Gebze Technical University

Computer Engineering

System Programing(CSE 344)

Midterm

Berkan AKIN

171044073

Problem

Two programs need to be written for a server and client. For each client connection request, the server will create a child process and communication will take place over this process. The clients will interact with the server in real time. When starting the server, the number of clients to be connected and the directory in which it will operate will be specified. A log file will be created for each operation. The server will process commands from the client and send them to the client. In summary   task is to design and implement a file server that enables multiple clients to connect, access and modify the contents of files in a specific directory

Solitions

System Design: Initially, the high-level architecture of the system should be designed. This involves determining the number of processes on both the server and client side and how they will communicate with each other. You could use a network protocol like FIFO for communication between the client and server. Additionally, you will need to determine which operations will be performed by multiple processes and which processes will be managed by signals.

File I/O Module: This module should perform reading and writing operations in various file formats. This includes being able to handle different types of files such as text files and binary files. Also, the management of large files (larger than 10 MB) should be considered.

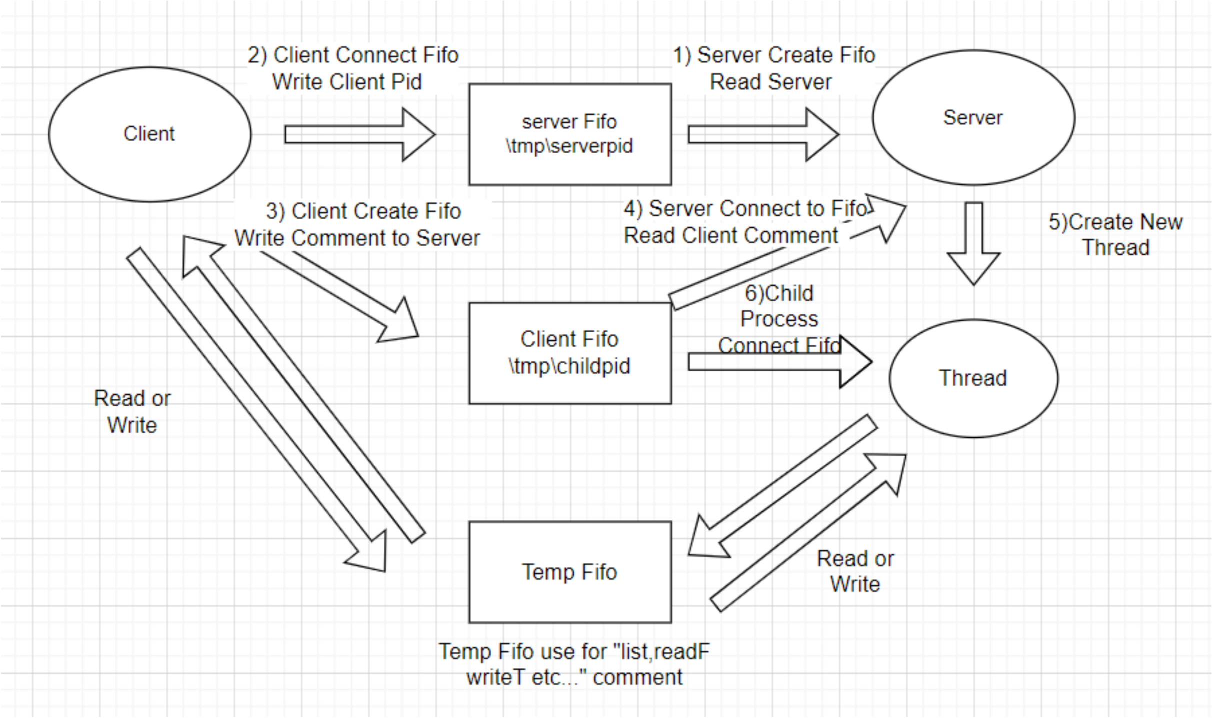
Synchronization Module: This module should be designed to prevent race conditions and provide mutual exclusion. When resource sharing is required between processes, one process should wait for others to complete and no changes should be made on the resource during this time. This ensures the preservation of data integrity and consistency of the file system.

Integration of File I/O and Synchronization Modules: These two modules should be integrated to form a file access system that can be accessed by multiple processes at the same time.

Testing: You should verify the system by testing the simultaneous access to files by multiple processes. This will help confirm that data consistency is maintained and race conditions are prevented.

These general steps should assist in creating a file server designed to meet most of the specified requirements. However, if a specific programming language or framework is being used, the solution might be slightly different. For instance, some languages naturally provide multi-process support, while others require specific libraries or plugins.

System Design



1) First, the server creates a FIFO for the clients to connect.

2) The client sends the address of the FIFO it created to the server.

3) "The client is ready to write commands by opening the FIFO file it created.

4) The server starts listening for incoming commands from the client by connecting to the FIFO created by the client.

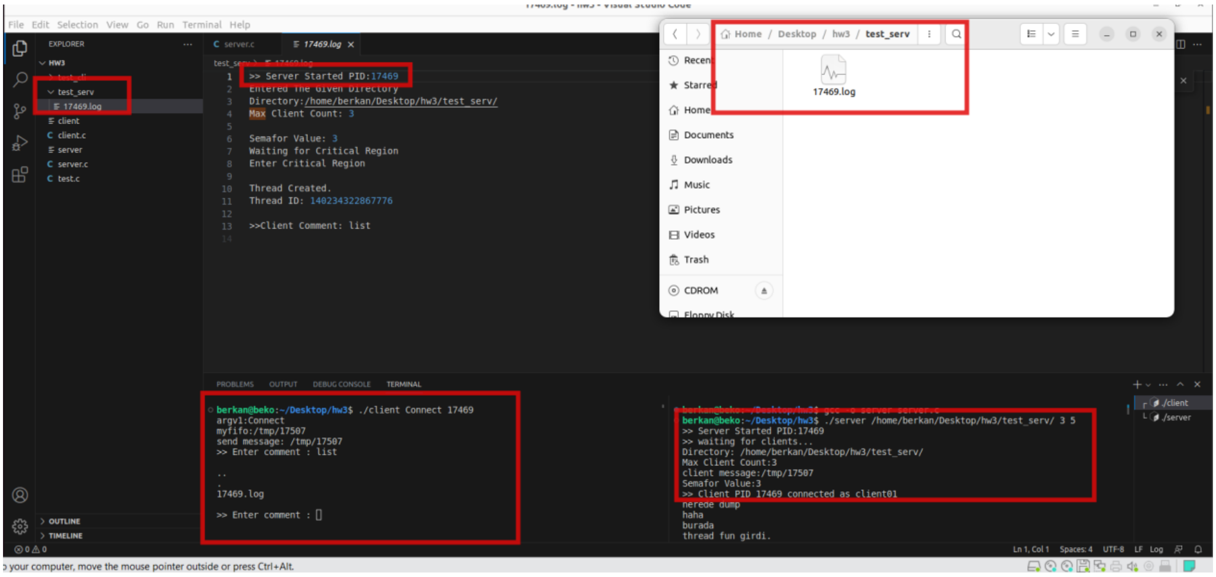
5) After reading the new FIFO address, the server creates a new Thread. The client establishes communication with the thread.

6) The thread connects to the FIFO created by the client. The client and server communicate through this FIFO.

Temp Fifo: These are temporary FIFOs created during the execution of commands.(list,readF,upload …).

Requirements

1) The server will operate in the directory specified at startup.

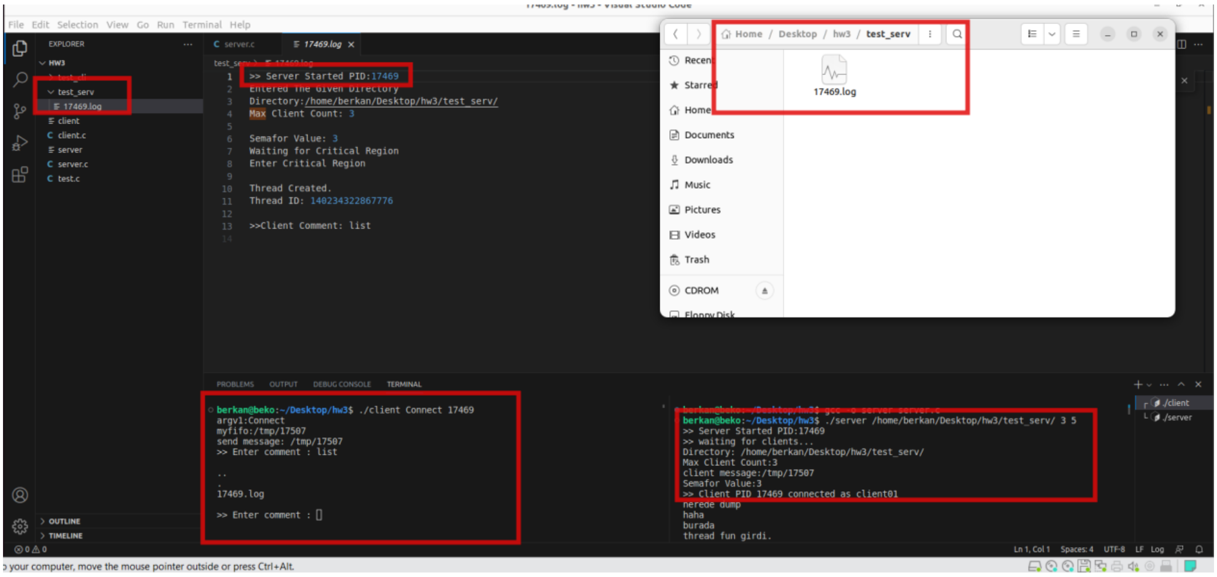

17469.log - hw3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
HW3
test_serv
17469.log
client
client.c
server
server.c
test.c
OUTLINE
TIMELINE
server.c
17469.log
Home / Desktop / hw3 / test_serv
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
CDROM
Floppy Disk
17469.log
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./client
./server
Ln 1, Col 1 Spaces: 4 UTF-8 LF Log
your computer, move the mouse pointer outside or press Ctrl+Alt.

```
>> Server Started PID:17469
Entered The Given Directory
Directory:/home/berkan/Desktop/hw3/test_serv/
Max Client Count: 3

Semafor Value: 3
Waiting for Critical Region
Enter Critical Region

Thread Created.
Thread ID: 140234322867776

>>Client Comment: list
```

```
berkan@beko:~/Desktop/hw3$ ./client Connect 17469
argv1:Connect
myfifo:/tmp/17507
send message: /tmp/17507
>> Enter comment : list

..
.
17469.log

>> Enter comment : 
```

```
berkan@beko:~/Desktop/hw3$ gcc -o server server.c
berkan@beko:~/Desktop/hw3$ ./server /home/berkan/Desktop/hw3/test_serv/ 3 5
>> Server Started PID:17469
>> waiting for clients...
Directory: /home/berkan/Desktop/hw3/test_serv/
Max Client Count:3
client message:/tmp/17507
Semafor Value:3
>> Client PID 17469 connected as client01
nerede dump
haha
burada
thread fun girdi.
```

❖ As you can see, entries have been made into the specified directory, and the log file has been created there.



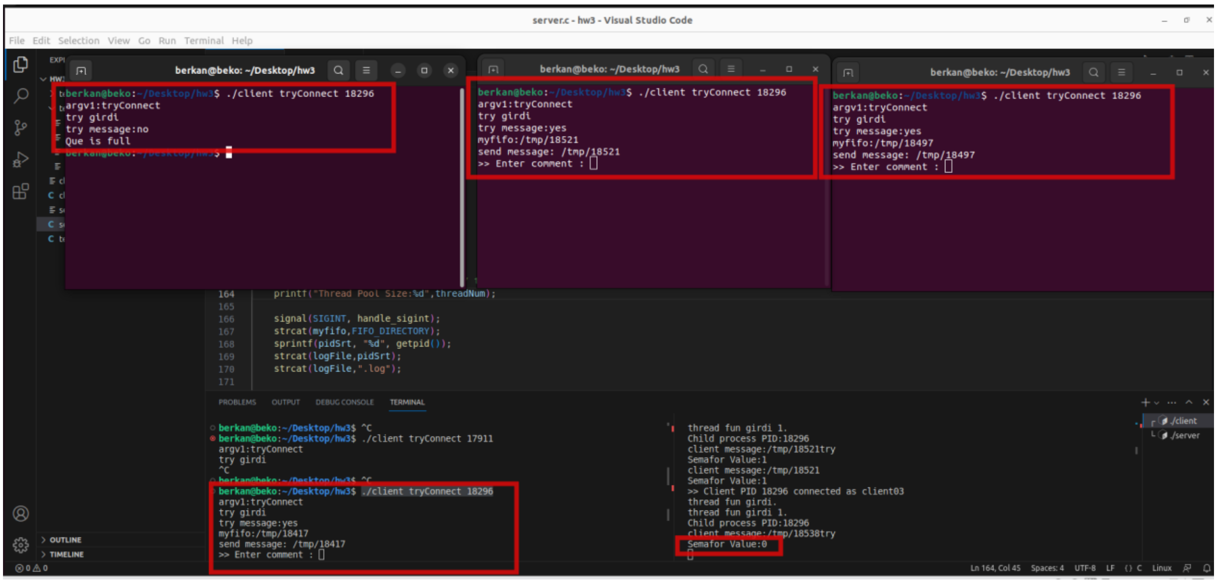
```
>> Server Started PID:17469
Entered The Given Directory
Directory:/home/berkan/Desktop/hw3/test_serv/
Max Client Count: 3

Semafor Value: 3
Waiting for Critical Region
Enter Critical Region

Thread Created.
Thread ID: 140234322867776

>>Client Comment: list
```

```
berkan@beko:~/Desktop/hw3$ ./client Connect 17469
argv1:Connect
myfifo:/tmp/17507
send message: /tmp/17507
>> Enter comment : list

..
.
17469.log

>> Enter comment : 
```

```
berkan@beko:~/Desktop/hw3$ gcc -o server server.c
berkan@beko:~/Desktop/hw3$ ./server /home/berkan/Desktop/hw3/test_serv/ 3 5
>> Server Started PID:17469
>> waiting for clients...
Directory: /home/berkan/Desktop/hw3/test_serv/
Max Client Count:3
client message:/tmp/17507
Semafor Value:3
>> Client PID 17469 connected as client01
nerede dump
haha
burada
thread fun girdi.
```

2)The server should be able to accept connections from as many clients as the maximum number of clients specified in the input when starting the server.

- The maximum number of clients that can connect to the server was given as 3. When 3 clients are connected, it rejects the 4th client's connection. The reason for the server rejecting the connection is because the client input is **‘tryConnect’**.


server.c - hw3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
berkan@beko: ~/Desktop/hw3
berkan@beko: ~/Desktop/hw3
berkan@beko: ~/Desktop/hw3
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./client
./server
OUTLINE
TIMELINE
Ln 164, Col 45 Spaces: 4 UTF-8 LF {} C Linux

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/hw3$
```

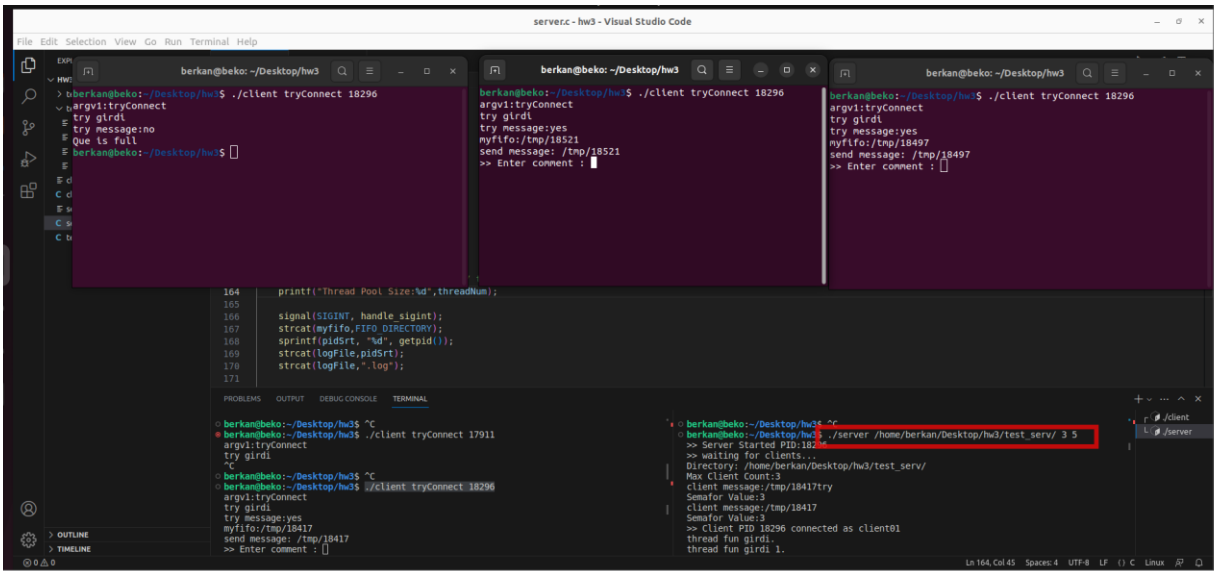
```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18521
send message: /tmp/18521
>> Enter comment :
```

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18497
send message: /tmp/18497
>> Enter comment :
```

```
    printf("Thread Pool Size:%d",threadNum);

    signal(SIGINT, handle_sigint);
    strcat(myfifo,FIFO_DIRECTORY);
    sprintf(pidSrt, "%d", getpid());
    strcat(logFile,pidSrt);
    strcat(logFile,".log");
```

```
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 17911
argv1:tryConnect
try girdi
^C
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18417
send message: /tmp/18417
>> Enter comment :
```

```
thread fun girdi 1.
Child process PID:18296
client message:/tmp/18521try
Semafor Value:1
client message:/tmp/18521
Semafor Value:1
>> Client PID 18296 connected as client03
thread fun girdi.
thread fun girdi 1.
Child process PID:18296
client message:/tmp/18538try
Semafor Value:0
```


server.c - hw3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
berkan@beko: ~/Desktop/hw3
berkan@beko: ~/Desktop/hw3
berkan@beko: ~/Desktop/hw3
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./client
./server
OUTLINE
TIMELINE
Ln 164, Col 45 Spaces: 4 UTF-8 LF {} C Linux

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/hw3$
```

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18521
send message: /tmp/18521
>> Enter comment :
```

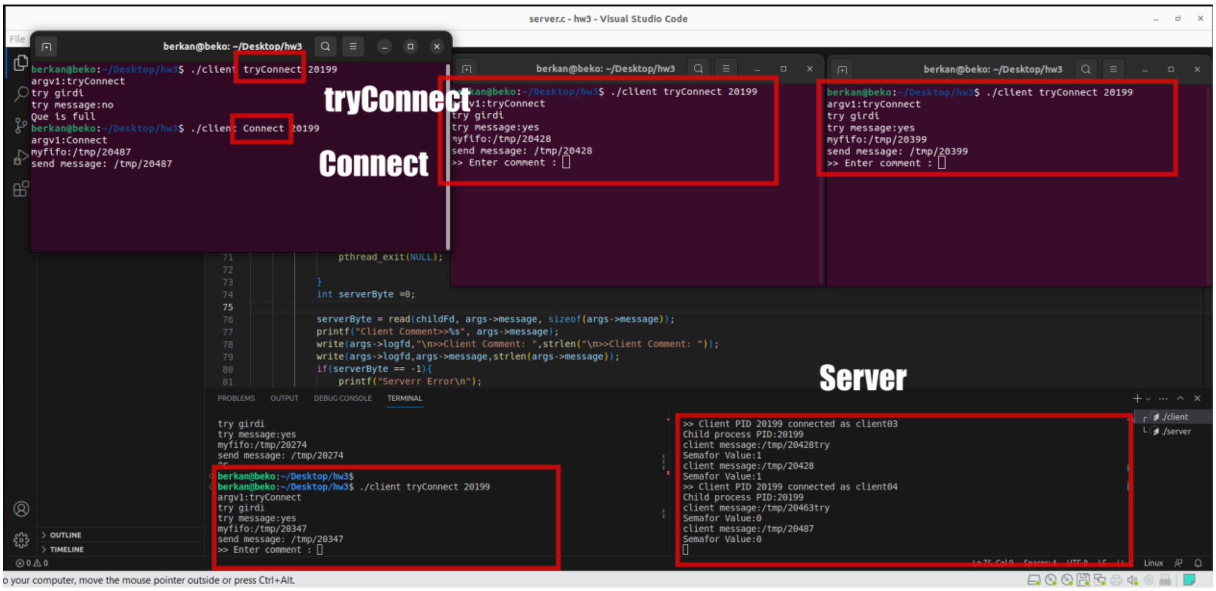
```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18497
send message: /tmp/18497
>> Enter comment :
```

```
    printf("Thread Pool Size:%d",threadNum);

    signal(SIGINT, handle_sigint);
    strcat(myfifo,FIFO_DIRECTORY);
    sprintf(pidSrt, "%d", getpid());
    strcat(logFile,pidSrt);
    strcat(logFile,".log");
```

```
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 17911
argv1:tryConnect
try girdi
^C
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 18296
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/18417
send message: /tmp/18417
>> Enter comment :
```

```
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./server /home/berkan/Desktop/hw3/test_serv/ 3 5
>> Server Started PID:18296
>> waiting for clients...
Directory: /home/berkan/Desktop/hw3/test_serv/
Max Client Count:3
client message:/tmp/18417try
Semafor Value:3
client message:/tmp/18417
Semafor Value:3
>> Client PID 18296 connected as client01
thread fun girdi.
thread fun girdi 1.
```

- "When we enter '**connected**' in the client input, the server takes it into the connection queue. When one client's connection drops, the next client in line connects.



```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 20199
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/hw3$ ./client Connect 20199
argv1:Connect
myfifo:/tmp/20487
send message: /tmp/20487
```

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 20199
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/20428
send message: /tmp/20428
>> Enter comment : 
```

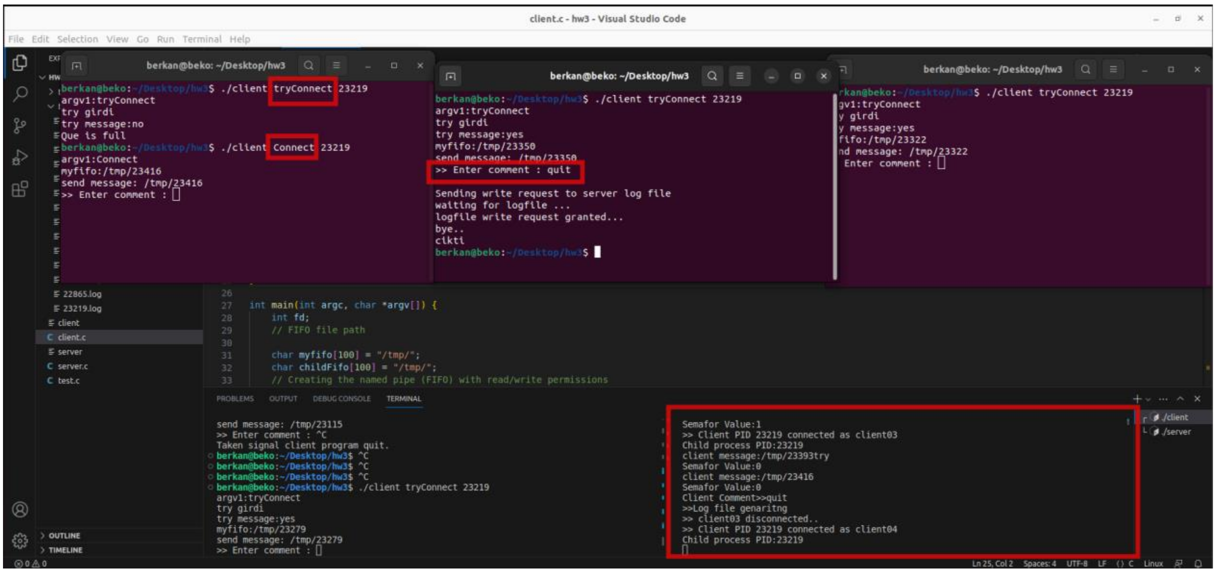
```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 20199
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/20399
send message: /tmp/20399
>> Enter comment : 
```

```
            pthread_exit(NULL);

        }
        int serverByte =0;

        serverByte = read(childFd, args->message, sizeof(args->message));
        printf("Client Comment>>%s", args->message);
        write(args->logfd,"\n>>Client Comment: ",strlen("\n>>Client Comment: "));
        write(args->logfd,args->message,strlen(args->message));
        if(serverByte == -1){
            printf("Serverr Error\n");
```

```
try girdi
try message:yes
myfifo:/tmp/20274
send message: /tmp/20274
^C
berkan@beko:~/Desktop/hw3$
berkan@beko:~/Desktop/hw3$ ./client tryConnect 20199
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/20347
send message: /tmp/20347
>> Enter comment : 
```

```
>> Client PID 20199 connected as client03
Child process PID:20199
client message:/tmp/20428try
Semafor Value:1
client message:/tmp/20428
Semafor Value:1
>> Client PID 20199 connected as client04
Child process PID:20199
client message:/tmp/20463try
Semafor Value:0
client message:/tmp/20487
Semafor Value:0
```

o your computer, move the mouse pointer outside or press Ctrl+Alt.



```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 23219
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/hw3$ ./client Connect 23219
argv1:Connect
myfifo:/tmp/23416
send message: /tmp/23416
>> Enter comment : 
```

```
berkan@beko:~/Desktop/hw3$ ./client tryConnect 23219
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/23350
send message: /tmp/23350
>> Enter comment : quit

Sending write request to server log file
waiting for logfile ...
logfile write request granted...
bye..
cikti
berkan@beko:~/Desktop/hw3$ 
```

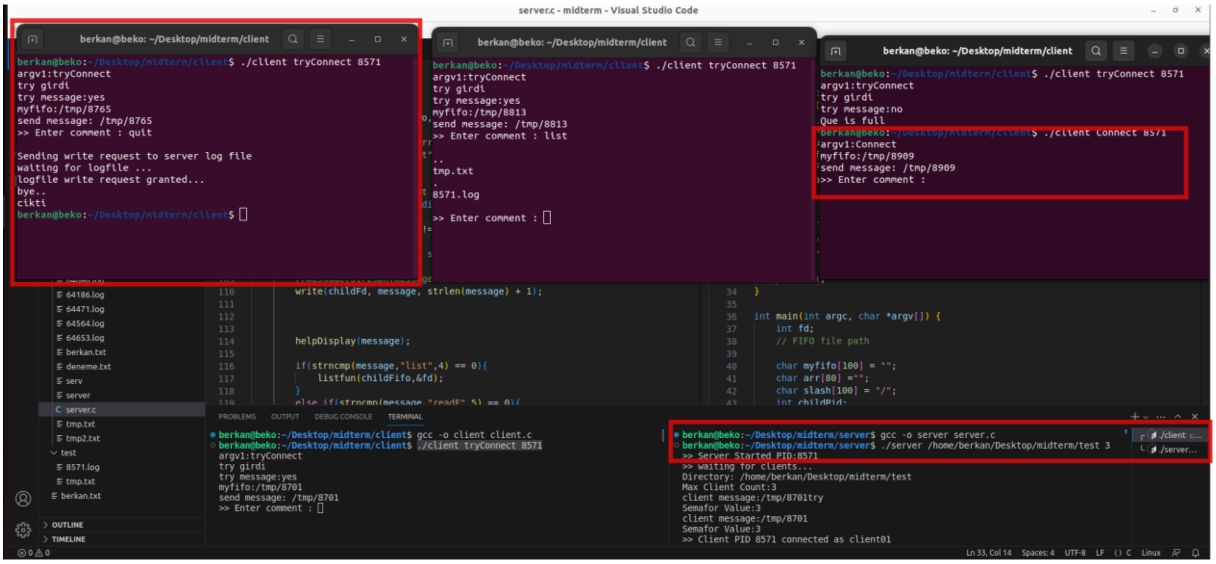
```
rkan@beko:~/Desktop/hw3$ ./client tryConnect 23219
gv1:tryConnect
y girdi
y message:yes
fifo:/tmp/23322
nd message: /tmp/23322
 Enter comment : 
```

```
int main(int argc, char *argv[]) {
    int fd;
    // FIFO file path

    char myfifo[100] = "/tmp/";
    char childFifo[100] = "/tmp/";
    // Creating the named pipe (FIFO) with read/write permissions
```

```
send message: /tmp/23115
>> Enter comment : ^C
Taken signal client program quit.
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 23219
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/23279
send message: /tmp/23279
>> Enter comment : 
```

```
Semafor Value:1
>> Client PID 23219 connected as client03
Child process PID:23219
client message:/tmp/23393try
Semafor Value:0
client message:/tmp/23416
Semafor Value:0
Client Comment>>quit
>>Log file genaritng
>> client03 disconnected..
>> Client PID 23219 connected as client04
Child process PID:23219
```

- In the image below, one client exits and the other is allowed to enter commands.



```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8765
send message: /tmp/8765
>> Enter comment : quit

Sending write request to server log file
waiting for logfile ...
logfile write request granted...
bye..
cikti
berkan@beko:~/Desktop/midterm/client$
```

```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8813
send message: /tmp/8813
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment :
```

```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/midterm/client$ ./client Connect 8571
argv1:Connect
myfifo:/tmp/8909
send message: /tmp/8909
>> Enter comment :
```

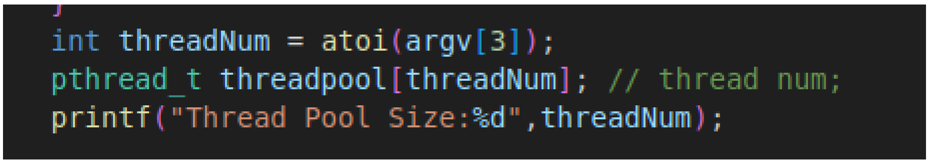
```
    write(childFd, message, strlen(message) + 1);



    helpDisplay(message);

    if(strncmp(message,"list",4) == 0){
        listfun(childFifo,&fd);
    }
    else if(strncmp(message,"readF",5) == 0){
```

```
}

int main(int argc, char *argv[]) {
    int fd;
    // FIFO file path

    char myfifo[100] = "";
    char arr[80] ="";
    char slash[100] = "/";
    int childPid;
```

```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment :
```

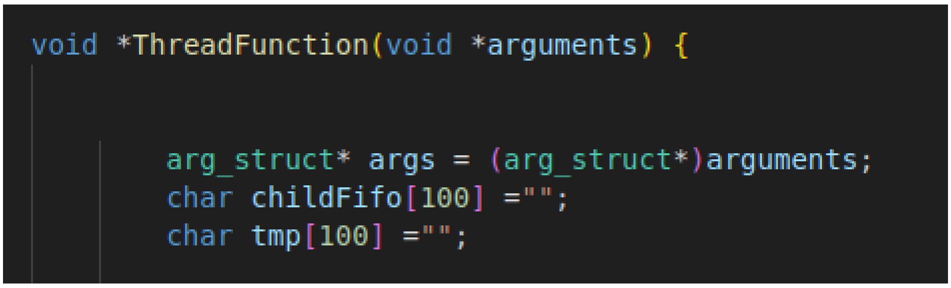
```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

- Tread Code Sinnipet
  thread pool



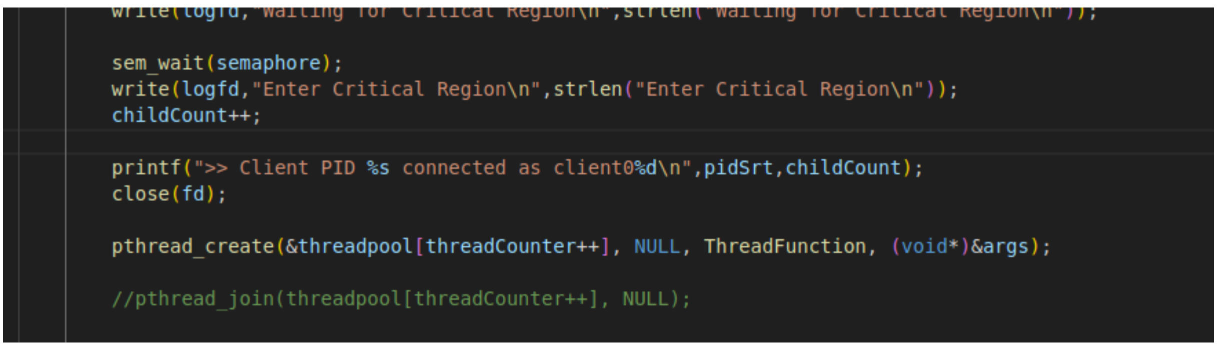
```
}
int threadNum = atoi(argv[3]);
pthread_t threadpool[threadNum]; // thread num;
printf("Thread Pool Size:%d",threadNum);
```

- Every client connecto one thread. Below image is Thread fun



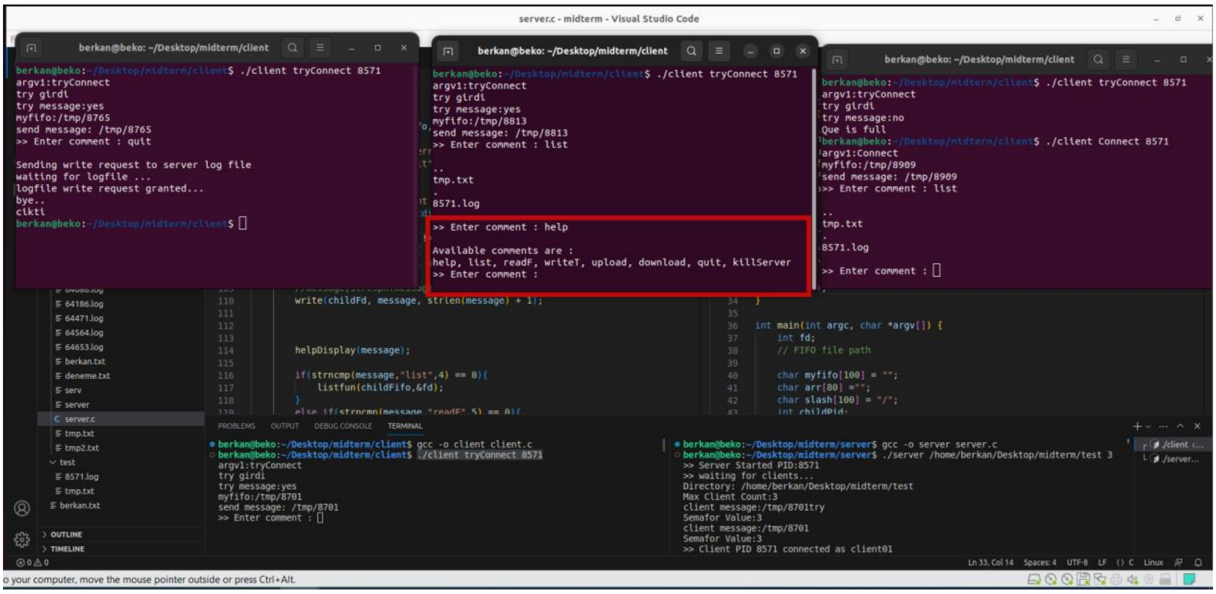
```
void *ThreadFunction(void *arguments) {


        arg_struct* args = (arg_struct*)arguments;
        char childFifo[100] ="";
        char tmp[100] ="";
```

- If the semaphore is available, a thread is created. If not, it waits for the other clients to exit. Then it creates a thread. The thread starts running.



```
write(logfd,"Waiting for Critical Region\n",strlen("Waiting for Critical Region\n"));

sem_wait(semaphore);
write(logfd,"Enter Critical Region\n",strlen("Enter Critical Region\n"));
childCount++;

printf(">> Client PID %s connected as client0%d\n",pidSrt,childCount);
close(fd);

pthread_create(&threadpool[threadCounter++], NULL, ThreadFunction, (void*)&args);

//pthread_join(threadpool[threadCounter++], NULL);
```

3) display the list of possible client requests



```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8765
send message: /tmp/8765
>> Enter comment : quit

Sending write request to server log file
waiting for logfile ...
logfile write request granted...
bye..
cikti
berkan@beko:~/Desktop/midterm/client$ 
```

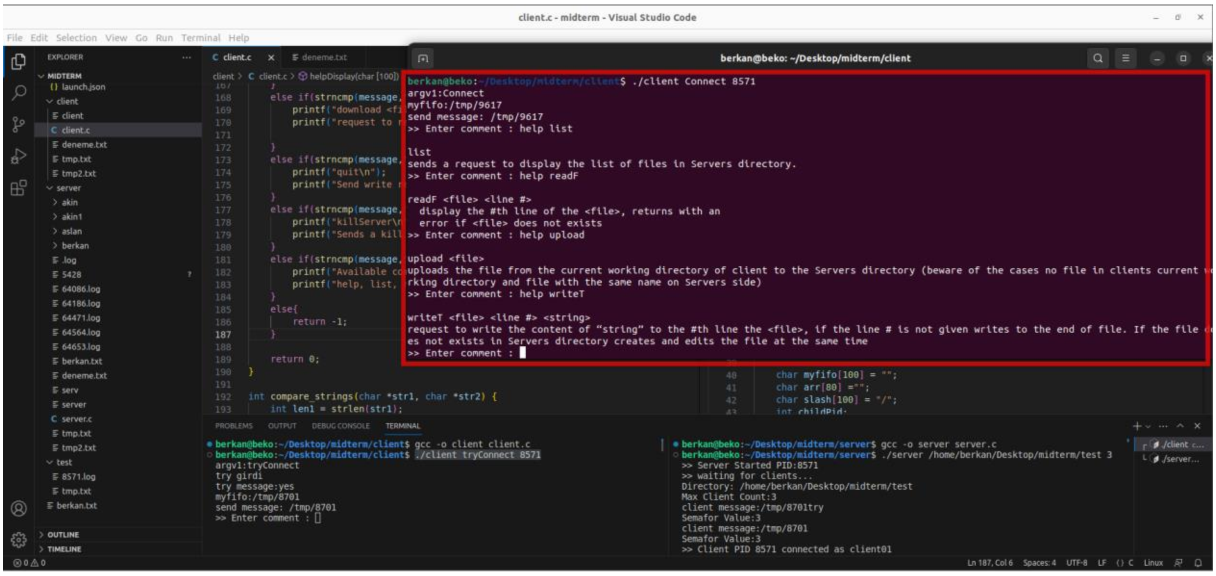
```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8813
send message: /tmp/8813
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : help

Available comments are :
help, list, readF, writeT, upload, download, quit, killServer
>> Enter comment :
```

```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/midterm/client$ ./client Connect 8571
argv1:Connect
myfifo:/tmp/8909
send message: /tmp/8909
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

- ❖ The visuals of how the commands work.

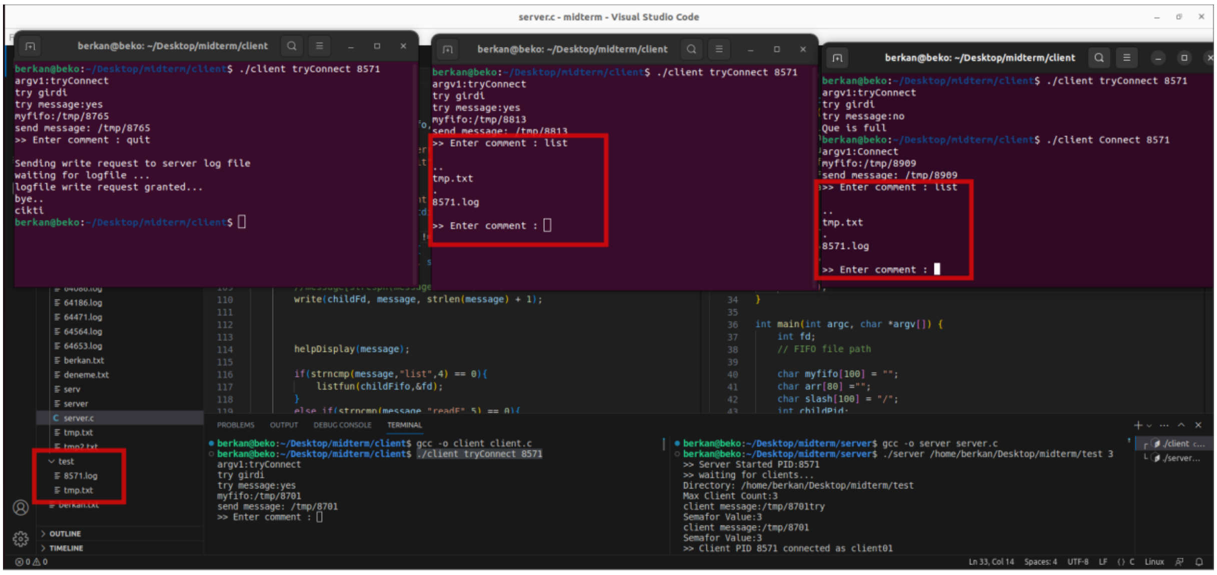


```
berkan@beko:~/Desktop/midterm/client$ ./client Connect 8571
argv1:Connect
myfifo:/tmp/9617
send message: /tmp/9617
>> Enter comment : help list

list
sends a request to display the list of files in Servers directory.
>> Enter comment : help readF

readF <file> <line #>
  display the #th line of the <file>, returns with an
  error if <file> does not exists
>> Enter comment : 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 w
rking directory and file with the same name on Servers side)
>> Enter comment : 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 d
es not exists in Servers directory creates and edits the file at the same time
>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

4)Show the files in the directory when the '**list**' command is given.



```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8765
send message: /tmp/8765
>> Enter comment : quit

Sending write request to server log file
waiting for logfile ...
logfile write request granted...
bye..
cikti
berkan@beko:~/Desktop/midterm/client$ 
```

```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8813
send message: /tmp/8813
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:no
Que is full
berkan@beko:~/Desktop/midterm/client$ ./client Connect 8571
argv1:Connect
myfifo:/tmp/8909
send message: /tmp/8909
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : 
```

```
    write(childFd, message, strlen(message) + 1);



    helpDisplay(message);

    if(strncmp(message,"list",4) == 0){
        listfun(childFifo,&fd);
    }
    else if(strncmp(message,"readF",5) == 0){
```

```
}

int main(int argc, char *argv[]) {
    int fd;
    // FIFO file path

    char myfifo[100] = "";
    char arr[80] ="";
    char slash[100] = "/";
    int childPid;
```

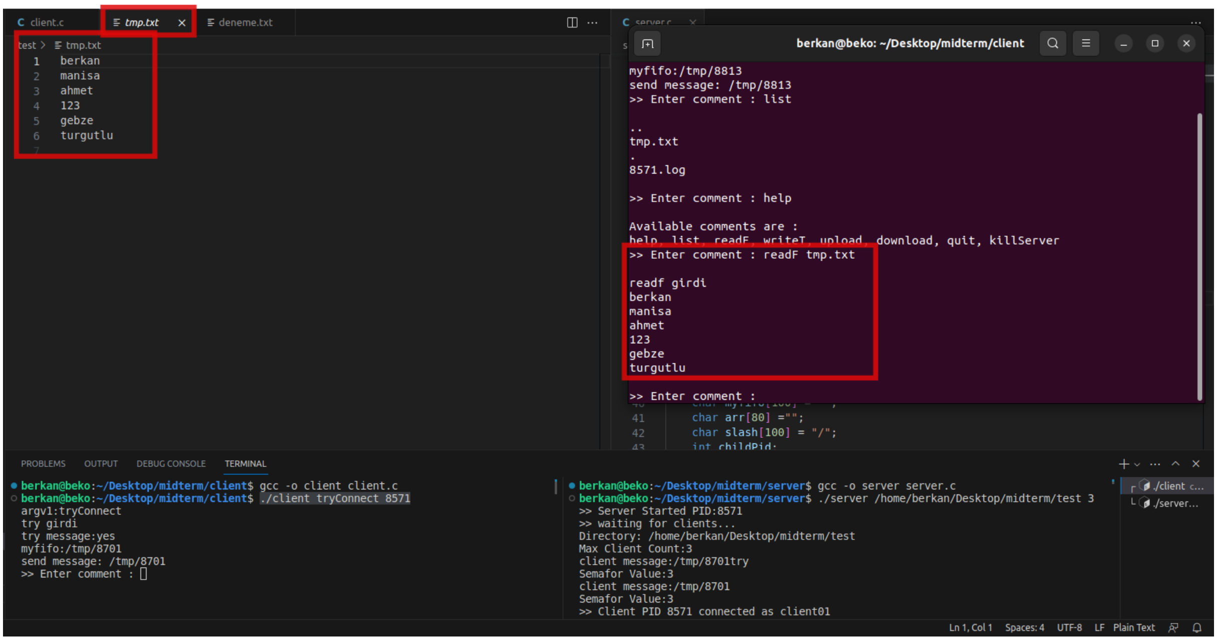
```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

5) Requests to display the # line of the <file>, if no line number is given the whole contents of the file is requested (and displayed on the client side)

readF <file> <line #>

❖ The result without giving a number.



```
berkan
manisa
ahmet
123
gebze
turgutlu
```

```
myfifo:/tmp/8813
send message: /tmp/8813
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : help

Available comments are :
help, list, readF, writeT, upload, download, quit, killServer
>> Enter comment : readF tmp.txt

readf girdi
berkan
manisa
ahmet
123
gebze
turgutlu

>> Enter comment :
```

```
    char arr[80] ="";
    char slash[100] = "/";
    int childPid;
```

```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : 
```

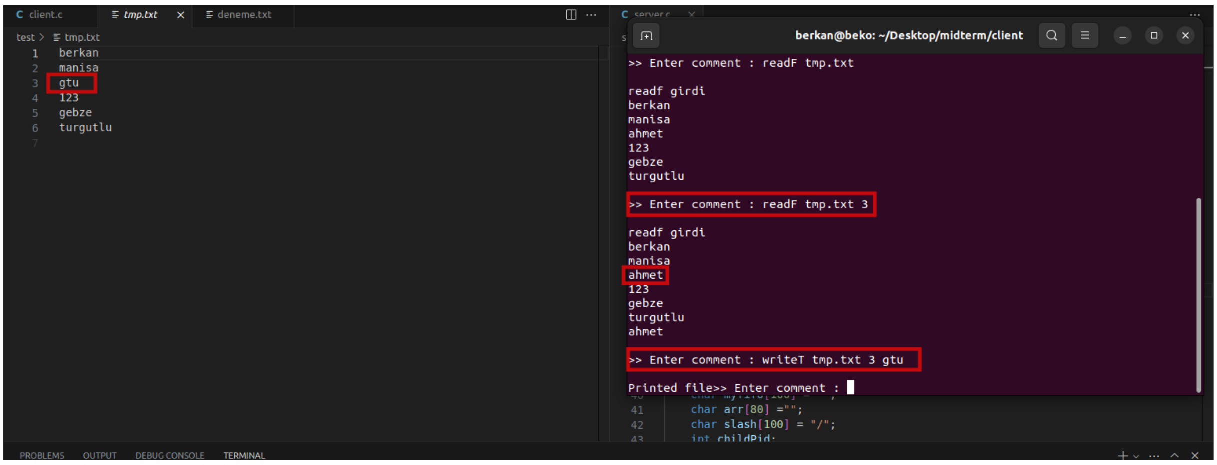
```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

Note: Data cannot be fetched by giving a line number.

6) 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 directory creates and edits the file at the same time.

writeT <file> <line #> <string>

- The word 'ahmet' in the 3rd line has been replaced with the word 'gtu'.

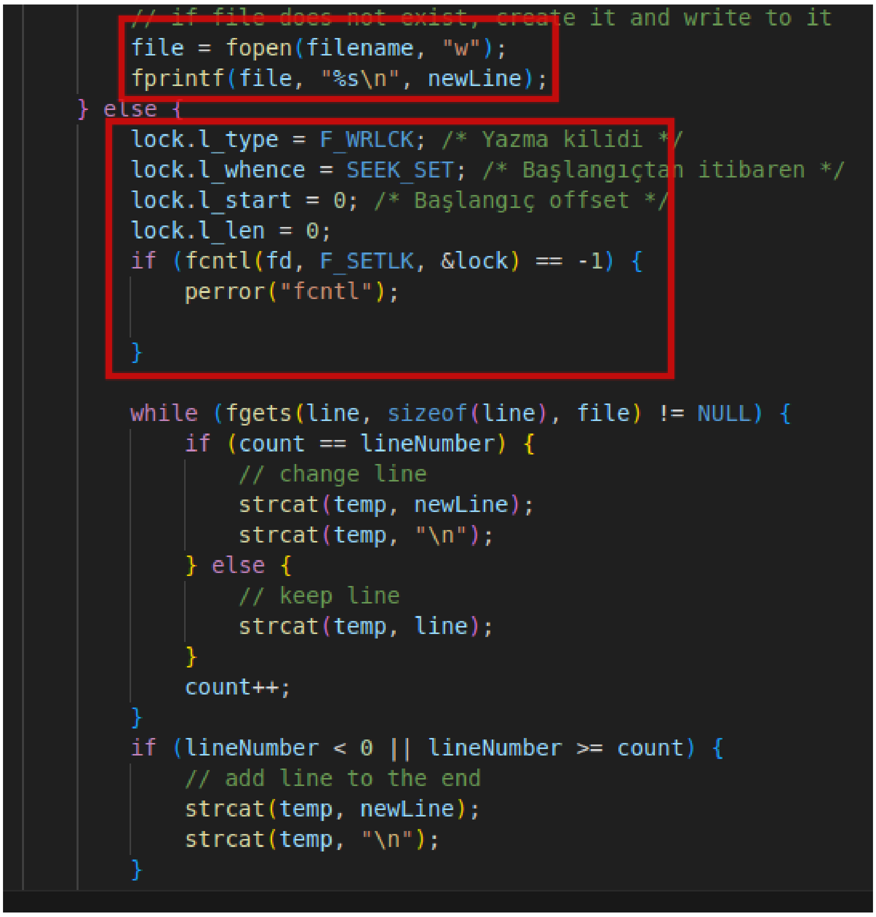


```
berkan
manisa
gtu
123
gebze
turgutlu
```

```
>> Enter comment : readF tmp.txt

readf girdi
berkan
manisa
ahmet
123
gebze
turgutlu

>> Enter comment : readF tmp.txt 3

readf girdi
berkan
manisa
ahmet
123
gebze
turgutlu
ahmet

>> Enter comment : writeT tmp.txt 3 gtu

Printed file>> Enter comment : 
```

```
    char arr[80] ="";
    char slash[100] = "/";
    int childPid;
```

- When there is no file, a file is created. "When a file is opened, another process cannot perform operations on the same file. The file has been locked with a **File Lock**.

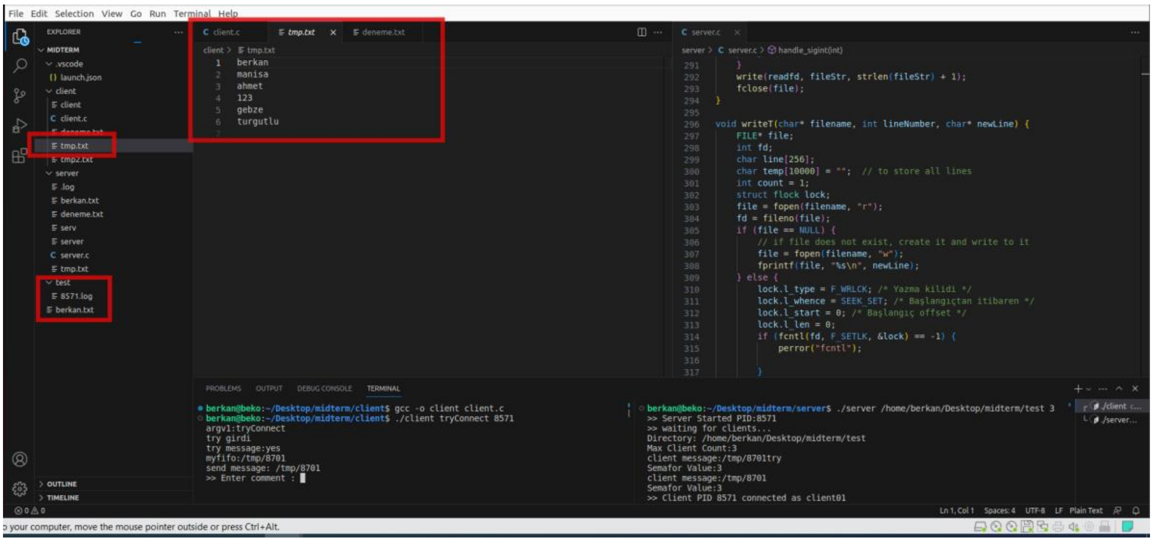


```
        // if file does not exist, create it and write to it
        file = fopen(filename, "w");
        fprintf(file, "%s\n", newLine);
    } else {
        lock.l_type = F_WRLCK; /* Yazma kilidi */
        lock.l_whence = SEEK_SET; /* Başlangıçtan itibaren */
        lock.l_start = 0; /* Başlangıç offset */
        lock.l_len = 0;
        if (fcntl(fd, F_SETLK, &lock) == -1) {
            perror("fcntl");

        }

        while (fgets(line, sizeof(line), file) != NULL) {
            if (count == lineNumber) {
                // change line
                strcat(temp, newLine);
                strcat(temp, "\n");
            } else {
                // keep line
                strcat(temp, line);
            }
            count++;
        }
        if (lineNumber < 0 || lineNumber >= count) {
            // add line to the end
            strcat(temp, newLine);
            strcat(temp, "\n");
        }
```

Not: When no number is given, it doesn't write to the end of the file.

7)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 Servers side)
upload <file>

- As seen in the first situation, the test directory is empty.



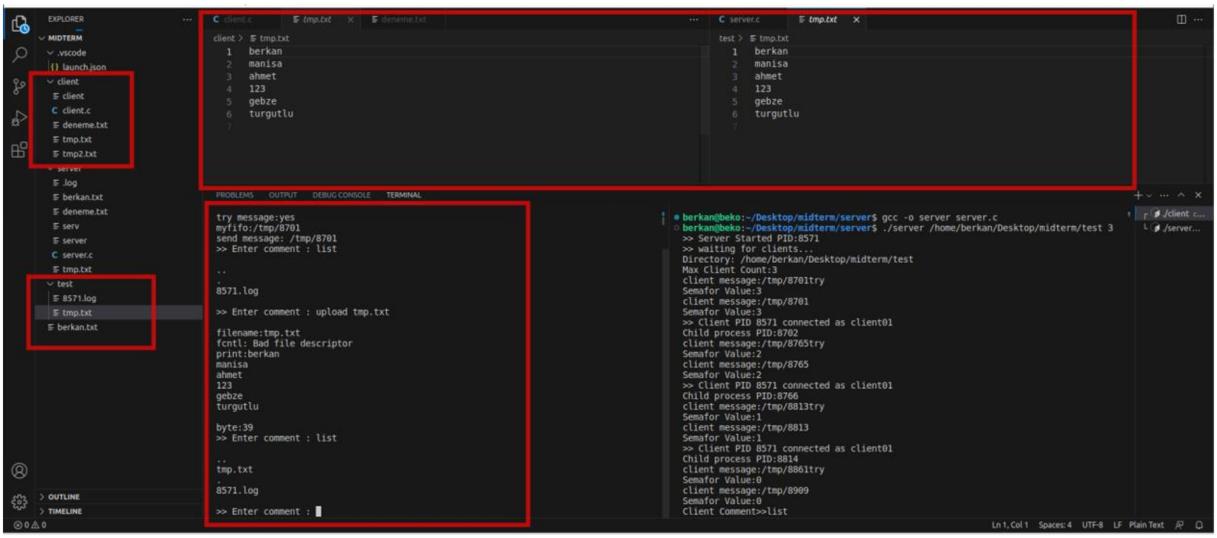
```
berkan
manisa
ahmet
123
gebze
turgutlu
```

```c
    }
    write(readfd, fileStr, strlen(fileStr) + 1);
    fclose(file);
}

void writeT(char* filename, int lineNumber, char* newLine) {
    FILE* file;
    int fd;
    char line[256];
    char temp[10000] = "";  // to store all lines
    int count = 1;
    struct flock lock;
    file = fopen(filename, "r");
    fd = fileno(file);
    if (file == NULL) {
        // if file does not exist, create it and write to it
        file = fopen(filename, "w");
        fprintf(file, "%s\n", newLine);
    } else {
        lock.l_type = F_WRLCK; /* Yazma kilidi */
        lock.l_whence = SEEK_SET; /* Başlangıçtan itibaren */
        lock.l_start = 0; /* Başlangıç offset */
        lock.l_len = 0;
        if (fcntl(fd, F_SETLK, &lock) == -1) {
            perror("fcntl");

        }
```

```
berkan@beko:~/Desktop/midterm/client$ gcc -o client client.c
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 8571
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
```

❖ When looked at with the 'list' command in the command line, the tmp.txt file does not appear at first. After running the 'upload tmp.txt' command and then running the 'list' command, it can be seen that the tmp.txt file has been created under the test directory.



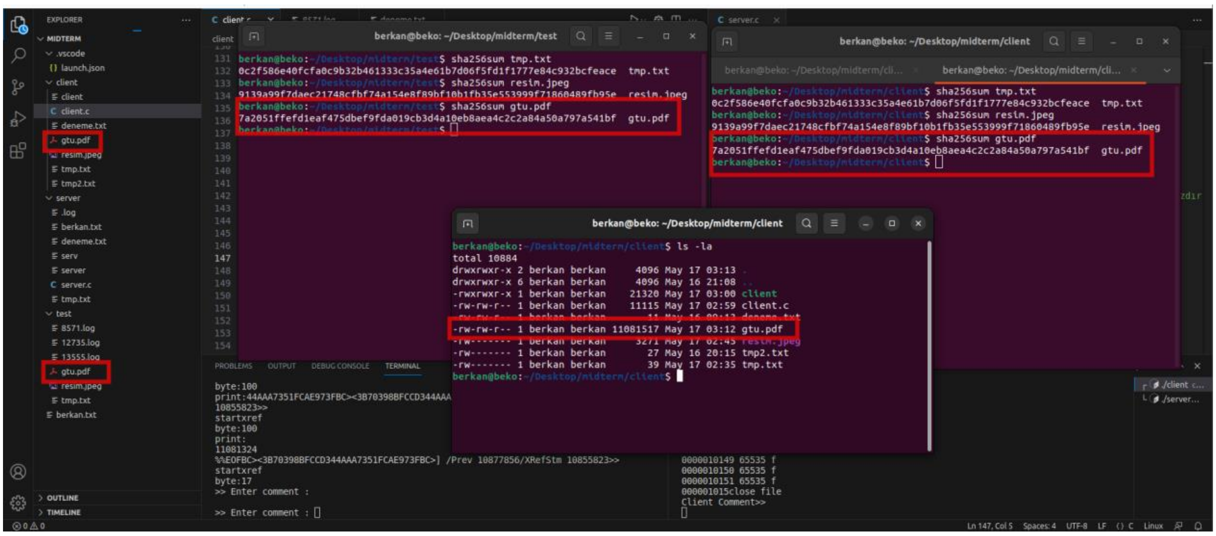
```
berkan
manisa
ahmet
123
gebze
turgutlu
```

```
berkan
manisa
ahmet
123
gebze
turgutlu
```

```
try message:yes
myfifo:/tmp/8701
send message: /tmp/8701
>> Enter comment : list

..
.
8571.log

>> Enter comment : upload tmp.txt

filename:tmp.txt
fcntl: Bad file descriptor
print:berkan
manisa
ahmet
123
gebze
turgutlu

byte:39
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : 
```

```
berkan@beko:~/Desktop/midterm/server$ gcc -o server server.c
berkan@beko:~/Desktop/midterm/server$ ./server /home/berkan/Desktop/midterm/test 3
>> Server Started PID:8571
>> waiting for clients...
Directory: /home/berkan/Desktop/midterm/test
Max Client Count:3
client message:/tmp/8701try
Semafor Value:3
client message:/tmp/8701
Semafor Value:3
>> Client PID 8571 connected as client01
Child process PID:8702
client message:/tmp/8765try
Semafor Value:2
client message:/tmp/8765
Semafor Value:2
>> Client PID 8571 connected as client01
Child process PID:8766
client message:/tmp/8813try
Semafor Value:1
client message:/tmp/8813
Semafor Value:1
>> Client PID 8571 connected as client01
Child process PID:8814
client message:/tmp/8861try
Semafor Value:0
client message:/tmp/8909
Semafor Value:0
Client Comment>>list
```

- ❖ We were able to copy a 10MB file with 'upload', and after conducting a hash check, we confirmed it's the same file.



berkan@beko: ~/Desktop/midterm/test

```
berkan@beko:~/Desktop/midterm/test$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/test$ sha256sum resim.jpeg
9139a99f7daec21748cfbf74a154e8f89bf10b1fb35e553999f71860489fb95e  resim.jpeg
berkan@beko:~/Desktop/midterm/test$ sha256sum gtu.pdf
7a2051ffefd1eaf475dbef9fda019cb3d4a10eb8aea4c2c2a84a50a797a541bf  gtu.pdf
berkan@beko:~/Desktop/midterm/test$ 
```

berkan@beko: ~/Desktop/midterm/client

```
berkan@beko:~/Desktop/midterm/client$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/client$ sha256sum resim.jpeg
9139a99f7daec21748cfbf74a154e8f89bf10b1fb35e553999f71860489fb95e  resim.jpeg
berkan@beko:~/Desktop/midterm/client$ sha256sum gtu.pdf
7a2051ffefd1eaf475dbef9fda019cb3d4a10eb8aea4c2c2a84a50a797a541bf  gtu.pdf
berkan@beko:~/Desktop/midterm/client$ 
```

berkan@beko: ~/Desktop/midterm/client

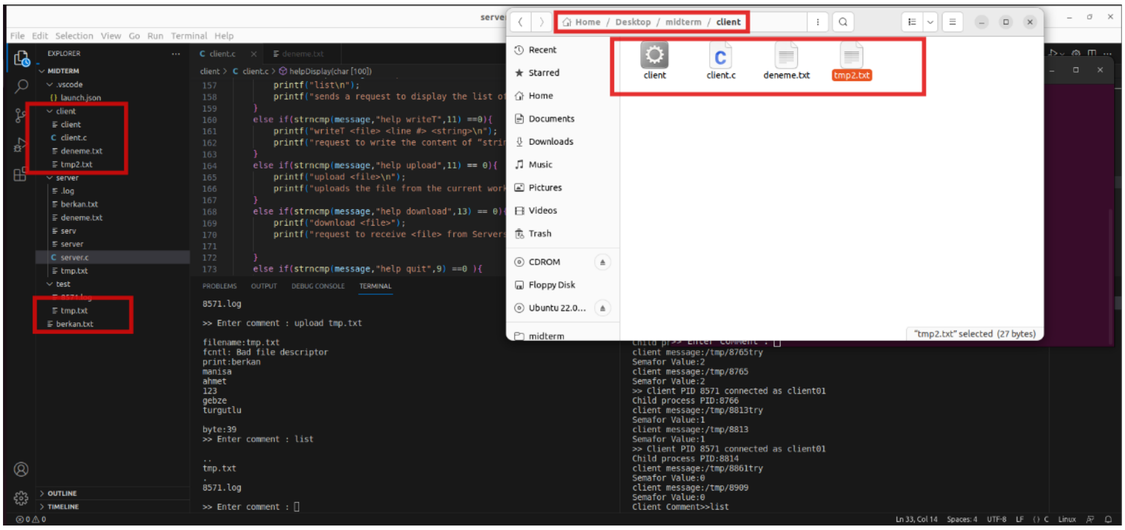
```
berkan@beko:~/Desktop/midterm/client$ ls -la
total 10884
drwxrwxr-x 2 berkan berkan     4096 May 17 03:13 .
drwxrwxr-x 6 berkan berkan     4096 May 16 21:08 ..
-rwxrwxr-x 1 berkan berkan    21320 May 17 03:00 client
-rw-rw-r-- 1 berkan berkan    11115 May 17 02:59 client.c
-rw-rw-r-- 1 berkan berkan       11 May 16 09:13 deneme.txt
-rw-rw-r-- 1 berkan berkan 11081517 May 17 03:12 gtu.pdf
-rw------- 1 berkan berkan     3271 May 17 02:45 resim.jpeg
-rw------- 1 berkan berkan       27 May 16 20:15 tmp2.txt
-rw------- 1 berkan berkan       39 May 17 02:35 tmp.txt
berkan@beko:~/Desktop/midterm/client$ 
```

```
byte:100
print:44AAA7351FCAE973FBC><3B70398BFCCD344AAA
10855823>>
startxref
byte:100
print:
11081324
%%EOFBC><3B70398BFCCD344AAA7351FCAE973FBC>] /Prev 10877856/XRefStm 10855823>>
startxref
byte:17
>> Enter comment :

>> Enter comment : 
```

```
0000010149 65535 f
0000010150 65535 f
0000010151 65535 f
000001015close file
Client Comment>>
```

7) Request to receive <file> from Servers directory to client side. **download <file>**

- ❖ It appears that there is a tmp.txt in the directory where the client is running. In the next image, we will copy the tmp.txt file from the test directory to the client directory with the 'download' command.



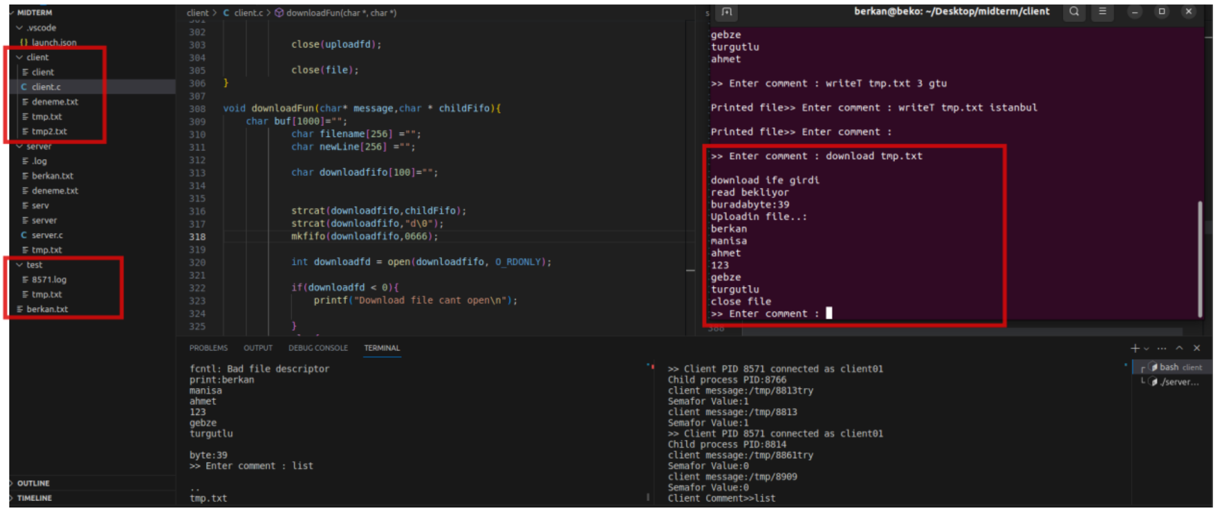
client > client.c > helpDisplay(char [100])

```c
            printf("list\n");
            printf("sends a request to display the list o
        }
        else if(strncmp(message,"help writeT",11) ==0){
            printf("writeT <file> <line #> <string>\n");
            printf("request to write the content of "strin
        }
        else if(strncmp(message,"help upload",11) == 0){
            printf("upload <file>\n");
            printf("uploads the file from the current work
        }
        else if(strncmp(message,"help download",13) == 0){
            printf("download <file>");
            printf("request to receive <file> from Servers

        }
        else if(strncmp(message,"help quit",9) ==0 ){
```

```
8571.log

>> Enter comment : upload tmp.txt

filename:tmp.txt
fcntl: Bad file descriptor
print:berkan
manisa
ahmet
123
gebze
turgutlu

byte:39
>> Enter comment : list

..
tmp.txt
.
8571.log

>> Enter comment : 
```

```
client message:/tmp/8765try
Semafor Value:2
client message:/tmp/8765
Semafor Value:2
>> Client PID 8571 connected as client01
Child process PID:8766
client message:/tmp/8813try
Semafor Value:1
client message:/tmp/8813
Semafor Value:1
>> Client PID 8571 connected as client01
Child process PID:8814
client message:/tmp/8861try
Semafor Value:0
client message:/tmp/8909
Semafor Value:0
Client Comment>>list
```

- ❖ The download was successfully received. The tmp.txt file was created in the client directory.



```c
                close(uploadfd);

                close(file);
}

void downloadFun(char* message,char * childFifo){
    char buf[1000]="";
        char filename[256] ="";
        char newLine[256] ="";

        char downloadfifo[100]="";


        strcat(downloadfifo,childFifo);
        strcat(downloadfifo,"d\0");
        mkfifo(downloadfifo,0666);

        int downloadfd = open(downloadfifo, O_RDONLY);

        if(downloadfd < 0){
            printf("Download file cant open\n");

        }
```

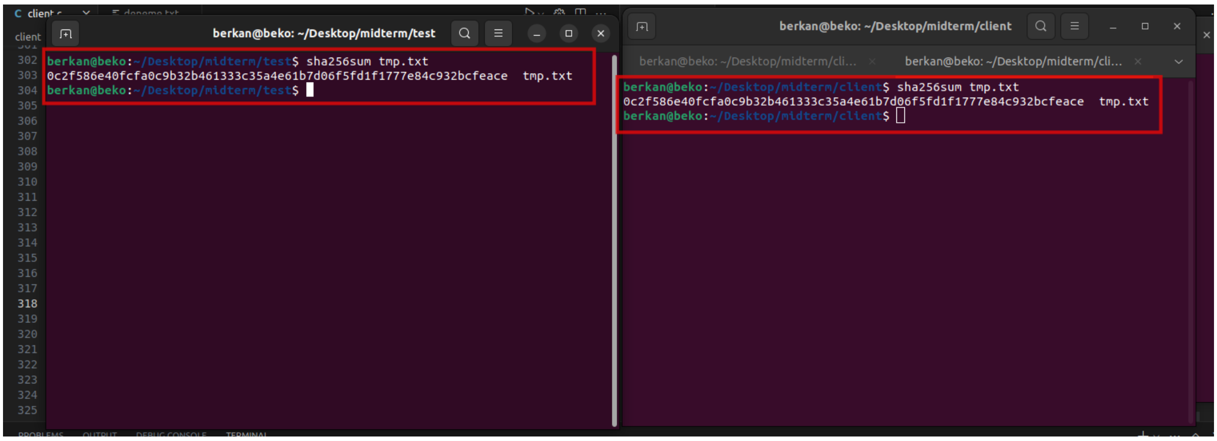
berkan@beko: ~/Desktop/midterm/client

```
gebze
turgutlu
ahmet

>> Enter comment : writeT tmp.txt 3 gtu

Printed file>> Enter comment : writeT tmp.txt istanbul

Printed file>> Enter comment :

>> Enter comment : download tmp.txt

download ife girdi
read bekliyor
buradabyte:39
Uploadin file..:
berkan
manisa
ahmet
123
gebze
turgutlu
close file
>> Enter comment :
```

```
fcntl: Bad file descriptor
print:berkan
manisa
ahmet
123
gebze
turgutlu

byte:39
>> Enter comment : list

..
tmp.txt
```

```
>> Client PID 8571 connected as client01
Child process PID:8766
client message:/tmp/8813try
Semafor Value:1
client message:/tmp/8813
Semafor Value:1
>> Client PID 8571 connected as client01
Child process PID:8814
client message:/tmp/8861try
Semafor Value:0
client message:/tmp/8909
Semafor Value:0
Client Comment>>list
```

- ❖ After the 'download' command was executed, the hash values of the copied file and the original file were compared and found to be the same.


client.c
deneme.txt
client
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

berkan@beko: ~/Desktop/midterm/test

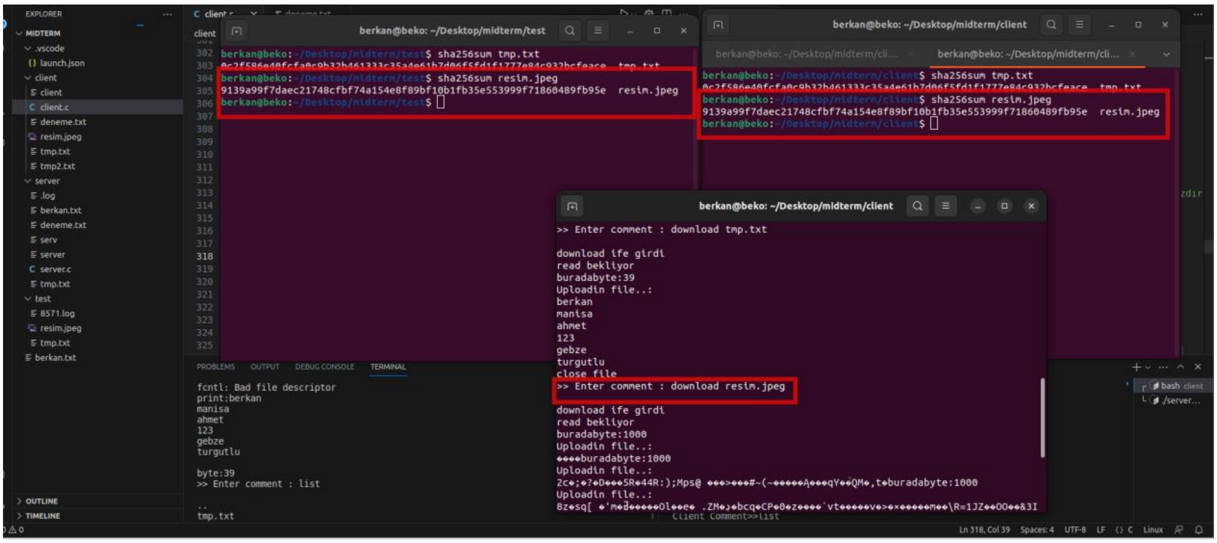
```
berkan@beko:~/Desktop/midterm/test$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/test$
```

berkan@beko: ~/Desktop/midterm/client

berkan@beko: ~/Desktop/midterm/cli...
berkan@beko: ~/Desktop/midterm/cli...

```
berkan@beko:~/Desktop/midterm/client$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/client$
```

- ❖ A file transfer has been made with a JPEG image file. The 'download' command has been tested with different file extensions and seen to work successfully. The hash values were checked and found to be the same.



berkan@beko: ~/Desktop/midterm/test

```
berkan@beko:~/Desktop/midterm/test$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/test$ sha256sum resim.jpeg
9139a99f7daec21748cfbf74a154e8f89bf10b1fb35e553999f71860489fb95e  resim.jpeg
berkan@beko:~/Desktop/midterm/test$
```

berkan@beko: ~/Desktop/midterm/client

berkan@beko: ~/Desktop/midterm/cli...
berkan@beko: ~/Desktop/midterm/cli...

```
berkan@beko:~/Desktop/midterm/client$ sha256sum tmp.txt
0c2f586e40fcfa0c9b32b461333c35a4e61b7d06f5fd1f1777e84c932bcfeace  tmp.txt
berkan@beko:~/Desktop/midterm/client$ sha256sum resim.jpeg
9139a99f7daec21748cfbf74a154e8f89bf10b1fb35e553999f71860489fb95e  resim.jpeg
berkan@beko:~/Desktop/midterm/client$
```

berkan@beko: ~/Desktop/midterm/client

```
>> Enter comment : download tmp.txt

download ife girdi
read bekliyor
buradabyte:39
Uploadin file..:
berkan
manisa
ahmet
123
gebze
turgutlu
close file
>> Enter comment : download resim.jpeg

download ife girdi
read bekliyor
buradabyte:1000
Uploadin file..:
����buradabyte:1000
Uploadin file..:
2c�;�?�D���5R�44R:);Mps@ ���>���#~(~�����A���qY��QM�,t�buradabyte:1000
Uploadin file..:
8z�sq[ �'m�d�����Ol���� .ZM�,�bcq�CP�0�z���� `vt�����v�>�×�����m��\R=1JZ��OO��&3I
```

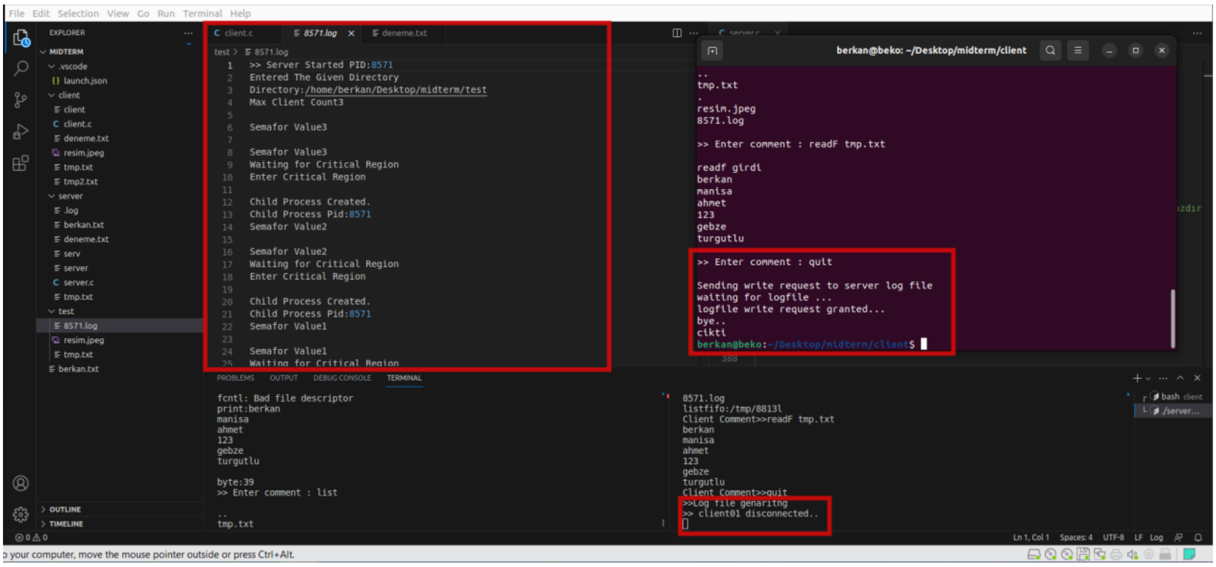
```
fcntl: Bad file descriptor
print:berkan
manisa
ahmet
123
gebze
turgutlu

byte:39
>> Enter comment : list

..
tmp.txt
```

```
Client Comment>>list
```

8) Send write request to Server side log file and quits.

quit

- ❖ The 'quit' command was given to the client and the connection with the server was terminated. After the connection was severed, the semaphore was incremented by one.


File Edit Selection View Go Run Terminal Help
EXPLORER
MIDTERM
client.c
8571.log
deneme.txt
test > 8571.log
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
berkan@beko: ~/Desktop/midterm/client
bash client
./server...
OUTLINE
TIMELINE

```
>> Server Started PID:8571
Entered The Given Directory
Directory:/home/berkan/Desktop/midterm/test
Max Client Count3

Semafor Value3

Semafor Value3
Waiting for Critical Region
Enter Critical Region

Child Process Created.
Child Process Pid:8571
Semafor Value2

Semafor Value2
Waiting for Critical Region
Enter Critical Region

Child Process Created.
Child Process Pid:8571
Semafor Value1

Semafor Value1
Waiting for Critical Region
```

```
..
tmp.txt
.
resim.jpeg
8571.log

>> Enter comment : readF tmp.txt

readf girdi
berkan
manisa
ahmet
123
gebze
turgutlu

>> Enter comment : quit

Sending write request to server log file
waiting for logfile ...
logfile write request granted...
bye..
cikti
berkan@beko:~/Desktop/midterm/client$
```

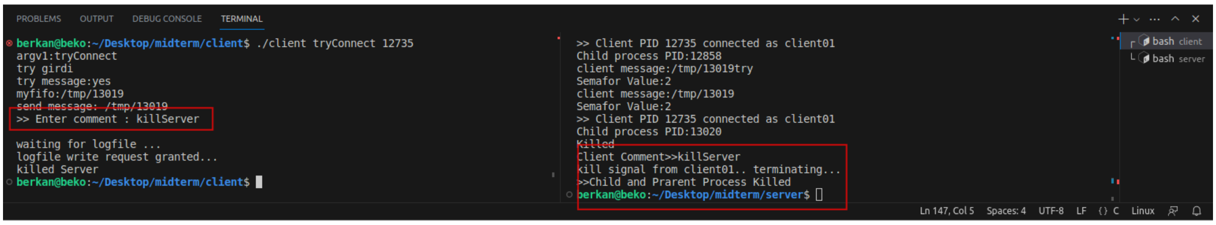
```
fcntl: Bad file descriptor
print:berkan
manisa
ahmet
123
gebze
turgutlu

byte:39
>> Enter comment : list

..
tmp.txt
```

```
8571.log
listfifo:/tmp/8813l
Client Comment>>readF tmp.txt
berkan
manisa
ahmet
123
gebze
turgutlu
Client Comment>>quit
>>Log file genaritng
>> client01 disconnected..
```

9) Sends a kill request to the Serve

killServer

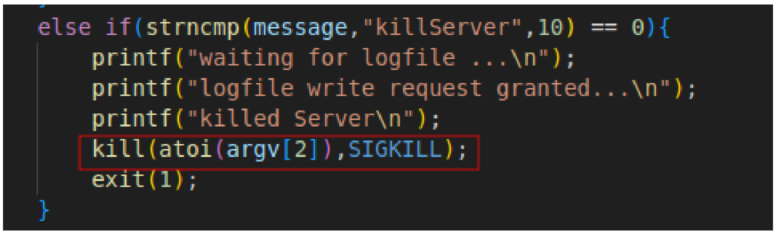
- ❖ We killed the server process by sending a kill signal to the server with the 'killServer' command.



```
berkan@beko:~/Desktop/midterm/client$ ./client tryConnect 12735
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/13019
send message: /tmp/13019
>> Enter comment : killServer

waiting for logfile ...
logfile write request granted...
killed Server
berkan@beko:~/Desktop/midterm/client$
```

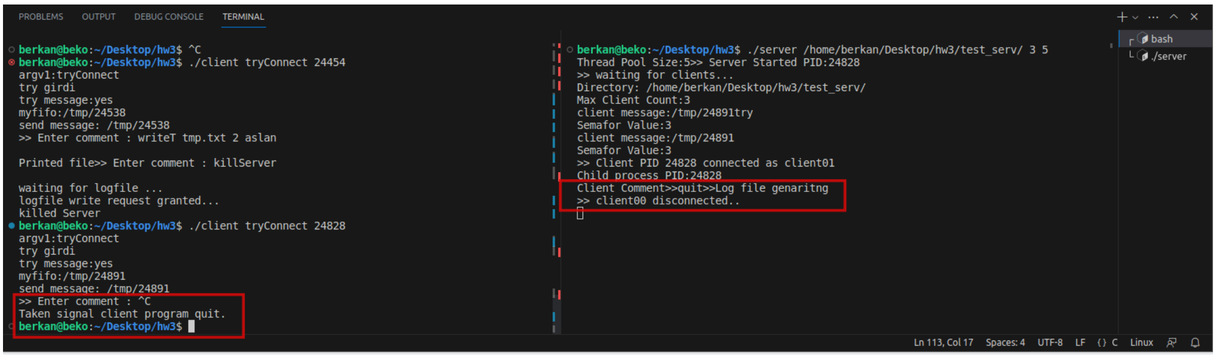
```
>> Client PID 12735 connected as client01
Child process PID:12858
client message:/tmp/13019try
Semafor Value:2
client message:/tmp/13019
Semafor Value:2
>> Client PID 12735 connected as client01
Child process PID:13020
Killed
Client Comment>>killServer
kill signal from client01.. terminating...
>>Child and Prarent Process Killed
berkan@beko:~/Desktop/midterm/server$
```

- ❖ Code snippet



```
else if(strncmp(message,"killServer",10) == 0){
    printf("waiting for logfile ...\n");
    printf("logfile write request granted...\n");
    printf("killed Server\n");
    kill(atoi(argv[2]),SIGKILL);
    exit(1);
}
```

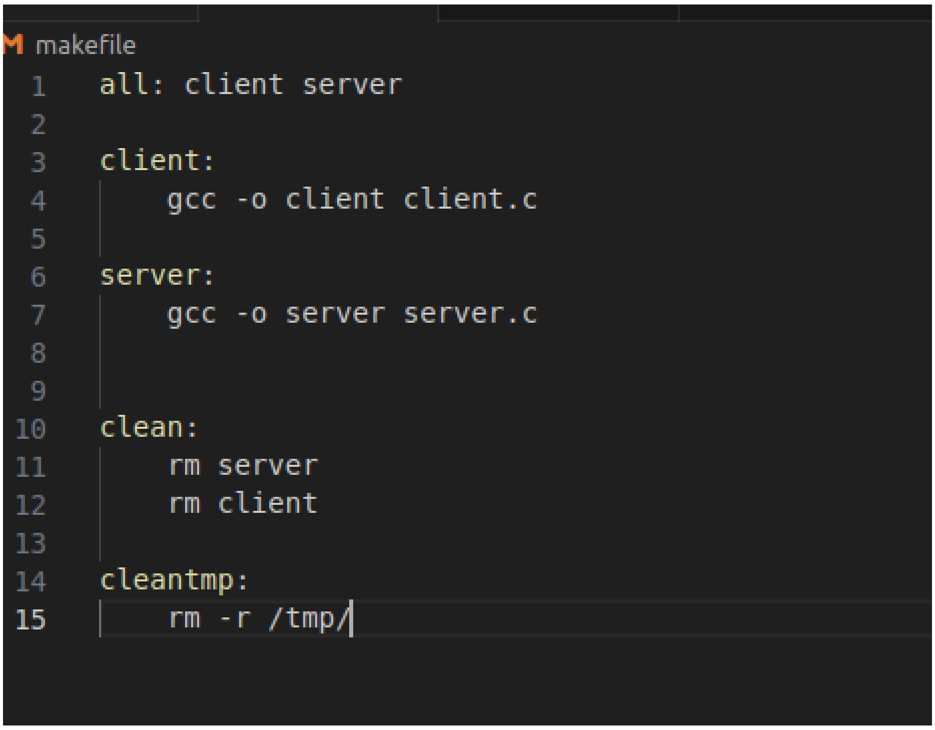
- The client stopped itself when it received the Ctrl+C signal. The server, on the other hand, disconnected its connection.



```
berkan@beko:~/Desktop/hw3$ ^C
berkan@beko:~/Desktop/hw3$ ./client tryConnect 24454
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/24538
send message: /tmp/24538
>> Enter comment : writeT tmp.txt 2 aslan

Printed file>> Enter comment : killServer

waiting for logfile ...
logfile write request granted...
killed Server
berkan@beko:~/Desktop/hw3$ ./client tryConnect 24828
argv1:tryConnect
try girdi
try message:yes
myfifo:/tmp/24891
send message: /tmp/24891
>> Enter comment : ^C
Taken signal client program quit.
berkan@beko:~/Desktop/hw3$
```

```
berkan@beko:~/Desktop/hw3$ ./server /home/berkan/Desktop/hw3/test_serv/ 3 5
Thread Pool Size:5>> Server Started PID:24828
>> waiting for clients...
Directory: /home/berkan/Desktop/hw3/test_serv/
Max Client Count:3
client message:/tmp/24891try
Semafor Value:3
client message:/tmp/24891
Semafor Value:3
>> Client PID 24828 connected as client01
Child process PID:24828
Client Comment>>quit>>Log file genaritng
>> client00 disconnected..
```

10) File cleanup is being done with the makefile.



```
all: client server

client:
	gcc -o client client.c

server:
	gcc -o server server.c


clean:
	rm server
	rm client

cleantmp:
	rm -r /tmp/
```

11)

Summary

Unfulfilled requirements are noted under the respective items. File transfers with a 10MB file have been successfully performed using 'download' and 'upload'. Files have been examined through their hash values. Using semaphores, clients have been put in a queue. After one client closes, the next client in the queue starts working. The functioning of the program is demonstrated in the images above.

Not: You need to press enter twice after each input. (If the client cannot connect to the server, try again.Buffurı temizlemek için her input sonrası 2 kere entera tıklayın. Client servera bağlanmazsa tekrar deneyince bağlanıyor.)