C.File Transfer

AIM:

To write a java program for file transfer using TCP Sockets.

Algorithm

Server

Step1: Import java packages and create class file server.

Step2: Create a new server socket and bind it to the port.

Step3: Accept the client connection

Step4: Get the file name and stored into the BufferedReader.

Step5: Create a new object class file and realine.

Step6: If file is exists then FileReader read the content until EOF is reached.

Step7: Stop the program.

Client

Step1: Import java packages and create class file server.

Step2: Create a new server socket and bind it to the port.

Step3: Now connection is established.

Step4: The object of a BufferReader class is used for storing data content which has been retrieved from socket object.

Step5 The connection is closed.

Step6: Stop the program.

Program

File Server :

```
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
```

```
public class FileServer
{
   public static void main(String[] args) throws Exception
        {
             //Initialize Sockets
             ServerSocket ssock = new ServerSocket(5000);
             Socket socket = ssock.accept();
            //The InetAddress specification
                InetAddress IA = InetAddress.getByName("localhost");

        //Specify the file
             File file = new File("e:\\Bookmarks.html");
             FileInputStream fis = new FileInputStream(file);
                BufferedInputStream bis = new BufferedInputStream(fis);
                //Get socket's output stream
                OutputStream os = socket.getOutputStream();
               //Read File Contents into contents array
                byte[] contents;
               long fileLength = file.length();
                long current = 0;
                long start = System.nanoTime();
                while(current!=fileLength){
                          int size = 10000;
                          if(fileLength - current >= size)
                             current += size;
                          else{
                                    size = (int)(fileLength - current);
                                    current = fileLength;
                           }
                    contents = new byte[size];
                bis.read(contents, 0, size);
                 os.write(contents);
                System.out.print("Sending file ... "+(current*100)/fileLength+"% complete!");
                }
            os.flush();
         //File transfer done. Close the socket connection!
        socket.close();
        ssock.close();
        System.out.println("File sent succesfully!");
        } }
```

File Client

```
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.InetAddress;
import java.net.Socket;

public class FileClient {
   public static void main(String[] args) throws Exception{
      //Initialize socket
            Socket socket = new Socket(InetAddress.getByName("localhost"), 5000);
               byte[] contents = new byte[10000];
      //Initialize the FileOutputStream to the output file's full path.
            FileOutputStream fos = new FileOutputStream("e:\\Bookmarks1.html");
              BufferedOutputStream bos = new BufferedOutputStream(fos);
              InputStream is = socket.getInputStream();
       //No of bytes read in one read() call
             int bytesRead = 0;
            while((bytesRead=is.read(contents))!=-1)
                  bos.write(contents, 0, bytesRead);
            bos.flush();
            socket.close();
            System.out.println("File saved successfully!");
   }
}
```

Output

server

```
E:\nwlab>java FileServer
Sending file ... 9% complete!
Sending file ... 19% complete!
Sending file ... 28% complete!
Sending file ... 38% complete!
Sending file ... 47% complete!
Sending file ... 57% complete!
Sending file ... 66% complete!
Sending file ... 76% complete!
Sending file ... 86% complete!
Sending file ... 95% complete!
```

Sending file ... 100% complete!
File sent successfully!

E:\nwlab>**client**
E:\nwlab>java FileClient
File saved successfully!

E:\nwlab>

.

Viva questions:

1. What are the types of protocol?
2. Define socket.
3. What information is needed to create a TCP Socket?
4. What are the two important TCP Socket classes?
5. What is the difference between the File and Random Access File classes?
6. What are some advantages and disadvantages of Java Sockets?
7. How to build FileInputStream object with byte array as a parameter
8. How to read file in byte array with FileInputStream

RESULT

Thus the java program file transfer application using TCP Sockets was executed