

NETWORKS 1 - CS3530

ASSIGNMENT - 1

SAI HARSHA KOTTAPALLI

CS17BTECH11036

Question 1

I have implemented the following features:

- a. Checking if a file exists on server
- b. sending file from client to server
- c. receiving file from server to client
- d. Encrypting data which is sent via server and decrypting data which client receives.

This is scalable where multiple clients can have common storage on server for common files, hence storage requirement is considerably reduced.

For file transfer, checking is self-explanatory.

For sending, i read 1024 bits of data every iteration and send it to the server.

For receiving, i create the necessary file and write the contents down to it.

Encrypting data adds security to our Server client model. Anyone who is listening to the data being sent would only see garbage data just from viewing them.

I have used caesar cipher for encryption and corresponding decryption.

I have used simple keywords like “ok” and “no” for checking for valid responses in the Server client Model.

Usage:

```
$ gcc Assgn1-1Client.c -o client
```

```
$ gcc Assgn1-1Server.c -o server
```

```
# Run server
```

```
$ sudo ./server 876
```

```
$ ./client 127.0.0.1 875 check bla
```

```
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1 $ ./client 127.0.0.1 875 check bla
Received: ok
file exists!!%
```

```
$ ./client 127.0.0.1 875 send bla
```

```
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1/test $ ./client 127.0.0.1 875 receive bla
Received: ok
Received: ok
```

```
$ ./client 127.0.0.1 875 receive bla
```

```
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1/test $ ./client 127.0.0.1 875 send bla
Received: ok
Received: ok
```

References:

<https://stackoverflow.com/questions/230062/whats-the-best-way-to-check-if-a-file-exists-in-c>

Question 2 - NORMAL MODE

Server:

We create a monitor each for the clients.

Based on the length of data received we can decide if client is disconnected(length is 0) or to forward the message to the other client.

Client:

I have used pthreads to run two parallel instances of receiver and sender respectively.

Here, the sender is responsible for taking input from user and writing it to socket's file descriptor. fflush() is necessary to flush the output buffer.

scanf() acts as a blocker in this function

Here, the receiver is responsible for reading from the socket's file descriptor.

fflush() is necessary to flush the output buffer.

Usage:

```
$ gcc Assgn1-2Client.c -o client -lpthread
```

```
$ gcc Assgn1-2Server.c -o server -lpthread
```

```
$ ./client 127.0.0.1 876
```

```
$ sudo ./server 876
```

```
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1$ ./client 127.0.0.1 876
# : Hello!
# :
From Client : Hey!
# : 
```

```
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1$ ./client 127.0.0.1 876
# :
From Client : Hello!
# : Hey!
# : 
```

Question 3

The programs are enhancements to the given demo code.

Server:

The `getpeername()` call returns the name of the peer connected to socket descriptor `socket`. `namelen`

We make two calls for `inet_pton()`. One to determine if client is of ipv4 or ipv6.

Client:

`getaddrinfo()` is responsible for getting appropriate ipaddr and socket type/details.

Usage:

```
$ gcc Assgn1-3Client.c -o client
```

```
$ gcc Assgn1-3Server.c -o server
```

```
$ sudo ./server 875
```

```
$ ./client ip6-localhost 875 hi
```

```
$ ./client localhost 875 hi
```

```
X archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1 ./client localhost 875 hi
Received: hi
archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1 ./client ip6-localhost 875 hi
Error finding Host --> Temporary failure in name resolution
X archelaus@archelaus ~/Documents/iith cse/5th Sem/networks-1/assgn-1 ./client ip6-localhost 875 hi
Received: hi
```

References:

<http://man7.org/linux/man-pages/man3/getaddrinfo.3.html>

https://www.mkssoftware.com/docs/man3/gai_strerror.3.asp

http://osr600doc.xinuos.com/en/SDK_netapi/sockC.TheIPv6sockaddrstructure.html

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.bpxbd00/getpeer.htm