# Description

## Client and Server

### Overview

The relationship between the client and server is that the server act as a tracker. It keep track of all the peers that has the file, all new peers to join, and remove the peer once they leave.

### Communication

Communication between the peers and the server is very simple.



A single serializable class contains all the commands and interfaces needed between the server and the client. Whenever this object is being serialized and passed onto the receiving end, a command is given and also additional information that will help the receiver perform the action dictates in the command. Below are the tables of the commands and the data that they carry with them.

**Commands from client to server**

|  |  |
| --- | --- |
| Register Request | |
| Client IP |  |
| Client port |  |
| Number of Files |  |
| Name of the files | This is a list containing the name of the files that the client wishes to register |
| Length of each of the files | This is a list containing the length of all the files |
|  |  |
|  |  |
| File List Request | |
| Client IP |  |
| Client Port |  |
|  |  |
|  |  |
| File Location Request | |
| Name of the file | This is the name of the file that the client wishes to obtain |
|  |  |
|  |  |
| Leave Request | |
| Client IP |  |
| Client Port |  |
|  |  |
|  |  |
| Data Request | |
| Name of file | This is the name of the file that the client wishes to obtain |
| Segment of file | This indicates which segment of the file the server should send |

Commands from Server to Client

|  |  |
| --- | --- |
| Register Reply | |
| List of booleans | This tells the client which of the files were successfully registered |
|  |  |
|  |  |
| File List Reply | |
| List of the files | The server will exclude any files that the client already has |
| Number of files |  |
| List of file lengths |  |
|  |  |
|  |  |
| File Location Reply | |
| Size of the file |  |
| No of end points |  |
| List of IP addresses | These are the IP addresses of the clients that has the file |
| List of Ports | These are the port numbers of the clients that has the file |
|  |  |
|  |  |
| Leave Reply | |
| None |  |
|  |  |
|  |  |
| Data Reply | |
| Hash for the file | This is sh 256 hash |
| segment of file | This tells the client which segment this is |
| Resulting data segment | This is data segment to send back |

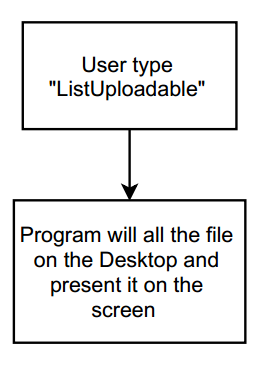
## Client to Client

Information flowing between client and client follows the exact same means as information between server and client. The two additional commands in the Server Client Message object provide clients with the ability to talk to each other.

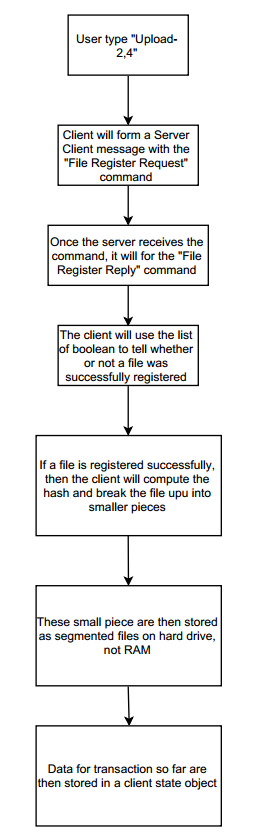
## Controls

Below are the figures depicting the action of the program as the user enters commands onto the console.

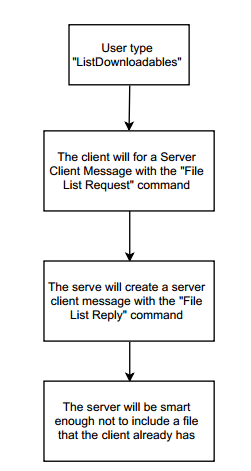
**“ListUploadables” Command**



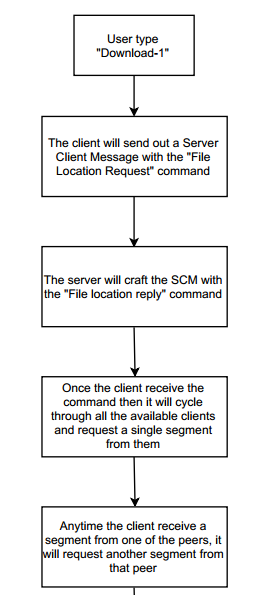
**“Upload-##” Command**

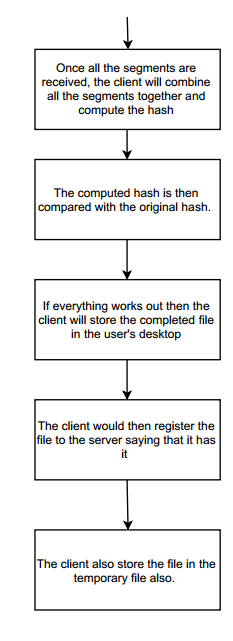


**“ListDownloadables” command**

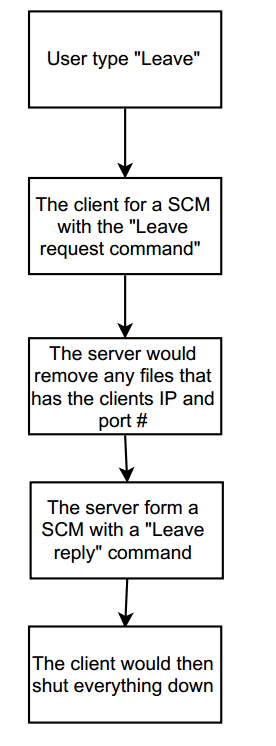


**“Download-#” Command**





**“Leave” Command**



# What works and doesn’t

## Working

* Interfaces: All the interfaces are implemented and are fully functional.
* Parallelism: Communication between the server and clients are 100% parallel. The server will support up to 40 parallel connection at a time.
* Error checking – Sh-256 Hash is performed and check on all the files
* Download completion – When a file is done download, the client will automatically register the client as a potential seed for that file.
* Storage of the file – The resulting files are successfully stored onto the desktop.
* Successful removal of the peer when the peer leaves the network

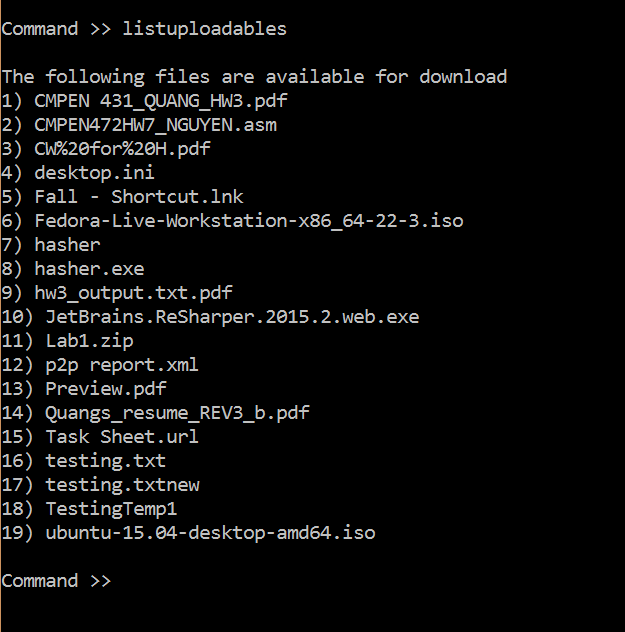
## Doesn’t work

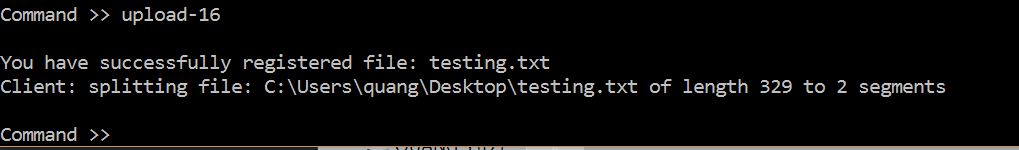
* Failure tolerance mechanism – No failure tolerance mechanism were implemented
* Status – The program does not support the display of the progress of the download thus far.
* Problems with network connection when move to Wifi. Connections works fine when the program is just doing internal networking, but it only work some of the time when using it with wifi. I suspect that this is because of the asynchronous function not being ok with the long delay.

# Sample output

## Client #1

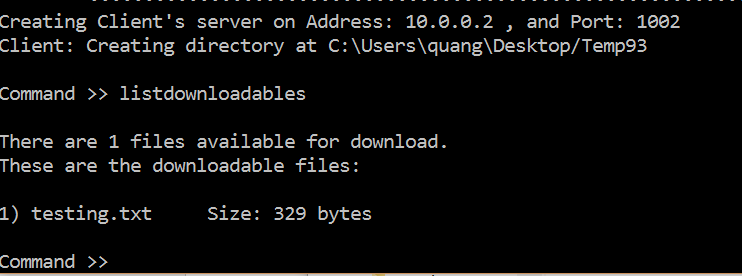
In this case, the client fires the program and decided to upload a file. See below for the command to bring up the list of files to upload and the command to upload file #16.

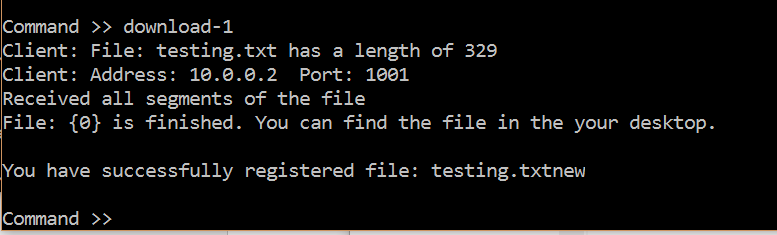




## Client #2

In this case, client #2 just enters the program and he/she query to see if any files are available for download. Since he/she sees that the file Testing.txt is available for download, he/she will download it. Below are the commands to do so and the output associate with it.





## Leaving

Lastly, when a client wants to quit, he/she can use the “leave” command. Please see below for an example of the leave command

