

Problem Statement

- Implement a file transfer protocol using TCP socket.
- Implement a DNS server using UDP socket.

Design description

Purpose of the program: The program tries to create an environment to transfer files using File Transfer Protocol (FTP) over TCP socket and to create an environment to create a DNS Server that processes a client's request for an ip or domain name.

Structural design:

- **FTP**
 - The server is started and asked for a port number to bind to.
 - The client is started and following things are asked from the user
 - The input file to be requested
 - The output file to write to
 - The server's ip address
 - The server' listening port number
 - The request was sent to the server via TCP socket.
 - The server sends an acknowledgement of receiving the message. If the requested file does not exist, appropriate notification is sent by the server.
 - The server sends the requested file and the file is received by the client.
- **DNS**
 - The server is started and asked for a port number to bind to.
 - The client is started and following things are asked from the user
 - The domain name for which ip is needed or the ip address for which domain name is needed.
 - The server's IP address.
 - The server's listening port.
 - The request was sent to the server via UDP socket.
 - The server processes the query and sends the required information. If the domain name or ip address is not present, error message is sent to the client.

Input and output format:

- **FTP**
 - The input file is named "input.txt".
 - The output file is names "output.txt"
- **DNS**
 - Client asks the user to enter its request ip or domain name.

Code Snippet

FTP

Server:

```
import socket
import os

port = int(input("Enter port number: "))

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind(('', port))
sock.listen(5)
c, addr = sock.accept()

# Receiving file request from the server
filename = c.recv(1024).decode()
fileValid = False

# Sending notification to the client that the file is present
if os.path.isfile(filename):
    c.send("Done".encode())
    fileValid = True

# Requesting for some other filename to send
while not fileValid:
    c.send("Error".encode())
    filename = c.recv(1024)
    if os.path.isfile(filename):
        c.send("Done".encode())
        fileValid = True

# Sending the requested file
with open(filename, 'rb') as f:
    print('Sending file...')
    lines = f.read(1024)
    while lines:
        c.send(lines)
        lines = f.read(1024)

# Closing the socket
print('File transfer complete')
c.shutdown(socket.SHUT_WR)
```

Client:

```
import socket

# Receiving different values required
filename = str(input('Enter name of file to receive: '))
filename2 = str(input('Enter name of file in which received data should be put: '))
ipaddr = input('Enter ip of server: ')
port = int(input('Enter listening port of the server: '))

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.connect((ipaddr, port))
ack = False

# Requesting file from the server
while not ack:
    sock.send(filename.encode())
    if sock.recv(1024).decode() == "Done": ack = True
    else:
        print('No file named ' + filename + 'present')
        filename = str(input('Enter name of file to receive: '))

# Receiving file from the server
with open(filename2, 'wb') as f:
    print('Receiving...')
    while(True):
        lines=sock.recv(1024)
        if not lines: break
        f.write(lines)

# Closing the socket
print('File received successfully.')
sock.shutdown(socket.SHUT_WR)
```

DNS**Server:**

```
import socket

# Define the mapping
dntoip={'www.xyzzyx.com':'123.90.0.1',
        'www.twh.com':'123.91.23.1',
        'www.hpx.com':'123.98.56.1'}

iptodn={'123.90.0.1':'www.xyzzyx.com',
        '123.91.23.1':'www.twh.com',
        '123.98.56.1':'www.hpx.com'}
```

```
# Accept ip address
port=int(input("Enter port of server machine: "))

sock=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind(('', port))

# Server must always run
print('Server Running')
while(True):

    iporname, addr=sock.recvfrom(1024)
    iporname=iporname.decode()

    if(iporname in dntoip): # If the dn is given return ip
        dataToSend='Required IP is: '+dntoip[iporname]
    elif(iporname in iptodn):
        dataToSend='Required DN is: '+iptodn[iporname]
    else:
        dataToSend='Invalid request'

    sock.sendto(dataToSend.encode(), 0, addr)
    print(dataToSend)
sock.close()
```

Client:

```
import socket

# Accept ip address
name=str(input("Enter domain name or ip: "))
ipaddr=str(input("Enter ip of dns server: "))
port=int(input("Enter port of server: "))

sock=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.connect((ipaddr, port))

# Send the domain name or ip
sock.sendto(name.encode(), 0, (ipaddr, port))

# Wait for the server to send
iporname, addr=sock.recvfrom(1024)
iporname=iporname.decode()
print(iporname)

sock.close()
```

Output

FTP

```
output.txt X
home > lp2917 > Documents > JU > 6th Sem > Computer Networks > Lab > ass7 > output.txt
1 Hi. My name is Priyank Lohariwal. 001710501055
2

input.txt X
home > lp2917 > Documents > JU > 6th Sem > Computer Networks > Lab > ass7 > input.txt
1 Hi. My name is Priyank Lohariwal. 001710501055
2
```

```
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 ftp_client.py
Enter name of file to receive: error_file
Enter name of file in which received data should be put: output.txt
Enter ip of server: localhost
Enter listening port of the server: 8000
No file named error_file present
Enter name of file to receive: input.txt
Receiving...
File received successfully.
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$
```

```
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 ftp_server.py
Enter port number: 8000
Sending file...
File transfer complete
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$
```

DNS

```
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 dns_server.py
Enter port of server machine: 8000
Server Running
Required IP is: 123.90.0.1
Invalid request
Required IP is: 123.98.56.1
```

```
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 dns_client.py
Enter domain name or ip: www.xzyzyx.com
Enter ip of dns server: localhost
Enter port of server: 8000
Required IP is: 123.90.0.1
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 dns_client.py
Enter domain name or ip: www.hpz.com
Enter ip of dns server: localhost
Enter port of server: 8000
Invalid request
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$ python3 dns_client.py
Enter domain name or ip: www.hpx.com
Enter ip of dns server: localhost
Enter port of server: 8000
Required IP is: 123.98.56.1
lp2917@devil:~/Documents/JU/6th Sem/Computer Networks/Lab/ass7$
```

Results and Analysis

File Transfer Protocol

The file was successfully transferred. The TCP protocol is found appropriate to be used for the implementation of a file transfer application. This is because TCP allows the sequential transfer of messages. Thus, the order of bits in a file is not lost or altered in any means, which is extremely important for transferring files.

Domain Name System

The requests were successfully answered. For invalid request, an error message was displayed. The UDP protocol is found appropriate for implementing a DNS server. This is because the UDP protocol follows the "single request, single reply" principle. Thus, a client cannot make a query before the previous query is answered by the server, which is the criteria for the proper working of a DNS server. It also ensures that the query of one client is not answered to another.

Overall, the implementation of the assignment is more or less correct. However, provisions can be made for a full duplex transfer.

Comments

The assignment was easy from a student's point of view. This task helped understand the concepts of different sockets present out there and their applications.
