

Developed an application for Remote File System(RFS) using sockets for client and server.

The Application supports 5 commands:

1. CWD - retrieve the path of current working directory
2. LS - lists all files and folder in current working directory
3. CD <dir> - changes the directory to <dir> as specified by client
4. DWD <file> - download the <file> from server to client
5. UPD <file> - uploads the <file> from client to remote server

Commands (3-5) throws an exception if they fail due to any runtime error and gives '*status-OK*' if they execute properly.

The application is developed in three layers:

1. File Service
2. Crypto Service
3. Networking

File Service: This is the topmost layer, uses Crypto service to encrypt and decrypt the data and executes the commands using OS API's. Uses FTP protocol to transfer files and data.

Crypto Service: This layer encrypts and decrypts the data before sending and on receiving respectively. Thus, it protects the original data and makes the connection more secure.

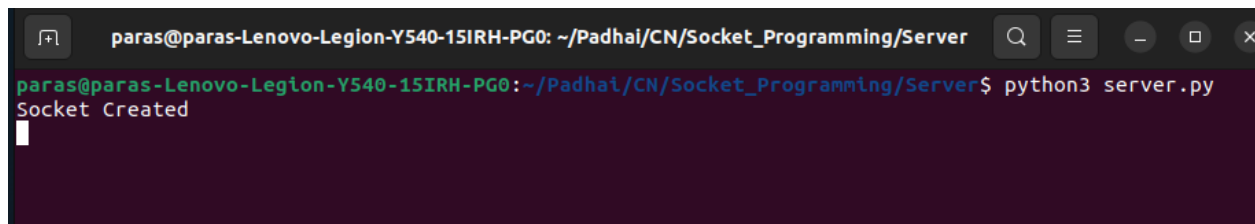
The three types of encryption available in crypto layer are:

1. Plain text - No change
2. Substitute - Alphanumeric character are substituted with a value decided by fixed offset
3. Transpose - Reverses the data word by word

Makes use of encryption protocols.

Network Layer: Network layer uses TCP connection to connect once and transfer multiple times without errors and in-sequence arrival of packets. IPv4 is used for the layer.

1. Running server file.

A terminal window with a dark purple background. The title bar shows the user 'paras' on a 'paras-Lenovo-Legion-Y540-15IRH-PG0' machine, with the current directory being '~/Padhai/CN/Socket_Programming/Server'. The terminal shows the command 'python3 server.py' being executed, followed by the output 'Socket Created' and a cursor on the next line.

```
paras@paras-Lenovo-Legion-Y540-15IRH-PG0: ~/Padhai/CN/Socket_Programming/Server
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Server$ python3 server.py
Socket Created
```

2. Running Client file and successful connection with server

```
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /
bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
█
```

3. Execution of CWD command. Have used a substitute with offset =2 as encryption hence the message is encrypted that way.

```
Status-NOK
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /
bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
CWD
CWD
CWD
b'/'jqog/rctcu/Rcfjck/EP/Uqemgv_Rtqitcookpi/Ugtxgt'
/home/paras/Padhai/CN/Socket_Programming/Server
```

4. Execution of LS command

```
/home/paras/Padhai/CN/Socket_Programming/Server
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
LS
LS
LS
b"['Etarvq.ra', 'jgnnq.vzv', 'dag.vzv', 'ugtxgt.ra', '__racejg_']"
['Crypto.py', 'hello.txt', 'bye.txt', 'server.py', '__pycache__']
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ █
```

5. Execution of CD <path> command

```
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
CD ../
CD ../
CD ../
b'/'jqog/rctcu/Rcfjck/EP/Uqemgv_Rtqitcookpi'
/home/paras/Padhai/CN/Socket_Programming
status-OK
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ █
```

Ln 62, Col 25 Spaces: 4

6. File structure before execution of DWD command. Note: hello.txt is not available in Client directory.

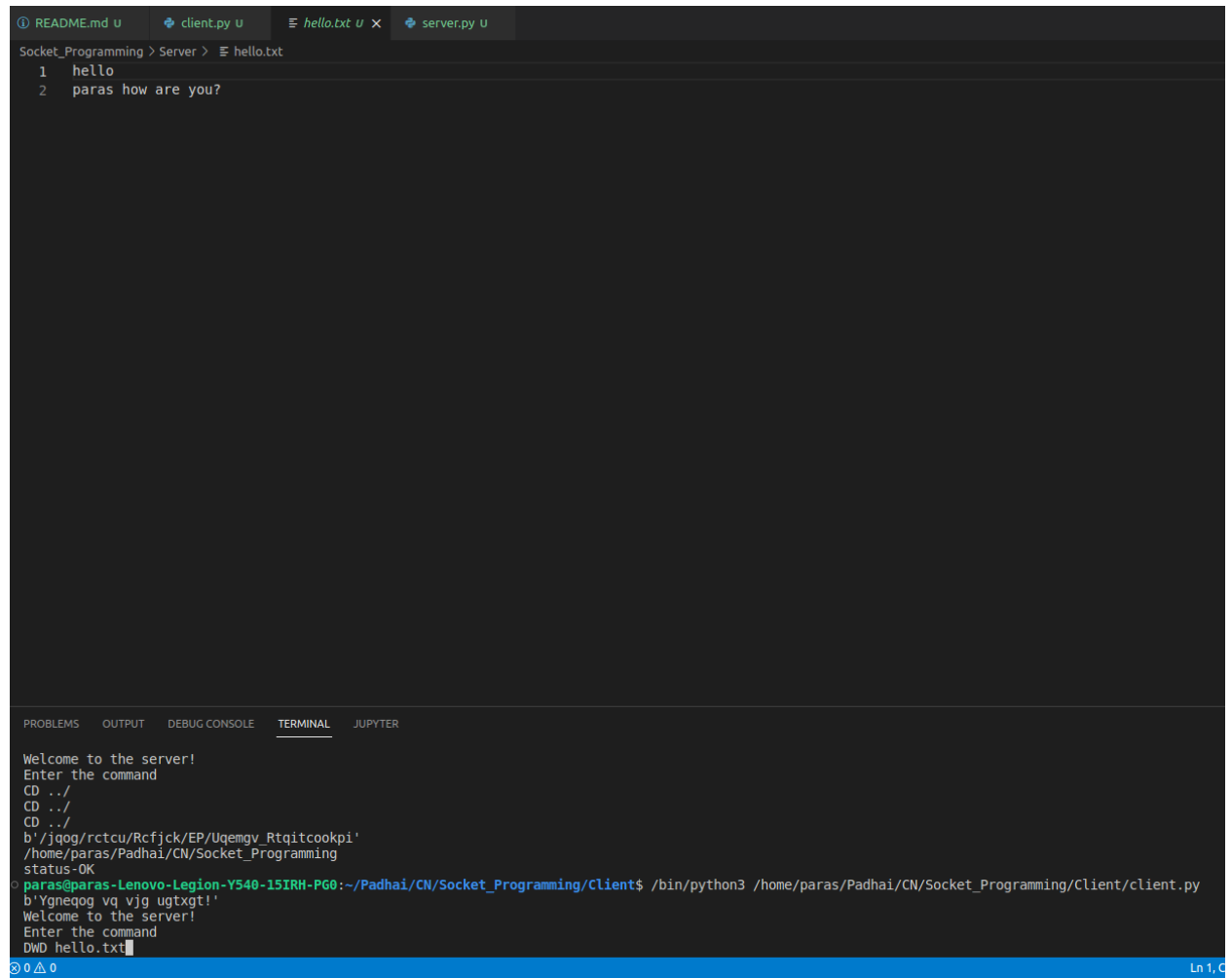
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'Socket_Programming'. The structure is as follows:

- Socket_Programming
 - Client
 - __pycache__
 - client.py
 - Crypto.py
 - test1.txt
 - Server
 - __pycache__
 - bye.txt
 - Crypto.py
 - hello.txt
 - server.py
 - README.md

On the right, the Editor pane shows the content of 'README.md'. The visible text includes:

```
try:
    fi
    #
    fi
    #
    #
    #
    wi
```

7. Execution of DWD hello.txt



The screenshot shows a code editor with four tabs: README.md, client.py, hello.txt, and server.py. The 'hello.txt' tab is active, displaying the following content:

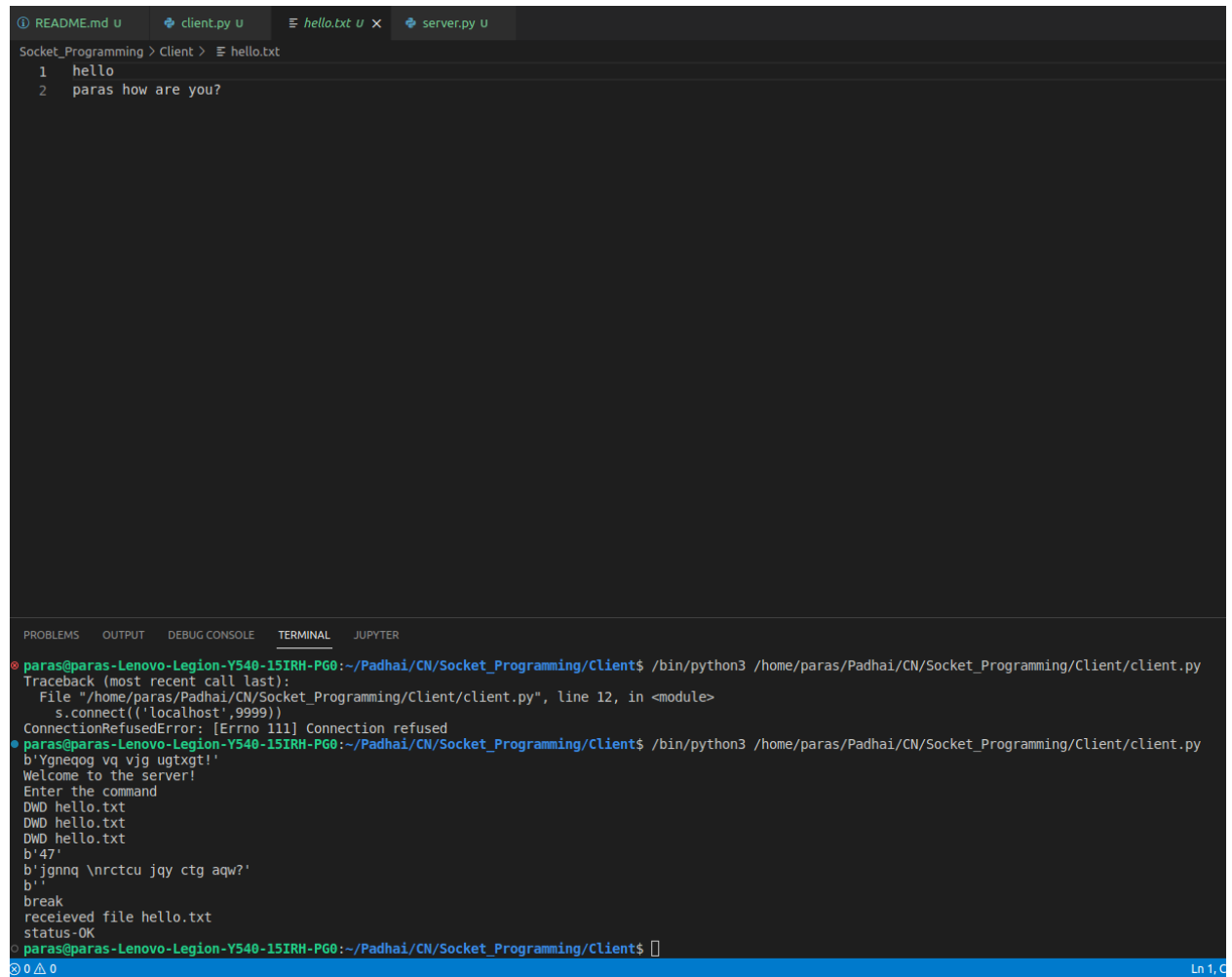
```
1 hello
2 paras how are you?
```

Below the editor is a terminal window with the following output:

```
Welcome to the server!
Enter the command
CD ../
CD ../
CD ../
b'jqog/rctcu/Rcfjck/EP/Uqemv_Rtqitcookpi'
/home/paras/Padhai/CN/Socket_Programming
status-OK
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
DWD hello.txt
```

The terminal window also shows a status bar at the bottom with the text "Ln 1, C" and a small icon.

8. We can see that hello.txt file is downloaded in Client directory



The screenshot shows a Jupyter Notebook interface with a dark theme. At the top, there is a file explorer bar with tabs for 'README.md', 'client.py', 'hello.txt', and 'server.py'. The 'hello.txt' tab is selected, showing its contents:

```
1 hello
2 paras how are you?
```

Below the file explorer is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'JUPYTER'. The 'TERMINAL' tab is active, displaying the following output:

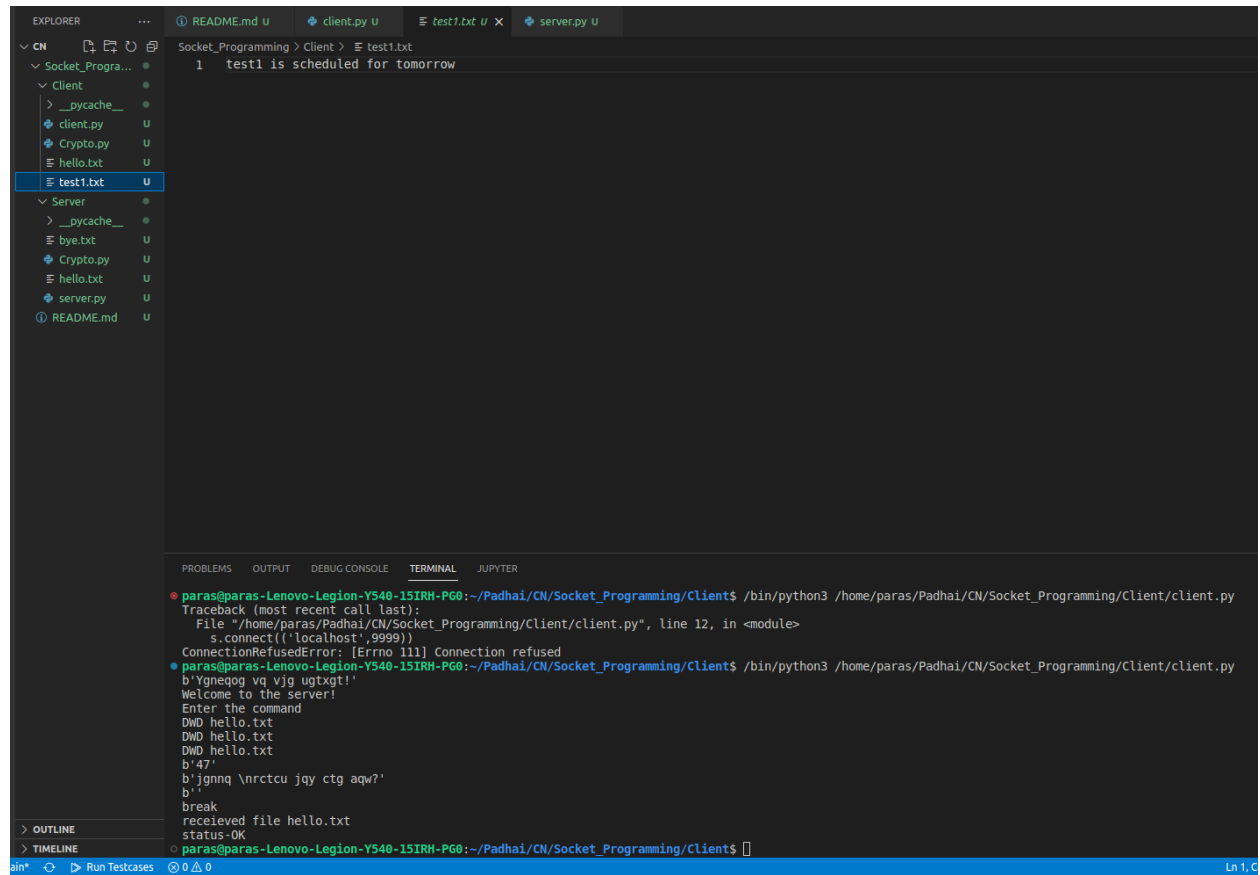
```
paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
Traceback (most recent call last):
  File "/home/paras/Padhai/CN/Socket_Programming/Client/client.py", line 12, in <module>
    s.connect(('localhost', 9999))
ConnectionRefusedError: [Errno 111] Connection refused

paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxt!'
Welcome to the server!
Enter the command
DWD hello.txt
DWD hello.txt
DWD hello.txt
b'47'
b'jgnnq \nrctcu jqy ctg aqw?'
b''
break
receieved file hello.txt
status-OK

paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$
```

The terminal output shows a successful connection to the server, followed by a series of commands and responses. The final status is 'status-OK'.

9. File tree before execution of UPD test1.txt. Note: test1.txt is not present inside the server directory



The screenshot shows a VS Code editor interface. On the left, the Explorer panel displays a file tree for a project named 'Socket_Programming'. The tree structure is as follows:

- Socket_Programming
 - Client
 - __pycache__
 - client.py
 - Crypto.py
 - hello.txt
 - test1.txt
 - Server
 - __pycache__
 - bye.txt
 - Crypto.py
 - hello.txt
 - server.py
 - README.md

The 'test1.txt' file under the 'Client' directory is selected. The main editor area shows the content of 'test1.txt', which is:

```
1 test1 is scheduled for tomorrow
```

At the bottom, the Terminal panel is active, showing the output of running a Python script. The first command results in a 'ConnectionRefusedError: [Errno 111] Connection refused'. The second command results in a successful connection to the server, where the client sends a message and receives a response.

```
paras@paras-Lenovo-Legion-Y540-15IRH-P60:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
Traceback (most recent call last):
  File "/home/paras/Padhai/CN/Socket_Programming/Client/client.py", line 12, in <module>
    s.connect(("localhost", 9999))
ConnectionRefusedError: [Errno 111] Connection refused
paras@paras-Lenovo-Legion-Y540-15IRH-P60:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
Welcome to the server!
Enter the command
DWD hello.txt
DWD hello.txt
DWD hello.txt
b'47'
b'jgnnq \nrctcu jay ctg aqw?'
b'
break
received file hello.txt
status-OK
paras@paras-Lenovo-Legion-Y540-15IRH-P60:~/Padhai/CN/Socket_Programming/Client$
```

10. Result of upload command. We can see the contents of test1.txt uploaded to server directory

```
EXPLORER
Socket_Progra...
  Client
    > __pycache__
    client.py
    Crypto.py
    hello.txt
    test1.txt
  Server
    > __pycache__
    bye.txt
    Crypto.py
    hello.txt
    server.py
    test1.txt
  README.md

Socket_Programming > Server > test1.txt
1 test1 is scheduled for tomorrow

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
b'47'
b'jgnnq \nrctcu jgy ctg aqw?'
b''
break
received file hello.txt
status-OK
● paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$ /bin/python3 /home/paras/Padhai/CN/Socket_Programming/Client/client.py
b'Ygneqog vq vjg ugtxgt!'
Welcome to the server!
Enter the command
UPD test1.txt
UPD test1.txt
UPD test1.txt
31
test1 is scheduled for tomorrow
break
file uploaded test1.txt
status-OK
○ paras@paras-Lenovo-Legion-Y540-15IRH-PG0:~/Padhai/CN/Socket_Programming/Client$
```