Communication Networks
Assignment - Socket Programming

Submitted By:

Aditya Aggarwal (20161129)

This Assignment is regarding creating a File Sharing Protocol similar to DropBox with the following features:

- Know the files present on each server's machine in the designated shared folder using TCP protocol as default.Download files from the shared folder.
- File transfer should incorporate MD5 checksum to handle file transfer errors.

In order to start the server just run python server.py and in another terminal run python client.py. The code is implemented in Python2.

Command 1: IndexGet
Request to display the files in the shared folder which is assumed to be the same directory as the code.
Flags: **shortlist**, **longlist**

shortlist: Returns the Name, Size, Timestamp, Extension of all the files in the shared folder within the given timestamp.
IndexGet shortlist <start_timestamp> <end_timestamp>
Example Query:

```
--> IndexGet shortlist 2019-03-20 00:00:00 2019-03-23 22:00:00
Name: Hello.txt     Size: 16     Timestamp: 2019-03-23 21:20:26     Extension: .txt
Name: server.py     Size: 7597   Timestamp: 2019-03-23 21:17:46     Extension: .py
Name: Assignment.pdf     Size: 65110   Timestamp: 2019-03-23 15:56:59     Extension: .pdf
Name: client.py     Size: 2841   Timestamp: 2019-03-23 17:28:10     Extension: .py
```

Bonus: Return extension specific files.(Example .pdf, .txt)
*IndexGet shortlist <start_timestamp> <end_timestamp> *.txt*
Example Query:

```
--> IndexGet shortlist 2019-03-20 00:00:00 2019-03-23 22:00:00 *.txt
Name: Hello.txt     Size: 16     Timestamp: 2019-03-23 21:20:26     Extension: .txt
```

If no such files exist then the server returns nothing.

longlist: Returns the Name, Size, Timestamp, Extension of all the files in the shared folder.
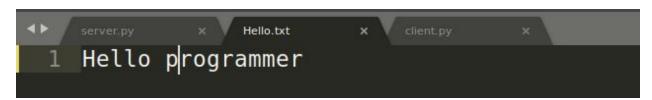
IndexGet longlist

Example Query:

```
aditya@aditya:~/Desktop/test$ python client.py
Socket object created successfully
Sat Mar 23 21:19:39 2019
Thank you for connecting
--> IndexGet longlist
Name: Book.pdf    Size: 9243279    Timestamp: 2019-03-14 05:42:49    Extension: .pdf
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:19:12    Extension: .txt
Name: Little_Theorem.pdf    Size: 374463    Timestamp: 2019-02-06 21:39:24    Extension: .pdf
Name: Chapter_3_V6.01.ppt    Size: 10322944    Timestamp: 2019-03-14 09:20:26    Extension: .ppt
Name: server.py    Size: 7597    Timestamp: 2019-03-23 21:17:46    Extension: .py
Name: Book2.pdf    Size: 19567290    Timestamp: 2019-03-13 20:27:42    Extension: .pdf
Name: Assignment.pdf    Size: 65110    Timestamp: 2019-03-23 15:56:59    Extension: .pdf
Name: client.py    Size: 2841    Timestamp: 2019-03-23 17:28:10    Extension: .py
-->
```

Bonus: Return longlist for only *.txt file containing word "Programmer" in it.

*IndexGet longlist *.txt*

Example Query:

```
--> IndexGet longlist *.txt
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
```

Contents of file Hello.txt

```
server.py    ×    Hello.txt    ×    client.py    ×
1   Hello programmer
```

Command 2: FileHash

Indicates that the client wants to check if any of the files on the other end have been changed.

Flags: **verify, checkall**

verify: Checks for the specific file name provided as command line argument and return its 'checksum' and 'last modified' timestamp.

FileHash verify <filename>

Example Query:

Client:

```
aditya@aditya:~/Desktop/test$ python client.py
Socket object created successfully Help    All changes saved in Drive
Sat Mar 23 20:23:02 2019
Thank you for connecting
--> FileHash verify server.py
Hash: 9b78f4025da3061b2fb25c93496755ea    Last Modified: 2019-03-23 19:58:08
-->
```

Server:

```
aditya@aditya:~/Acads/Sem6/CN$ python server.py
Socket successfully created ls  Add-ons  Help    All changes saved in Drive
Connection received from: ('127.0.0.1', 56960)
Hash sent for the file server.py
```

In case the requested file is not present in the shared folder, it returns an Error.

```
aditya@aditya:~/Desktop/test$ python client.py
Socket object created successfully Help    All changes saved in Drive
Sat Mar 23 20:23:02 2019
Thank you for connecting
--> FileHash verify server.py
Hash: 9b78f4025da3061b2fb25c93496755ea    Last Modified: 2019-03-23 19:58:08
--> FileHash verify file.py
Requested file does not exist.
-->
```

checkall: Perform verify flag on all the files in the shared directory. Thus it returns File Name, Checksum and Last Modified Timestamp for all the files.
FileHash checkall
Example Query:
Client:

```
--> FileHash checkall
```

Output received by Client

```
File Name: Book.pdf    Hash: 4b6b1e27e2c01d0d97d0d147df56114c    Last Modified: 2019-03-14 05:42:49
File Name: Little_Theorem.pdf    Hash: 7e56d0fcd9ef2f590369ea24a9ef1aaa    Last Modified: 2019-02-06 21:39:24
File Name: Chapter_3_V6.01.ppt    Hash: 04fcaa4d81e4c6f11497e3e87fcd2ee9    Last Modified: 2019-03-14 09:20:26
File Name: server.py    Hash: 9b78f4025da3061b2fb25c93496755ea    Last Modified: 2019-03-23 19:58:08
File Name: Book2.pdf    Hash: 5c4480ad460ca303410de770e695d25d    Last Modified: 2019-03-13 20:27:42
File Name: Assignment.pdf    Hash: 2e17d52a02fdb43db743c9480ac8758f    Last Modified: 2019-03-23 15:56:59
File Name: client.py    Hash: e2309f70bf1e36bf4bca121f612ae8a7    Last Modified: 2019-03-23 17:28:10
```

Command 3: FileDownload
Used to download the files from the shared folder of connected users. In case no flag is mentioned TCP is considered as default.
Flags: **TCP,UDP**

TCP: Creates a default TCP connection and then downloads the mentioned file.
FileDownload <filename> TCP
Example Query:

```
--> FileDownload server.py TCP
File downloaded
File Name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    Hash: 9b78f4025da3061b2fb25c93496755ea
-->
```
○ The flag variable can take the value TCP or UDP depe
on the users request.
○ If a socket is not available , it should be created and b
clients must use this socket for the file transfer.

```
--> FileDownload server.py
File downloaded
File Name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    Hash: 9b78f4025da3061b2fb25c93496755ea
-->
```

A fourth argument can also be mentioned as the new path of the file at the client's end. In case it is not specified it is taken same as the source name.
FileDownload <filename> TCP <destinationpath>

Example Query:

```
--> FileDownload server.py TCP
File downloaded
File Name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    Hash: 9b78f4025da3061b2fb25c93496755ea
--> FileDownload server.py TCP new_server.py
File downloaded
File Name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    Hash: 9b78f4025da3061b2fb25c93496755ea
-->
```

If the specified file is not present error messages are returned.

Example Query:

```
--> FileDownload file.py TCP
Error opening file or file does not exist
-->
```

UDP: All the above functions are also implemented for the UDP protocol.

Here are a few examples:

FileDownload <filename> UDP

Example Query:

```
--> FileDownload server.py UDP
File name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    hash: 9b78f4025da3061b2fb25c93496755ea
File Downloaded
-->
```

```
--> FileDownload server.py UDP
File name: server.py    Size: 7820    Timestamp: 2019-03-23 19:58:08    hash: 9b78f4025da3061b2fb25c93496755ea
File Downloaded
--> FileDownload file.py UDP
Error opening file or file does not exist
-->
```

Appropriate Messages also appear on the server side and proper error handling is also done in case of incorrect commands.

Command 4: exit

Disconnects the client from server while the server waits for a new client.

```
aditya@aditya:~/Acads/Sem6/CN$ python server.py
Socket successfully created
Connection received from: ('127.0.0.1', 43558)
Client aborted connection
```

Client:

```
aditya@aditya:~/Desktop/test$ python client.py
Socket object created successfully
Sat Mar 23 21:37:23 2019
Thank you for connecting
--> IndexGet shortlist 2019-03-20 00:00:00 2019-03-24 00:00:00
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    Extension: .py
Name: Assignment.pdf    Size: 65110    Timestamp: 2019-03-23 15:56:59    Extension: .pdf
--> IndexGet shortlist 2019-03-20 00:00:00 2019-03-24 00:00:00 *.txt
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt

--> IndexGet longlist
Name: Book.pdf    Size: 9243279    Timestamp: 2019-03-14 05:42:49    Extension: .pdf
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Name: Little_Theorem.pdf    Size: 374463    Timestamp: 2019-02-06 21:39:24    Extension: .pdf
Name: Chapter_3_V6.01.ppt    Size: 10322944    Timestamp: 2019-03-14 09:20:26    Extension: .ppt
Name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    Extension: .py
Name: Book2.pdf    Size: 19567290    Timestamp: 2019-03-13 20:27:42    Extension: .pdf
Name: Assignment.pdf    Size: 65110    Timestamp: 2019-03-23 15:56:59    Extension: .pdf

--> IndexGet longlist *.txt
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt

--> FileHash checkall
File Name: Book.pdf    Hash: 4b6b1e27e2c01d0d97d0d147df56114c    Last Modified: 2019-03-14 05:42:49
File Name: Hello.txt    Hash: 7b968dd957566e3ce04e6a9344a32e8e    Last Modified: 2019-03-23 21:20:26
File Name: Little_Theorem.pdf    Hash: 7e56d0fcd9ef2f590369ea24a9ef1aaa    Last Modified: 2019-02-06 21:39:24
File Name: Chapter_3_V6.01.ppt    Hash: 04fcaa4d81e4c6f11497e3e87fcd2ee9    Last Modified: 2019-03-14 09:20:26
File Name: server.py    Hash: e8a2a95212404cfd2d9743d530ccb61d    Last Modified: 2019-03-23 21:35:25
File Name: Book2.pdf    Hash: 5c4480ad460ca303410de770e695d25d    Last Modified: 2019-03-13 20:27:42
File Name: Assignment.pdf    Hash: 2e17d52a02fdb43db743c9480ac8758f    Last Modified: 2019-03-23 15:56:59
--> FileHash verify server.py
Hash: e8a2a95212404cfd2d9743d530ccb61d    Last Modified: 2019-03-23 21:35:25
--> FileDownload server.py TCP
File downloaded
File Name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    Hash: e8a2a95212404cfd2d9743d530ccb61d
--> FileDownload server.py UDP ./new_server.py
File name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    hash: e8a2a95212404cfd2d9743d530ccb61d
File Downloaded
--> exit
aditya@aditya:~/Desktop/test$
```

Server:

```
aditya@aditya:~/Acads/Sem6/CN$ python server.py
Socket successfully created
Connection received from: ('127.0.0.1', 43578)
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    Extension: .py
Name: Assignment.pdf    Size: 65110    Timestamp: 2019-03-23 15:56:59    Extension: .pdf
Sent details successfully
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Sent details successfully
Name: Book.pdf    Size: 9243279    Timestamp: 2019-03-14 05:42:49    Extension: .pdf
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Name: Little_Theorem.pdf    Size: 374463    Timestamp: 2019-02-06 21:39:24    Extension: .pdf
Name: Chapter_3_V6.01.ppt    Size: 10322944    Timestamp: 2019-03-14 09:20:26    Extension: .ppt
Name: server.py    Size: 7667    Timestamp: 2019-03-23 21:35:25    Extension: .py
Name: Book2.pdf    Size: 19567290    Timestamp: 2019-03-13 20:27:42    Extension: .pdf
Name: Assignment.pdf    Size: 65110    Timestamp: 2019-03-23 15:56:59    Extension: .pdf
Sent details successfully
Name: Hello.txt    Size: 16    Timestamp: 2019-03-23 21:20:26    Extension: .txt
Sent detail successfully
Hash sent for all the files.
Hash sent for the file server.py
server.py sent through TCP.
server.py sent through UDP.
^[Client aborted connection
Invalid Command
Client Disconnected
```