CSE344 MIDTERM REPORT

SÜLEYMAN KORAMAZ
1901042615

SYSTEM

SERVER SIDE

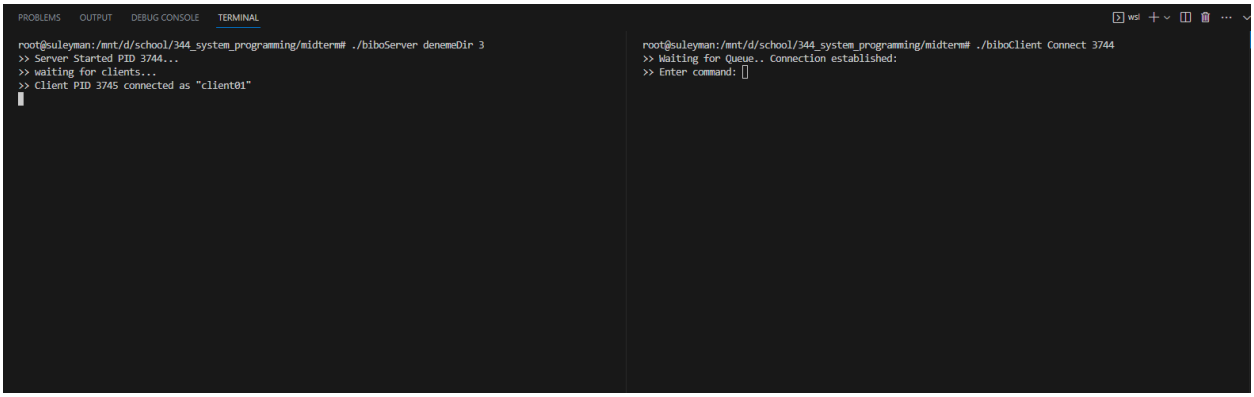
1- Create semaphore.
2- Create dictionary, specified with argv[1].
3- Go to the dictionary with chdir.
4- Create log file.
5- Set the signals.
6- Create FIFO in /tmp.
7- Create shared memory.
8- Attach the sharedQueue to shared memory.
9- Set the semaphore.
10- Get client_pid which try to connect.
11- If queue is full, make it to wait for slot int the queue with semaphore.
12- If queue is not full, create FIFO for connected client.
13- Create FIFO for connected client response messages.
14- Add client to sharedQueue.
15- Create child process for client with fork.
16- Call handleClientRequest function until it returns 1 or 2 which returned when request is quit – killServer.
17- In the handleClientRequest function, write request to log file.
18- Send request to process function.
19- Complete request in process function:
    a- write excepted string to fifo (help).
    b- write available commands to fifo (list)
    c- Read specified file and write it to fifo (readF)
    d- Write taken string to specified file (writeF)
    e- Move specified file in Server folder to client side by using rename method (download)
    f- Move specified file in client side to Server folder by using rename method (upload)
    g- Terminate client and remove it from sharedQueue (quit)
    h- Terminate all clients in the sharedQueue and terminate Server (killServer)
20- terminateAllClients function terminate all clients in the sharedQueue with kill function by using client_pid.

CLIENT SIDE

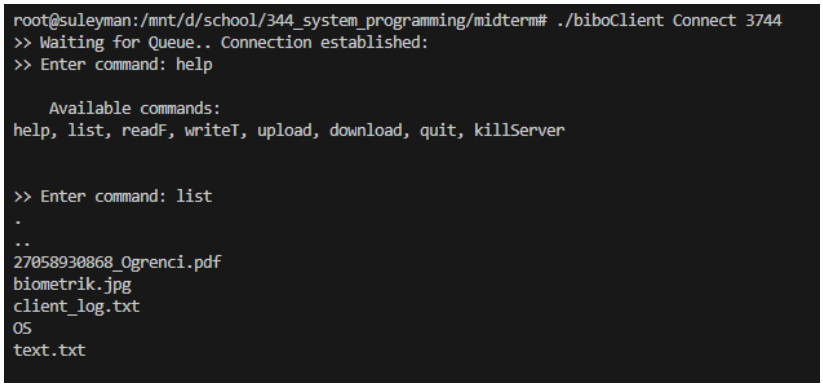
1- Set the signals.
2- Create FIFO
3- Open server fifo and write client pid to it. It is kind a connection request.
4- If server queue is full, it will wait until a empty slot.
5- After connection, call handleUserInput function.
6- This function in while loop until terminate signal or quit request.
7- It takes user input as a request, write it to client fifo and take response with reading client response fifo.

RESULTS

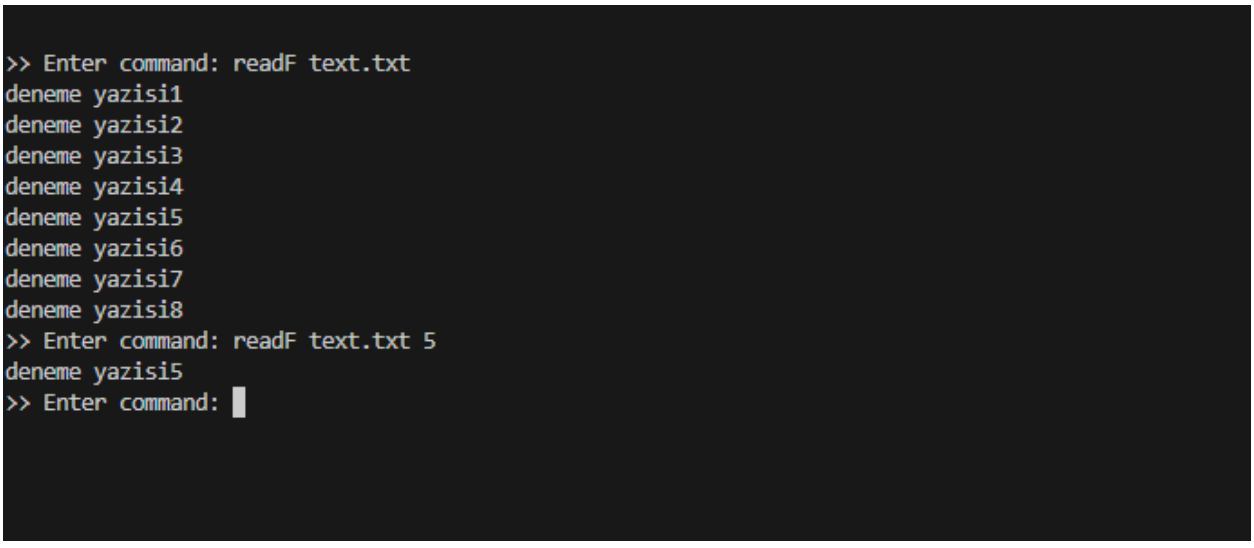
Connection:



help – list



readF



RESULTS

writeF

```
>> Enter command: writeF text.txt 4 hello world

    write process completed succesfully..

>> Enter command: readF text.txt
deneme yazisi1
deneme yazisi2
deneme yazisi3
hello world
deneme yazisi5
deneme yazisi6
deneme yazisi7
deneme yazisi8
>> Enter command:
```

download:

```
>> Enter command: list
.
..
27058930868_Ogrenci.pdf
biometrik.jpg
client_log.txt
OS
text.txt

>> Enter command: download text.txt

    file transfer request received. Beginning file transfer:
    123 bytes transferred:

>> Enter command: download 27058930868_Ogrenci.pdf

    file transfer request received. Beginning file transfer:
    103488 bytes transferred:

>> Enter command: list
.
..
biometrik.jpg
client_log.txt
OS

>> Enter command:
```

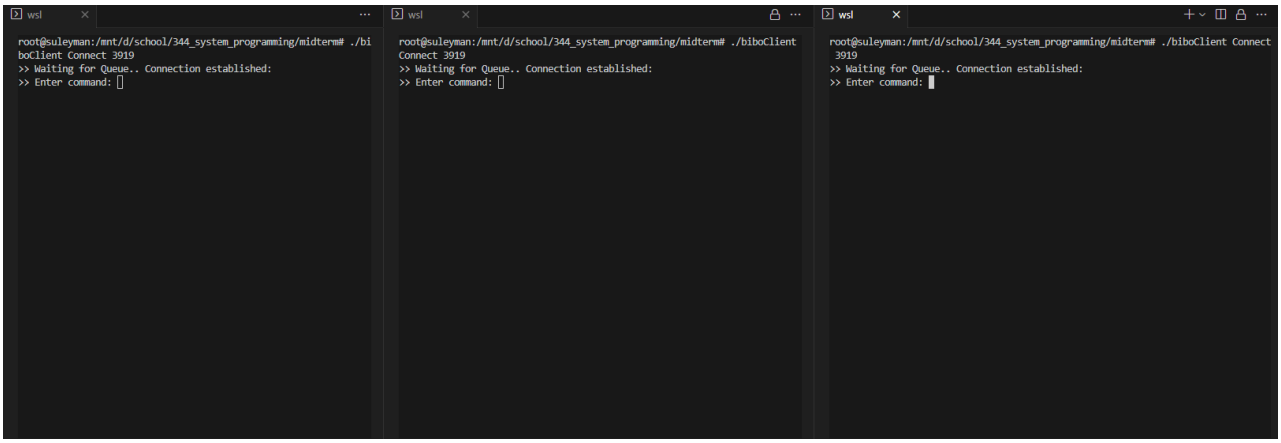
upload:

```
>> Enter command: upload 27058930868_Ogrenci.pdf

    file transfer request received. Beginning file transfer:
    103488 bytes transferred:

>> Enter command: list
.
..
27058930868_Ogrenci.pdf
biometrik.jpg
client_log.txt
OS

>> Enter command:
```

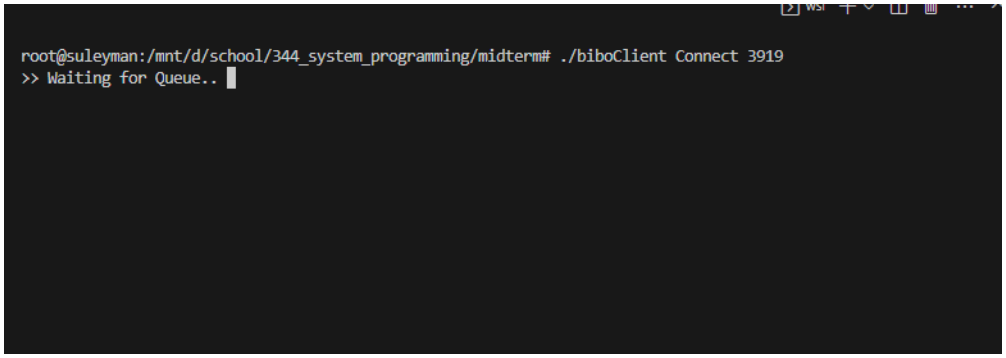
RESULTS

Max client, wait in queue, quit, when the slot is empty connect test:

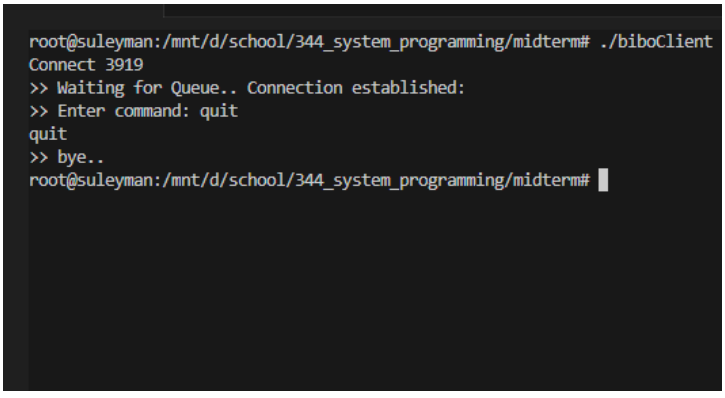
client01, client02, client03:



client04:



client02 quit:

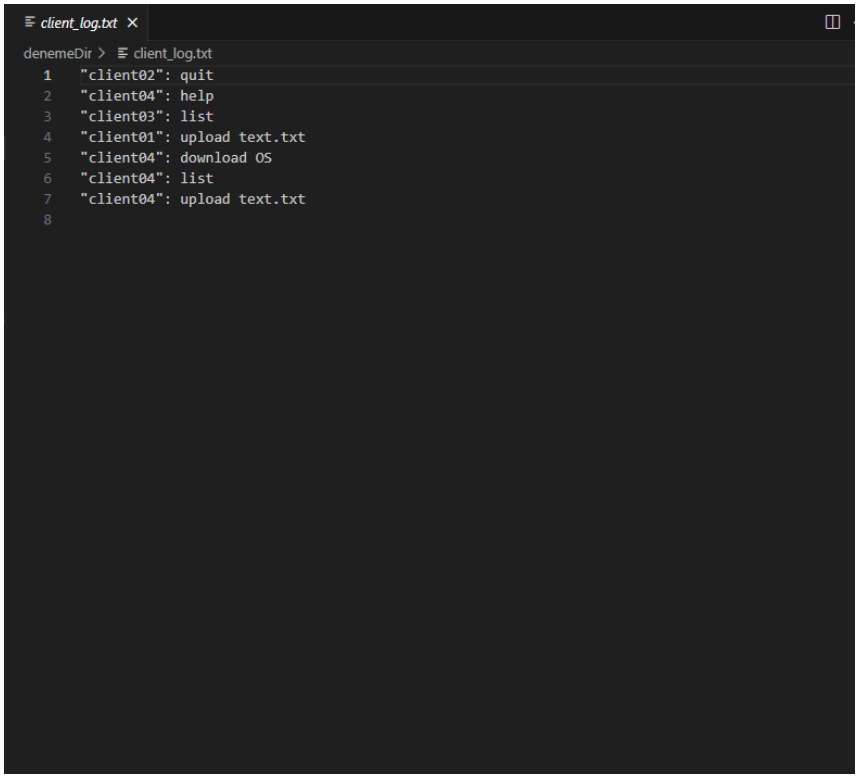


RESULTS

Server output:

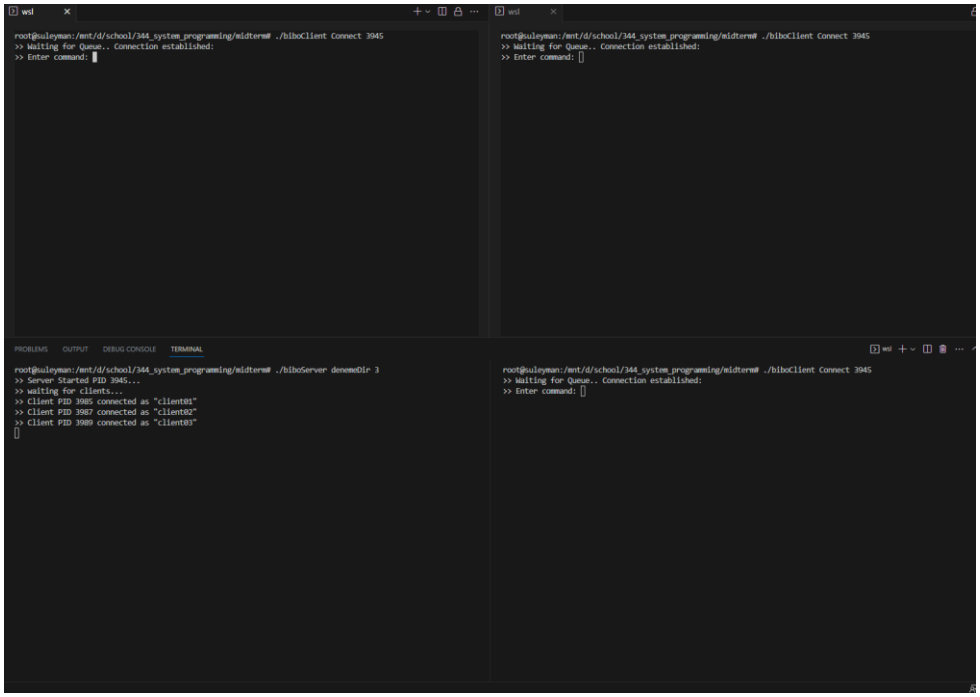
```
root@suleyman:/mnt/d/school/344_system_programming/midterm# ./biboServer denemeDir 3
>> Server Started PID 3919...
>> waiting for clients...
>> Client PID 3924 connected as "client01"
>> Client PID 3928 connected as "client02"
>> Client PID 3930 connected as "client03"
>> Connection request PID 3932... Que FULL
>> "client03" disconnected...
>> Client PID 3932 connected as "client04"
```

After a few requests, logfile:



RESULTS

Before killServer:



After killServer: