CSE-344 SYSTEM PROGRRAMING SPRING 2024

MIDTERM PROJECT

REPORT

ONUR ATASEVER

210104004087

LOGIC:

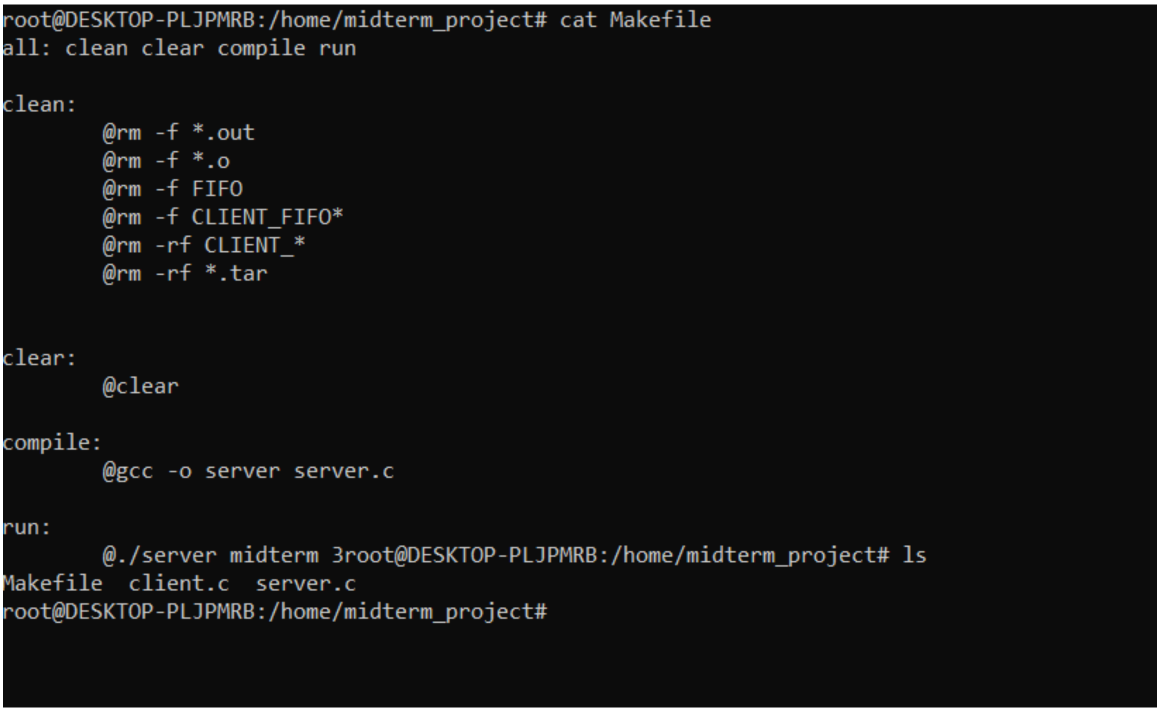
In this client-server model Project, there are mainly two types FIFO's which are general and spesific. If any client wants to connect to the server, client have to use general connection FIFO, if the necessary conditions are met, the connection is established and private FIFOs are created. Private FIFOs names based on Client's program id. So for each server-client connection there are two unique FIFO for communication. First FIFO for server to write and client to read Second FIFO for client to write and server to read. FIFO's are designed as fixed size communication. The server side writes up to a certain byte and the client side reads that much.

Each time a connection is established, the id of the client and the newly created process is stored. Thus, they can be terminated in special cases. For example when a client request server to kill the server or when any interrupt (CTRL+C) happens, to terminate processes, stored process ids are used.

Other important part is critical sections. Shared memory was used so that each process on the server side could know which file it was. A struct was kept in each element of the created shared memory array. This Struct has the file name and lock information of this file within itself. When any file operations happens, it checks if given file name is in the array. If it is not it adds, if it is according to lock informations, it gets blocked or it goes to critical region and it changes lock state which is in shared memory. The first element of the shared memory array is log.txt.

The general FIFO which is for connection requests, it is defined as NON-BLOCKING. So program always continue. In this program's logic, if nothing can be read from FIFO (which means it returns -1), it checks client queue. If queue is empty it goes while loop again. But if there is some clients in the queue, it checks the number of connected clients and according to number, it accepts clients from queue. But, if some connection request are read from FIFO, it checks if queue is empty or not. If it is empty and number of connected clients are less than max size, request is accepted and it connects. If the number of connected clients are equal to max size, client is added to the queue. So, it checks FIFO, it prioritizes those in the queue. But of course if the connectin type is tryConnect and the server is full, it does not wait, response is sent directly and client finishes.

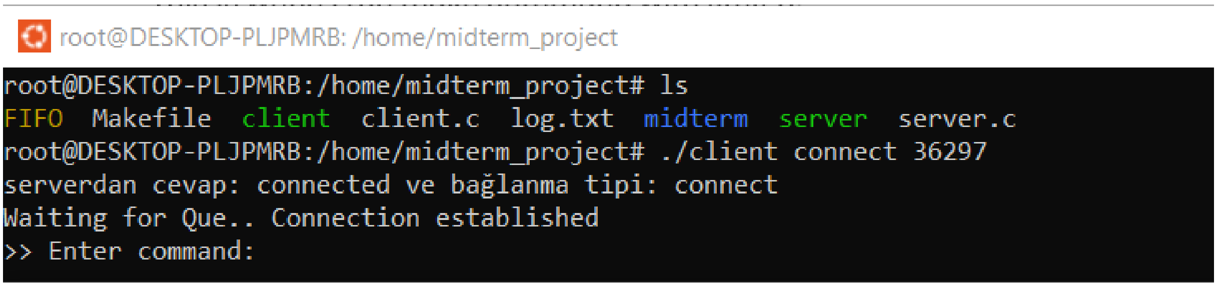
Three different signal handler used with sigaction. First one sigchld_handler which is for the ending of child processes. Designed to prevent them from becoming zombie processes. And it decreases the number of connected clients. And it calls a function to remove dead client and child process from arrays. Second one is a normal signal_handler. This function reacts to SIGUSR1 signal. This signal is used to handle killServer function. While a child process is running, if any killServer command occurs, child send this signal with kill function to inform parent process. Child process can not kill all the server because it does not know all the connected clients but parent does. Parent process handle this signal and it kill's all the processes which are in the client and children array. And the last signal handler is sigint_handler. When CTRL+C interrupt occurs, this handler kill all the processes which are in the the client and children array. And bot hare free's allocated memory and unlinks the FIFO.



```
root@DESKTOP-PLJPMRB:/home/midterm_project# cat Makefile
all: clean clear compile run

clean:
        @rm -f *.out
        @rm -f *.o
        @rm -f FIFO
        @rm -f CLIENT_FIFO*
        @rm -rf CLIENT_*
        @rm -rf *.tar


clear:
        @clear

compile:
        @gcc -o server server.c

run:
        @./server midterm 3root@DESKTOP-PLJPMRB:/home/midterm_project# ls
Makefile  client.c  server.c
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

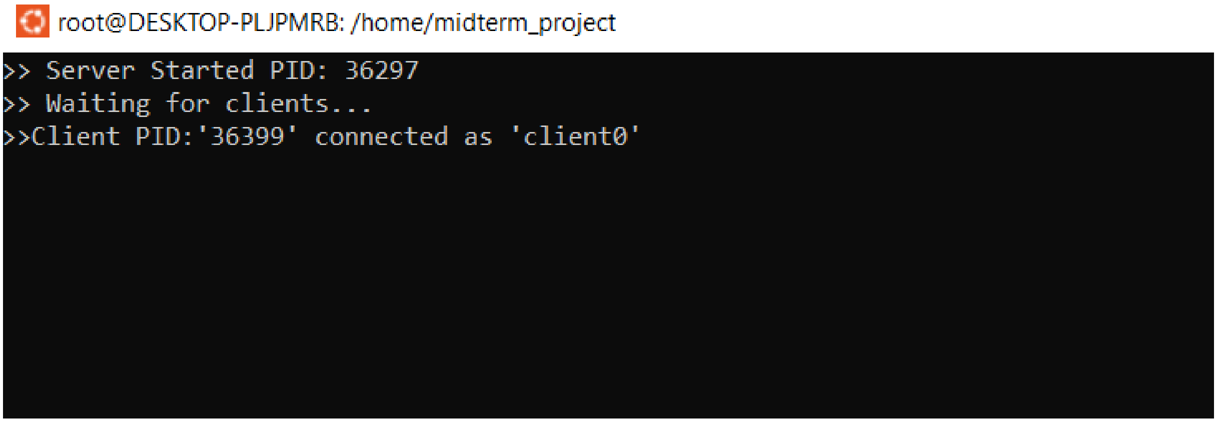
In the directory there are only three folder.

make command with limit 3 and Client connected with tryConnect command:



root@DESKTOP-PLJPMRB: /home/midterm_project

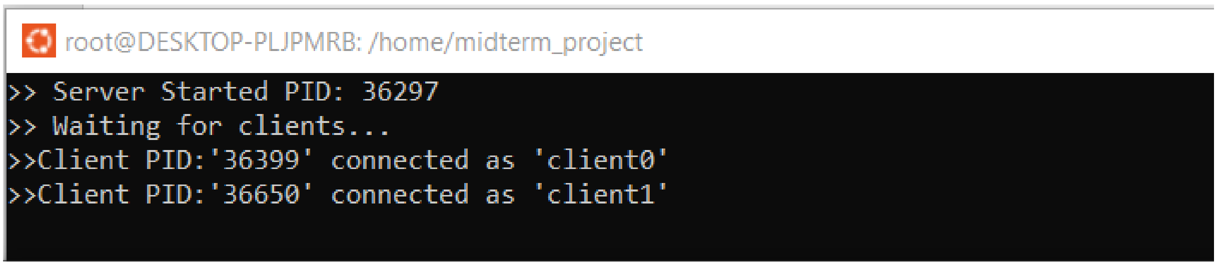
```
root@DESKTOP-PLJPMRB:/home/midterm_project# ls
FIFO  Makefile  client  client.c  log.txt  midterm  server  server.c
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 36297
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```



root@DESKTOP-PLJPMRB: /home/midterm_project

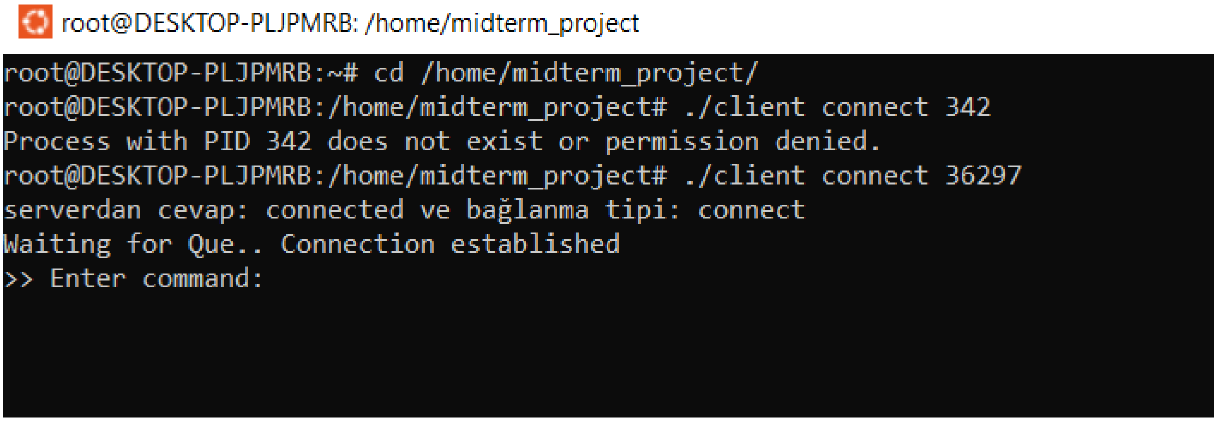
```
>> Server Started PID: 36297
>> Waiting for clients...
>>Client PID:'36399' connected as 'client0'
```

It tries to connect with connect command but with wrong server pid, then correct one:



root@DESKTOP-PLJPMRB: /home/midterm_project

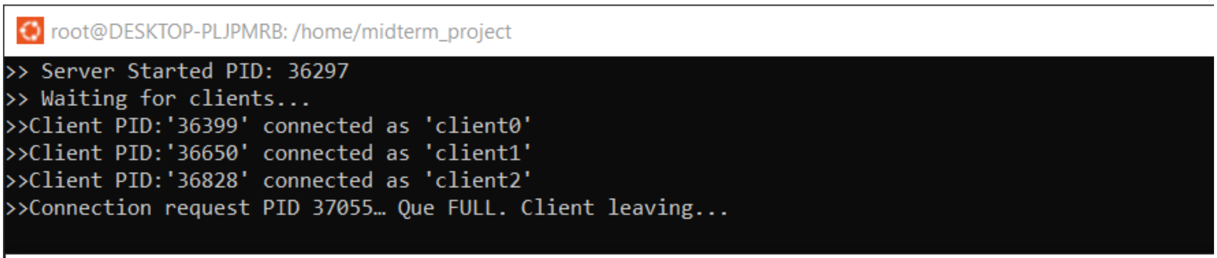
```
>> Server Started PID: 36297
>> Waiting for clients...
>>Client PID:'36399' connected as 'client0'
>>Client PID:'36650' connected as 'client1'
```



root@DESKTOP-PLJPMRB: /home/midterm_project

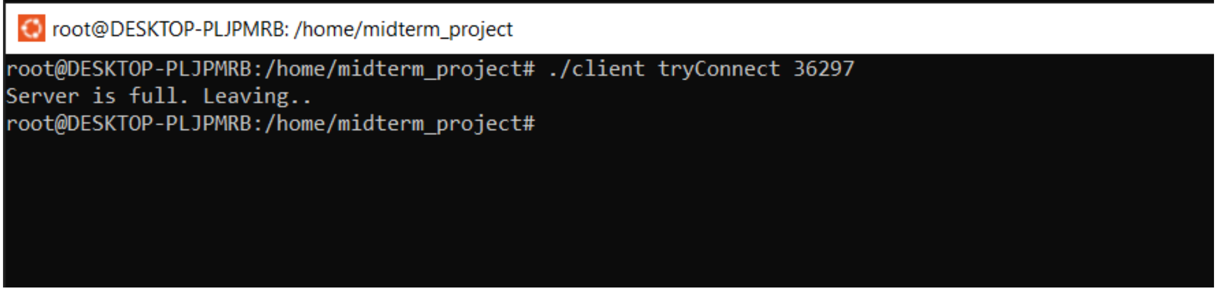
```
root@DESKTOP-PLJPMRB:~# cd /home/midterm_project/
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 342
Process with PID 342 does not exist or permission denied.
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 36297
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```

When the server is full, client tries to connect with tryConnect command:



root@DESKTOP-PLJPMRB: /home/midterm_project

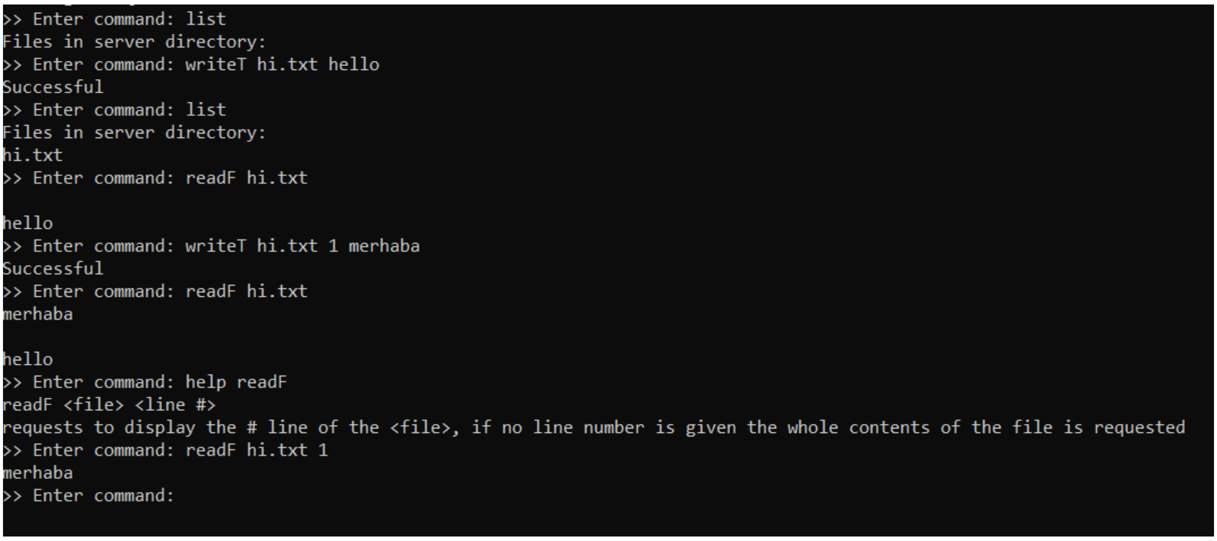
```
>> Server Started PID: 36297
>> Waiting for clients...
>>Client PID:'36399' connected as 'client0'
>>Client PID:'36650' connected as 'client1'
>>Client PID:'36828' connected as 'client2'
>>Connection request PID 37055… Que FULL. Client leaving...
```



root@DESKTOP-PLJPMRB: /home/midterm_project

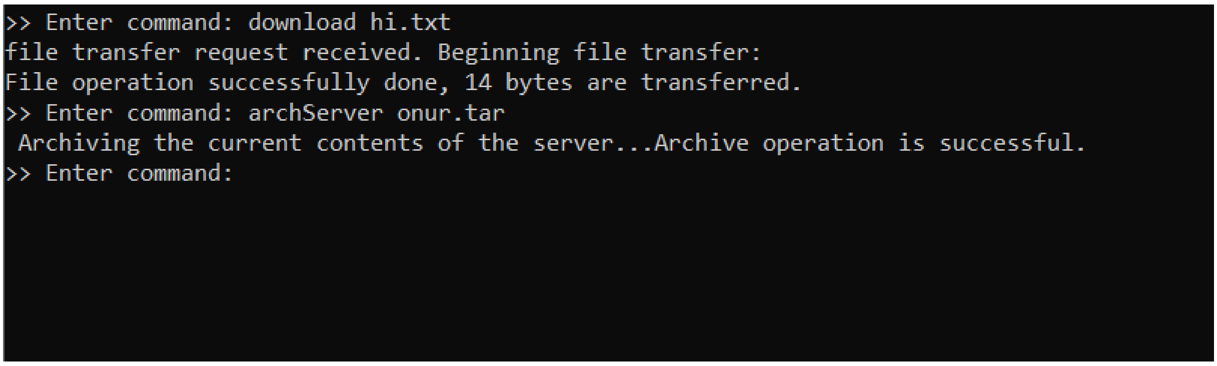
```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client tryConnect 36297
Server is full. Leaving..
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

At the beginning there is nothing in the directory, then with write, file is created and it is displayed with list. All the file and on eline is read with readF command. Help command is displayed.



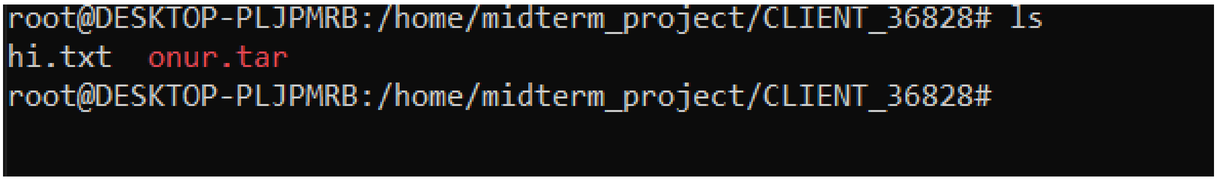
```
>> Enter command: list
Files in server directory:
>> Enter command: writeT hi.txt hello
Successful
>> Enter command: list
Files in server directory:
hi.txt
>> Enter command: readF hi.txt

hello
>> Enter command: writeT hi.txt 1 merhaba
Successful
>> Enter command: readF hi.txt
merhaba

hello
>> Enter command: help readF
readF <file> <line #>
requests to display the # line of the <file>, if no line number is given the whole contents of the file is requested
>> Enter command: readF hi.txt 1
merhaba
>> Enter command:
```

Download and archive is done and they are stored in the client directory.

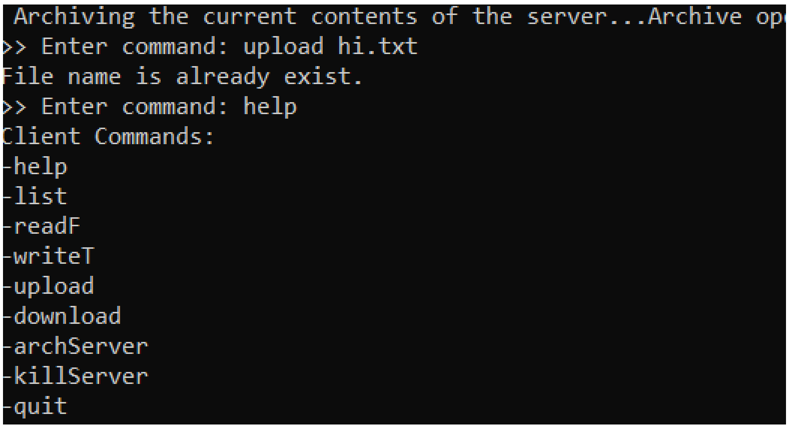


```
>> Enter command: download hi.txt
file transfer request received. Beginning file transfer:
File operation successfully done, 14 bytes are transferred.
>> Enter command: archServer onur.tar
 Archiving the current contents of the server...Archive operation is successful.
>> Enter command:
```

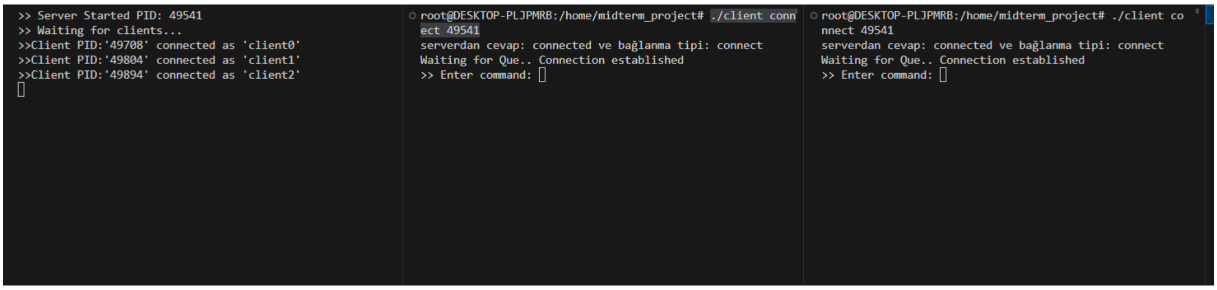
Client directory:



```
root@DESKTOP-PLJPMRB:/home/midterm_project/CLIENT_36828# ls
hi.txt  onur.tar
root@DESKTOP-PLJPMRB:/home/midterm_project/CLIENT_36828#
```



```
 Archiving the current contents of the server...Archive op
>> Enter command: upload hi.txt
File name is already exist.
>> Enter command: help
Client Commands:
-help
-list
-readF
-writeT
-upload
-download
-archServer
-killServer
-quit
```

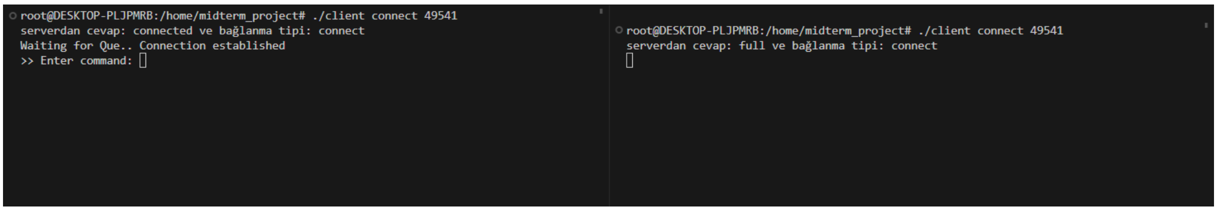


```
>> Server Started PID: 49541
>> Waiting for clients...
>>Client PID:'49708' connected as 'client0'
>>Client PID:'49804' connected as 'client1'
>>Client PID:'49894' connected as 'client2'
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client conn
ect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client co
nnect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```

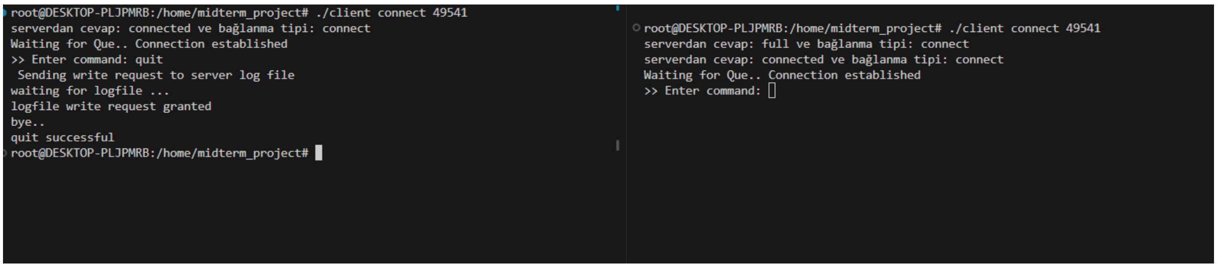
And the third and fourth connection. Fourth is waiting



```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: full ve bağlanma tipi: connect
```

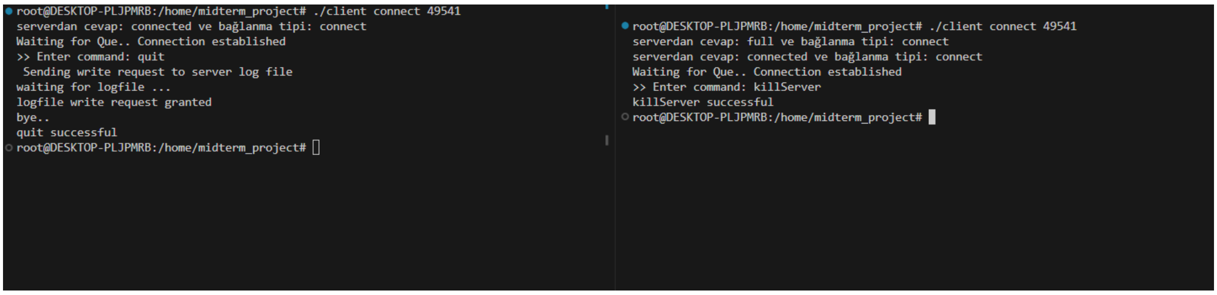
When third is quit fourth comes:



```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: quit
 Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..
quit successful
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: full ve bağlanma tipi: connect
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command:
```

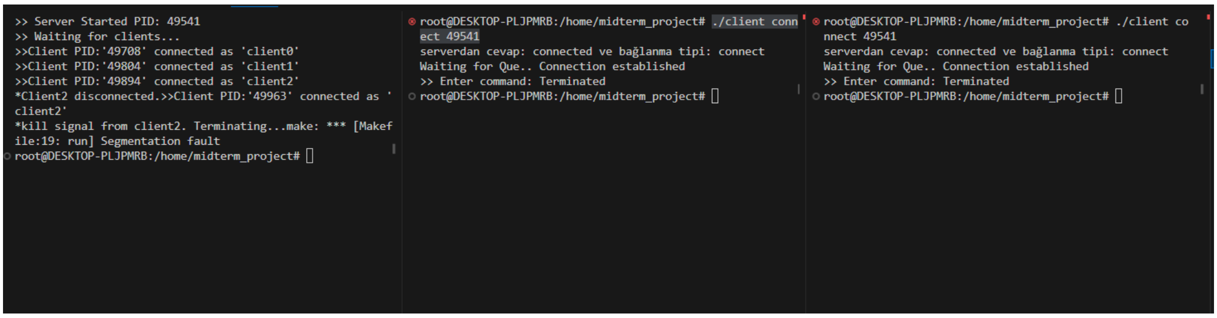
killServer:



```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: quit
 Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..
quit successful
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client connect 49541
serverdan cevap: full ve bağlanma tipi: connect
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: killServer
killServer successful
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

Other processes are terminated.

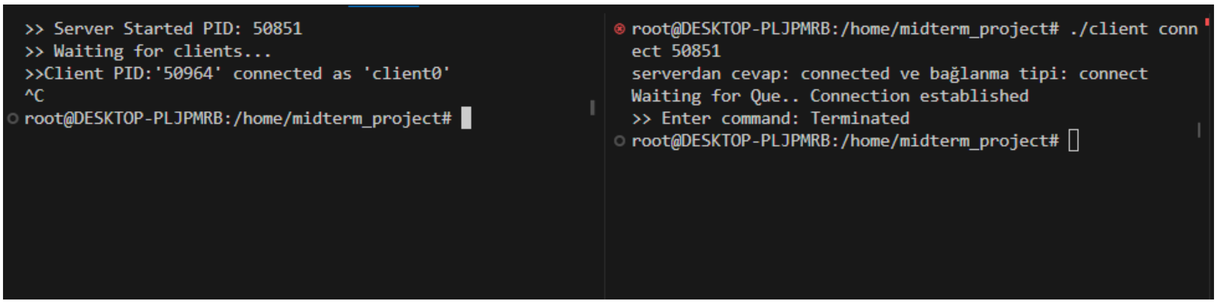


```
>> Server Started PID: 49541
>> Waiting for clients...
>>Client PID:'49708' connected as 'client0'
>>Client PID:'49804' connected as 'client1'
>>Client PID:'49894' connected as 'client2'
*Client2 disconnected.>>Client PID:'49963' connected as '
client2'
*kill signal from client2. Terminating...make: *** [Makef
ile:19: run] Segmentation fault
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client conn
ect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: Terminated
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client co
nnect 49541
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: Terminated
root@DESKTOP-PLJPMRB:/home/midterm_project#
```

With CTRL+C, other processes are terminated:



```
>> Server Started PID: 50851
>> Waiting for clients...
>>Client PID:'50964' connected as 'client0'
^C
root@DESKTOP-PLJPMRB:/home/midterm_project# 
```

```
root@DESKTOP-PLJPMRB:/home/midterm_project# ./client conn
ect 50851
serverdan cevap: connected ve bağlanma tipi: connect
Waiting for Que.. Connection established
>> Enter command: Terminated
root@DESKTOP-PLJPMRB:/home/midterm_project# 
```