To Run Program:

```
Make neHosServer  -> compile

Make neHosClient -> compile

./neHosServer <filename> <maxClient> -> run

./neHosClient connect/tryConnect serverPID -> run

Make clean -> log and exe files will be cleaned
```

Connection between Server and Client:

The communication between the client and server is facilitated through unnamed pipes known as FIFOs. The server maintains a well-known FIFO through which clients connect. When a client wishes to connect, it prepares a packet containing its PID, current working directory, and a wait bit indicating its preference for waiting in a queue. This packet is sent to the server via the well-known FIFO. The server then assesses the request, considering factors such as the status of the wait queue and the availability of child servers. Based on this assessment, the server responds to the client, indicating whether immediate service is possible or if a delay is expected.

Upon receiving the server's response, the client decides whether to wait or give up based on its connection option, specified as a command-line argument. If the client opts to connect, it opens a client request FIFO and waits until the server forks a child server, which then opens the other end of the client request FIFO. On the other hand, if the client chooses to give up, it terminates without establishing a connection with the server. This mechanism ensures efficient communication between the client and server while allowing clients to choose between waiting for service or terminating the connection attempt.

Request and Response:

Child servers in the system are managed by the main server through an array containing their PIDs. When a new child server is forked, its PID is added to this array. The main server periodically checks the number of child servers and the waiting queue for incoming client connections. If there are available slots for new child servers and waiting clients, the main server forks a new child server.

Each child server is responsible for handling communication with a specific client. Upon forking, a child server opens the write end of the client's FIFO, allowing the client to unblock and begin listening for commands. User commands sent by the client are converted into requests. The size of each request may vary depending on the command and its arguments. To ensure correct data transmission, the size of the request data is sent along with a header before transmitting the actual data.

Similarly, responses from the child server to the client are also sent with a header containing the size of the response data. This allows the client to correctly receive and interpret the response. By utilizing this method, child servers can effectively handle requests from clients and provide responses accordingly, facilitating efficient communication between clients and the server system.

For Sync:

To address race conditions in the server directory, the server implements the readers-writers problem solution. This involves creating structures called safe_file and safe_dir. These structures manage access to files in the server directory.

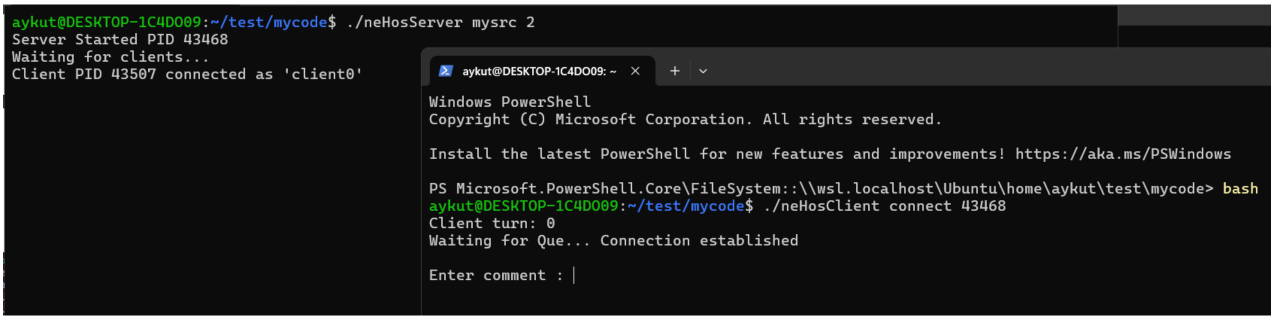
The main server initializes a shared memory region and sets up the safe_dir structure within it. Each child process can then access specific semaphores for unique files by accessing this shared memory region.

The safe_file structure contains the file's name, reader and writer counts, and semaphores. These semaphores ensure mutual exclusion, preventing race conditions when multiple processes attempt to access the same file simultaneously.

By implementing this approach, the server effectively manages access to files in the server directory, ensuring that concurrent operations are executed safely and without conflicts.

Application Results:

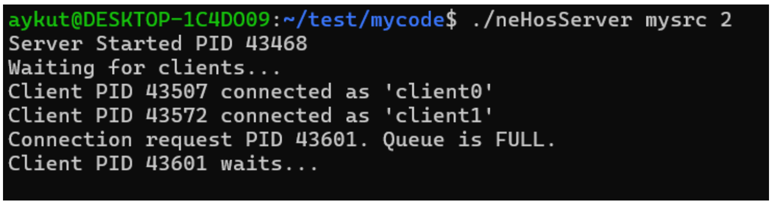
-Connect



```
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosServer mysrc 2
Server Started PID 43468
Waiting for clients...
Client PID 43507 connected as 'client0'
```

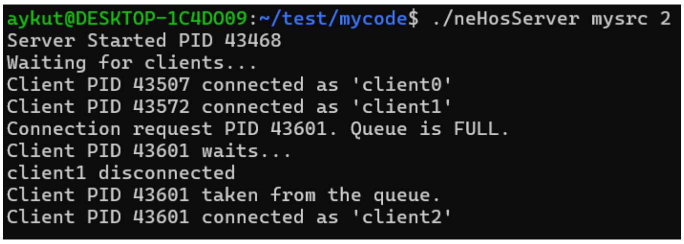
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\home\aykut\test\mycode> bash
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : |
```

-Enqueue



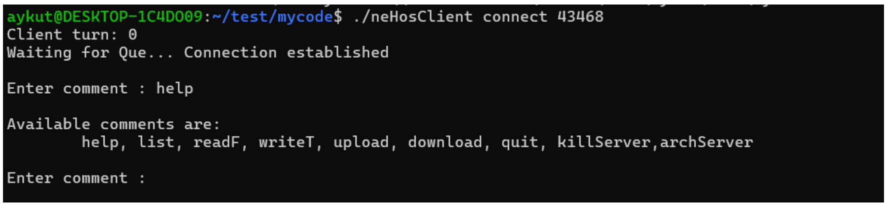
```
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosServer mysrc 2
Server Started PID 43468
Waiting for clients...
Client PID 43507 connected as 'client0'
Client PID 43572 connected as 'client1'
Connection request PID 43601. Queue is FULL.
Client PID 43601 waits...
```

-Dequeue



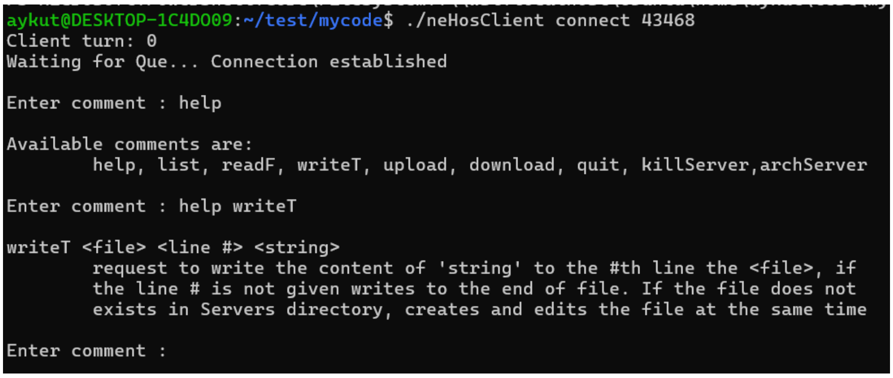
```
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosServer mysrc 2
Server Started PID 43468
Waiting for clients...
Client PID 43507 connected as 'client0'
Client PID 43572 connected as 'client1'
Connection request PID 43601. Queue is FULL.
Client PID 43601 waits...
client1 disconnected
Client PID 43601 taken from the queue.
Client PID 43601 connected as 'client2'
```

-Help command



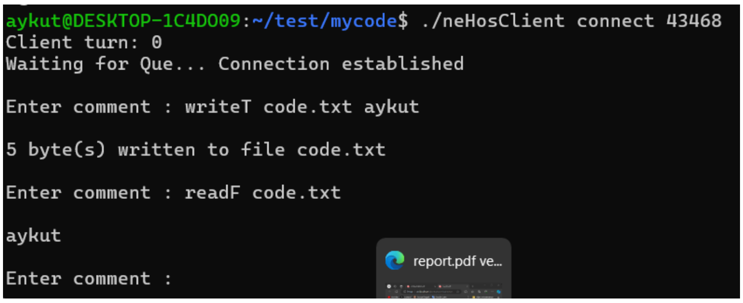
```
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : help

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

Enter comment :
```

-Help command with argument



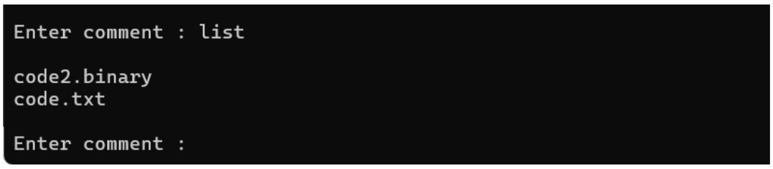
```
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : help

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

Enter comment : help writeT

writeT <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. If the file does not
        exists in Servers directory, creates and edits the file at the same time

Enter comment :
```

-WriteT and ReadF commands



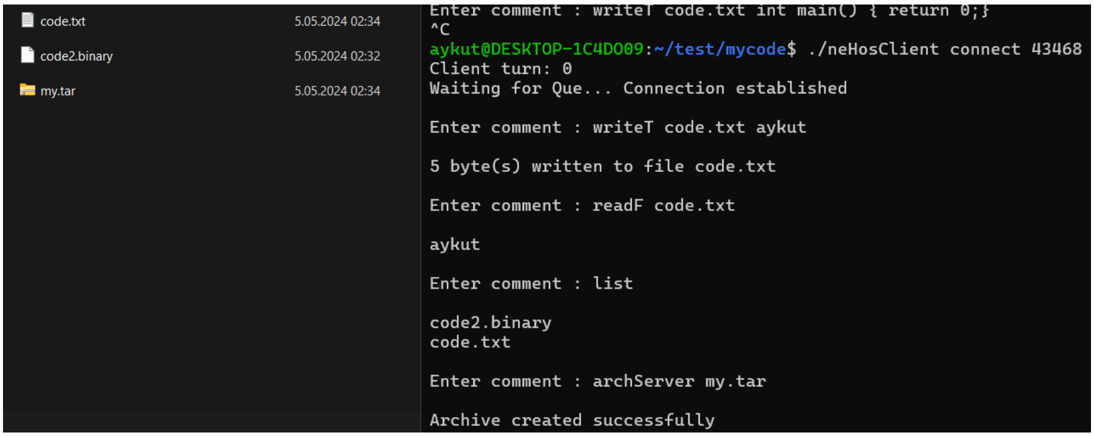
```
aykut@DESKTOP-1C4D009:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : writeT code.txt aykut

5 byte(s) written to file code.txt

Enter comment : readF code.txt

aykut

Enter comment :
```

-List command



```
Enter comment : list

code2.binary
code.txt

Enter comment :
```

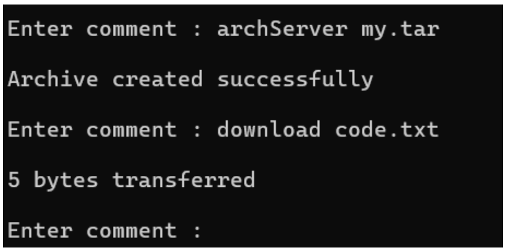
-ArchServer command



```
code.txt        5.05.2024 02:34
code2.binary    5.05.2024 02:32
my.tar          5.05.2024 02:34
```

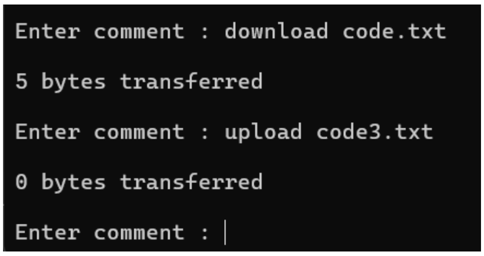
```
Enter comment : writeT code.txt int main() { return 0;}
^C
aykut@DESKTOP-1C4D009:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : writeT code.txt aykut

5 byte(s) written to file code.txt

Enter comment : readF code.txt

aykut

Enter comment : list

code2.binary
code.txt

Enter comment : archServer my.tar

Archive created successfully
```

-Download command



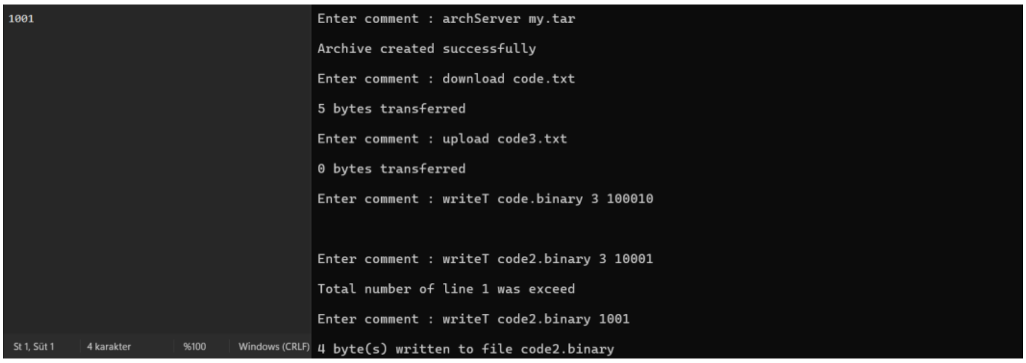
```
Enter comment : archServer my.tar

Archive created successfully

Enter comment : download code.txt

5 bytes transferred

Enter comment :
```

-Upload command



```
Enter comment : download code.txt

5 bytes transferred

Enter comment : upload code3.txt

0 bytes transferred

Enter comment : |
```

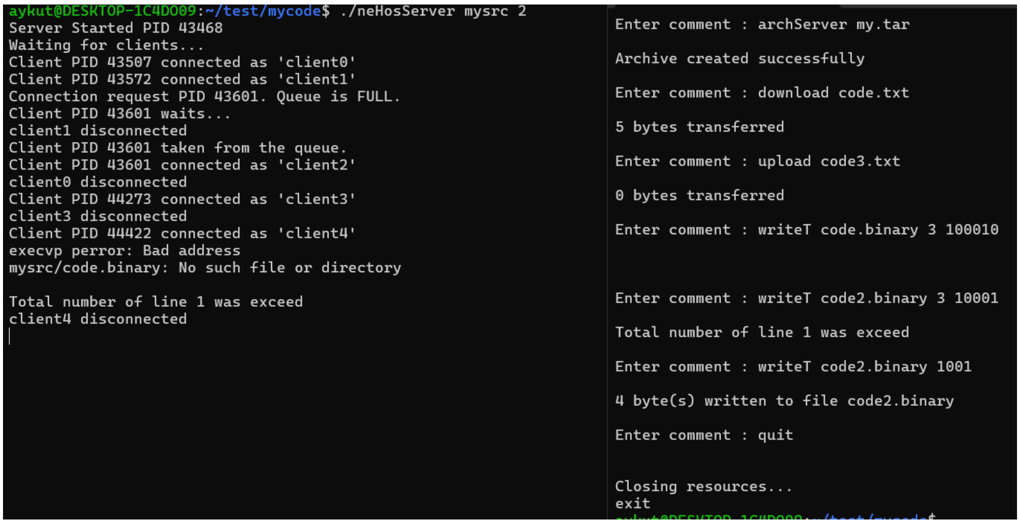
-writeF to binary file



```
1001
```

```
Enter comment : archServer my.tar

Archive created successfully

Enter comment : download code.txt

5 bytes transferred

Enter comment : upload code3.txt

0 bytes transferred

Enter comment : writeT code.binary 3 100010


Enter comment : writeT code2.binary 3 10001

Total number of line 1 was exceed

Enter comment : writeT code2.binary 1001

4 byte(s) written to file code2.binary
```

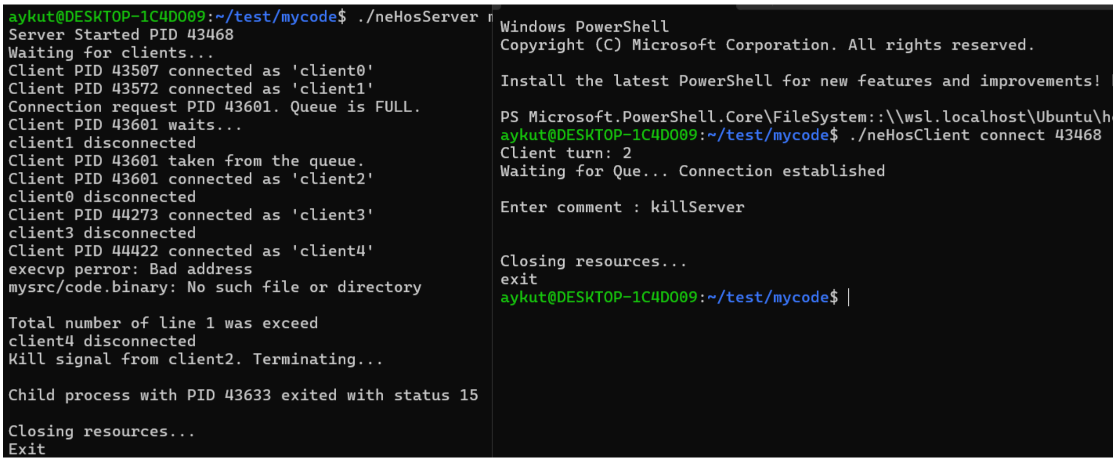
-quit command



```
aykut@DESKTOP-1C4D009:~/test/mycode$ ./neHosServer mysrc 2
Server Started PID 43468
Waiting for clients...
Client PID 43507 connected as 'client0'
Client PID 43572 connected as 'client1'
Connection request PID 43601. Queue is FULL.
Client PID 43601 waits...
client1 disconnected
Client PID 43601 taken from the queue.
Client PID 43601 connected as 'client2'
client0 disconnected
Client PID 44273 connected as 'client3'
client3 disconnected
Client PID 44422 connected as 'client4'
execvp perror: Bad address
mysrc/code.binary: No such file or directory

Total number of line 1 was exceed
client4 disconnected
|
```

```
Enter comment : archServer my.tar

Archive created successfully

Enter comment : download code.txt

5 bytes transferred

Enter comment : upload code3.txt

0 bytes transferred

Enter comment : writeT code.binary 3 100010


Enter comment : writeT code2.binary 3 10001

Total number of line 1 was exceed

Enter comment : writeT code2.binary 1001

4 byte(s) written to file code2.binary

Enter comment : quit


Closing resources...
exit
aykut@DESKTOP-1C4D009:~/test/mycode$
```

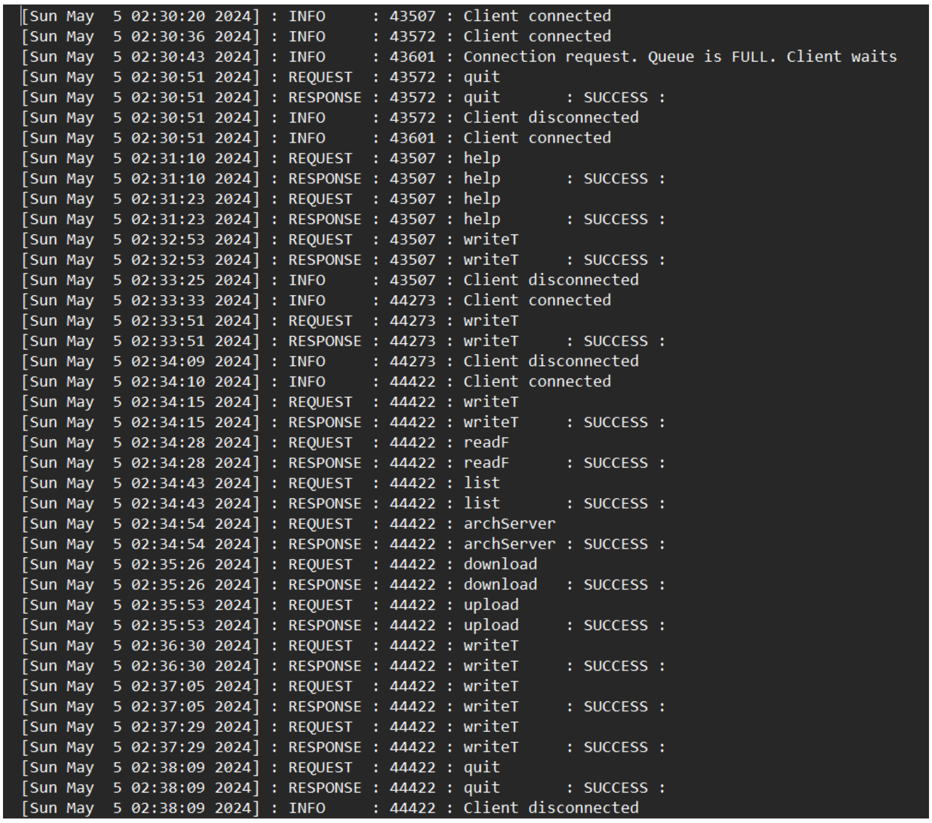
-killServer command



```
aykut@DESKTOP-1C4D009:~/test/mycode$ ./neHosServer r
Server Started PID 43468
Waiting for clients...
Client PID 43507 connected as 'client0'
Client PID 43572 connected as 'client1'
Connection request PID 43601. Queue is FULL.
Client PID 43601 waits...
client1 disconnected
Client PID 43601 taken from the queue.
Client PID 43601 connected as 'client2'
client0 disconnected
Client PID 44273 connected as 'client3'
client3 disconnected
Client PID 44422 connected as 'client4'
execvp perror: Bad address
mysrc/code.binary: No such file or directory

Total number of line 1 was exceed
client4 disconnected
Kill signal from client2. Terminating...

Child process with PID 43633 exited with status 15

Closing resources...
Exit
```

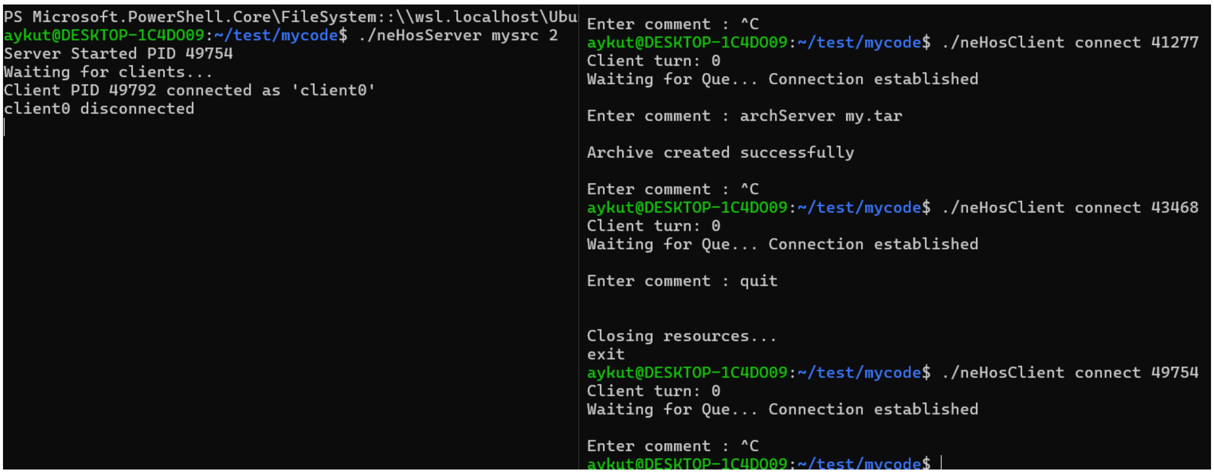
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!

PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu\h
aykut@DESKTOP-1C4D009:~/test/mycode$ ./neHosClient connect 43468
Client turn: 2
Waiting for Que... Connection established

Enter comment : killServer


Closing resources...
exit
aykut@DESKTOP-1C4D009:~/test/mycode$ |
```

-logfile output



```
[Sun May  5 02:30:20 2024] : INFO     : 43507 : Client connected
[Sun May  5 02:30:36 2024] : INFO     : 43572 : Client connected
[Sun May  5 02:30:43 2024] : INFO     : 43601 : Connection request. Queue is FULL. Client waits
[Sun May  5 02:30:51 2024] : REQUEST  : 43572 : quit
[Sun May  5 02:30:51 2024] : RESPONSE : 43572 : quit       : SUCCESS :
[Sun May  5 02:30:51 2024] : INFO     : 43572 : Client disconnected
[Sun May  5 02:30:51 2024] : INFO     : 43601 : Client connected
[Sun May  5 02:31:10 2024] : REQUEST  : 43507 : help
[Sun May  5 02:31:10 2024] : RESPONSE : 43507 : help       : SUCCESS :
[Sun May  5 02:31:23 2024] : REQUEST  : 43507 : help
[Sun May  5 02:31:23 2024] : RESPONSE : 43507 : help       : SUCCESS :
[Sun May  5 02:32:53 2024] : REQUEST  : 43507 : writeT
[Sun May  5 02:32:53 2024] : RESPONSE : 43507 : writeT     : SUCCESS :
[Sun May  5 02:33:25 2024] : INFO     : 43507 : Client disconnected
[Sun May  5 02:33:33 2024] : INFO     : 44273 : Client connected
[Sun May  5 02:33:51 2024] : REQUEST  : 44273 : writeT
[Sun May  5 02:33:51 2024] : RESPONSE : 44273 : writeT     : SUCCESS :
[Sun May  5 02:34:09 2024] : INFO     : 44273 : Client disconnected
[Sun May  5 02:34:10 2024] : INFO     : 44422 : Client connected
[Sun May  5 02:34:15 2024] : REQUEST  : 44422 : writeT
[Sun May  5 02:34:15 2024] : RESPONSE : 44422 : writeT     : SUCCESS :
[Sun May  5 02:34:28 2024] : REQUEST  : 44422 : readF
[Sun May  5 02:34:28 2024] : RESPONSE : 44422 : readF      : SUCCESS :
[Sun May  5 02:34:43 2024] : REQUEST  : 44422 : list
[Sun May  5 02:34:43 2024] : RESPONSE : 44422 : list       : SUCCESS :
[Sun May  5 02:34:54 2024] : REQUEST  : 44422 : archServer
[Sun May  5 02:34:54 2024] : RESPONSE : 44422 : archServer : SUCCESS :
[Sun May  5 02:35:26 2024] : REQUEST  : 44422 : download
[Sun May  5 02:35:26 2024] : RESPONSE : 44422 : download   : SUCCESS :
[Sun May  5 02:35:53 2024] : REQUEST  : 44422 : upload
[Sun May  5 02:35:53 2024] : RESPONSE : 44422 : upload     : SUCCESS :
[Sun May  5 02:36:30 2024] : REQUEST  : 44422 : writeT
[Sun May  5 02:36:30 2024] : RESPONSE : 44422 : writeT     : SUCCESS :
[Sun May  5 02:37:05 2024] : REQUEST  : 44422 : writeT
[Sun May  5 02:37:05 2024] : RESPONSE : 44422 : writeT     : SUCCESS :
[Sun May  5 02:37:29 2024] : REQUEST  : 44422 : writeT
[Sun May  5 02:37:29 2024] : RESPONSE : 44422 : writeT     : SUCCESS :
[Sun May  5 02:38:09 2024] : REQUEST  : 44422 : quit
[Sun May  5 02:38:09 2024] : RESPONSE : 44422 : quit       : SUCCESS :
[Sun May  5 02:38:09 2024] : INFO     : 44422 : Client disconnected
```

-ctrl+c signal output



```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubu
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosServer mysrc 2
Server Started PID 49754
Waiting for clients...
Client PID 49792 connected as 'client0'
client0 disconnected
```

```
Enter comment : ^C
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 41277
Client turn: 0
Waiting for Que... Connection established

Enter comment : archServer my.tar

Archive created successfully

Enter comment : ^C
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 43468
Client turn: 0
Waiting for Que... Connection established

Enter comment : quit


Closing resources...
exit
aykut@DESKTOP-1C4DO09:~/test/mycode$ ./neHosClient connect 49754
Client turn: 0
Waiting for Que... Connection established

Enter comment : ^C
aykut@DESKTOP-1C4DO09:~/test/mycode$
```