FileTransfer: Test Document

Nicole Jingco
Tomas Quat

Table of Contents

Summary

The following section covers the summary of the results of the GET and SEND requests.

GET Requests

Case	Expected	Passed?
Retrieve File from Server	File found in the server and retrieved back to the client	Yes
Retrieve File not found in server	Get Error message that file does not exist in the server followed by promoting a new request.	Yes

SEND Requests

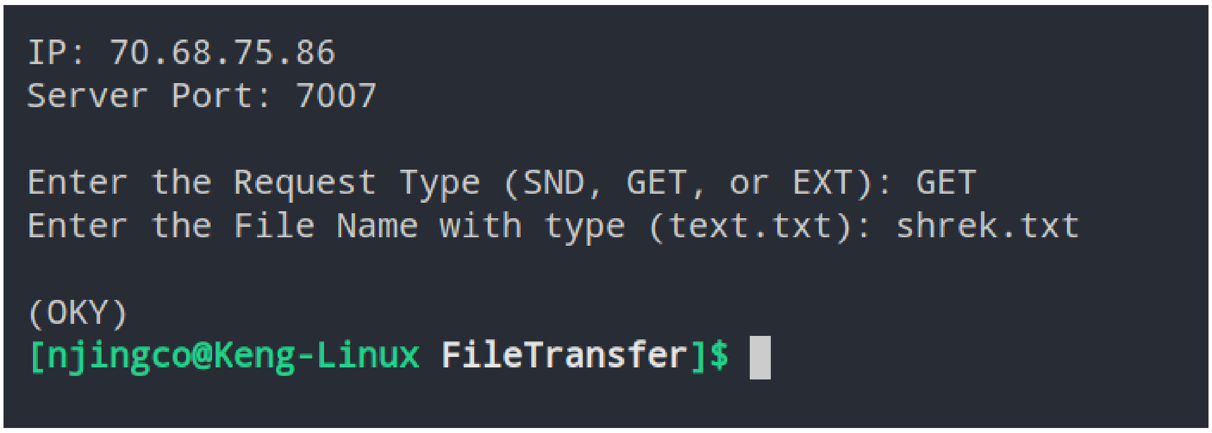
Case	Expected	Passed?
Send file to server	File sent and saved by server	Yes
Send file not found to server	Error Message is shown and requests to retype the file name	Yes

Other Functionality

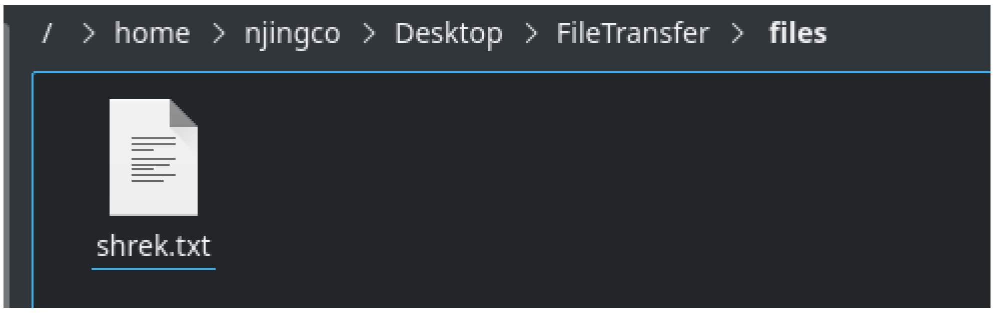
Case	Expected	Passed?
Request Port in Use	Error Message is shown and closes application	Yes
Client enters wrong command	Re-prompts the user for a new command	Yes
Connect to wrong Server IP	Error Message is shown and closes the application	Yes

Successful GET

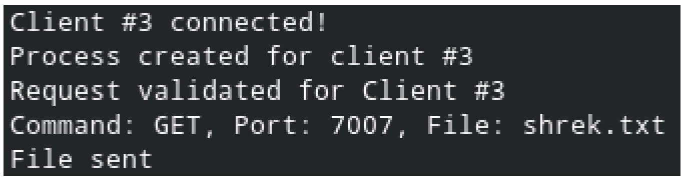
If the user enters valid ip, port, request and an existing file name in the server, it will write a file in their save folder.



```
IP: 70.68.75.86
Server Port: 7007

Enter the Request Type (SND, GET, or EXT): GET
Enter the File Name with type (text.txt): shrek.txt

(OKY)
[njingco@Keng-Linux FileTransfer]$
```



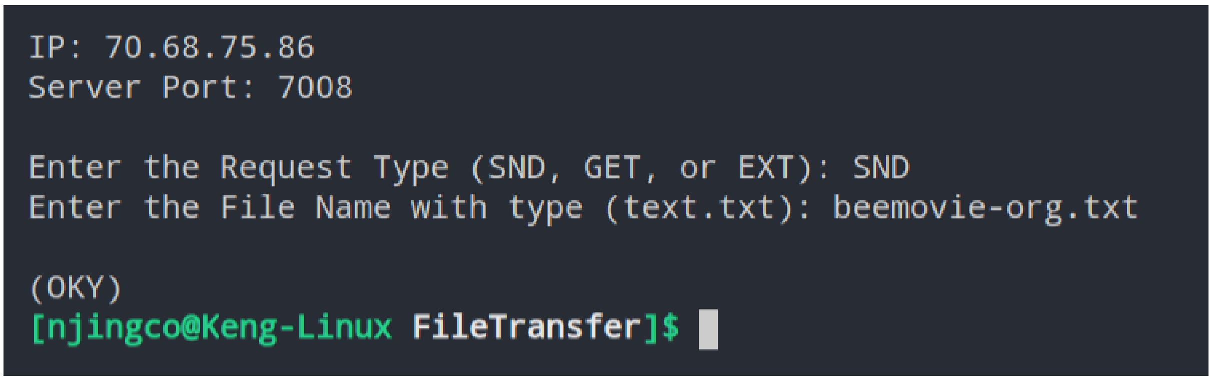
The following screenshot contains the server output over the course of a successful GET operation. A connected message and client number is printed when a client successfully connects to the server. Afterwards, a message is output upon successful creation of the clients handling process. If a client sends a valid request message, the validation message is printed followed by the components of the command including: Command Type, Data Port #, and Filename. Finally once the transfer has completed a message notifying that the job is done is output.



```
Client #3 connected!
Process created for client #3
Request validated for Client #3
Command: GET, Port: 7007, File: shrek.txt
File sent
```

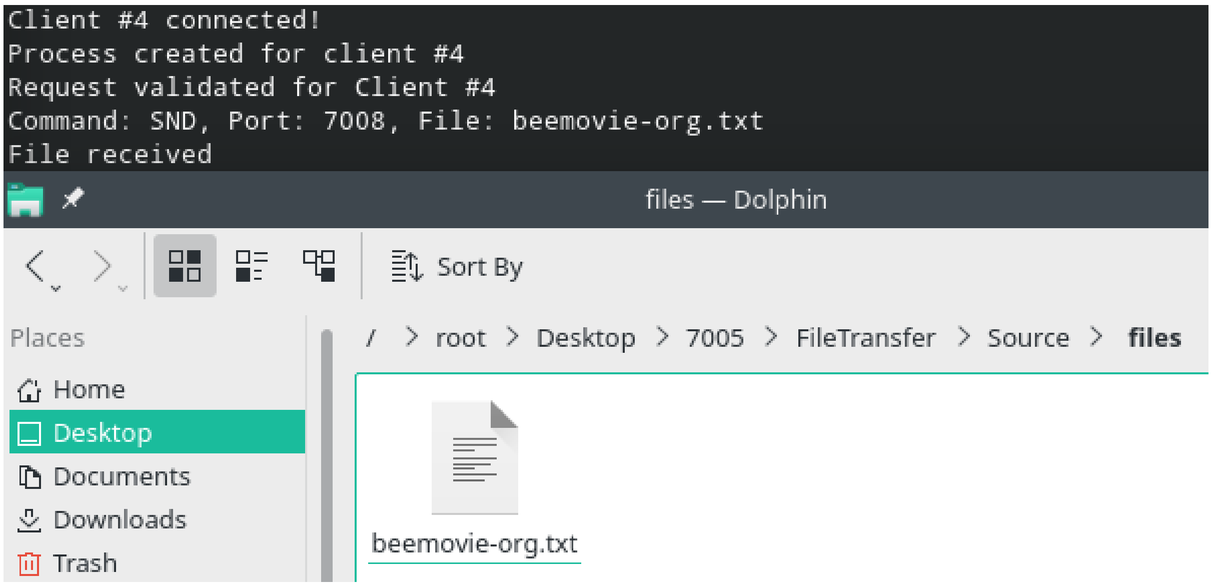
Successful SEND

If the user enters valid ip, port, request and an existing file name in their folder, it will write a file in the server's save folder.



```
IP: 70.68.75.86
Server Port: 7008

Enter the Request Type (SND, GET, or EXT): SND
Enter the File Name with type (text.txt): beemovie-org.txt

(OKY)
[njingco@Keng-Linux FileTransfer]$
```

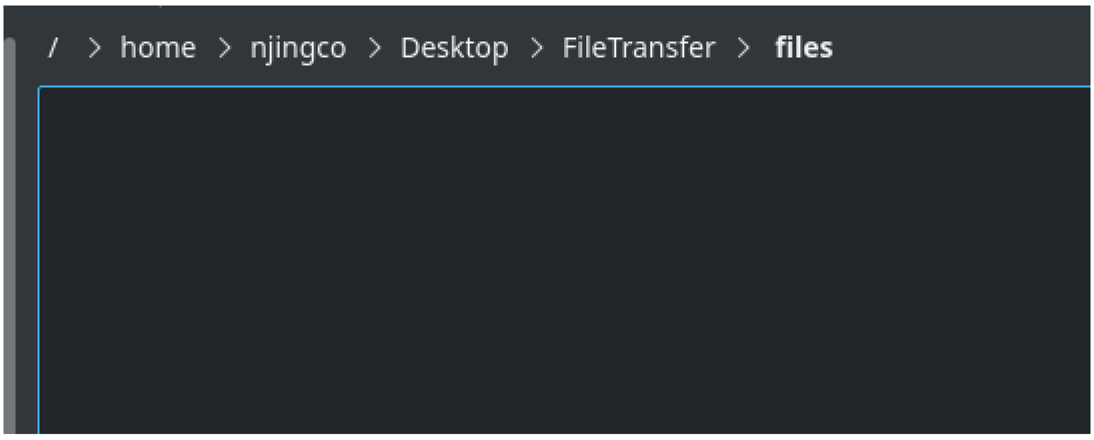
The following screenshot display the Server console output throughout the successful completion of a SND request. A connected message and client number is printed when a client successfully connects to the server. Afterwards, a message is output upon successful creation of the clients handling process. If a client sends a valid request message, the validation message is printed followed by the components of the command including: Command Type, Data Port #, and Filename. Once the transfer has been successfully completed a message notifying the completion is output.
Also included is the /files/ directory with the newly transferred file.



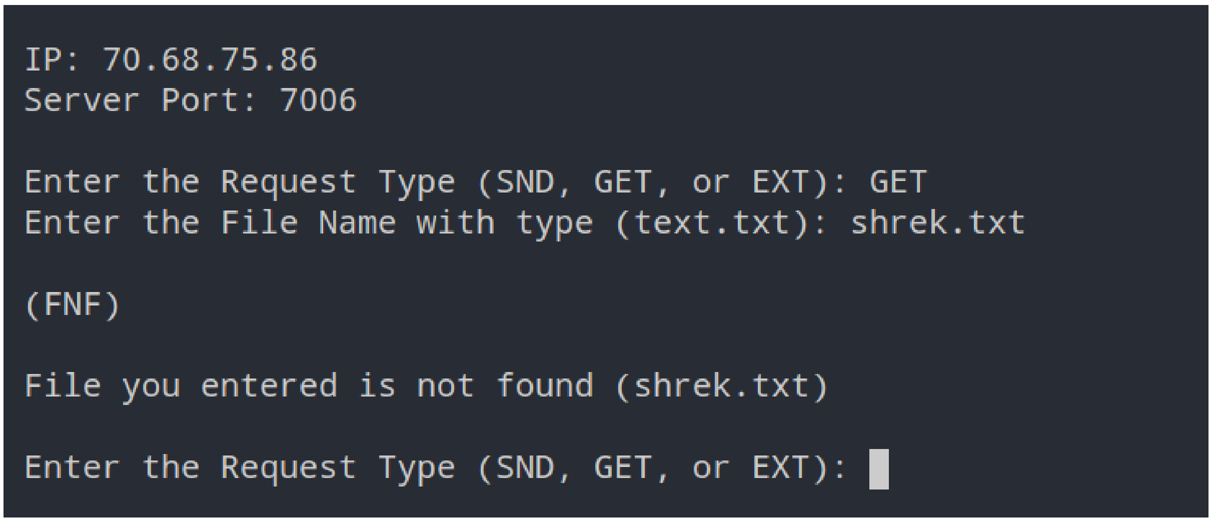
```
Client #4 connected!
Process created for client #4
Request validated for Client #4
Command: SND, Port: 7008, File: beemovie-org.txt
File received
```

File not Found (GET)

The following screenshot is what the save folder looks like before a GET Request.

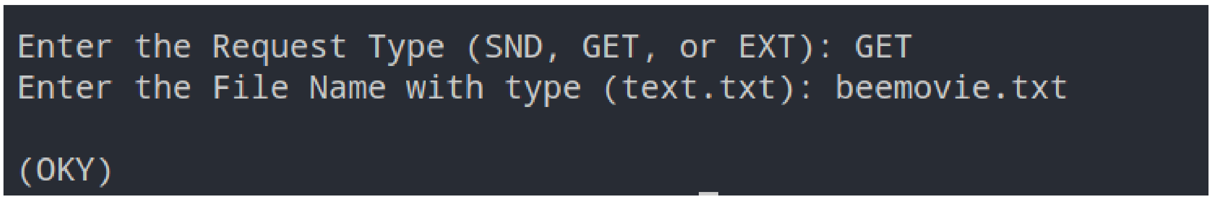


After the user enters the IP and Port number, it will prompt the user to enter a request type and a file name. If the GET request and file name, that the server does not have, is entered the server will send back a FNF meaning the file was not found. The program will also print out the file name input was not found in the server. It will then re-prompt the user to input the request type and file name again.



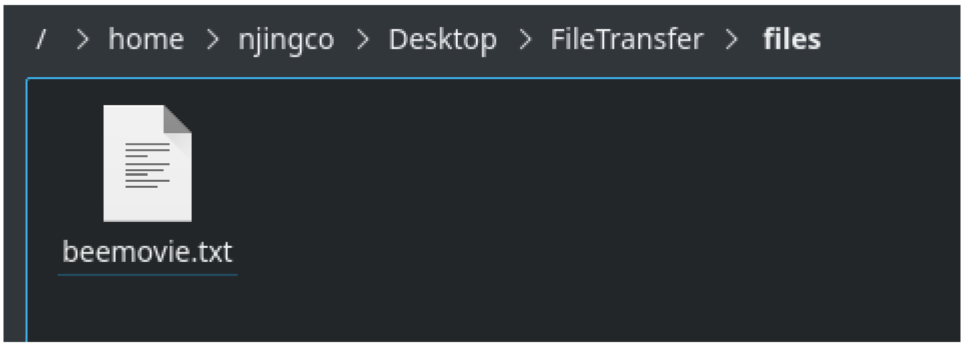
```
IP: 70.68.75.86
Server Port: 7006

Enter the Request Type (SND, GET, or EXT): GET
Enter the File Name with type (text.txt): shrek.txt

(FNF)

File you entered is not found (shrek.txt)

Enter the Request Type (SND, GET, or EXT): 
```

If found, the Server will send back an OKY signal meaning that the server has the file.



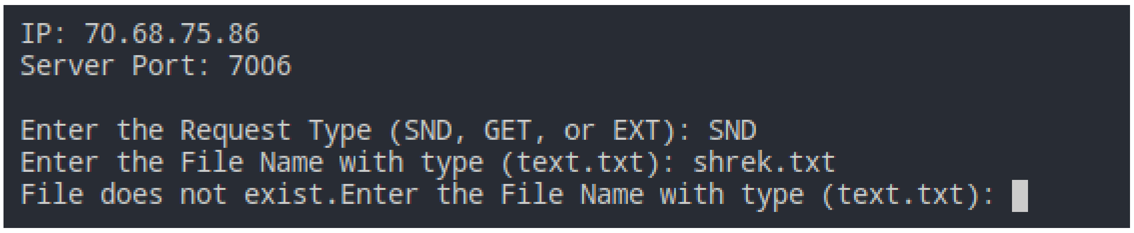
```
Enter the Request Type (SND, GET, or EXT): GET
Enter the File Name with type (text.txt): beemovie.txt

(OKY)
```

After the file is written, the file will show up in the save folder.



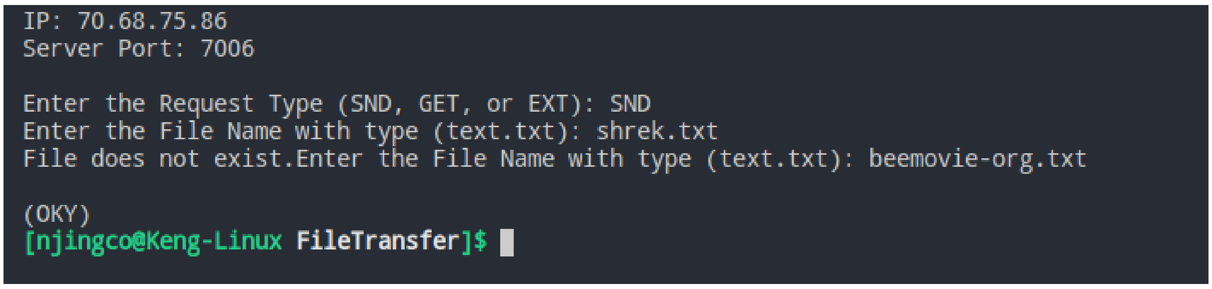
File Not Found (SND)

If the user enters a file that they do not have in their folder, it will re-prompt them again to enter a valid file name.



```
IP: 70.68.75.86
Server Port: 7006

Enter the Request Type (SND, GET, or EXT): SND
Enter the File Name with type (text.txt): shrek.txt
File does not exist.Enter the File Name with type (text.txt): 
```

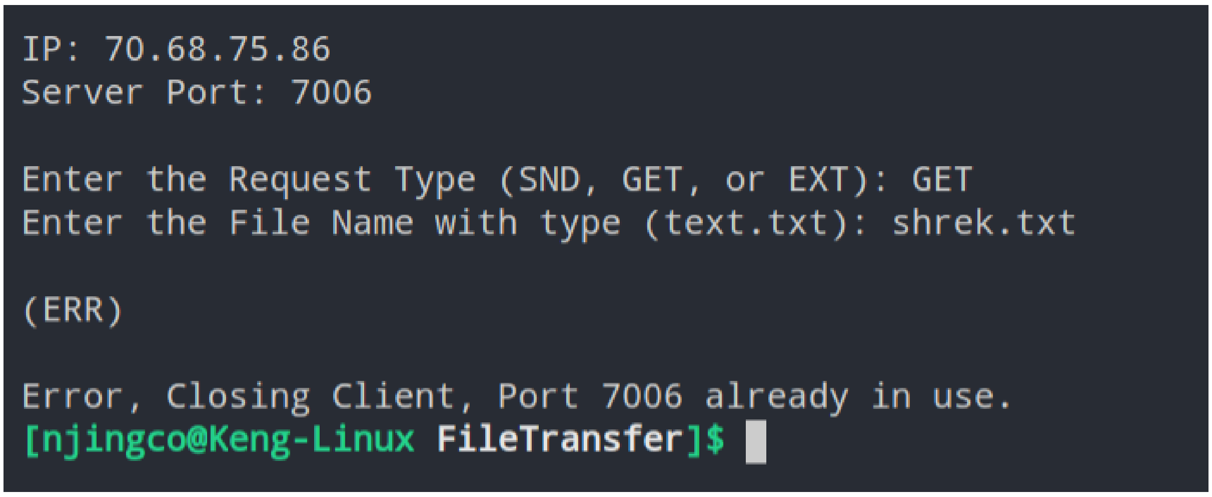
When a valid file is entered, it will send the file to the server as normal.



```
IP: 70.68.75.86
Server Port: 7006

Enter the Request Type (SND, GET, or EXT): SND
Enter the File Name with type (text.txt): shrek.txt
File does not exist.Enter the File Name with type (text.txt): beemovie-org.txt

(OKY)
[njingco@Keng-Linux FileTransfer]$ 
```

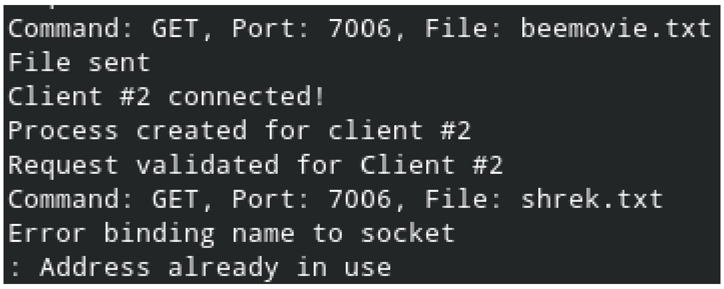
Request Port in Use

If the port requested for data transfer is in use, the server will send an error message to indicattole that they cannot use that port, then the program will close.



```
IP: 70.68.75.86
Server Port: 7006

Enter the Request Type (SND, GET, or EXT): GET
Enter the File Name with type (text.txt): shrek.txt

(ERR)

Error, Closing Client, Port 7006 already in use.
[njingco@Keng-Linux FileTransfer]$
```

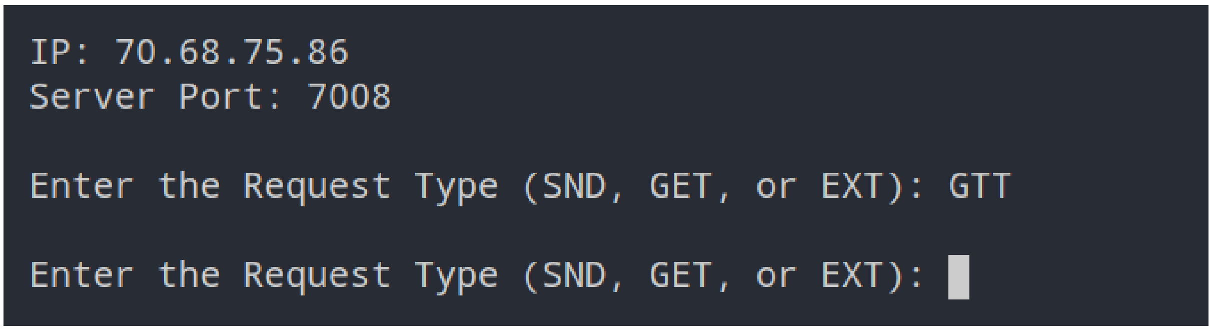
If a client connects and requests to use a data port already connected to, or recently used by, another client an error message is printed on the Server console, and a corresponding error message is sent back to the client.



```
Command: GET, Port: 7006, File: beemovie.txt
File sent
Client #2 connected!
Process created for client #2
Request validated for Client #2
Command: GET, Port: 7006, File: shrek.txt
Error binding name to socket
: Address already in use
```

Client wrong Command

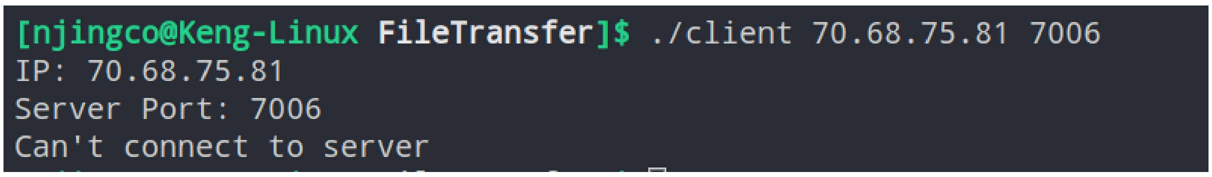
If the user enters an invalid request command, it will re-prompt them to enter the request again.



```
IP: 70.68.75.86
Server Port: 7008

Enter the Request Type (SND, GET, or EXT): GTT

Enter the Request Type (SND, GET, or EXT):
```

Client wrong IP

If the user enters the wrong IP, it will display a “Can’t connect to server” message and exit.



```
[njingco@Keng-Linux FileTransfer]$ ./client 70.68.75.81 7006
IP: 70.68.75.81
Server Port: 7006
Can't connect to server
```