Gebze Technical Univesity

CSE 344 System Programming

-

Final Assignment Report

Abdulsamed Aslan

200104004098

Problem overview

I was expected to develop a client – server side cloud data storing application. All the connected clients recevies the server files and contents. Also when a change occurs in a client, server should send the changes to other clients. And If any changes occur in server directory, it inform all the clients.

Solution

Server

Server program takes directory path and port number as command line argument. Then it opens the given directory and creates a socket using given port number.

Server accepts incoming client connections. Upon accepting a connection, it retrieves the client's IP address and prints a connection message. It then adds the client socket to a queue, signals waiting threads, and increments the count of active threads. The loop repeats indefinitely, processing new client connections concurrently.

```
while (1)
{
    addr_size = sizeof(client_addr);
    client_sock = accept(server_sock, (struct sockaddr*)&client_addr,
&addr_size);

    if (isSignalReceived()){
        cleanup();
        break;
    }

    char *client_address = inet_ntoa(client_addr.sin_addr);
    printf("[+] Client connected | IP ADRESS: %s |\n",client_address);

    pthread_mutex_lock(&que_mutex);
    ++total_active_threads;
    enqueue(queue, client_sock);
    pthread_cond_signal(&que_cond);
    pthread_mutex_unlock(&que_mutex);

}
```

Firstly Server stores all the file informations of the given server directory to a Array. Awaken thread firstly sends all the the stored data to own client to provide synchronization.

The watch_changes function monitors a directory for changes. It first checks for new files and directories using the added_new_file_check function. Then, it detects deleted files, removes them from the list, and notifies waiting threads. Next, it identifies modified files and updates their information. Finally, it repeats the process in a continuous loop, periodically checking for changes. The function effectively tracks additions, deletions, and modifications in the specified directory.

```
void *watcher_thread_function(void *arg){


    char * directory = dirname;

    mask_sig();
    save_initial(directory);
    while (1) {
        if (isSignalReceived())
            break;
        watch_changes(directory);
    }

    return NULL;
}
```

Receiving client requests

The receiver_thread_function is responsible for handling communication from the client side. It receives data from the client using the recv function and performs actions based on the received data. Inside a loop, it receives data from the client and constructs the full path of the file or directory. Depending on the received data, it performs actions such as adding or deleting files/directories, modifying files, or handling exit or disconnection requests. The function manages file operations like creating or removing files/directories and writing received content. After each iteration, the function resets the data structure. Finally, it returns, indicating the end of the thread.

```
while ((received_bytes = recv(client_sock, &socketData, sizeof(SocketData), 
0)) > 0)
```

Sending changes to clients

The sender_thread_function is a thread function responsible for handling client communication. It dequeues clients from a queue and creates receiver threads to handle their communication. If the thread is not busy, it waits for clients to be available in the queue. Once a client is dequeued, it creates a receiver thread to handle the client's communication. If the thread is already busy, it waits for a signal or a condition variable to be signaled before proceeding. Depending on the status of the last file change, the function sends appropriate messages to the client. After processing the file change, it signals a semaphore to indicate completion. The loop continues until a signal is received or the thread is ready to exit. If a client was being processed, the function waits for the receiver thread to join. Finally, the function returns.

Clients

Client program takes directory name, port number and ip adress (optionally) as command line arguments. It opens the given directory if it exists. Otherwise new directory is created.

After setting required socket protocols, it connects to server port using socket. When connection is establish, It waits the first equalization process. Once this process finished, server inform the client. Then client watcher thread starts to travers its directory to catch changes.

Sending changes to server

The senderFunction sends the relevant SocketData structure over a socket based on the file information and status. For an ADDED status and a non-directory file, it sends the file content in chunks along with the initial socketData. For a DELETED status, it sends the corresponding socketData. For a MODIFIED status and a non-directory file, it sends the modified file content in chunks along with the socketData. Finally, it sends the last socketData to indicate the completion of the operation.

Receiving changes from server

The receiverFunction thread receives data over a socket in chunks, constructs a full path using the received filename and a directory name, writes the path and status to a log file, and performs various actions based on the received status, such as adding or deleting files, modifying file content, signaling the watcher thread, or terminating the client. It ensures thread safety by using a mutex, clears the received data structure, and continues the process for the next data.

Writing to the logfile

The writeToLog function is responsible for writing log messages to the logfile. It takes the filename and status as input parameters. A message buffer is created with a fixed size. Based on the provided status, a log message is formatted and stored in the message buffer. The log message indicates the status of the file, such as "ADDED", "DELETED", "MODIFIED", or "FINISH_EQUALIZE".The log message is then written to the logfile using the write function.

Signals

I have implemented a dedicated thread specifically designed to handle signals. This thread is responsible for receiving signals while all other threads block them. Upon receiving a signal, the signal handler thread sets a variable to 1. The remaining threads periodically check the value of this variable, and if it is set to 1, they gracefully break their ongoing tasks. Finally, a cleanup function is invoked to deallocate any allocated memory.

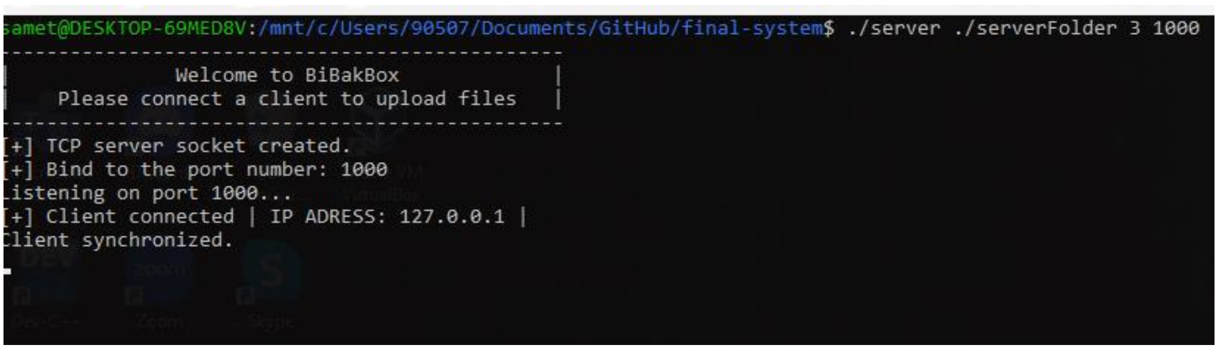
```
void* signal_handler_thread_func(void *arg){
    sigset_t mask;
    sigemptyset(&mask);
    siginfo_t info;

    sigaddset(&mask, SIGINT);
    sigaddset(&mask, SIGTSTP);

    if (sigwaitinfo(&mask, &info) == -1)
    {
        perror("sigwaitinfo");
        return NULL;
    }
    printf("Signal received... Program will terminate...\n");
    pthread_mutex_lock(&sig_mutex);
    signal_received = 1;
    pthread_mutex_unlock(&sig_mutex);


    return NULL;
}
```

If server program receives a signal, it inform all the clients to exit. Otherwise, If a client receives a signal it inform server program but only the client exit.

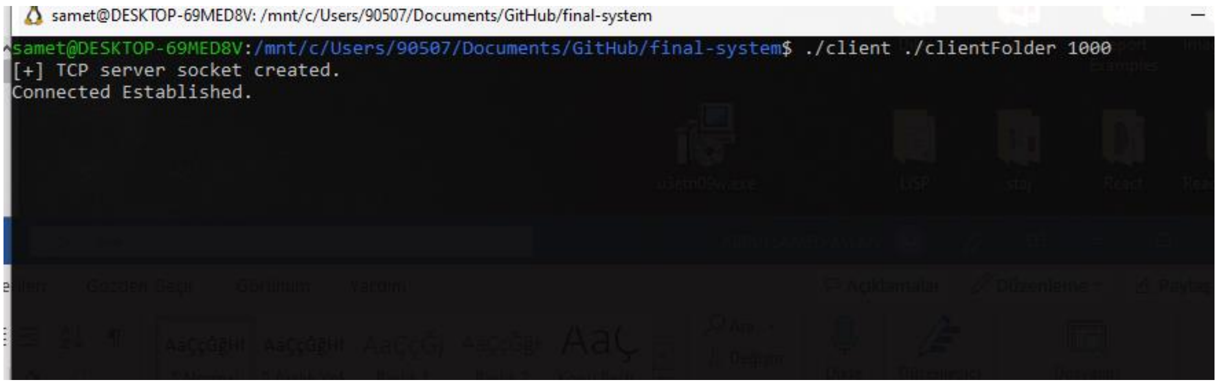
Test Cases

Case 1:

First connection synchronization

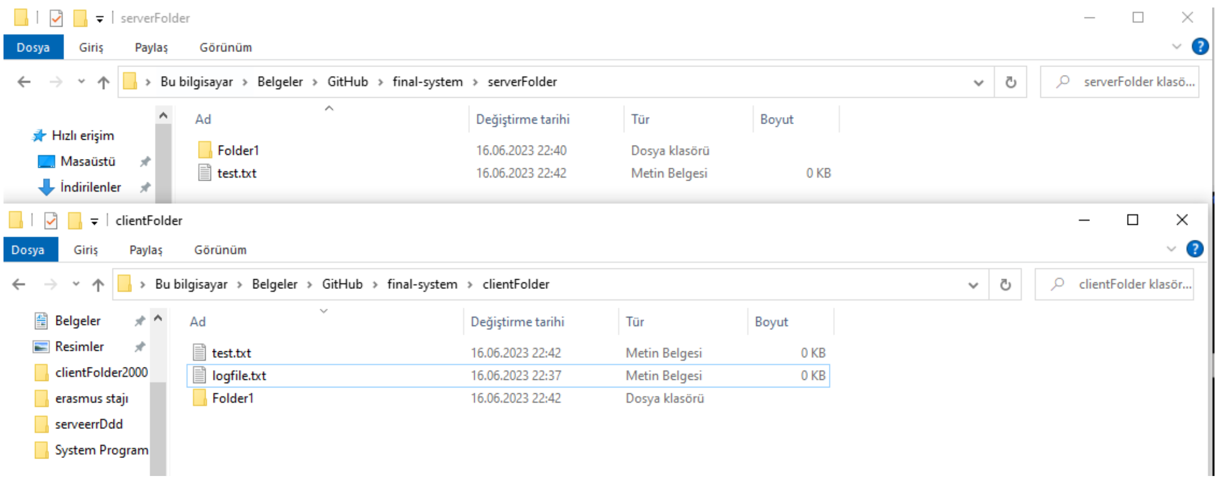


```
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./server ./serverFolder 3 1000
------------------------------------------------
|          Welcome to BiBakBox                 |
|     Please connect a client to upload files  |
------------------------------------------------
[+] TCP server socket created.
[+] Bind to the port number: 1000
Listening on port 1000...
[+] Client connected | IP ADRESS: 127.0.0.1 |
Client synchronized.
```



samet@DESKTOP-69MED8V: /mnt/c/Users/90507/Documents/GitHub/final-system

```
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./client ./clientFolder 1000
[+] TCP server socket created.
Connected Established.
```



serverFolder

Bu bilgisayar › Belgeler › GitHub › final-system › serverFolder

Ad	Değiştirme tarihi	Tür	Boyut
Folder1	16.06.2023 22:40	Dosya klasörü	
test.txt	16.06.2023 22:42	Metin Belgesi	0 KB

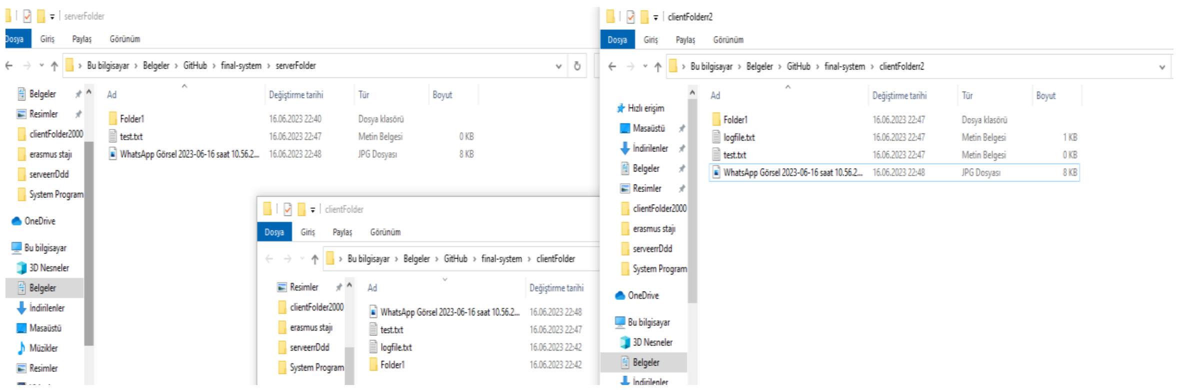
clientFolder

Bu bilgisayar › Belgeler › GitHub › final-system › clientFolder

Ad	Değiştirme tarihi	Tür	Boyut
test.txt	16.06.2023 22:42	Metin Belgesi	0 KB
logfile.txt	16.06.2023 22:37	Metin Belgesi	0 KB
Folder1	16.06.2023 22:42	Dosya klasörü	

Case2:

Server synchronized all the client when a client added a file.


serverFolder

Bu bilgisayar › Belgeler › GitHub › final-system › serverFolder

Ad	Değiştirme tarihi	Tür	Boyut
Folder1	16.06.2023 22:40	Dosya klasörü	
test.txt	16.06.2023 22:47	Metin Belgesi	0 KB
WhatsApp Görsel 2023-06-16 saat 10.56.2...	16.06.2023 22:48	JPG Dosyası	8 KB

clientFolder

Bu bilgisayar › Belgeler › GitHub › final-system › clientFolder

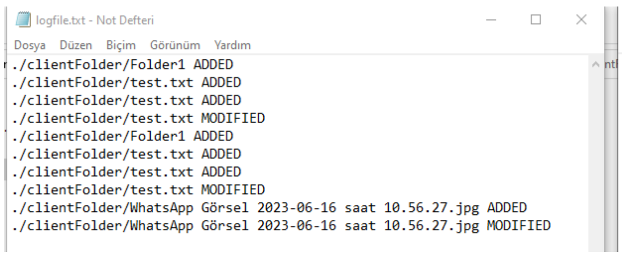
Ad	Değiştirme tarihi
WhatsApp Görsel 2023-06-16 saat 10.56.2...	16.06.2023 22:48
test.txt	16.06.2023 22:47
logfile.txt	16.06.2023 22:42
Folder1	16.06.2023 22:42

clientFolderr2

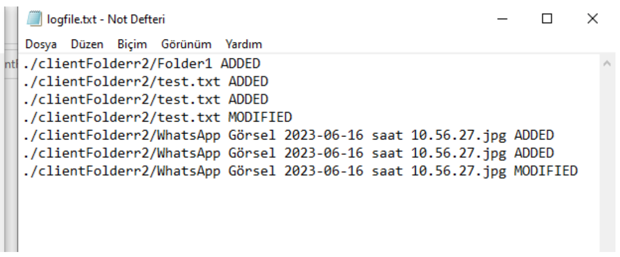
Bu bilgisayar › Belgeler › GitHub › final-system › clientFolderr2

Ad	Değiştirme tarihi	Tür	Boyut
Folder1	16.06.2023 22:47	Dosya klasörü	
logfile.txt	16.06.2023 22:47	Metin Belgesi	1 KB
test.txt	16.06.2023 22:47	Metin Belgesi	0 KB
WhatsApp Görsel 2023-06-16 saat 10.56.2...	16.06.2023 22:48	JPG Dosyası	8 KB

Client 1 and Client 2 Log files


logfile.txt - Not Defteri

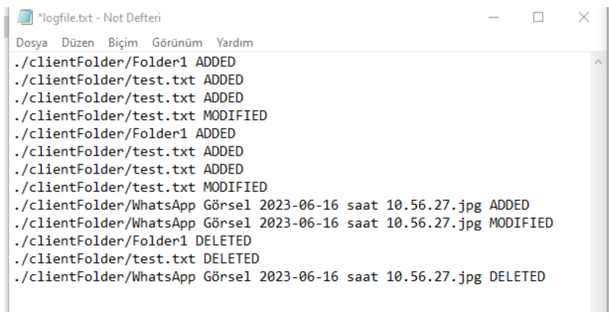
```
./clientFolder/Folder1 ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt MODIFIED
./clientFolder/Folder1 ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt MODIFIED
./clientFolder/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolder/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg MODIFIED
```


logfile.txt - Not Defteri

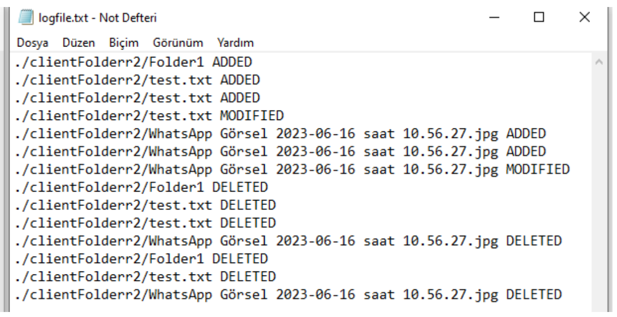
```
./clientFolderr2/Folder1 ADDED
./clientFolderr2/test.txt ADDED
./clientFolderr2/test.txt ADDED
./clientFolderr2/test.txt MODIFIED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg MODIFIED
```

Case 3:

Once Client 1 deletes all the files, Server provide the synchronization.


*logfile.txt - Not Defteri

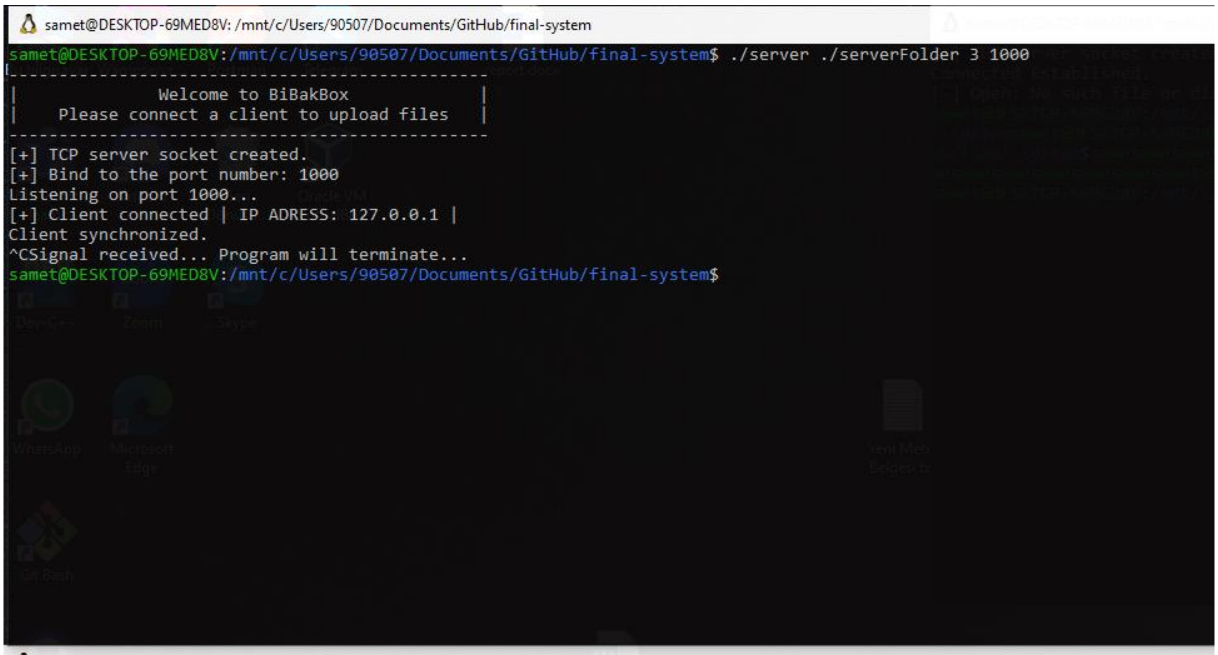
```
./clientFolder/Folder1 ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt MODIFIED
./clientFolder/Folder1 ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt ADDED
./clientFolder/test.txt MODIFIED
./clientFolder/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolder/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg MODIFIED
./clientFolder/Folder1 DELETED
./clientFolder/test.txt DELETED
./clientFolder/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg DELETED
```


logfile.txt - Not Defteri

```
./clientFolderr2/Folder1 ADDED
./clientFolderr2/test.txt ADDED
./clientFolderr2/test.txt ADDED
./clientFolderr2/test.txt MODIFIED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg ADDED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg MODIFIED
./clientFolderr2/Folder1 DELETED
./clientFolderr2/test.txt DELETED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg DELETED
./clientFolderr2/Folder1 DELETED
./clientFolderr2/test.txt DELETED
./clientFolderr2/WhatsApp Görsel 2023-06-16 saat 10.56.27.jpg DELETED
```

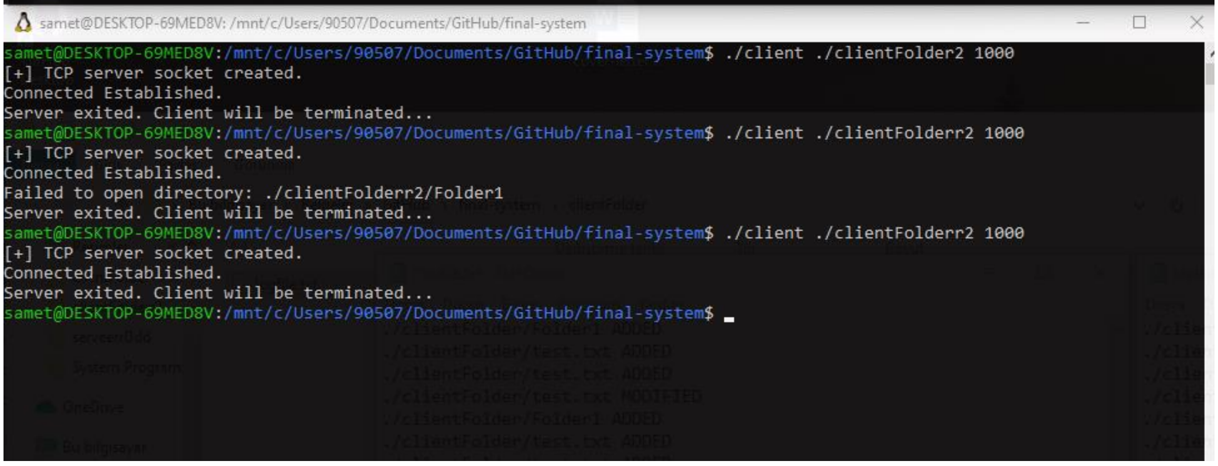
Case 4:

Servers received a signal



samet@DESKTOP-69MED8V: /mnt/c/Users/90507/Documents/GitHub/final-system

```
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./server ./serverFolder 3 1000
------------------------------------------------
|              Welcome to BiBakBox               |
|    Please connect a client to upload files     |
------------------------------------------------
[+] TCP server socket created.
[+] Bind to the port number: 1000
Listening on port 1000...
[+] Client connected | IP ADRESS: 127.0.0.1 |
Client synchronized.
^CSignal received... Program will terminate...
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$
```

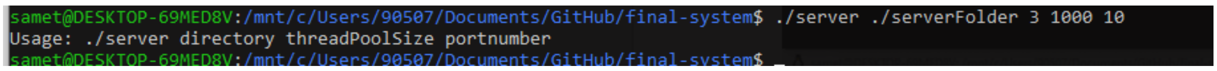


samet@DESKTOP-69MED8V: /mnt/c/Users/90507/Documents/GitHub/final-system

```
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./client ./clientFolder2 1000
[+] TCP server socket created.
Connected Established.
Server exited. Client will be terminated...
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./client ./clientFolderr2 1000
[+] TCP server socket created.
Connected Established.
Failed to open directory: ./clientFolderr2/Folder1
Server exited. Client will be terminated...
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./client ./clientFolderr2 1000
[+] TCP server socket created.
Connected Established.
Server exited. Client will be terminated...
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$
```

Case 5

Wrong command line arguments



```
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$ ./server ./serverFolder 3 1000 10
Usage: ./server directory threadPoolSize portnumber
samet@DESKTOP-69MED8V:/mnt/c/Users/90507/Documents/GitHub/final-system$
```

Compiling and Running

Compile the code

`make`

Run Server

`./server <dirname> <thread_poolSize> <portNumber>`

Run Server

`./client <dirname> <portNumber>`

Clean the unnecessary files

`make clean`