

Gebze Technical University

Department of Computer Engineering



Spring 2023 - CSE 344
Midterm Project Report

GONCA EZGİ ÇAKIR
151044054

1. Solution Approach

In summary, this program is implemented as a basic server client architecture. Communication between client and server provided by *server FIFOs*. Client processes handled by *children processes* on the server side by communicating through *client FIFOs* and the synchronization between processes are provided by *shared memory, counting semaphores* and *binary semaphores (mutex)*. In order to prevent mutual exclusion and race condition *reader - writer paradigm* is implemented by binary semaphores. *Signals* are handled on both client and server sides.

1.1. Client Side

Client connects to server by **./biboClient Connect/tryConnect serverPid** arguments. If the connection succeeds, client *sends request* to server such as help, list, upload, download, readF, writeT, quit, killServer and gets response till quited or terminated by SIGINT.

Client program implementation contents are listed below.

- Stores the argument in a struct name as *BiboClient*. Connect or tryConnect as *connectInfo*, client process id as *clientPid*, server process id as *serverPid* and *creates the client's fifo*.
- Client opens the server fifo by using serverPid variable and sends the biboClient struct to server side, then waits for response to start commanding.
- *If there is no space* in the client queue and the connectInfo is "Connect" then client *waits for another client to quit*, till then it can send any request to server.

If there is no space in the client queue and the connectInfo is "tryConnect" then server sends SIGUSR1 signal in order to *terminate that client*.

If there is a space in the client queue on the server side, client receives an information on the read end side of the server fifo. Then *starts to write requests to server fifo's* write side and wait for response again on the server fifo's read side.

- While client waits for a response server firstly sends the response size, client reads that information and allocates space for the response then server writes the response then client reads this response and stores to print to screen. All these read and write processes are provided on server fifo.
- If the client quits frees all its resources and terminates. If it receives SIGINT signal by Ctrl-C it writes to server fifo for the last time but changes the connectInfo to "kill" in order to handle closing all clients and server on the server side; frees all its resources and terminates.

Error Handling

- Given serverPid checked, if there is no fifo then the client receives and SIGUSR1 and terminates.
- If there is more or less argument than three client program prints an error message on screen and terminates.

1.2. Server Side

Client connects to server by **./biboServer directoryName maxClient** arguments. Server gets client connections and handles them by children processes, communicates through client fifos and sends related responses to each client. It synchronizes all resources and children processes till receives killServer request or terminated by SIGINT .

Server program implementation contents are listed below.

- Stores the argument in a struct name as *BiboServer*. Directory name as *dirName* and max client number as *maxClient* also creates the server fifo.
- Server directory created/opened then shared memory for children process counting semaphores, binary semaphores (mutex), counter variables is created and initialized. Client queue is created as separate shared memory and initialized.
- Server starts to wait for client connection if *there is a space* in the client queue and the then *client connects and starts to send requests*.

If *there is no space* in the client queue and the *connectInfo* is “*tryConnect*” then client can't connect, server sends SIGUSR1 signal in order to *terminate that client*.

If *there is no space* in the client queue and the *connectInfo* is “*Connect*” then client *waits for another client to quit*, till then it can send any request to server.

- When a client connects the client counter is incremented and the client id created in order to print server results.

Semaphore Usage

- Counting semaphore *sem_clients* is set to maxClient amount at the beginning; called *sem_wait()* each time a client connected and called *sem_post()* when a client disconnected.
- Binary semaphore *sem_queue* is set to 1 amount at the beginning; called *sem_wait()* each time before a queue is manipulated and called *sem_post()* after.
- Binary semaphore *sem_logfile* is set to 1 amount at the beginning; called *sem_wait()* each time before a logfile is manipulated and called *sem_post()* after.

- Binary semaphores *readTry*, *rmutex*, *rsc*, *wmutex* set to 1 amount at the beginning; and used in the reader writer paradigm when a file is readed or writed. For the *upload* and *download* commands *writer* is applied; for the *readF* reader is applied.

Command Details

- **help:** sets the response with a list of commands as a string and returns the string.
- **list:** lists the current directory content by using dirent struct and adds to a string till it reaches directory '.' or '..' and returns the string.
- **help + command:** sets the response with the related command's explanation as a string and returns the string
- **upload:** sets the source to client directory and destination to server directory. Reads the source file and creates the file in the destination and writes into it. If the file is already exist at the destination or not exist at the source it sets an error message as response.
- **download:** sets the source to server directory and destination to client directory. Reads the source file and creates the file in the destination and writes into it. If the file is already exist at the destination or not exist at the source it sets an error message as response.
- **readF:** reads the given file's content, sets into a string and returns the string. If there a line number given it only sets that line's content to string and returns the string. NOT SUPPORTED FOR LARGE FILES.
- **writeT:** NOT SUPPORTED.
- **quit:** details written to log file and child process ends, clients end when it requested to quit.
- **killServer:** calls signal handler in order to terminate all client processes, children processes and server process itself. All the children and client process pids previously stored in an array, so they are terminated according to these content.

Error Handling

- If there is more or less argument than three server program prints an error message on screen and terminates.
- Double instance is control by creating a temp file and removing at the end of the server process. So this temp file occurency at the beginning of the program; if it occurs that means server process already started; otherwise it is the only instance.

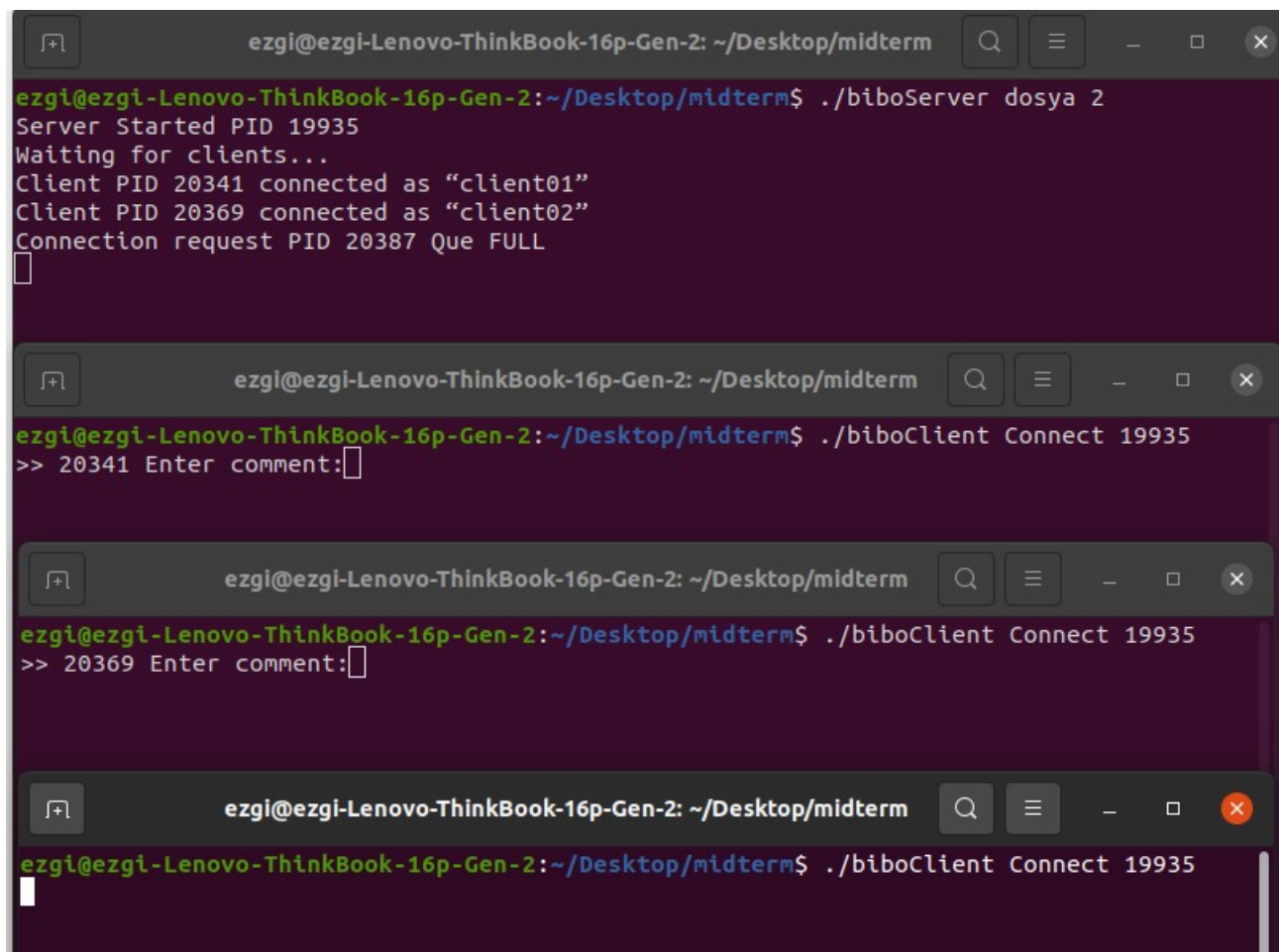
- Log file access is restricted fro clients by checking the filename argument at the beginning of the upload, download or readF commands. If the filename equals to log file name then request receives and related error message.

2. Test Results

Multiple test case senarios applied and screenshots are added. Below you will see some program input result.

NOTE: Client Pids printed before 'Enter comment' line just to show the test result much clearly.

2.1. Client Queue is Full, New Client Comes With 'Connect'



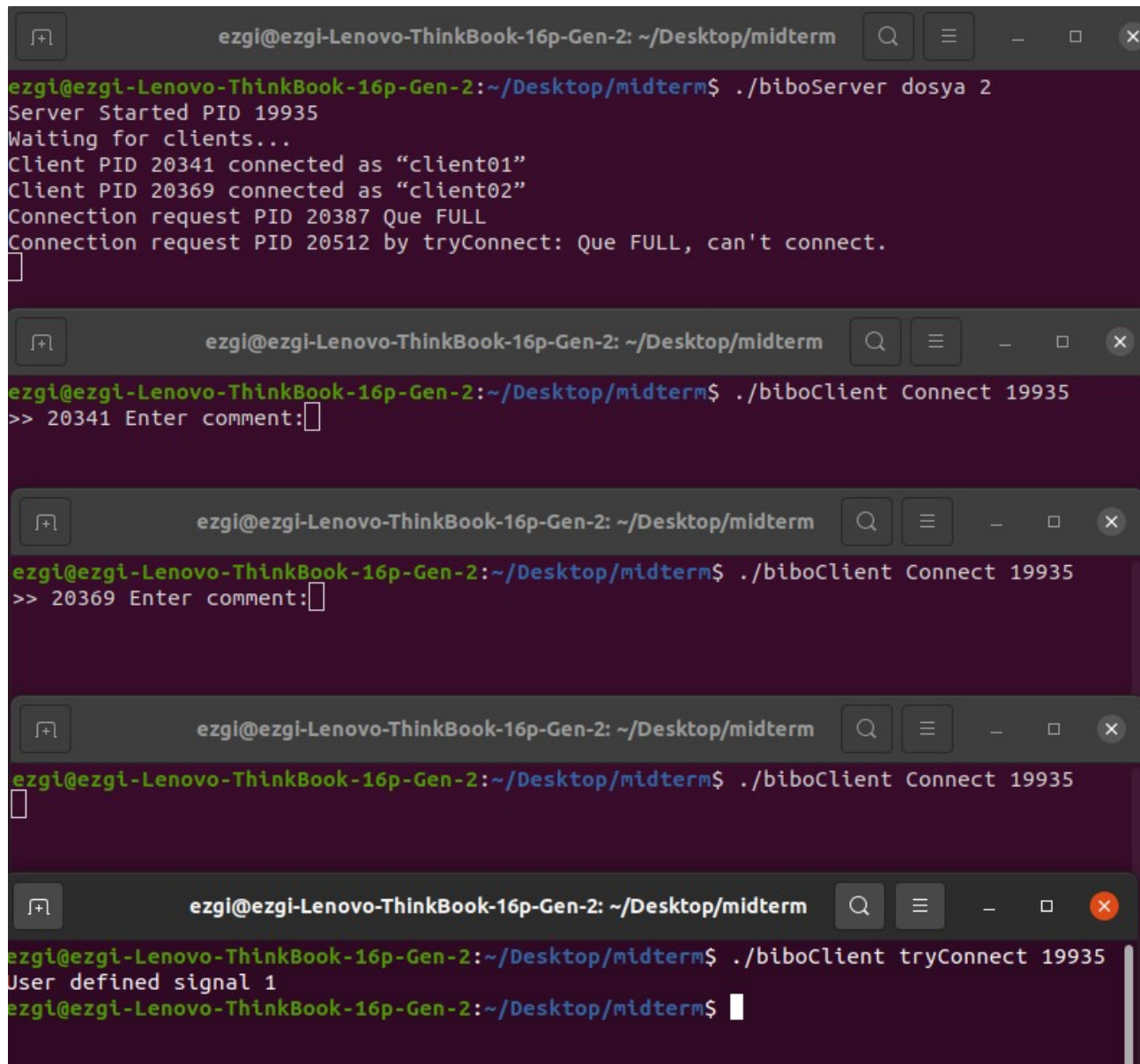
```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...
Client PID 20341 connected as "client01"
Client PID 20369 connected as "client02"
Connection request PID 20387 Que FULL
█

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20341 Enter comment:█

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20369 Enter comment:█

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
█
```

2.2. Client Queue is Full, New Client Comes With 'tryConnect'



```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...
Client PID 20341 connected as "client01"
Client PID 20369 connected as "client02"
Connection request PID 20387 Que FULL
Connection request PID 20512 by tryConnect: Que FULL, can't connect.

```

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20341 Enter comment:

```

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20369 Enter comment:

```

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935

```

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient tryConnect 19935
User defined signal 1
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$

```

2.3. When a Client Quits, Another Client From Queue is Started

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...
Client PID 20341 connected as "client01"
Client PID 20369 connected as "client02"
Connection request PID 20387 Que FULL
Connection request PID 20512 by tryConnect: Que FULL, can't connect.

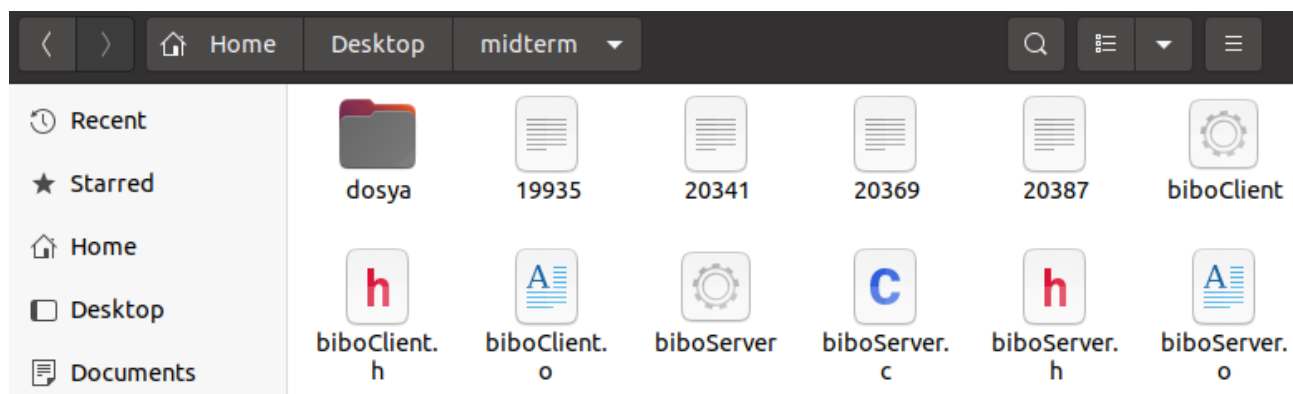
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20341 Enter comment:

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20369 Enter comment:

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient tryConnect 19935
User defined signal 1
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

FIFOs Before Terminating




```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...
Client PID 20341 connected as "client01"
Client PID 20369 connected as "client02"
Connection request PID 20387 Que FULL
Connection request PID 20512 by tryConnect: Que FULL, can't connect.
client01 disconnected.
Client PID 20387 connected as "client03"

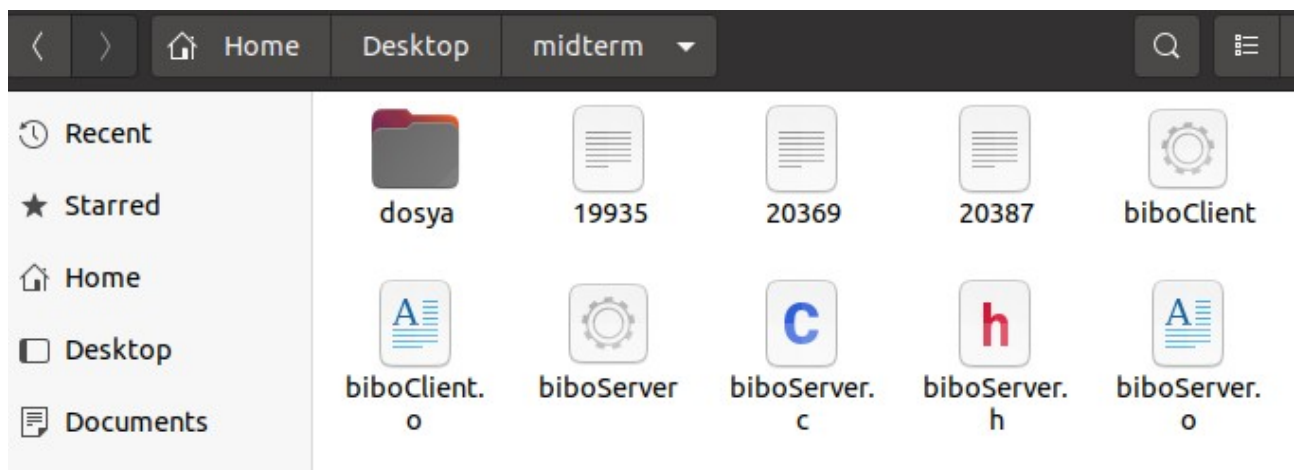
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20341 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.
bye.

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20369 Enter comment:

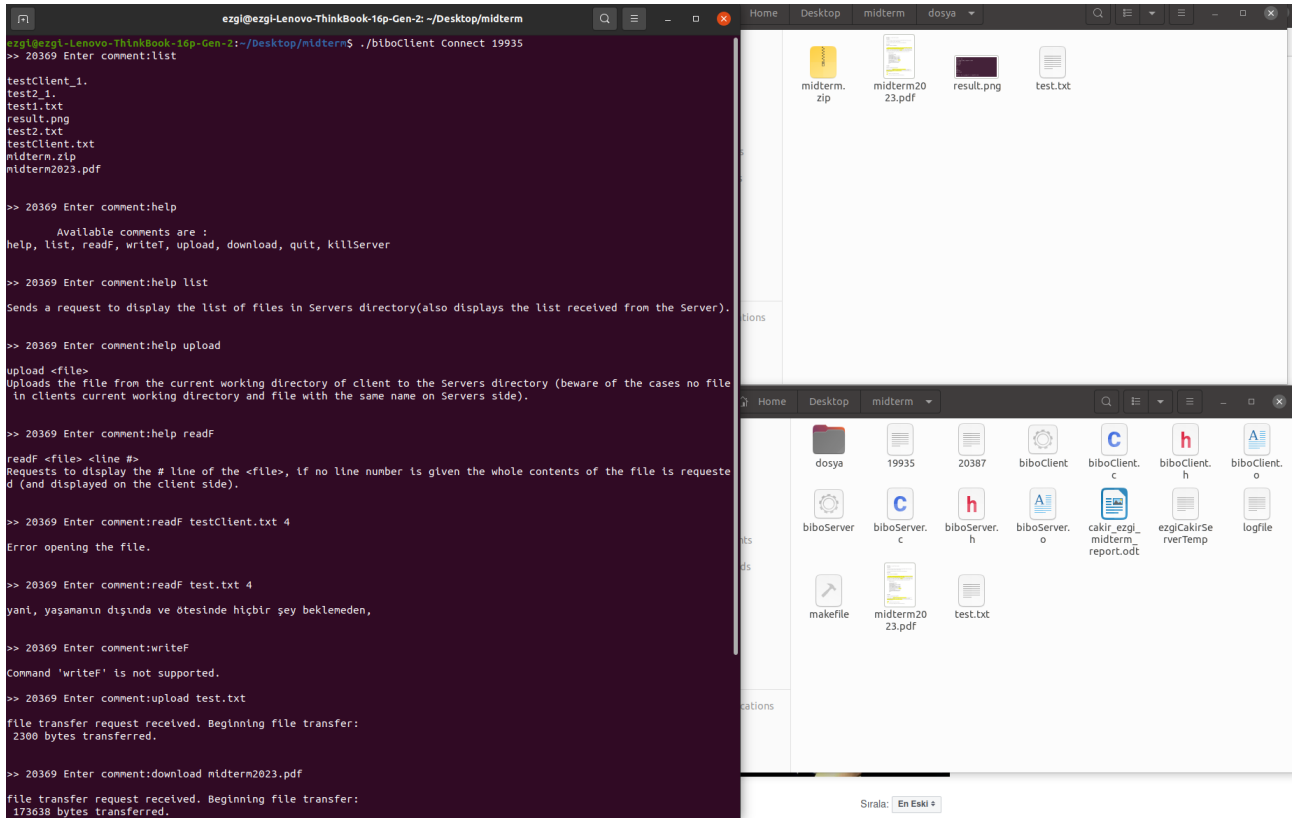
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20387 Enter comment:
```

FIFOs After Terminating One Client



2.4. Client Sends Requests

(Some cases are handled as error handlings)



2.5. Client Sends killServer

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...
Client PID 20341 connected as "client01"
Client PID 20369 connected as "client02"
Connection request PID 20387 Que FULL
Connection request PID 20512 by tryConnect: Que FULL, can't connect.
client01 disconnected.
Client PID 20387 connected as "client03"
burda2
filename:testClient.txt
burda2
filename:test.txt
client02 disconnected.
kill signal from client03. terminating.

SIGINT signal is caught.
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20341 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.
bye.

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
>> 20369 Enter comment:upload x.txt

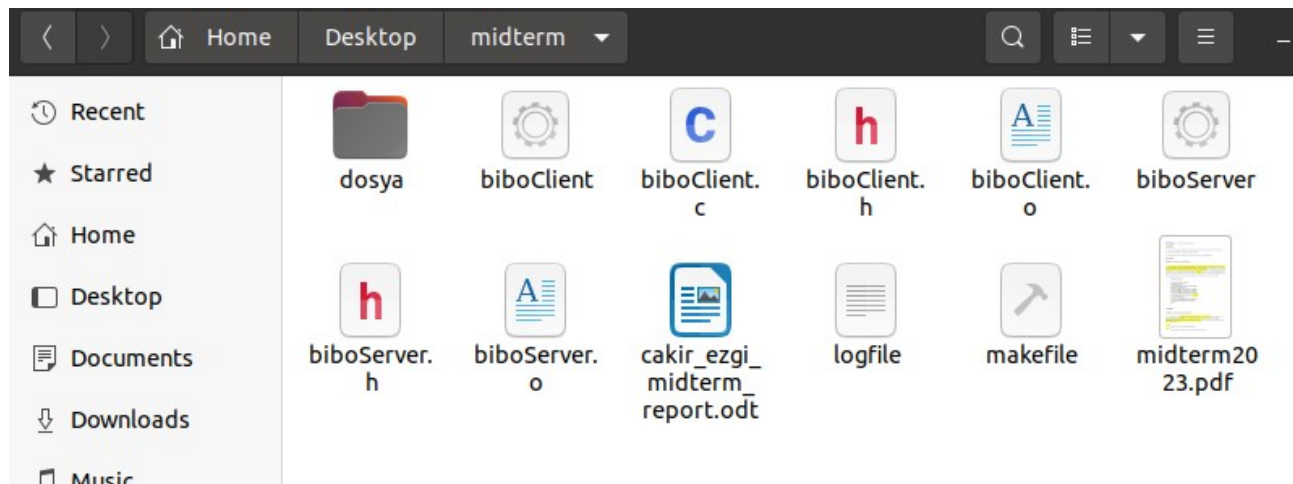
Given file name is not found.

>> 20369 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 19935
>> 20387 Enter comment:killServer
kill
SIGINT signal is caught.
kill
SIGINT signal is caught.
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

FIFOs After Terminating Server



ps aux - No Zombies

```

root      36142  0.0  0.0      0      0 ?        I<    06:23   0:00 [xfs_mru_cache]
root      36143  0.0  0.0      0      0 ?        I     06:23   0:00 [kworker/5:1-events]
root      36147  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsIO]
root      36148  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36149  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36150  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36151  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36152  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36153  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36154  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36155  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36156  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36157  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36158  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36159  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36160  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36161  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36162  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36163  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsCommit]
root      36164  0.0  0.0      0      0 ?        S     06:23   0:00 [jfsSync]
root      37430  0.1  0.0      0      0 ?        I     06:24   0:00 [kworker/2:0-events]
root      38032  0.0  0.0      0      0 ?        I     06:24   0:00 [kworker/3:1-events]
root      38090  0.0  0.0      0      0 ?        I     06:24   0:00 [kworker/0:0-events]
root      38456  0.0  0.0      0      0 ?        I     06:27   0:00 [kworker/10:2-events]
root      38476  0.1  0.0      0      0 ?        I     06:27   0:00 [kworker/6:1-events]
root      38596  0.0  0.0      0      0 ?        I     06:28   0:00 [kworker/4:1-events]
root      38608  0.0  0.0      0      0 ?        I     06:29   0:00 [kworker/14:1-events]
root      38609  0.0  0.0      0      0 ?        I     06:29   0:00 [kworker/8:1]
ezgi      38637  1.1  0.1 1028480 50616 ?        Sl    06:29   0:01 /usr/bin/gnome-screenshot --g
root      38664  0.0  0.0      0      0 ?        I     06:29   0:00 [kworker/5:0]
root      38668  0.0  0.0      0      0 ?        I     06:29   0:00 [kworker/2:1]
root      38673  0.0  0.0      0      0 ?        I     06:30   0:00 [kworker/3:0-events]
root      38686  0.0  0.0      0      0 ?        I     06:30   0:00 [kworker/13:1-mm_percpu_wq]
ezgi      38690  1.0  0.1 823980 60240 ?        Sl    06:30   0:00 /usr/bin/gedit --gapplication
root      38756  0.0  0.0      0      0 ?        I     06:31   0:00 [kworker/11:0-events]
root      38757  0.0  0.0      0      0 ?        I     06:31   0:00 [kworker/15:0-events]
root      38760  0.0  0.0      0      0 ?        I     06:31   0:00 [kworker/9:1]
ezgi      38769  0.0  0.0 14232 3588 pts/0    R+    06:31   0:00 ps aux
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$

```

2.6. Error Handling Cases – 1 (File Access, Invalid Command)

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 38551
>> 38579 Enter comment:upload deneme.txt

Given file name is not found.

>> 38579 Enter comment:download deneme.txt

Given file name is not found.

>> 38579 Enter comment:readF deneme.txt

Error opening the file.

>> 38579 Enter comment:read

Command 'read' is not supported.

>> 38579 Enter comment:blabla

Command 'blabla' is not supported.

>> 38579 Enter comment:upload logfile

Client doesn't have permission to acces to logfile.

>> 38579 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.
bye.
```

Log

```
Open ▼ [icon]
1 Client Pid: 38579, Client Id: client01, Request: upload
2 Client Pid: 38579, Client Id: client01, Request: download
3 Client Pid: 38579, Client Id: client01, Request: readF
4 Client Pid: 38579, Client Id: client01, Request: read
5 Client Pid: 38579, Client Id: client01, Request: blabla
6 Client Pid: 38579, Client Id: client01, Request: upload
7 Client Pid: 38579, Client Id: client01 quited.
```


2.7. Error Handling Cases – 2 (Invalid Number of Console Arguments)

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya
Usage: There should be 3 console arguments. [biboServer <dirname> <maxNumOfClients>]: Success
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2 hgh
Usage: There should be 3 console arguments. [biboServer <dirname> <maxNumOfClients>]: Success
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect
Usage: There should be 3 console arguments. [biboClient <Connect/tryConnect> ServerPID]: Success
Failed to open server fifo.: No such file or directory
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 3 ghj
Usage: There should be 3 console arguments. [biboClient <Connect/tryConnect> ServerPID]: Success
Failed to open server fifo.: No such file or directory
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

2.8. Error Handling Cases – 3 (Double Instance of Server)

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...

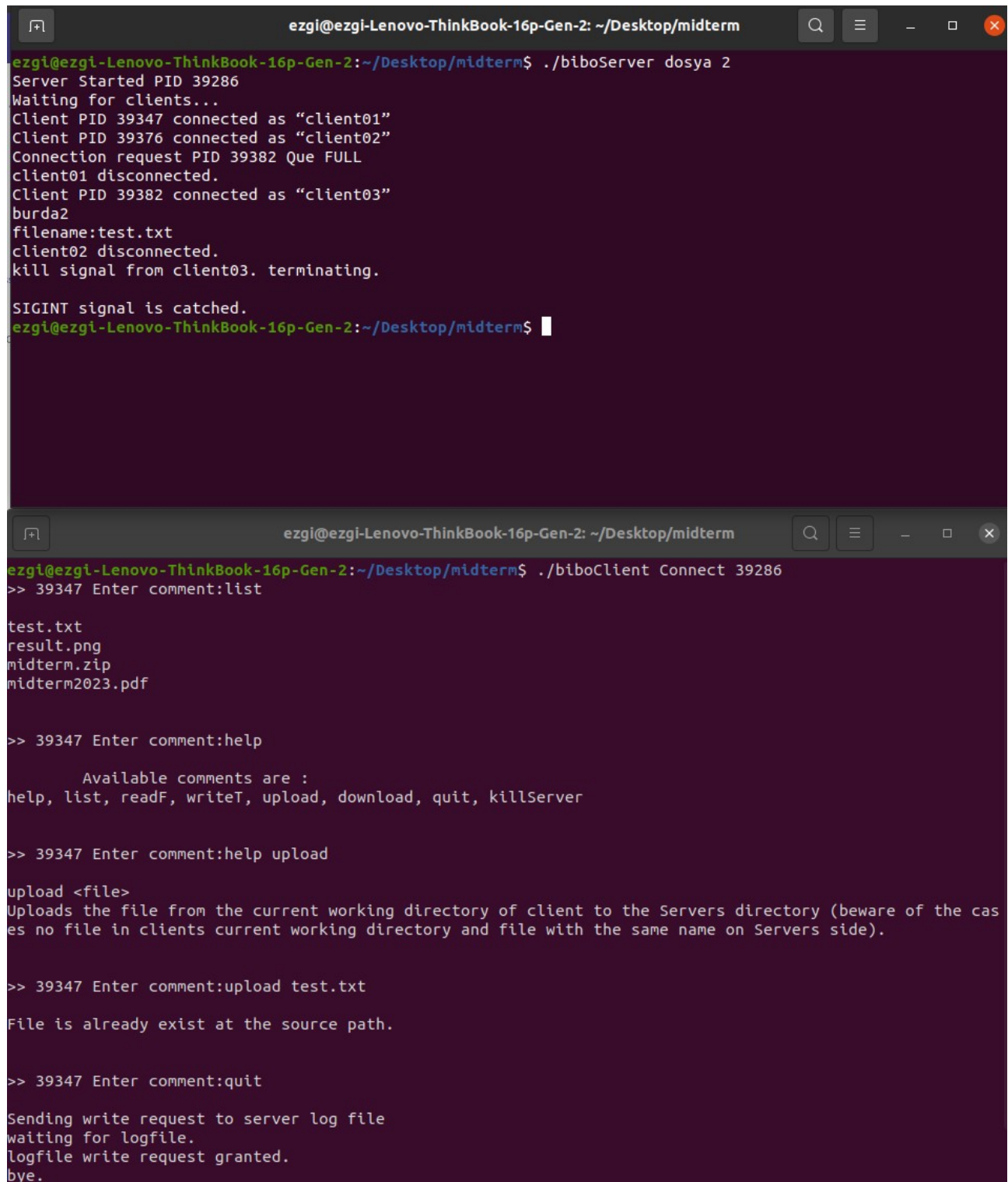
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Another instance of the server process is already running.
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

2.9. Error Handling Cases – 4 (Wrong Server Pid on Client Side)

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 19935
Waiting for clients...

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 199
Failed to open server fifo.: No such file or directory
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

2.10. Log File Example



The image displays two terminal windows from a Linux system, showing the execution of a custom network protocol between a server and a client.

The top terminal window shows the server's perspective. The user runs `./biboServer dosya 2`. The server starts with PID 39286 and enters a loop waiting for clients. It successfully connects three clients: 'client01' (PID 39347), 'client02' (PID 39376), and 'client03' (PID 39382). Client01 disconnects. Client02 sends a filename 'test.txt'. Client03 sends a 'kill' signal, which the server catches as a SIGINT and terminates.

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboServer dosya 2
Server Started PID 39286
Waiting for clients...
Client PID 39347 connected as "client01"
Client PID 39376 connected as "client02"
Connection request PID 39382 Que FULL
client01 disconnected.
Client PID 39382 connected as "client03"
burda2
filename:test.txt
client02 disconnected.
kill signal from client03. terminating.

SIGINT signal is caught.
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

The bottom terminal window shows the client's perspective. The user runs `./biboClient Connect 39286`. The client connects to the server and enters a loop of commands. It lists files (test.txt, result.png, midterm.zip, midterm2023.pdf), requests help, and attempts to upload 'test.txt', which fails because the file already exists at the source path. Finally, it sends a 'quit' command, which the server logs.

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 39286
>> 39347 Enter comment:list

test.txt
result.png
midterm.zip
midterm2023.pdf

>> 39347 Enter comment:help

Available comments are :
help, list, readF, writeT, upload, download, quit, killServer

>> 39347 Enter comment:help upload

upload <file>
Uploads the file from the current working directory of client to the Servers directory (beware of the cas
es no file in clients current working directory and file with the same name on Servers side).

>> 39347 Enter comment:upload test.txt

File is already exist at the source path.

>> 39347 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.
bye.
```



```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 39286
>> 39376 Enter comment:readF test.txt 10

beyaz gömleğinle bir laboratuvarıda

>> 39376 Enter comment:writeT

writeT is a valid command but it is not supported for this implementation.

>> 39376 Enter comment:write

Command 'write' is not supported.

>> 39376 Enter comment:quit

Sending write request to server log file
waiting for logfile.
logfile write request granted.
bye.

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$

ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ ./biboClient Connect 39286
>> 39382 Enter comment:help list

Sends a request to display the list of files in Servers directory(also displays the list received from the S
erver).

>> 39382 Enter comment:killServer
kill
SIGINT signal is caught.
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```

Log File

```
Open
1 Client Pid: 39347, Client Id: client01, Request: list
2 Client Pid: 39347, Client Id: client01, Request: help
3 Client Pid: 39347, Client Id: client01, Request: help
4 Client Pid: 39347, Client Id: client01, Request: upload
5 Client Pid: 39347, Client Id: client01 quited.
6 Client Pid: 39376, Client Id: client02, Request: readF
7 Client Pid: 39376, Client Id: client02, Request: writeT
8 Client Pid: 39376, Client Id: client02, Request: write
9 Client Pid: 39376, Client Id: client02 quited.
10 Client Pid: 39382, Client Id: client03, Request: help list
```

3. Makefile

You can see makefile content down below. It only compiles the codes with warning flags by “make” command and cleans .o files with “make clean” command.



The screenshot shows a code editor with four tabs: `biboClient.c`, `biboClient.h`, `biboServer.c`, and `biboServer.h`. The active tab is `makefile`. The content of the `makefile` is as follows:

```
1 all: midterm
2
3 midterm:
4     gcc -c biboClient.c
5     gcc -c biboServer.c
6     gcc biboClient.o -o biboClient -lrt -lm -lpthread -Wall -std=c99 -pedantic
7     gcc biboServer.o -o biboServer -lrt -lm -lpthread -Wall -std=c99 -pedantic
8
9 clean:
10     rm *.o biboClient biboServer
11
```



The screenshot shows a terminal window with the prompt `ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2: ~/Desktop/midterm`. The user has executed the following commands and received the following output:

```
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~$ cd Desktop/midterm/
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ make clean
rm *.o biboClient biboServer
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$ make
gcc -c biboClient.c
gcc -c biboServer.c
gcc biboClient.o -o biboClient -lrt -lm -lpthread -Wall -std=c99 -pedantic
gcc biboServer.o -o biboServer -lrt -lm -lpthread -Wall -std=c99 -pedantic
ezgi@ezgi-Lenovo-ThinkBook-16p-Gen-2:~/Desktop/midterm$
```