

README

Name: Eashan Sapre

UNI: es4069

This is a network program that allows clients and servers to interact using commands. The program has two modes of operation: server mode and client mode.

Instructions:

Install the requirements using `pip install -r requirements.txt`

Open terminal or command prompt at the folder in which the `FileApp.py` file is located.

`FileApp.py` - Main python program file to run as client or server.

`Util.py` - utility file containing variable names

Run the below commands to start a server or client respectively.

The server and client can be started using the command line arguments "`-s`" and "`-c`" respectively.

```
FileApp -c <name> <server-ip> <server-port> <client-udp-port> <client-tcp-port
```

Eg. `python3 Network.py -c sam 127.0.0.1 4000 2100 3100`

Registration

The client can successfully register with an available username using the program. Upon successful registration, the client's local table is initialized. However, if the requested username is already taken by another client, the server rejects the request.

```
Eg. (base) bravo99@Saps-MacBook-Air client_1 % python3 FileApp.py -c joey 127.0.0.1 4000 5100 4100
```

```
>>> Welcome, You are registered.
```

```
>>> Client Table Updated
```

```
Eg. (base) bravo99@Saps-MacBook-Air ~ % python3 FileApp.py -s 4000
```

Running on IP:127.0.0.1

Running on port:4000

File Offering

The "setdir" command works as stated in the specification. The "offer" command will fail with an appropriate error message if no "setdir" command has succeeded.

```
>>>setdir /
```

```
>>> Successfully set <./> as the directory for searching offered files.
```

```
>>>offer pqr.txt abc.txt
```

```
>>>offer abc.txt
```

File Listing

The program can correctly list file offerings using the table with proper formatting. It will display a proper message when no files are being offered, and the file is updated when the client table is updated.

```
>>>list
```

File | Name | IP | Port

```
('abc.txt', 'joey', '127.0.0.1', 4100)
```

```
('mcq.txt', 'sam', '127.0.0.1', 3100)
```

File Transfer

Clients can successfully request and receive a file offered by another client using request command.>>> request filename client_name

```
>>>request abc.txt joey
```

De-Registration

The "dereg" command de-registers the client without exiting the client program.

```
>>>dereg joey
```

```
>>> you are offline. Bye
```

Sample Output

Ex1 . multiple clients are registered with server

```
Last login: Mon Mar 27 14:24:55 on ttys004
(base) bravo99@Shris-Air 4111_file_transfer_socket_programming_project % python3
FileApp.py -s 4000
Running on IP:192.168.1.156
Running on port:4000
client registered: eashan
client registered: martin
client registered: joey
client registered: sam

Last login: Mon Mar 27 14:23:12 on ttys002
(base) bravo99@Shris-Air client_2 % python3 FileApp.py -c martin 192.168.1.156 4000 2100 3100
>>> Welcome, You are registered.
>>> Client Table Updated
>>>

Last login: Mon Mar 27 14:27:26 on ttys000
(base) bravo99@Shris-Air client_3 % python3 FileApp.py -c eashan 192.168.1.156 4000 1000 7100
>>> Welcome, You are registered.
>>> Client Table Updated
>>>
```

Ex 2. File offered by Client 1 and Client 2. Table list is updated across all clients.

The screenshot shows a Mac desktop with three terminal windows open, illustrating a file transfer process between multiple clients using socket programming.

Terminal 1 (Client 1):

```
000 2100 3100
>>> Welcome, You are registered.
>>> Client Table Updated
>>>client table update received
>>>
setdir ./
>>> Successfully set <./> as the directory for searching offered files.
>>>offer work.pdf
>>> nickname details sent
>>>client table update received
>>>
```

Terminal 2 (Client 2):

```
000 2100 3100
>>> Welcome, You are registered.
>>> Client Table Updated
>>>client table update received
>>>
setdir ./
>>> Successfully set <./> as the directory for searching offered files.
>>>offer work.pdf
>>> nickname details sent
>>>client table update received
>>>
```

Terminal 3 (Client 3):

```
(base) bravo99@Shris-Air client_3 % python3 FileApp.py -c eashan 192.168.1.156 4000 8100 6100 -80x...
000 1000 7100
>>> Welcome, You are registered.
>>> Client Table Updated
>>>client table update received
>>>
client table update received
>>>
list
+-----+
| FILENAME | OWNER | IP ADDRESS | TCP PORT |
+-----+
| 7.pdf | sam | 192.168.1.156 | 6100 |
| work.pdf | martin | 192.168.1.156 | 3100 |
+-----+
```

The screenshots show the progression of file offers and table updates between the three clients, with the table list being updated across all clients in real-time.

Ex 3. Requesting and downloading file

```
Terminal Shell Edit View Window Help Mon 27 Mar 2:33 PM

4111_file_transfer_socket_programming_project — python3 FileApp.py -s ...
client_2 — python3 FileApp.py -c martin 192.168.1.156 4000 2100 3100 — 8...
client_3 — python3 FileApp.py -c eashan 192.168.1.156 4000 1000 7100 — ...

b'ACK'
ACK received by server
b'ACK'
ACK received by server
No ACK from client joey
No ACK from client joey
b'ACK'
ACK received by server
[('7.pdf', 'sam', '192.168.1.156', 6100)]
new file offereing received
ack sent
filename received. Updating table
Table Updated
b'ACK'
ACK received by server
b'ACK'
ACK received by server
b'ACK'
ACK received by server
('7.pdf', 'sam', '192.168.1.156', 6100)

>>>list
File | Name | IP | Port

('7.pdf', 'sam', '192.168.1.156', 6100)

>>>Table update recievied
>>>list
File | Name | IP | Port

('7.pdf', 'sam', '192.168.1.156', 6100)

('work.pdf', 'martin', '192.168.1.156', 3100)

>>>Table update recievied
>>>request work.pdf martin
>>> Connection with client established.
>>> Downloading work.pdf.
>>> Successfully Downloaded.
>>>[]

Last login: Mon Mar 27 14:23:10 on ttys002
(base) bravo99@Shris-Air:client_2 % python3 FileApp.py -c martin 192.168.1.156 4000 2100 3100
It is
>>> Welcome, You are registered.
>>> Client Table Updated
>>>list
File | Name | IP | Port

assific ('7.pdf', 'sam', '192.168.1.156', 6100)
ination
e fold
>>>Table update received
>>>setdir .
>>> Successfully set <./> as the directory for searching offered files.
>>>offer work.pdf
tion
ts can
>>> nickname details sent
>>> Request for file recieved
>>> Sending
es. T
along
>>>Table update recievied

>>> nickname details sent
>>> Client Table Updated
g the
>>>setdir .
>>> Successfully set <./> as the directory for searching offered files.
>>>offer 7.pdf
ment
>>> nickname details sent
-s I
ve arg
: proc
rver
ent
lient
forma
2 po
liste
1
```

Ex 4. Deregistration of client 1 and table updates

The image shows two macOS Terminal windows side-by-side, illustrating a file transfer application using sockets and Python 3.

Terminal Window 1 (Left):

- Shows a client named "client_3" interacting with a server.
- Commands sent:
 - File list request: `list`
 - Offered files:
 - 7.pdf (IP: 192.168.1.156, Port: 6100)
 - work.pdf (IP: 192.168.1.156, Port: 3100)

Terminal Window 2 (Right):

 - Shows a client named "client_1" interacting with a server.
 - Commands received:
 - Nickname details sent: `>>> nickname details sent`
 - Client table update received: `>>> client table update received`
 - Offered files:
 - 7.pdf (IP: 192.168.1.156, Port: 6100)
 - work.pdf (IP: 192.168.1.156, Port: 3100)

System Status:

 - Top bar: Mon 27 Mar 3:14 PM
 - Bottom dock: Various Mac OS X icons (Calculator, Mail, Safari, etc.)