

Exp No: 7

PROGRAM FOR FILE TRANSFER

Date:

AIM

ALGORITHM

220071601028

CODE

Server.java

```
import java.io.*;
import java.net.*;

public class Server {

    private static DataInputStream dataInputStream=null;
    private static DataOutputStream dataOutputStream=null;

    public static void main(String[] args) throws IllegalArgumentException {

        if (args.length != 1) {
            throw new IllegalArgumentException("One argument is required. Path
to store the recieving file (exclude format).");
        }

        try {

            ServerSocket serverSocket = new ServerSocket(900);

            Socket socket = serverSocket.accept();
            System.out.println("Connection Established");

            dataInputStream= new DataInputStream(socket.getInputStream());
            dataOutputStream = new DataOutputStream(socket.getOutputStream());

            receiveFile(args[0]);

            dataInputStream.close();
            dataOutputStream.close();
            socket.close();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void receiveFile(String path) throws Exception {

        int bytesRead;

        String file_format = dataInputStream.readUTF();

        FileOutputStream fileOutputStream = new FileOutputStream(path + "." + file_format);
        long sizeLeft = dataInputStream.readLong();

        byte[] buffer = new byte[4 * 1024];
```

```

        do {
            bytesRead = dataInputStream.read(buffer, 0, (int) Math.min(buffer.length,
sizeLeft));
            fileOutputStream.write(buffer);
            sizeLeft -= bytesRead; // subtracting bytes read from size
        } while (sizeLeft > 0 && bytesRead != -1);

        System.out.println("File is Received and Stored at " + path + "." + file_format);
        fileOutputStream.close();
    }
}

```

Client.java

```

import java.io.*;
import java.net.Socket;

public class Client {

    private static DataOutputStream dataOutputStream = null;
    private static DataInputStream dataInputStream = null;

    public static void main(String[] args) throws IllegalArgumentException {

        if (args.length != 1) {
            throw new IllegalArgumentException("One argument is required. Path
of the file to be sent.");
        }

        try {

            Socket socket = new Socket("localhost",900);

            dataInputStream = new DataInputStream(socket.getInputStream());
            dataOutputStream = new DataOutputStream(socket.getOutputStream());

            System.out.println("Sending the File to the Server");
            sendFile(args[0]);

            dataInputStream.close();
            dataOutputStream.close();

        } catch (Exception e) {
            e.printStackTrace();
        }

    }

    private static void sendFile(String path) throws Exception {

        int bytesRead;
        File file = new File(path);
    }
}

```

```

String[] splittedStr = path.split("\\.", 0);
dataOutputStream.writeUTF(splittedStr[splittedStr.length - 1]);

FileInputStream fileInputStream = new FileInputStream(file);
int sizeLeft = (int) file.length();
dataOutputStream.writeLong(sizeLeft);

byte[] buffer = new byte[4 * 1024];

do {
    bytesRead = fileInputStream.read(buffer, 0, (int) Math.min(buffer.length, sizeLeft));
    dataOutputStream.write(buffer, 0, bytesRead);
    dataOutputStream.flush();
    sizeLeft -= bytesRead; // subtracting bytes read from size
} while (sizeLeft > 0 && bytesRead != -1);




fileInputStream.close();
dataOutputStream.close();
}
}

```



OUTPUT

Before:

Client:

 Client.class	04-04-2024 00:06	CLASS File	2 KB
 Client.java	04-04-2024 00:15	Java Source File	2 KB
 img.jpeg	03-04-2024 23:10	JPEG File	352 KB

Server:

 Server.class	04-04-2024 00:06	CLASS File	3 KB
 Server.java	04-04-2024 00:05	Java Source File	2 KB

Executing the program:

Client:

```

Computer-Networks\7\Client>java Client ./img.jpeg
Sending the File to the Server

```

Server:




```

Computer-Networks\7\Server>java Server ./img-recieved
Connection Established
File is Received and Stored at ./img-recieved.jpeg




```

After:

Client:

 Client.class	04-04-2024 00:06	CLASS File	2 KB
 Client.java	04-04-2024 00:15	Java Source File	2 KB
 img.jpeg	03-04-2024 23:10	JPEG File	352 KB

Server:

 img-recieved.jpeg	18-04-2024 13:50	JPEG File	352 KB
 Server.class	04-04-2024 00:06	CLASS File	3 KB
 Server.java	04-04-2024 00:05	Java Source File	2 KB

RESULT