa
bbbbbbbbb

ccccccccc
ddddddd
eeeee
f
cccc
ffff
ggggggggggggg

>archServer
Invalid command. Enter 'help' to see the command
>archServer topuBurada.tar
Downloading file. Please wait... It can take a few minutes if it is a large file...
Downloading file. Please wait... It can take a few minutes if it is a large file...
Downloading file. Please wait... It can take a few minutes if it is a large file...
>
```

topuBurada.tar — Location: /ARCH_DIR/

Name	Size	Type	Modified
clientfile.txt	72 bytes	plain text d...	04 Mayıs 2024, 19:22
file1.txt	44 bytes	plain text d...	04 Mayıs 2024, 19:22
file2.txt	19 bytes	plain text d...	04 Mayıs 2024, 19:22

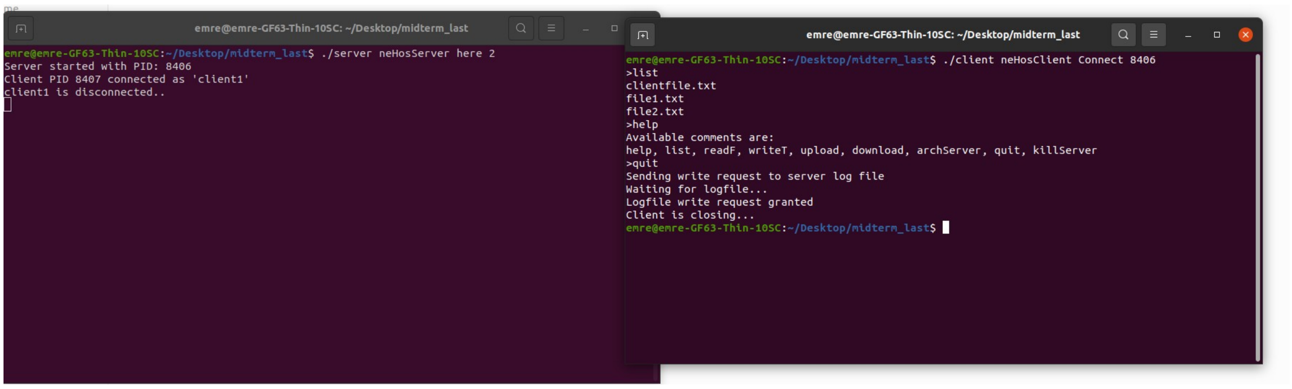
As it is seen directly, server side files are archieved in the given tar name “topuBurada.tar”. It contains a folder containing all server side files. The files are downloaded from the server one-by-one as it is written in the PDF file.

3.12) “help” command working result:



```
>help
Available comments are:
help, list, readF, writeT, upload, download, archServer, quit, killServer
>help list
list: sends a request to display the list of files in Servers directory (also displays the list received from the Server)
>help readF
readF <file> <line #>: display the #th line of the <file>, returns with an error if <file> does not exists
>help writeT
writeT <file> <line #> <string>: request to write the content of “string” to the #th line the <file>, if the line # is not given writes to the end of file. If the file does not exists in Servers director
y creates and edits the file at the same time
>help upload
upload <file>: uploads the file from the current working directory of client to the Servers directory (beware of the cases no file in clients current working directory and file with the same name on Serv
ers side)
>help download
download <file>: request to receive <file> from Servers directory to client side
>help archServer
archServer <fileName>.tar: Using fork, exec and tar utilities create a child process that will collect all the files currently available on the the Server side and store them in the <filename>.tar archiv
e
>help quit
quit: Send write request to Server side log file and quits
>help killServer
killServer: Sends a kill request to the Server
```

3.13) “quit” command working result:



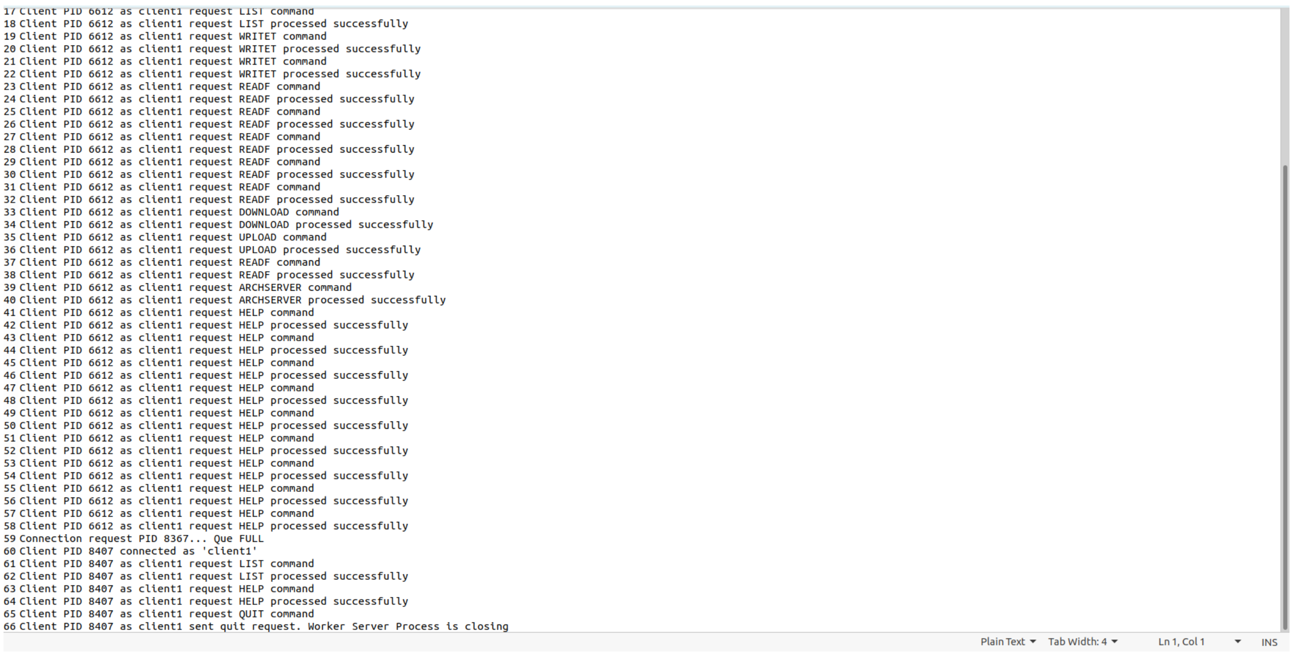
emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 8406
Client PID 8407 connected as 'client1'
client1 is disconnected..
```

emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last

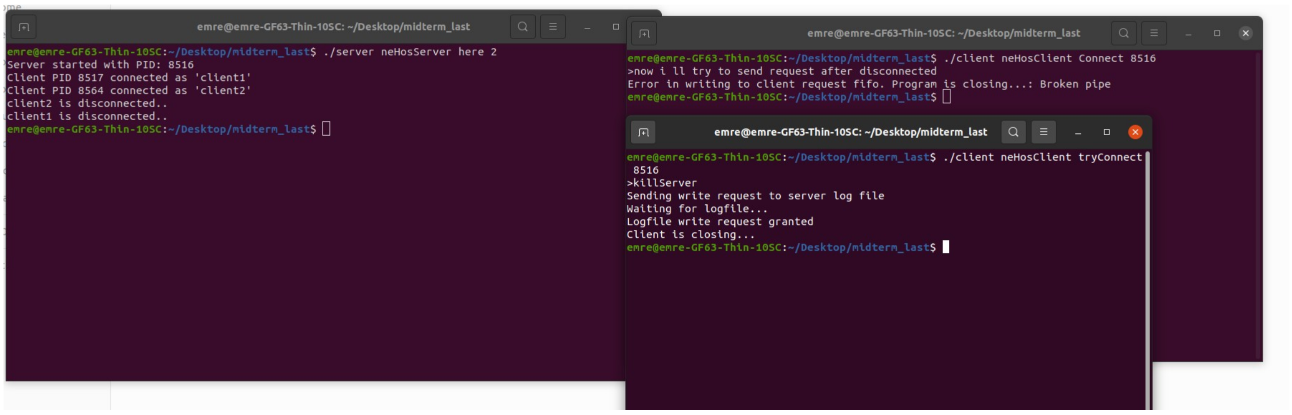
```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 8406
>list
clientfile.txt
file1.txt
file2.txt
>help
Available comments are:
help, list, readF, writeT, upload, download, archServer, quit, killServer
>quit
Sending write request to server log file
Waiting for logfile...
Logfile write request granted
Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

As you can see, when quit command is sent the client is closed and the server worker process is also closed. You can also check the log file’s content below.



```
Client PID 6612 as client1 request LIST command
Client PID 6612 as client1 request LIST processed successfully
Client PID 6612 as client1 request WRITET command
Client PID 6612 as client1 request WRITET processed successfully
Client PID 6612 as client1 request WRITET command
Client PID 6612 as client1 request WRITET processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request DOWNLOAD command
Client PID 6612 as client1 request DOWNLOAD processed successfully
Client PID 6612 as client1 request UPLOAD command
Client PID 6612 as client1 request UPLOAD processed successfully
Client PID 6612 as client1 request READF command
Client PID 6612 as client1 request READF processed successfully
Client PID 6612 as client1 request ARCHSERVER command
Client PID 6612 as client1 request ARCHSERVER processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Client PID 6612 as client1 request HELP command
Client PID 6612 as client1 request HELP processed successfully
Connection request PID 8367... Que FULL
Client PID 8407 connected as 'client1'
Client PID 8407 as client1 request LIST command
Client PID 8407 as client1 request LIST processed successfully
Client PID 8407 as client1 request HELP command
Client PID 8407 as client1 request HELP processed successfully
Client PID 8407 as client1 request QUIT command
Client PID 8407 as client1 sent quit request. Worker Server Process is closing
```

3.14) “killServer” command working result:



emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./server neHosServer here 2
Server started with PID: 8516
Client PID 8517 connected as 'client1'
Client PID 8564 connected as 'client2'
client2 is disconnected..
client1 is disconnected..
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

emre@emre-GF63-Thin-10SC: ~/Desktop/midterm_last

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient Connect 8516
>now i ll try to send request after disconnected
Error in writing to client request fifo. Program is closing...: Broken pipe
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

```
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$ ./client neHosClient tryConnect
 8516
>killServer
Sending write request to server log file
Waiting for logfile...
Logfile write request granted
Client is closing...
emre@emre-GF63-Thin-10SC:~/Desktop/midterm_last$
```

When client sent “killServer” command, the server side and client side is closed. Server is completely closed by the way as it says like that in the PDF. The server closes all the child processes meanwhile as you can see, the other child process was closed successfully.