14. Concurrent File Server

Santhisenan A

March 11, 2018

1 Aim

Develop concurrent file server which will provide the file requested by client if it exists. If not server sends appropriate message to the client. Server should also send its process ID (PID) to clients for display along with file or the message.

2 Concurrent File Server

In computing, a file server (or fileserver) is a computer attached to a network that provides a location for shared disk access, i.e. shared storage of computer files (such as text, image, sound, video) that can be accessed by the workstations that are able to reach the computer that shares the access through a computer network. The term server highlights the role of the machine in the clientserver scheme, where the clients are the workstations using the storage. It is common that a file server does not perform computational tasks, and does not run programs on behalf of its clients. It is designed primarily to enable the storage and retrieval of data while the computation is carried out by the workstations.

A server can be iterative, i.e. it iterates through each client and serves one request at a time. Alternatively, a server can handle multiple clients at the same time in parallel, and this type of a server is called a concurrent server.

3 Code

3.1 server.py

```
import socket
import threading
import os

def recvFile(name, sock):
    filename = sock.recv(1024)
    if(os.path.isfile(filename)):
        sock.send("EXISTS" + str(os.path.getsize(filename)))
        userResponse = sock.recv(1024)
        if(userResponse[:2] == "OK"):
            with open(filename, "rb") as f:
                bytesToSend = f.read(1024)
                sock.send(bytesToSend)
                while(bytesToSend != ""):
                    bytesToSend = f.read(1024)
                    sock.send(bytesToSend)
    else:
        sock.send("ERR")
    sock.close()

HOST = '127.0.0.1'
PORT = 5000

s = socket.socket()
s.bind((HOST,PORT))

s.listen(10)

print "Server running..."
while True:
    conn, addr = s.accept()
    print "Client "+str(addr)+" connected"
    t = threading.Thread(target=recvFile, args=("retrFile", conn))
    t.start()

s.close()
```

3.2 client.py

```
    import socket
```

```
HOST = "127.0.0.1"
PORT = 5000

s = socket.socket()
s.connect((HOST, PORT))

filename = raw_input("Enter the filename: ")
if(filename != 'q'):
    s.send(filename)
    data = s.recv(1024)
    if(data[:6] == "EXISTS"):
        filesize = long(data[6:])
        message = raw_input("File is present in the server. Do you want to downl
        if(message == 'y'):
            s.send("OK")
            f = open("recv_" + filename, "wb")
            data = s.recv(1024)
            totalReceived = len(data)
            f.write(data)
            while totalReceived < filesize:
                data = s.recv(1024)
                totalReceived += len(data)
                f.write(data)
            print "Download complete"
    else:
        print "File is not present in the server"
s.close()
```

4 Output