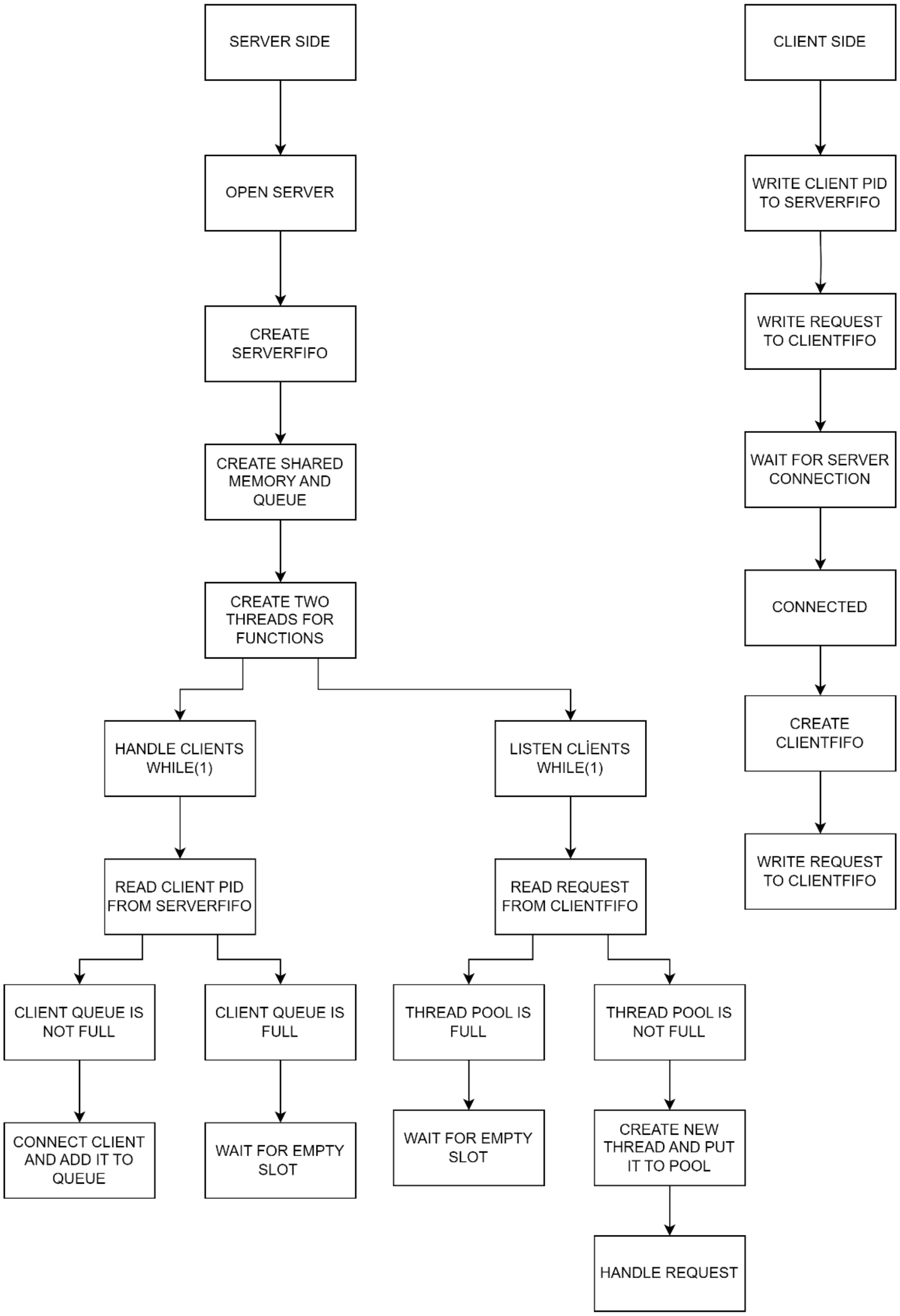
CSE344 HOMEWORK 4 REPORT

SÜLEYMAN KORAMAZ

1901042615

SYSTEM ARCHITECTURE


SERVER SIDE
OPEN SERVER
CREATE SERVERFIFO
CREATE SHARED MEMORY AND QUEUE
CREATE TWO THREADS FOR FUNCTIONS
HANDLE CLIENTS WHILE(1)
READ CLIENT PID FROM SERVERFIFO
CLIENT QUEUE IS NOT FULL
CONNECT CLIENT AND ADD IT TO QUEUE
CLIENT QUEUE IS FULL
WAIT FOR EMPTY SLOT
LISTEN CLİENTS WHILE(1)
READ REQUEST FROM CLIENTFIFO
THREAD POOL IS FULL
WAIT FOR EMPTY SLOT
THREAD POOL IS NOT FULL
CREATE NEW THREAD AND PUT IT TO POOL
HANDLE REQUEST
CLIENT SIDE
WRITE CLIENT PID TO SERVERFIFO
WRITE REQUEST TO CLIENTFIFO
WAIT FOR SERVER CONNECTION
CONNECTED
CREATE CLIENTFIFO
WRITE REQUEST TO CLIENTFIFO

DESIGN DECISIONS AND IMPLEMENTATION DETAILS

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.

SERVER SIDE

Server side is totally different from midterm server. In this homework, server is creating two threads for handle clients and listen clients in a loop. Handle clients function always reads server fifo until end of the program. When new client try to connect, check queue is full or not and connect client. Listen clients function always reads client fifo that common file that written by all clients. When request has arrived from any client, check thread pool is full or not. If pool is full, make that request wait to empty slot. If pool is not full or any thread be done, create new thread for that request and put it to thread pool.

SERVER IMPLEMENTATION

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. Create thread1 and thread2.
11. Thread1 is calling handleClients() function that is connects clients.
    a- Read client pid which is trying to connect in server fifo.
    b- If queue is full, make it wait to empty slot with semaphore.
    c- If queue is not full, connect it and add it to queue.
12. Thread2 is calling listenClients() function that is read requests from clients
    a- Open client fifo for read requests and open client response fifo for write responses.
    b- Always reads clients request in a loop.
    c- When any client write request, if thread pool if full, make request to wait.
    d- If it is not full create two threads.
    e- First thread for handle request, one thread for control first thread is done or not.
13. When any client quits, remove it from the queue and take next request in the semaphore queue and do same process.
14. When any client kill server, terminate all clients and terminate server.

FUNCTIONS

VOID HANDLESIGNAL(INT)

Function that terminates server and connected clients.

VOID TERMINATEALLCLIENTS()

Helper function that terminates all clients in queue.

INT PROCESS(CONST CHAR*, INT)

Helper function that handles request and send response. returns 1 if quit, returns 2 if kill server request arrived.

VOID* HANDLECLIENTREQUESTS(VOID*)

Function that handle client request; update log file, send request to process() function, handle quit and killServer returns.

VOID* HANDLECLIENTS(VOID*)

Function that works until program end, read new client’s pid from server fifo, handle connections and queue.

VOID* LISTENCLIENTS(VOID*)

Function that works until program end, read requests from client fifo, handle thread pool and creating threads.

VOID* MONITORTHREADCOMPLETION(VOID*)

Helper function for control thread be done or not.

INT REMOVECLIENTBYPID(INT)

Helper function for handle queue when any client quit.

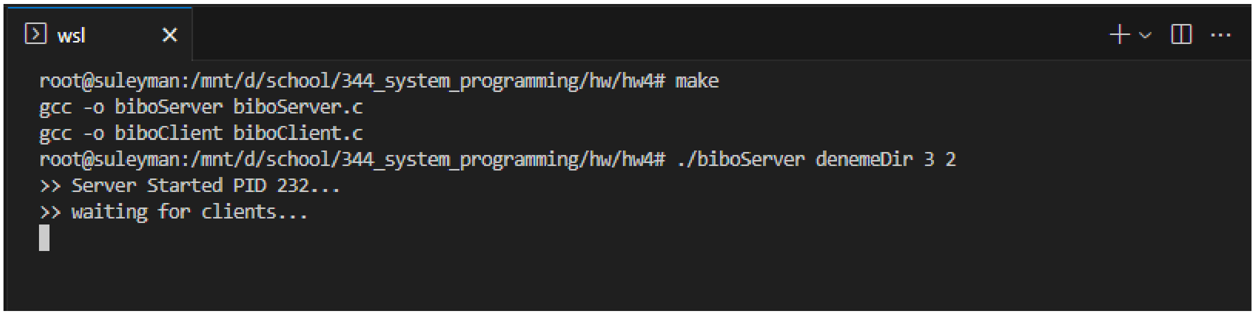
TEST RESULTS

TEST

1- Open server with max 3 clients and 2 pool size.
2- Connect client1.
3- Connect client2.
4- Connect client3.
5- Connect client4 (it will wait).
6- At the same time, send three requests with first three clients. Third request will wait for empty slot.

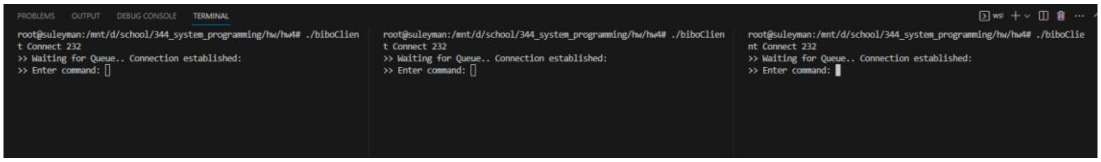
NOTE: I tested code with adding sleep(20) at the beginning of the handleClientRequests() function. It makes wait requests for me to send second and third requests.

STEP 1 (OPEN SERVER)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
```

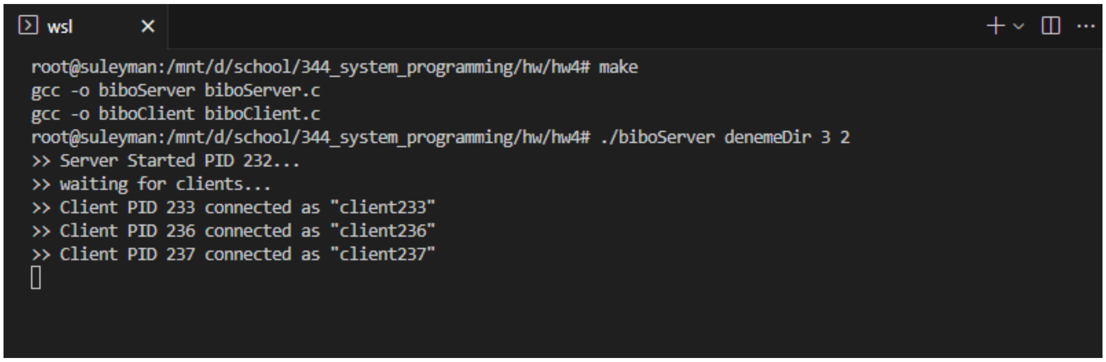
STEP 2,3,4 (CONNECT CLIENTS)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command:
```

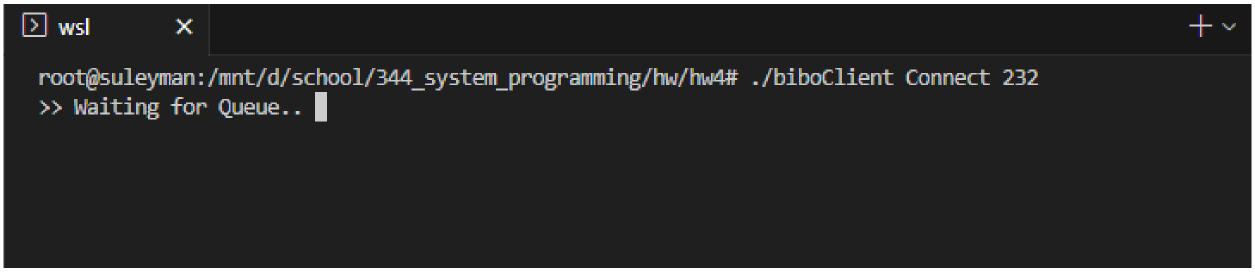
```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command:
```

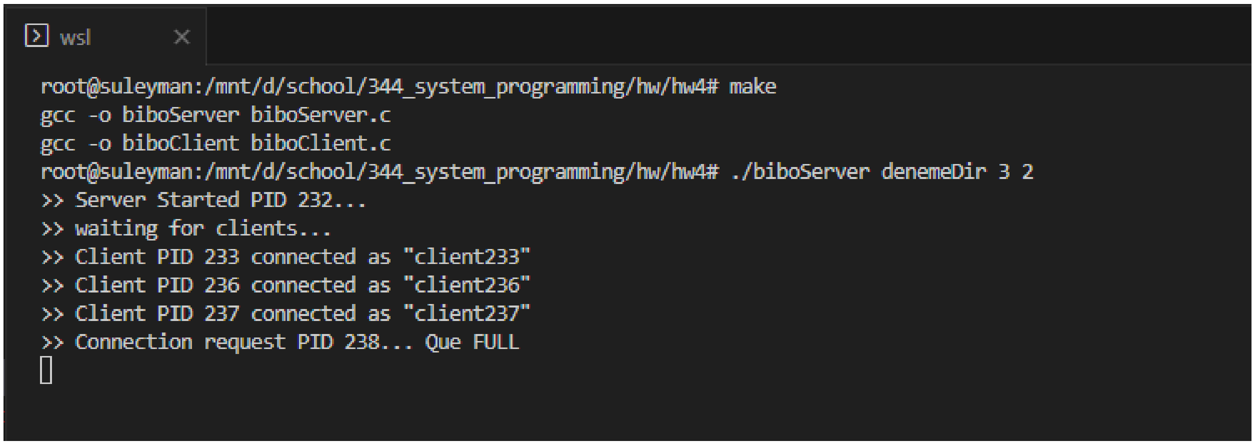


```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
```

STEP 5 (FOURTH CLIENT TRY TO CONNECT)

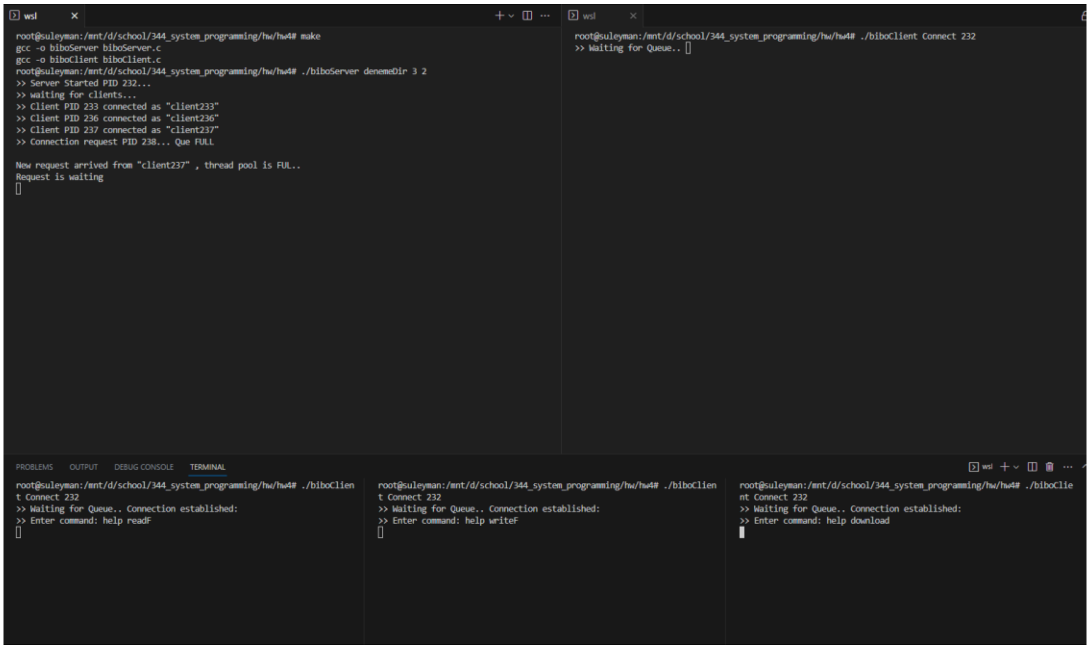


```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue..
```



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL
```

STEP 6.A (THREE CLIENTS SEND REQUEST, THIRD REQUEST IS WAITING)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL

New request arrived from "client237" , thread pool is FUL..
Request is waiting
```

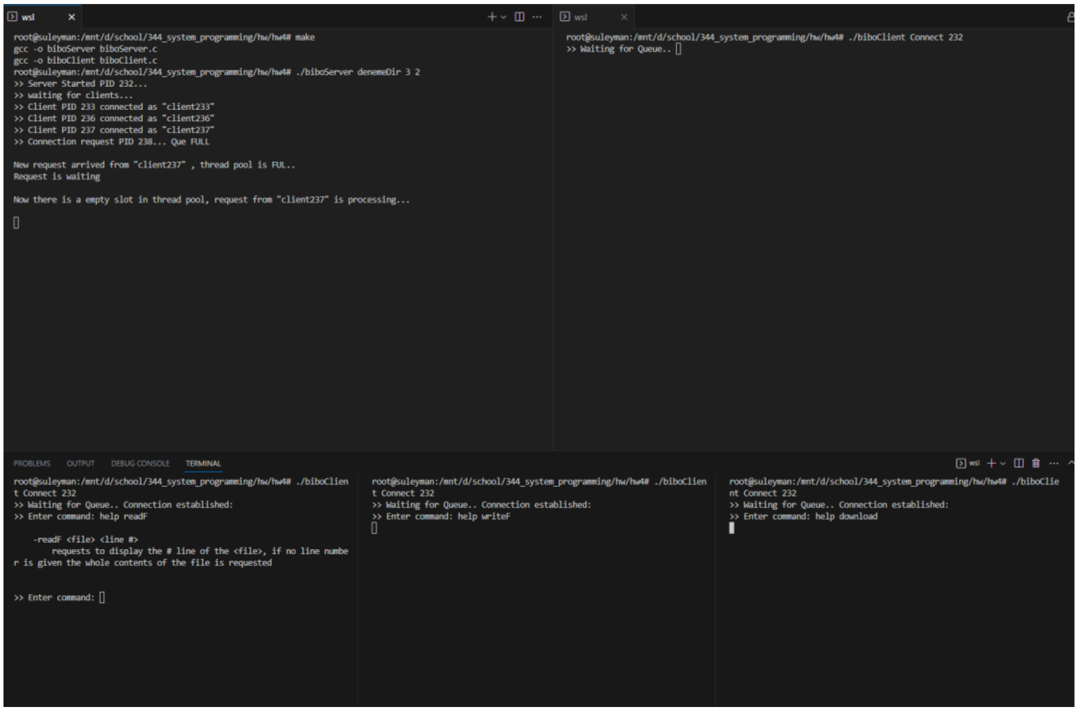
```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue..
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help readF
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help writeF
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help download
```

STEP 6.B (FIRST REQUEST HANDLED, THIRD REQUEST IS PROCESSING)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL

New request arrived from "client237" , thread pool is FUL..
Request is waiting

Now there is a empty slot in thread pool, request from "client237" is processing...

```

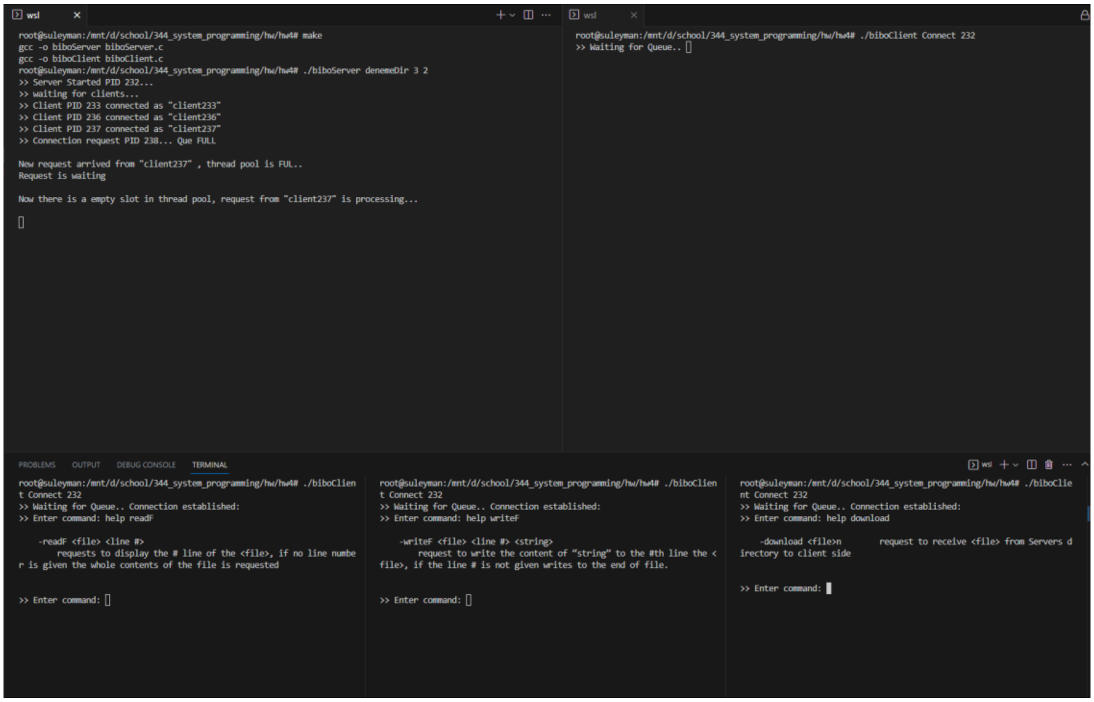
```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue..
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help readF

    -readF <file> <line #>
        requests to display the # line of the <file>, if no line numbe
r is given the whole contents of the file is requested


>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help writeF
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help download
```

STEP 6.C (ALL REQUESTS HANDLED)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL

New request arrived from "client237" , thread pool is FUL..
Request is waiting

Now there is a empty slot in thread pool, request from "client237" is processing...

```

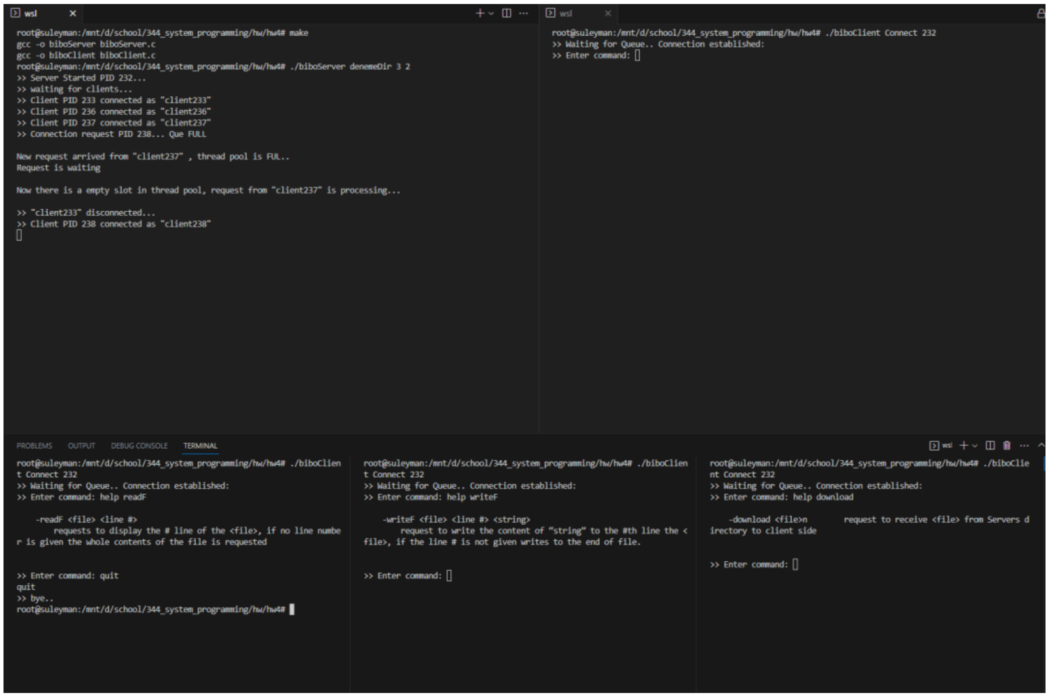
```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue..
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help readF

    -readF <file> <line #>
        requests to display the # line of the <file>, if no line numbe
r is given the whole contents of the file is requested


>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help writeF

    -writeF <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.


>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help download

    -download <file>n       request to receive <file> from Servers d
irectory to client side


>> Enter command:
```

STEP 6.D (ONE OF THE CLIENTS QUIT, CLIENT WHICH IS WAITING CONNECTED)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL

New request arrived from "client237" , thread pool is FUL..
Request is waiting

Now there is a empty slot in thread pool, request from "client237" is processing...

>> "client233" disconnected...
>> Client PID 238 connected as "client238"
```

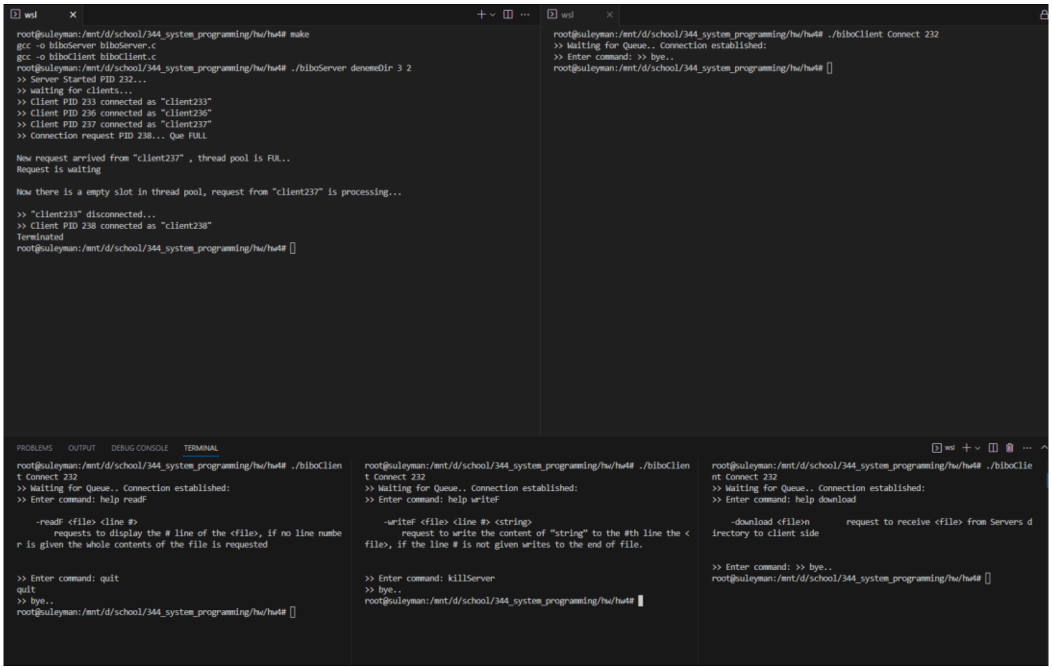
```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help readF

    -readF <file> <line #>
        requests to display the # line of the <file>, if no line numbe
r is given the whole contents of the file is requested


>> Enter command: quit
quit
>> bye..
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help writeF

    -writeF <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.


>> Enter command:
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help download

    -download <file>n        request to receive <file> from Servers d
irectory to client side


>> Enter command:
```

STEP6.E (ONE OF THE CLIENTS SEND KILL SERVER REQUEST)



```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# make
gcc -o biboServer biboServer.c
gcc -o biboClient biboClient.c
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboServer denemeDir 3 2
>> Server Started PID 232...
>> waiting for clients...
>> Client PID 233 connected as "client233"
>> Client PID 236 connected as "client236"
>> Client PID 237 connected as "client237"
>> Connection request PID 238... Que FULL

New request arrived from "client237" , thread pool is FUL..
Request is waiting

Now there is a empty slot in thread pool, request from "client237" is processing...

>> "client233" disconnected...
>> Client PID 238 connected as "client238"
Terminated
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClient Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: >> bye..
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help readF

    -readF <file> <line #>
        requests to display the # line of the <file>, if no line numbe
r is given the whole contents of the file is requested


>> Enter command: quit
quit
>> bye..
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClien
t Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help writeF

    -writeF <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.


>> Enter command: killServer
>> bye..
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```

```
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4# ./biboClie
nt Connect 232
>> Waiting for Queue.. Connection established:
>> Enter command: help download

    -download <file>n        request to receive <file> from Servers d
irectory to client side


>> Enter command: >> bye..
root@suleyman:/mnt/d/school/344_system_programming/hw/hw4#
```