CSE 344 System Programming Midterm Project Report

First of all, I would like to start the report by talking about the general structure of my code. In my programme, when the server starts, a special fifo is created for the server pid. Also, one fifo is created for each client connected. These fifos are also specific to client pids. After the server program starts, I open the necessary fifolari files and directories. Then an infinite loop opens, this loop waits for a continuous connection request from the client. If the connection request exceeds the maximum demand, it sorts and stores incoming connection requests according to a certain queuing algorithm. After the currently running processes are finished, requests are started to be pulled from the queue in order. This continues in this way.

After the client side sends the first connection request via the fifo with the same pid, if the connection status is appropriate, the client is informed of this via the fifo opened this time. Client prints something on the screen based on this response. Then the server creates a new child process for this connection and now the client and that child process enter into a command - response dilemma, until quit or killServer is entered as a command.

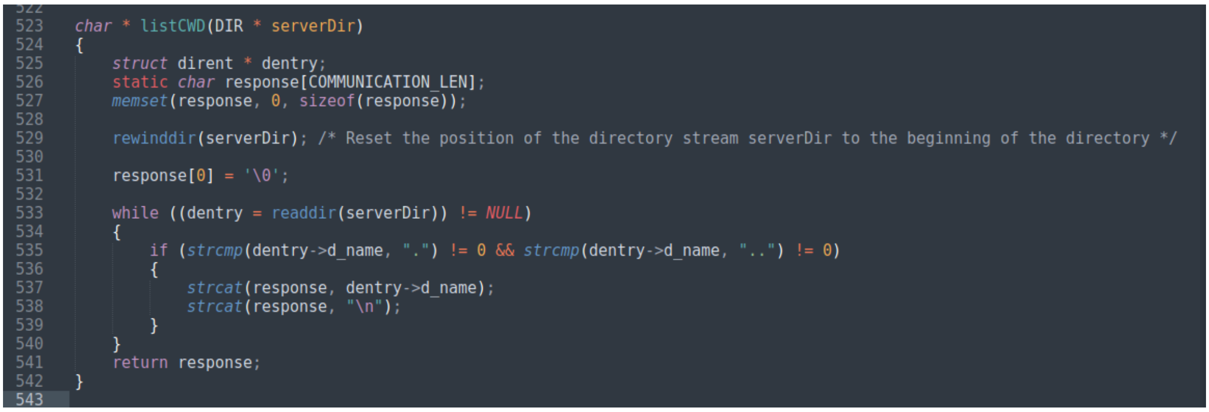
On the server side, each connected client and all the requests created for those clients are stored in a continuous array with their pids. When a SIGINT signal is received, signal_handler is called and all processes stored in these arrays are terminated with kill if they are still alive. Then all resources are cleaned.

Commands

Now let's explain the commands I wrote in order and how I handled them:

help: I did not do anything extreme for the help command, when help is entered, all commands are printed on the screen, and when entered with a command, the description of the entered command is printed on the screen.

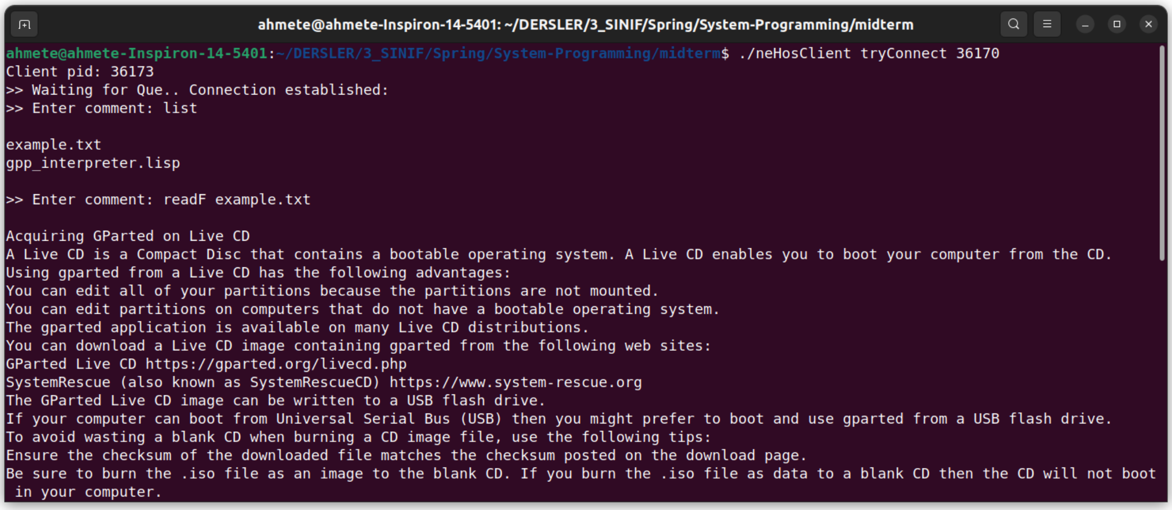
list: This command prints the names of all files and folders in the directory entered with the server.



```c
char * listCWD(DIR * serverDir)
{
    struct dirent * dentry;
    static char response[COMMUNICATION_LEN];
    memset(response, 0, sizeof(response));

    rewinddir(serverDir); /* Reset the position of the directory stream serverDir to the beginning of the directory */

    response[0] = '\0';

    while ((dentry = readdir(serverDir)) != NULL)
    {
        if (strcmp(dentry->d_name, ".") != 0 && strcmp(dentry->d_name, "..") != 0)
        {
            strcat(response, dentry->d_name);
            strcat(response, "\n");
        }
    }
    return response;
}
```

Here a check is made for the directories '.' and '..' to suppress only the elements of the current directory.

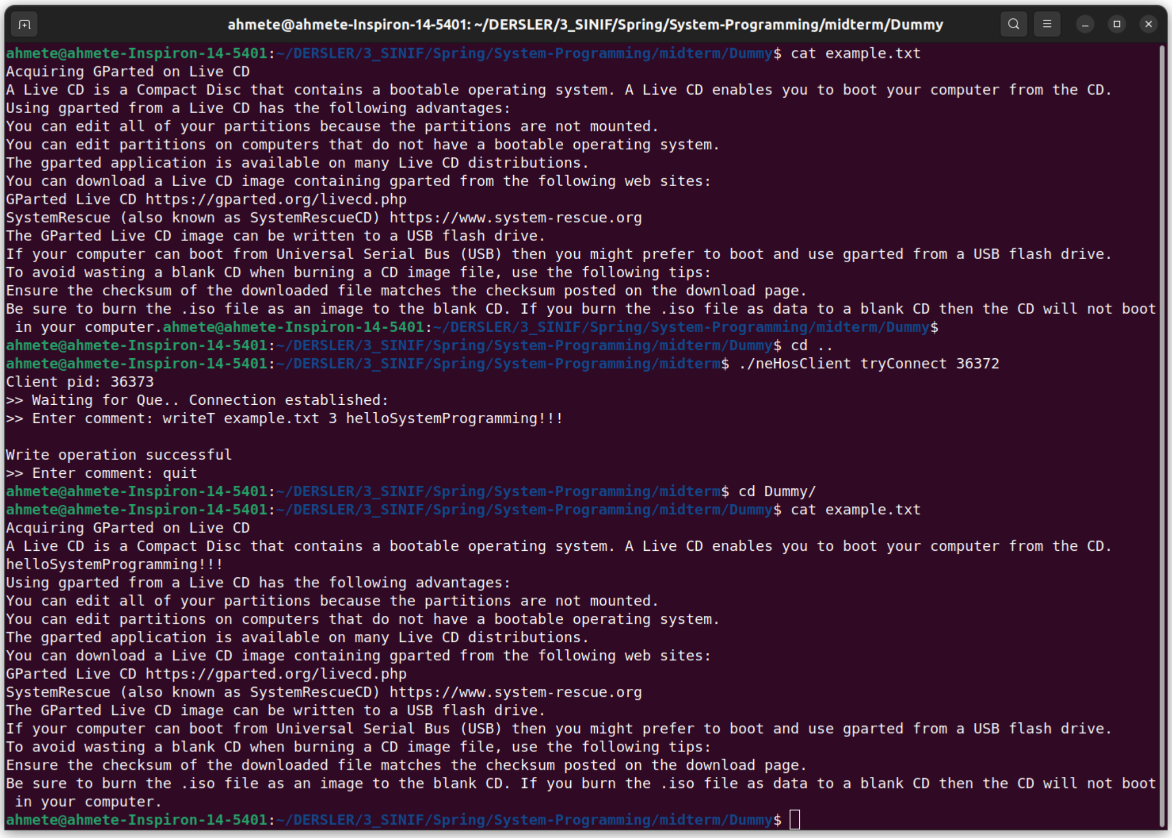
readF: This command opens the given file and reads the information in it based on the index value that comes as a parameter. If the parameter is 0, the whole file is read, otherwise the items in the given index are read and printed to the screen. Firstly, the number of lines of the file is calculated in a loop and if the index is incorrect, the client user is warned that an incorrect input has been entered.



ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient tryConnect 36170
Client pid: 36173
>> Waiting for Que.. Connection established:
>> Enter comment: list

example.txt
gpp_interpreter.lisp

>> Enter comment: readF example.txt

Acquiring GParted on Live CD
A Live CD is a Compact Disc that contains a bootable operating system. A Live CD enables you to boot your computer from the CD.
Using gparted from a Live CD has the following advantages:
You can edit all of your partitions because the partitions are not mounted.
You can edit partitions on computers that do not have a bootable operating system.
The gparted application is available on many Live CD distributions.
You can download a Live CD image containing gparted from the following web sites:
GParted Live CD https://gparted.org/livecd.php
SystemRescue (also known as SystemRescueCD) https://www.system-rescue.org
The GParted Live CD image can be written to a USB flash drive.
If your computer can boot from Universal Serial Bus (USB) then you might prefer to boot and use gparted from a USB flash drive.
To avoid wasting a blank CD when burning a CD image file, use the following tips:
Ensure the checksum of the downloaded file matches the checksum posted on the download page.
Be sure to burn the .iso file as an image to the blank CD. If you burn the .iso file as data to a blank CD then the CD will not boot
 in your computer.
```

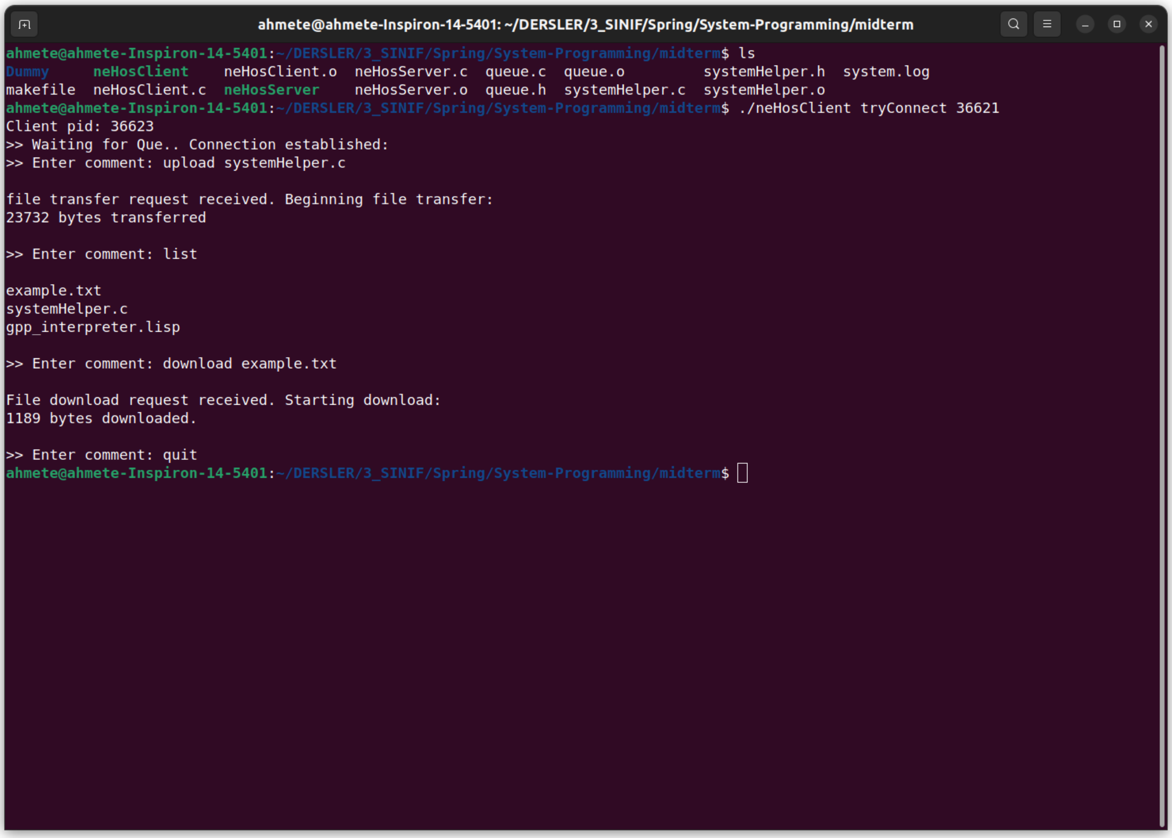
writeT: This command adds a line to the given file according to the specified index. The content of this insertion is also given as a parameter. If the number of lines is not given, I write the text directly to the end of the file. Also, if the specified file does not exist, a file with that name is created and the operations are performed on it. Another thing I want to point out is: writeT <file> <line #> <string> for this command, since I separated all the elements according to their spaces, I accepted the last parameter string with no spaces between them.



ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy

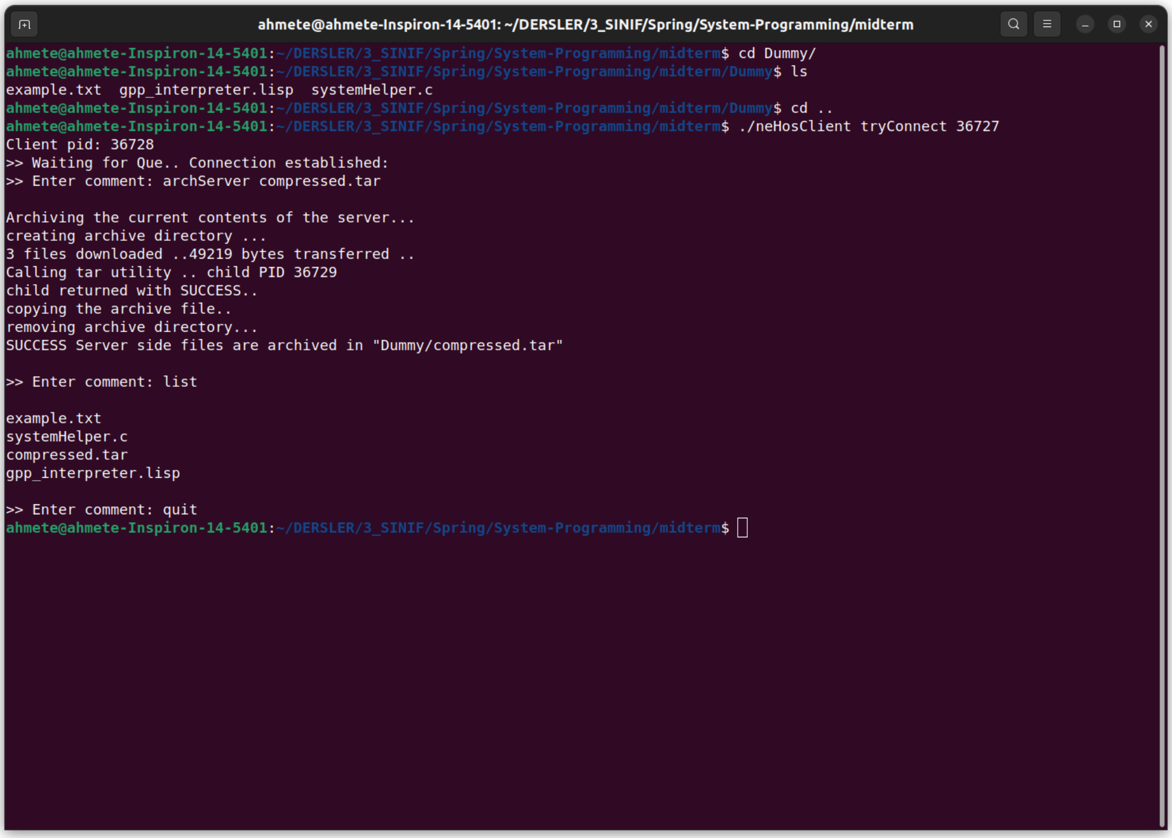
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ cat example.txt
Acquiring GParted on Live CD
A Live CD is a Compact Disc that contains a bootable operating system. A Live CD enables you to boot your computer from the CD.
Using gparted from a Live CD has the following advantages:
You can edit all of your partitions because the partitions are not mounted.
You can edit partitions on computers that do not have a bootable operating system.
The gparted application is available on many Live CD distributions.
You can download a Live CD image containing gparted from the following web sites:
GParted Live CD https://gparted.org/livecd.php
SystemRescue (also known as SystemRescueCD) https://www.system-rescue.org
The GParted Live CD image can be written to a USB flash drive.
If your computer can boot from Universal Serial Bus (USB) then you might prefer to boot and use gparted from a USB flash drive.
To avoid wasting a blank CD when burning a CD image file, use the following tips:
Ensure the checksum of the downloaded file matches the checksum posted on the download page.
Be sure to burn the .iso file as an image to the blank CD. If you burn the .iso file as data to a blank CD then the CD will not boot
 in your computer.ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ cd ..
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient tryConnect 36372
Client pid: 36373
>> Waiting for Que.. Connection established:
>> Enter comment: writeT example.txt 3 helloSystemProgramming!!!

Write operation successful
>> Enter comment: quit
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ cd Dummy/
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ cat example.txt
Acquiring GParted on Live CD
A Live CD is a Compact Disc that contains a bootable operating system. A Live CD enables you to boot your computer from the CD.
helloSystemProgramming!!!
Using gparted from a Live CD has the following advantages:
You can edit all of your partitions because the partitions are not mounted.
You can edit partitions on computers that do not have a bootable operating system.
The gparted application is available on many Live CD distributions.
You can download a Live CD image containing gparted from the following web sites:
GParted Live CD https://gparted.org/livecd.php
SystemRescue (also known as SystemRescueCD) https://www.system-rescue.org
The GParted Live CD image can be written to a USB flash drive.
If your computer can boot from Universal Serial Bus (USB) then you might prefer to boot and use gparted from a USB flash drive.
To avoid wasting a blank CD when burning a CD image file, use the following tips:
Ensure the checksum of the downloaded file matches the checksum posted on the download page.
Be sure to burn the .iso file as an image to the blank CD. If you burn the .iso file as data to a blank CD then the CD will not boot
 in your computer.
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ 
```

upload & download: I transfer data for upload and download in 1024 bytes each to handle large files. I also used shared memory for these two operations.


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ls
Dummy     neHosClient    neHosClient.o  neHosServer.c  queue.c  queue.o         systemHelper.h  system.log
makefile  neHosClient.c  neHosServer    neHosServer.o  queue.h  systemHelper.c  systemHelper.o
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient tryConnect 36621
Client pid: 36623
>> Waiting for Que.. Connection established:
>> Enter comment: upload systemHelper.c

file transfer request received. Beginning file transfer:
23732 bytes transferred

>> Enter comment: list

example.txt
systemHelper.c
gpp_interpreter.lisp

>> Enter comment: download example.txt

File download request received. Starting download:
1189 bytes downloaded.

>> Enter comment: quit
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```

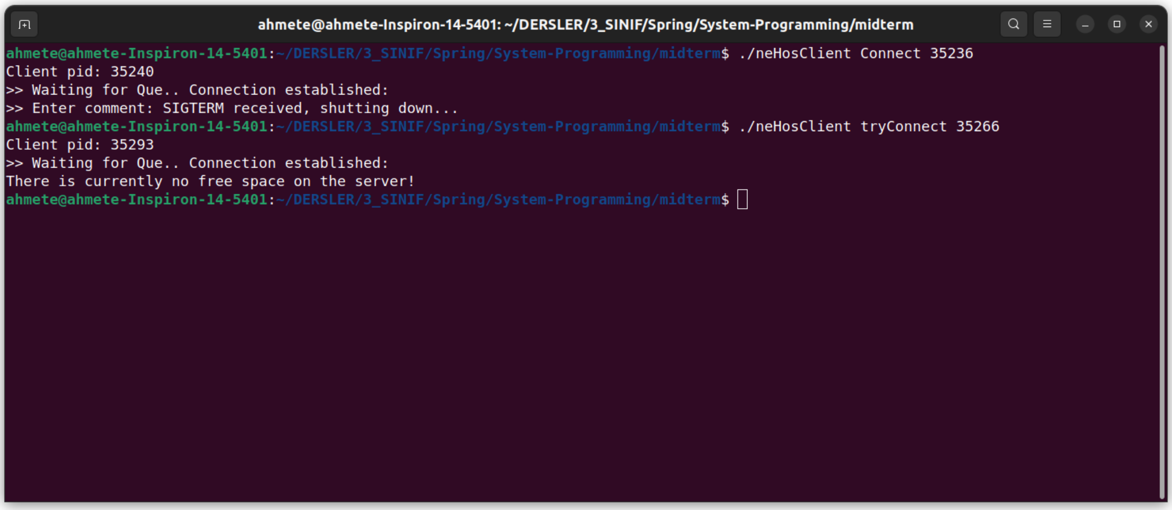
archServer: This command compresses the files in the server directory into .tar and puts them back into the server directory.



ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

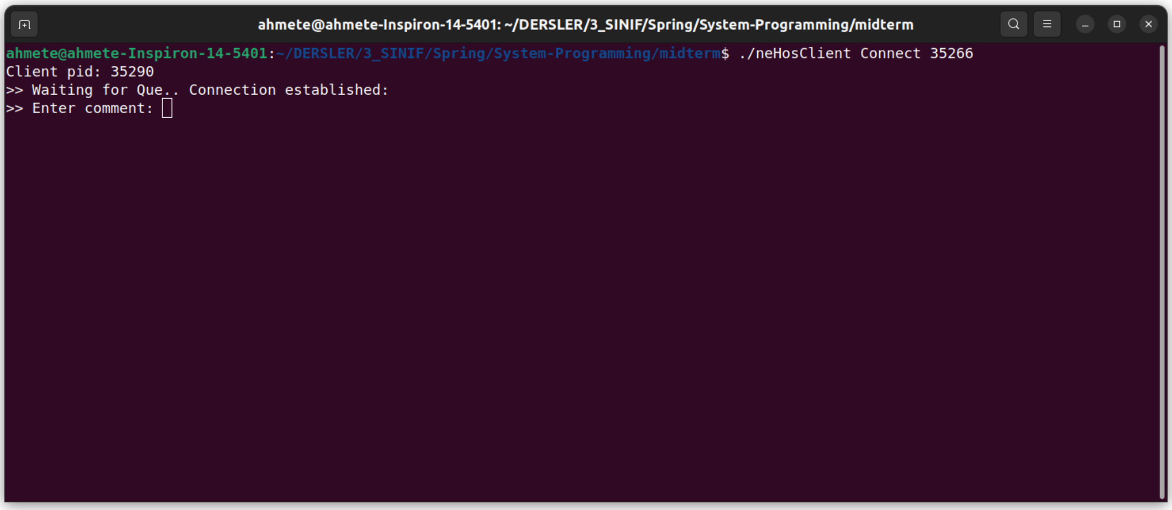
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ cd Dummy/
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ ls
example.txt  gpp_interpreter.lisp  systemHelper.c
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm/Dummy$ cd ..
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient tryConnect 36727
Client pid: 36728
>> Waiting for Que.. Connection established:
>> Enter comment: archServer compressed.tar

Archiving the current contents of the server...
creating archive directory ...
3 files downloaded ..49219 bytes transferred ..
Calling tar utility .. child PID 36729
child returned with SUCCESS..
copying the archive file..
removing archive directory...
SUCCESS Server side files are archived in "Dummy/compressed.tar"

>> Enter comment: list

example.txt
systemHelper.c
compressed.tar
gpp_interpreter.lisp

>> Enter comment: quit
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```

Other outputs:



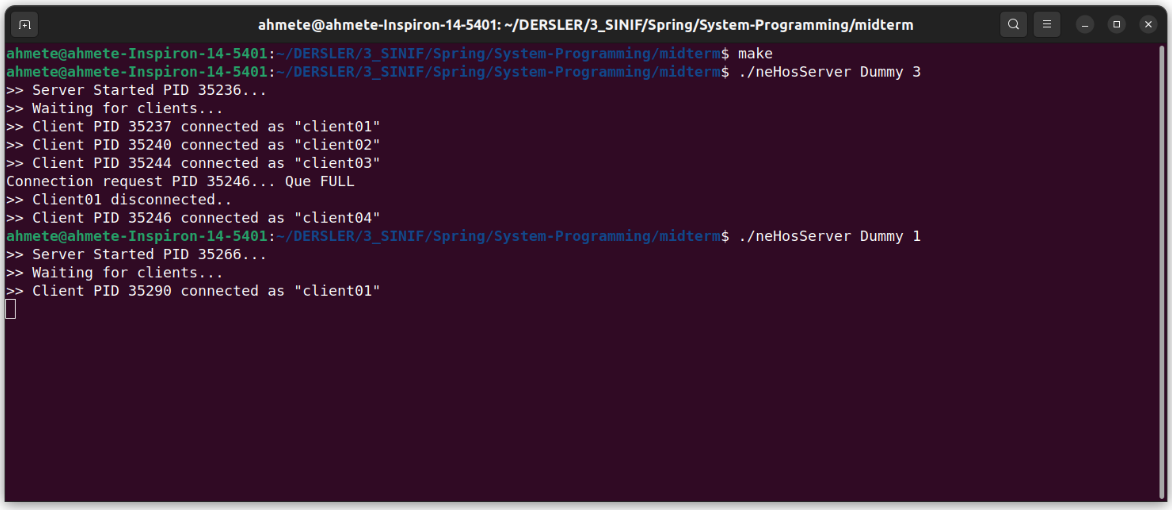
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35240
>> Waiting for Que.. Connection established:
>> Enter comment: SIGTERM received, shutting down...
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient tryConnect 35266
Client pid: 35293
>> Waiting for Que.. Connection established:
There is currently no free space on the server!
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```



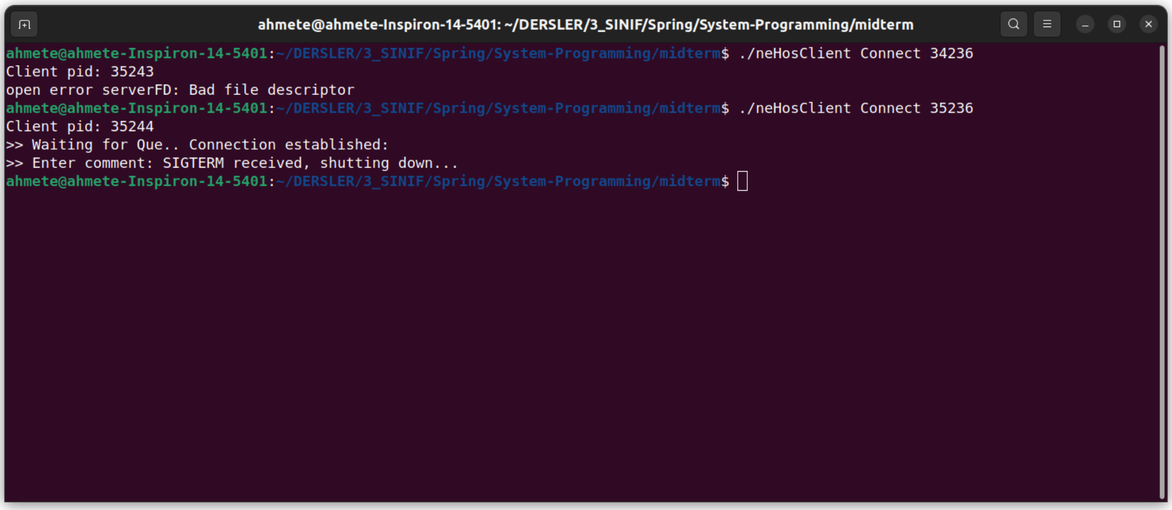
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35266
Client pid: 35290
>> Waiting for Que.. Connection established:
>> Enter comment: 
```



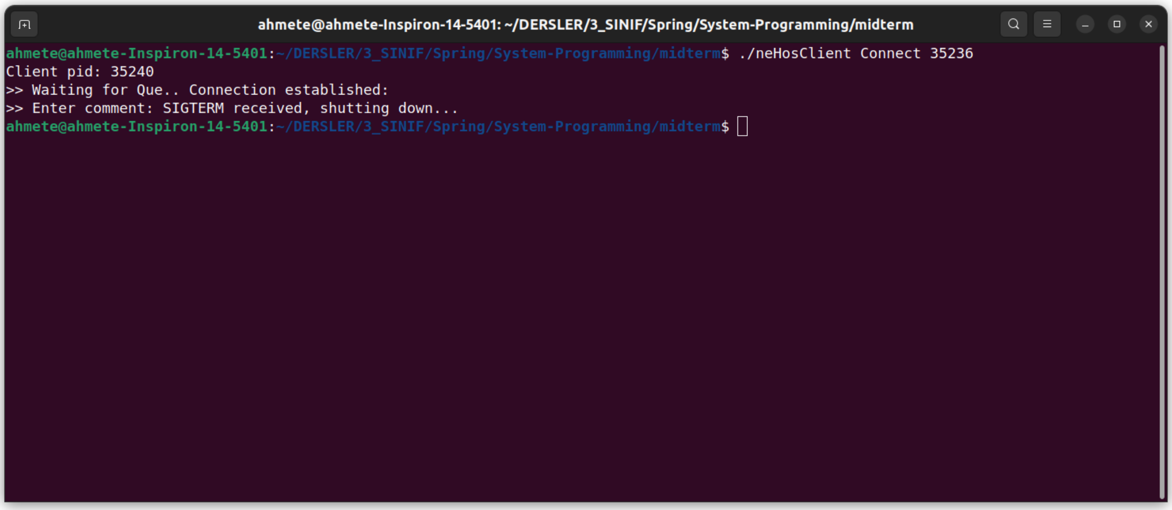
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35236...
>> Waiting for clients...
>> Client PID 35237 connected as "client01"
>> Client PID 35240 connected as "client02"
>> Client PID 35244 connected as "client03"
Connection request PID 35246... Que FULL
>> Client01 disconnected..
>> Client PID 35246 connected as "client04"
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 1
>> Server Started PID 35266...
>> Waiting for clients...
>> Client PID 35290 connected as "client01"
```



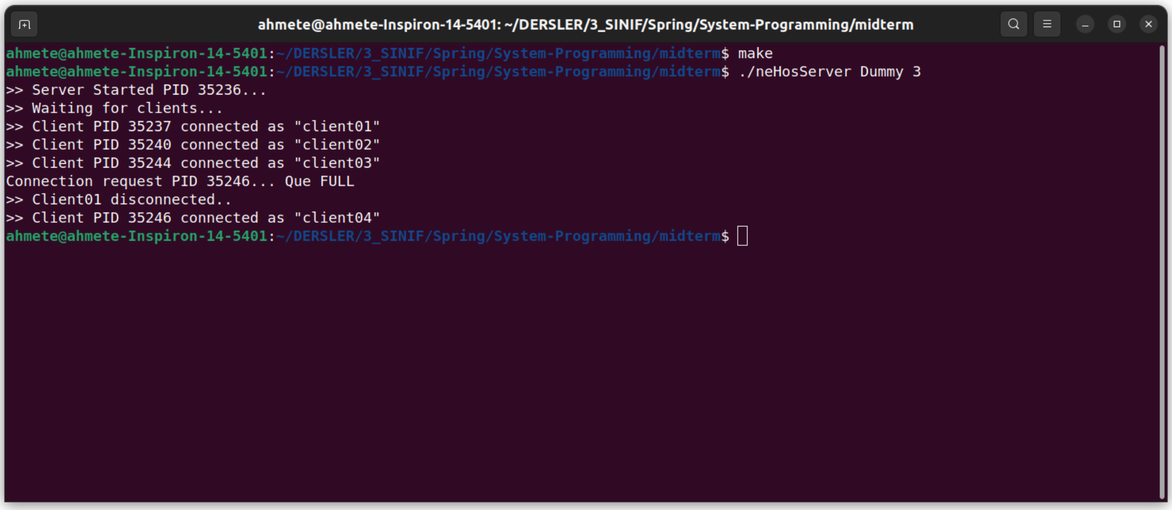
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 34236
Client pid: 35243
open error serverFD: Bad file descriptor
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35244
>> Waiting for Que.. Connection established:
>> Enter comment: SIGTERM received, shutting down...
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```



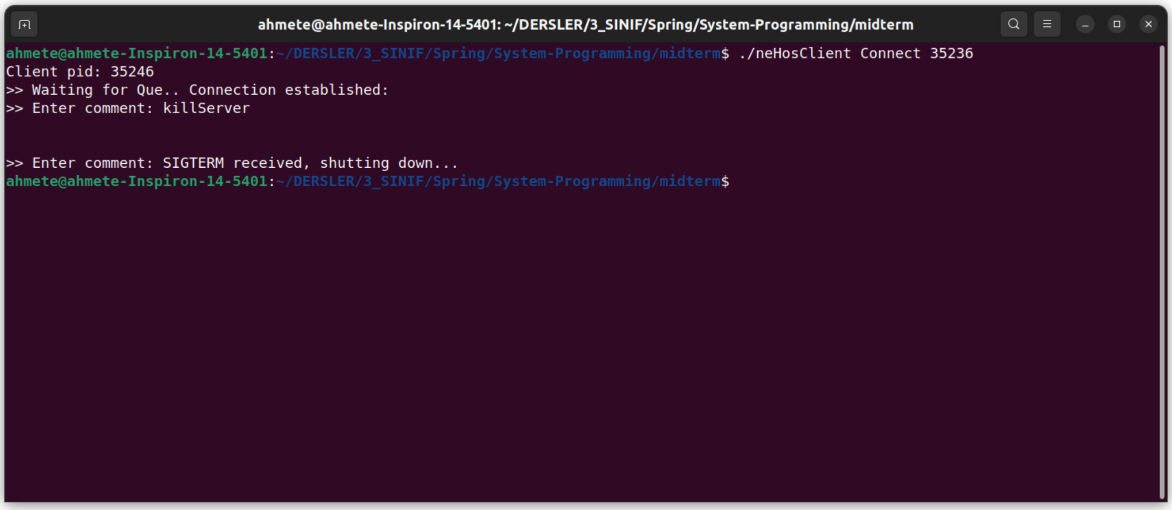
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35240
>> Waiting for Que.. Connection established:
>> Enter comment: SIGTERM received, shutting down...
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```

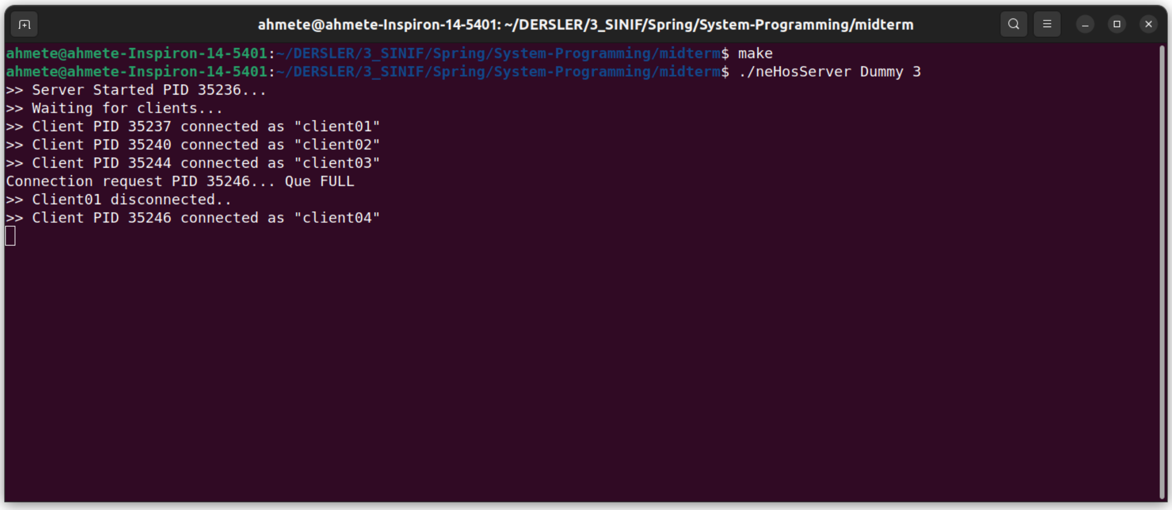


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

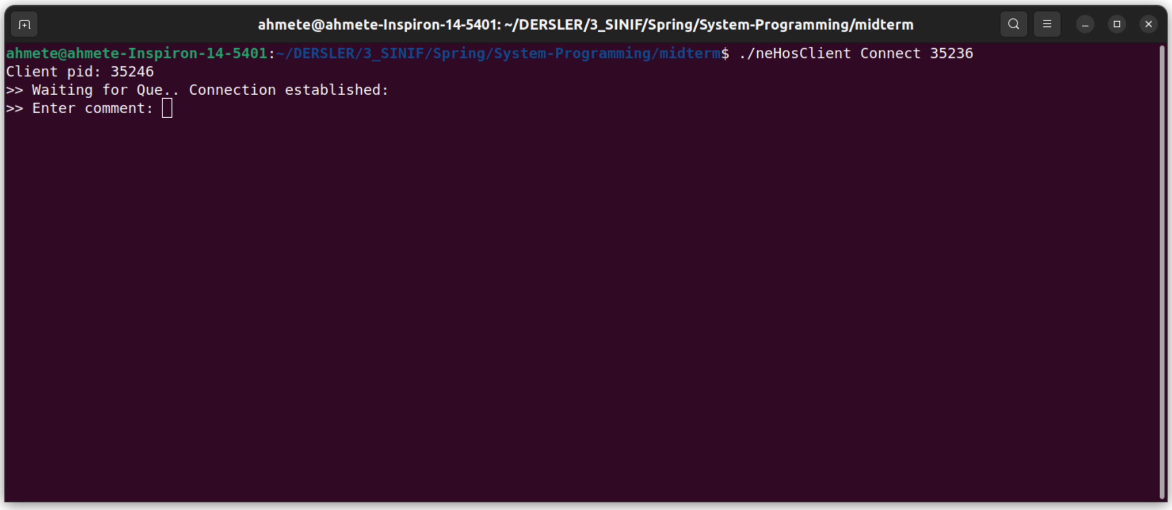
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35236...
>> Waiting for clients...
>> Client PID 35237 connected as "client01"
>> Client PID 35240 connected as "client02"
>> Client PID 35244 connected as "client03"
Connection request PID 35246... Que FULL
>> Client01 disconnected..
>> Client PID 35246 connected as "client04"
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

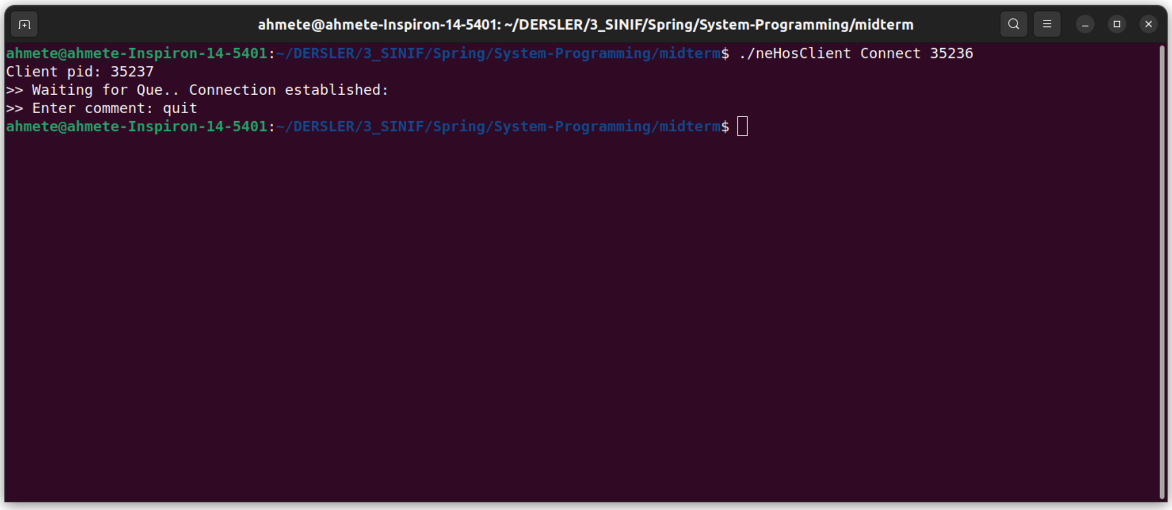
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35246
>> Waiting for Que.. Connection established:
>> Enter comment: killServer


>> Enter comment: SIGTERM received, shutting down...
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35236...
>> Waiting for clients...
>> Client PID 35237 connected as "client01"
>> Client PID 35240 connected as "client02"
>> Client PID 35244 connected as "client03"
Connection request PID 35246... Que FULL
>> Client01 disconnected..
>> Client PID 35246 connected as "client04"
```

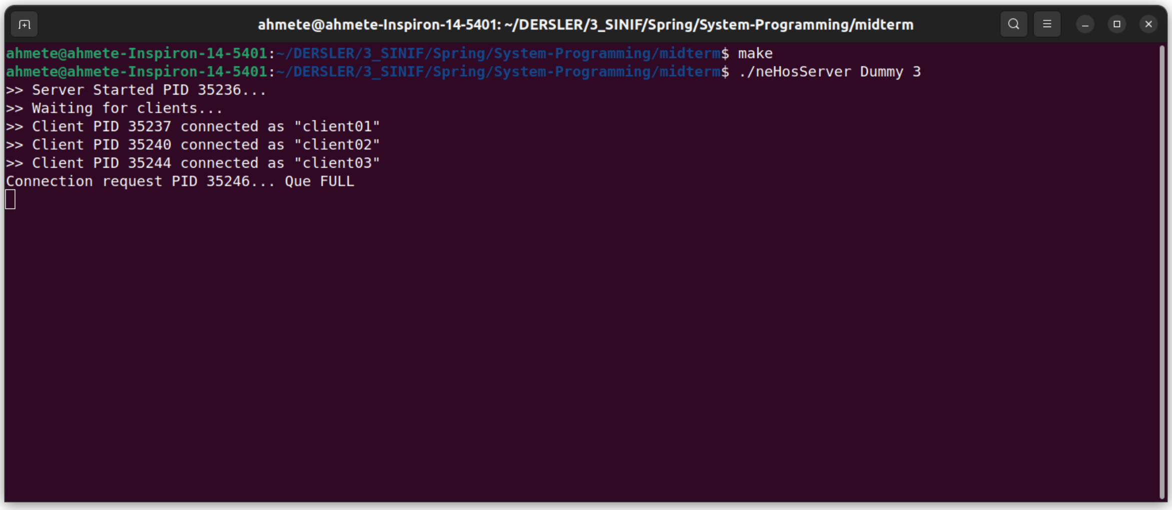

ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35246
>> Waiting for Que.. Connection established:
>> Enter comment: 
```



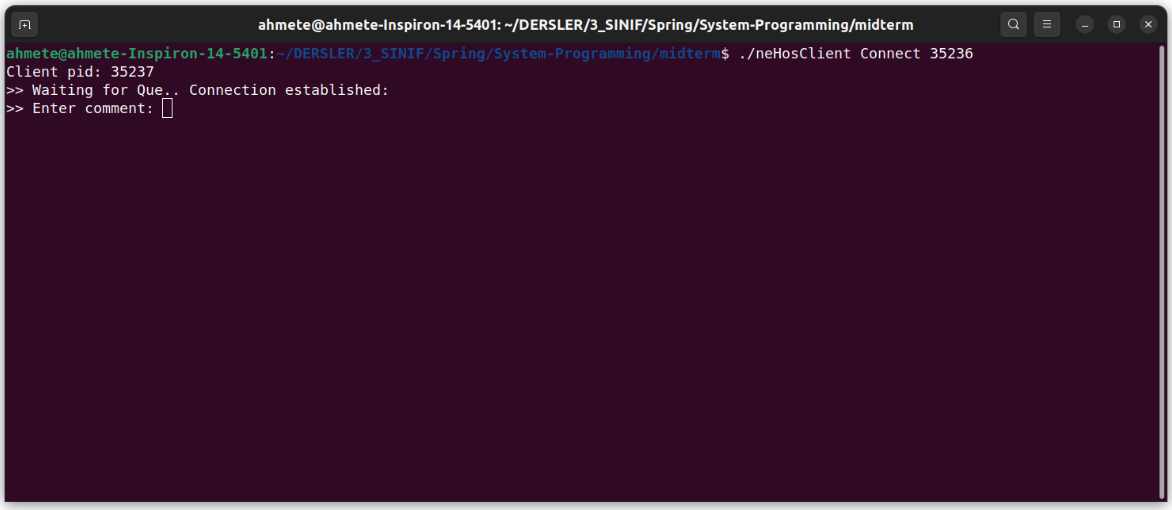
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35237
>> Waiting for Que.. Connection established:
>> Enter comment: quit
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```



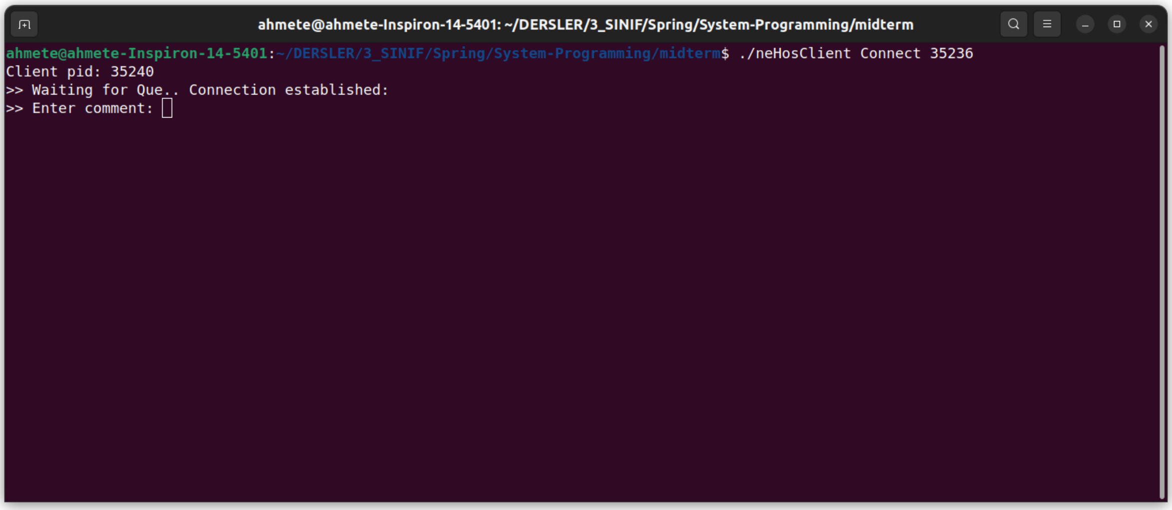
ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35236...
>> Waiting for clients...
>> Client PID 35237 connected as "client01"
>> Client PID 35240 connected as "client02"
>> Client PID 35244 connected as "client03"
Connection request PID 35246... Que FULL
```

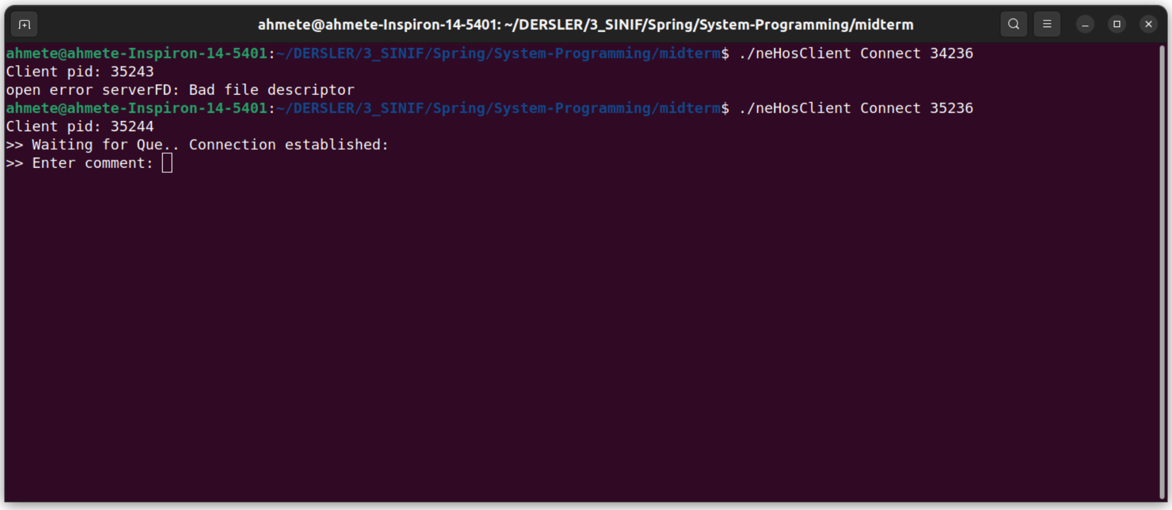


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

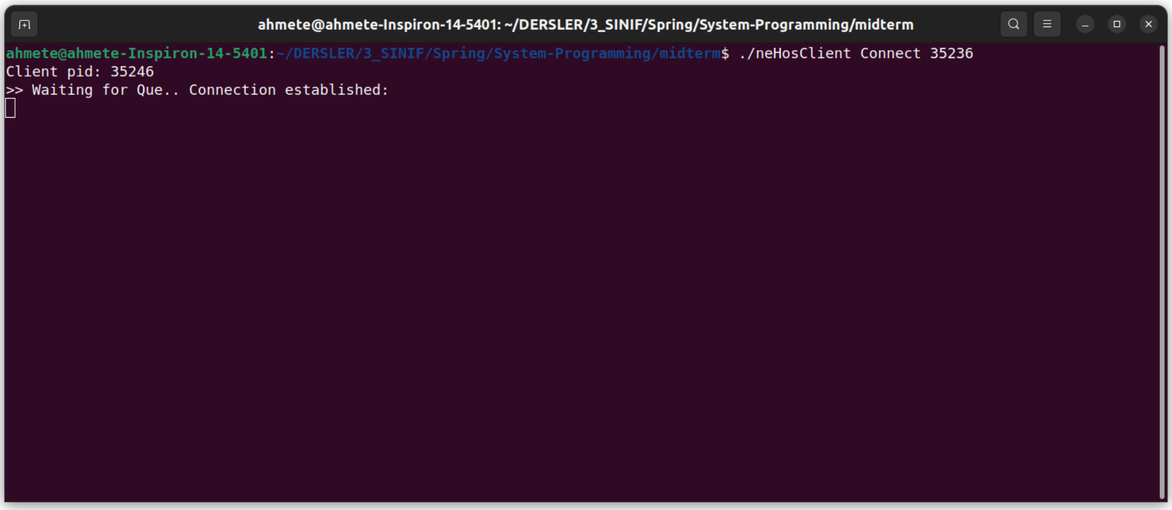
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35237
>> Waiting for Que.. Connection established:
>> Enter comment: 
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

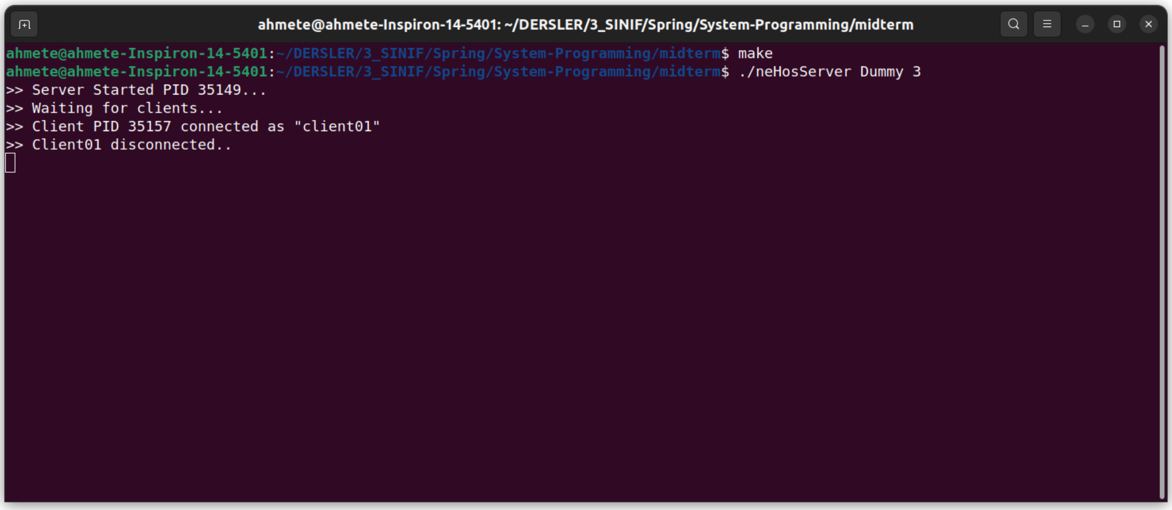
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35240
>> Waiting for Que.. Connection established:
>> Enter comment: 
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

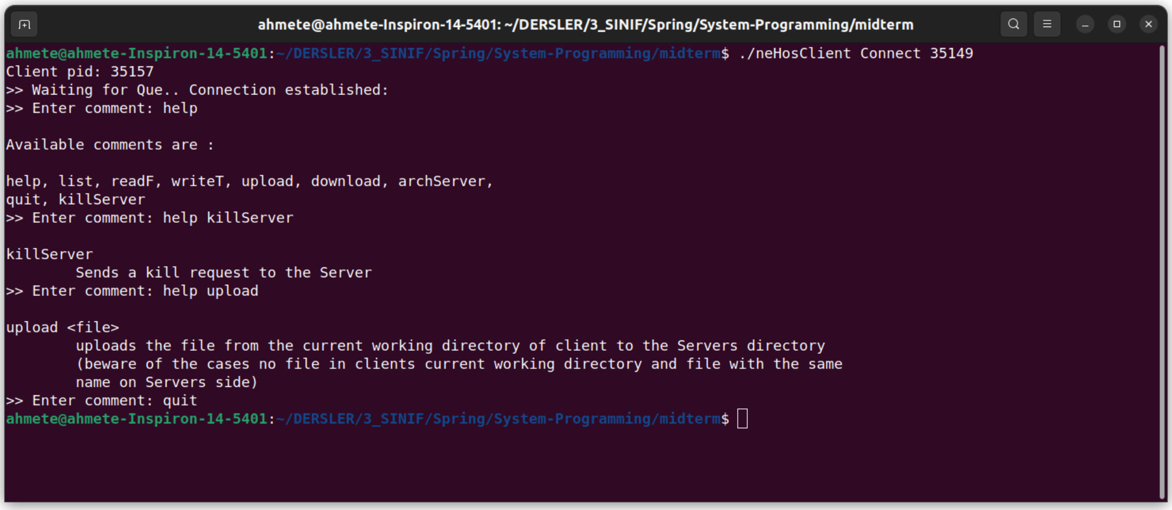
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 34236
Client pid: 35243
open error serverFD: Bad file descriptor
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35244
>> Waiting for Que.. Connection established:
>> Enter comment: 
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

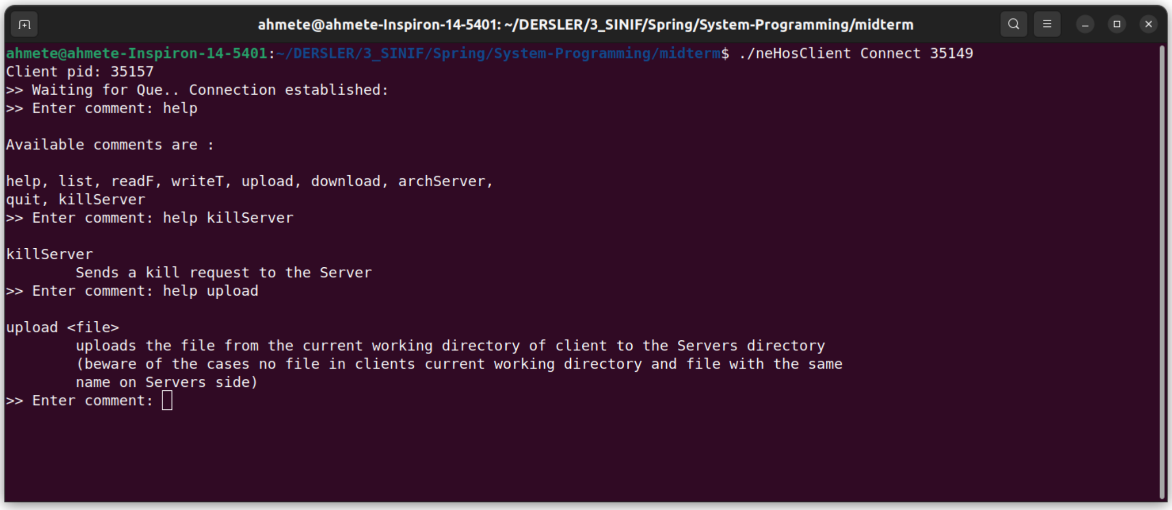
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35236
Client pid: 35246
>> Waiting for Que.. Connection established:
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

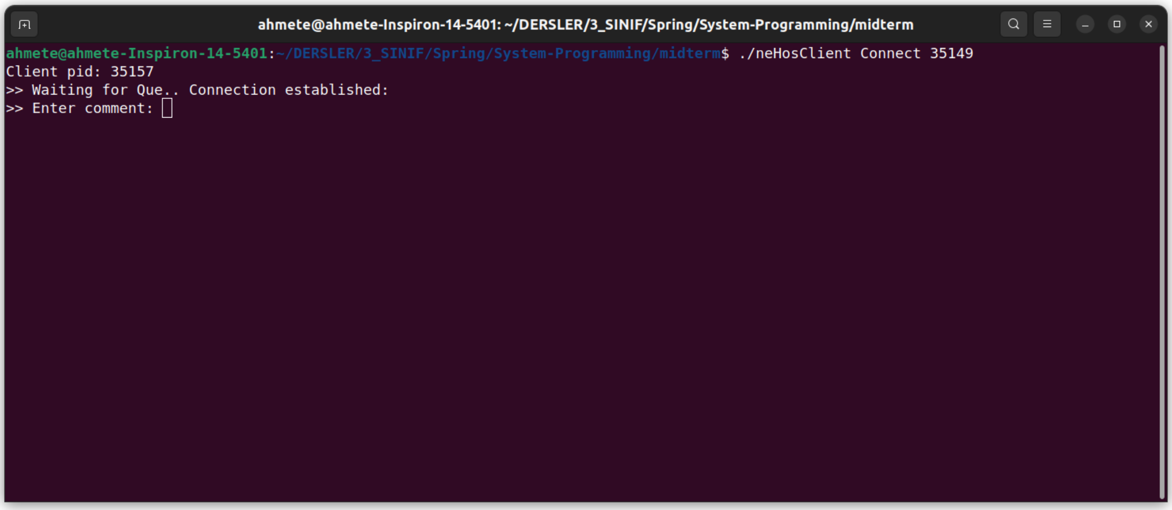
```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35149...
>> Waiting for clients...
>> Client PID 35157 connected as "client01"
>> Client01 disconnected..
```


ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35149
Client pid: 35157
>> Waiting for Que.. Connection established:
>> Enter comment: help

Available comments are :

help, list, readF, writeT, upload, download, archServer,
quit, killServer
>> Enter comment: help killServer

killServer
        Sends a kill request to the Server
>> Enter comment: help upload

upload <file>
        uploads the file from the current working directory of client to the Servers directory
        (beware of the cases no file in clients current working directory and file with the same
        name on Servers side)
>> Enter comment: quit
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ 
```

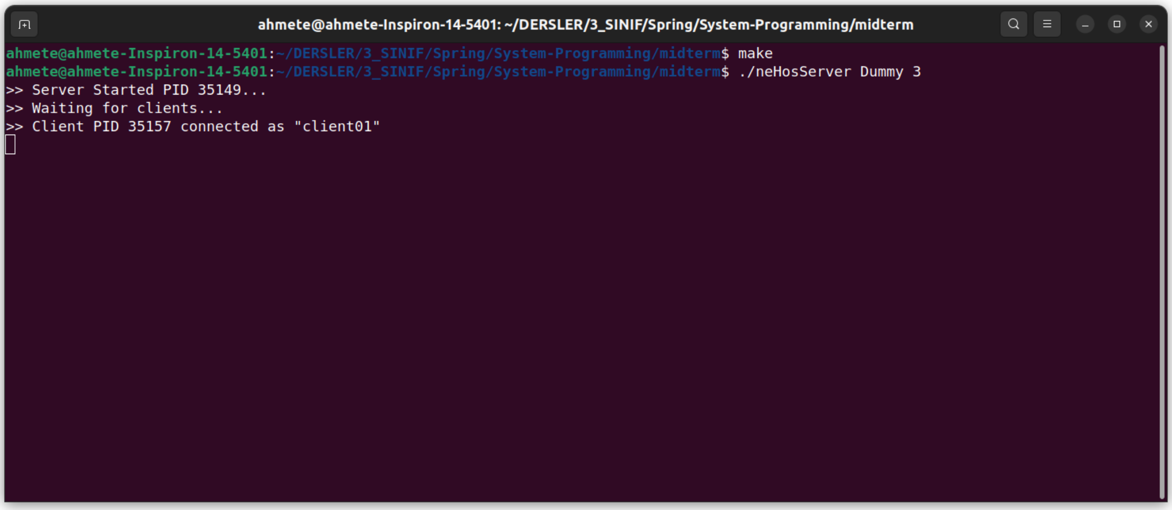

ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35149
Client pid: 35157
>> Waiting for Que.. Connection established:
>> Enter comment: help

Available comments are :

help, list, readF, writeT, upload, download, archServer,
quit, killServer
>> Enter comment: help killServer

killServer
        Sends a kill request to the Server
>> Enter comment: help upload

upload <file>
        uploads the file from the current working directory of client to the Servers directory
        (beware of the cases no file in clients current working directory and file with the same
        name on Servers side)
>> Enter comment: 
```



ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosClient Connect 35149
Client pid: 35157
>> Waiting for Que.. Connection established:
>> Enter comment: 
```



ahmete@ahmete-Inspiron-14-5401: ~/DERSLER/3_SINIF/Spring/System-Programming/midterm

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ make
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/3_SINIF/Spring/System-Programming/midterm$ ./neHosServer Dummy 3
>> Server Started PID 35149...
>> Waiting for clients...
>> Client PID 35157 connected as "client01"
```