I have neither given nor received unauthorized assistance on this work.

Simple File Upload and Download Service

Project by – Jayvardhan Chinchwade-10014264, Pramod Hanagandi-1001398156

Abstract

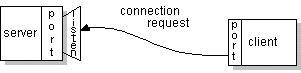
Networks are the backbone of Computer systems which help us in all the types of communication. In this project we are implementing a simple file transfer application using Java programming language and the concept of Sockets. Sockets are one of the most basic concepts of network programming which provide a medium of communication between a client and a server. This application also has the features to delete and rename a file which has been uploaded to the server.

Concept of Sockets

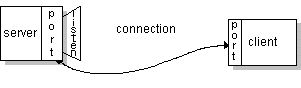
A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

In simple terms we can say that sockets provide a medium which is used to communicate between two systems. There are two components of sockets viz. a client and a server. A server initiates the socket and waits for a client to start a communication. When a client connects to the socket the data is transferred. Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

On the client-side: The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.



If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.



On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server. The client and server can now communicate by writing to or reading from their sockets. [1]

Implementation

In our project we have implemented the same abovementioned concept of sockets to design and develop an application which carries the task of Uploading, Downloading, Deleting and Renaming a file on the Server based on the Client Server model. This project is based on the message oriented communication. In the server side, we have initiated a ServerSocket object denoting the port number and then invoke the accept() method of the ServerSocket class which listens to the socket for incoming communication requests.

Now, in the Client code we have initiated the Socket object specifying the server name and port number which it wants to connect to. After establishing a connection between a Client and the Server, we have used InputStream and OutputStream to send and receive data from client to server and also from server to client.

References:

[1] Oracle.com