19CSE301 Computer Networks

Assignment (08-08-2021)

Submitted By,
Sandeep Rajakrishnan
CB.EN.U4CSE19650
CSE F

1. File transfer program using sockets.

- Client will send a csv file. It will be read modified by server and send back to client.(Salary should be increased by 10% in the employ csv file)
- Client will read the modified file and display.

SOLUTION AND CODE WORKFLOW

- Server has started and is waiting for the client to connect.
- Client Connects to the server.
- Server is waiting for the client to send a CSV file
- Client creates a CSV file named data.csv by entering the headers and the data
- Client reads the data.csv and stores the data into a byte array[].
- Client sends the byte array to the server and waits for the server's response.
- Server receives the byte array containing the data , sent by the client.
- Server creates its own copy of data.csv and writes the data from the byte array to the data.csv.
- Server reads its copy of data.csv line by line and writes each line into a new CSV file named temp.csv , after updating the salary.
- Now the server has its own copy of data.csv and the updated salary file named temp.csv
- Server deletes its copy of data.csv and renames the "temp.csv" to "data.csv".
- Server sends the updated copy of "data.csv" back to client.
- Client receives the updated data in the form of a byte array.
- Client creates a CSV file named updatedData.csv and writes the data from the received byte array into the updatedData.csv.
- Now the client has the "data.csv" and the "updatedData.csv".
- Client deletes the old "data.csv" and renames the "updatedData.csv" to "data.csv".
- Now client has the file "data.csv" which contains the data of employees with salaries increased by 10%.

OUTPUT PASTED AFTER THE CODE

CODE

CLIENT SIDE CODE

```
import java.io.*;
import java.net.*;
import java.nio.file.Files;

class client {

  public static void main(String[] args) {
    try {
      Socket s = new Socket("localhost", 3000); // server IP , port
      System.out.println();
      System.out.println("Connection established with server");

      DataInputStream din = new DataInputStream(s.getInputStream());
      DataOutputStream dout = new DataOutputStream(s.getOutputStream());

      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

      File file = new File("data.csv");
      FileWriter fileWritter;

      String line = "";

      // keep reading until "Over" is input
      boolean headerPresent = false;
      DataInputStream input = new DataInputStream(System.in);

      // Creating and inputting data into csv
      System.out.println();
      System.out.println("Create the CSV file : ");

      while (!line.equals("Over")) {
        try {
          fileWritter = new FileWriter(file.getName(), true);

          BufferedWriter bw = new BufferedWriter(fileWritter);

          if (headerPresent == false) {
            System.out.print("Enter the headers : ");
            headerPresent = true;
          } else System.out.print("Enter the next row of data : ");
          line = input.readLine();

          if (!line.equals("Over")) bw.write(line + "\n");
          bw.close();

        } catch (IOException i) {
```

```
          System.out.println(i);
        }
      }
      System.out.println();
      System.out.println("Sending the file to the server to update the salary by
10%");

      // Reading data from the file created above
      // storing it into byte array and sending the byte array to server

      FileInputStream fr = new FileInputStream("data.csv");
      byte b[] = new byte[20002];
      fr.read(b, 0, b.length);
      dout.write(b, 0, b.length); // writing to server
      dout.flush();
      fr.close();

      System.out.println("Waiting for server's response...");

      // Receiving the updated file from server and storing it into byte array
      // storing the data from the byte array into a new csv

      byte fromServer[] = new byte[20002];
      din.read(fromServer, 0, fromServer.length); // Reading from server

      System.out.println("Updated file received from the server");

      FileOutputStream fstr = new FileOutputStream("updatedData.csv");
      fstr.write(fromServer, 0, fromServer.length);
      fstr.close();

      // Renaming updated file and deleting old file

      File f1 = new File("updatedData.csv");
      File f2 = new File("data.csv");
      boolean delete = f2.delete();
      boolean success = f1.renameTo(f2);

      String str = new String(fromServer);
      System.out.println();
      System.out.println("Updated file received from the server contains the
following data : \n");

      System.out.println(str);

      s.close();

    } catch (Exception e) {
      System.out.println("Lost Connection");
    }
  }
```

SERVER SIDE CODE

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

class server {
  public static void main(String[] args) {
    try {

      ServerSocket ss = new ServerSocket(3000);
      System.out.println();
      System.out.println("Waiting for connection... ");

      Socket s = ss.accept();

      System.out.println("Connection established with client");

      System.out.println("Waiting for client to send the file...");

      DataInputStream din = new DataInputStream(s.getInputStream());
      DataOutputStream dout = new DataOutputStream(s.getOutputStream());

      ///  BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

      byte b[] = new byte[20002];

      din.read(b, 0, b.length);
      System.out.println("File received from the client");

      FileOutputStream fr = new FileOutputStream("data.csv");

      fr.write(b, 0, b.length);

      fr.close();
      System.out.println();
      System.out.println("File received from the client contains the following data
: \n");

      // Creating the Updated Data file

      File newFile = new File("temp.csv");
      String name = "", department = "", salary = "";
      try {
        Scanner x = new Scanner(new File("data.csv"));
        x.useDelimiter("[,\n]");

        while (x.hasNext()) {

          name = x.next();
```

```
          department = x.next();
          salary = x.next();
          System.out.println(name + "," + department + "," + salary);

          try {

            int updatedSalary = Integer.parseInt(salary);
            updatedSalary = (int)(updatedSalary + (int)(updatedSalary * 0.1));
            String textToAppend = name + "," + department + "," +
String.valueOf(updatedSalary) + "\n";
            BufferedWriter writer = new BufferedWriter(
              new FileWriter("temp.csv", true));

            writer.write(textToAppend);
            writer.close();
          } catch (Exception e) {
            String textToAppend = name + "," + department + "," + salary + "\n";
            BufferedWriter writer = new BufferedWriter(
              new FileWriter("temp.csv", true));

            writer.write(textToAppend);
            writer.close();
          }

        }

        x.close();

      } catch (Exception e) {
        System.out.println();
        System.out.println("File has been updated : Salary increased by 10%");

      }

      System.out.println("Sending updated file back to client...");

      // Sending updated file back to client
      FileInputStream fstr = new FileInputStream("temp.csv");
      byte sendToClient[] = new byte[20002];
      fstr.read(sendToClient, 0, sendToClient.length);
      dout.write(sendToClient, 0, sendToClient.length);
      dout.flush();
      fstr.close();

      System.out.println("Updated file sent back to client.\n");

      // Deleting old file and renaming temp file

      File file = new File("temp.csv");
      File file2 = new File("data.csv");
      boolean delete = file2.delete();
```

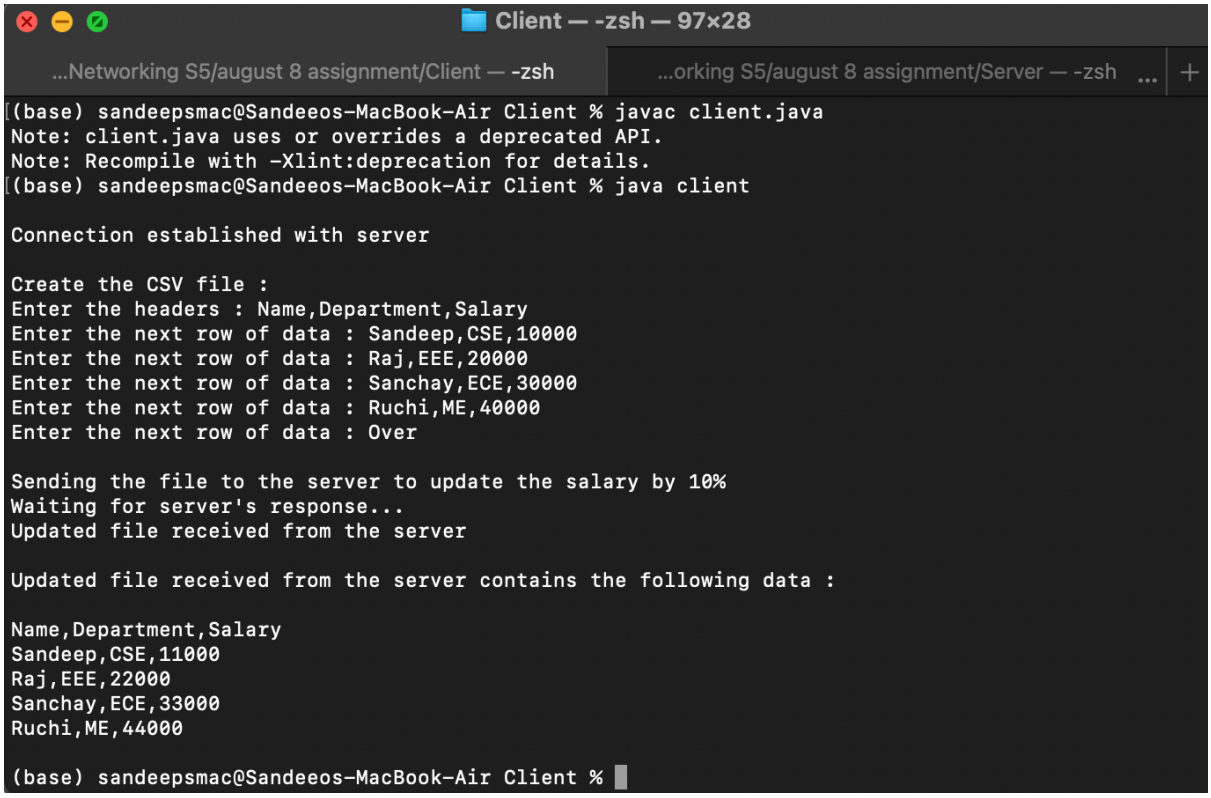
```
      boolean success = file.renameTo(file2);

      s.close();
      ss.close();

    } catch (Exception e) {
      System.out.println(e);
      System.out.println("Lost Connection");
```

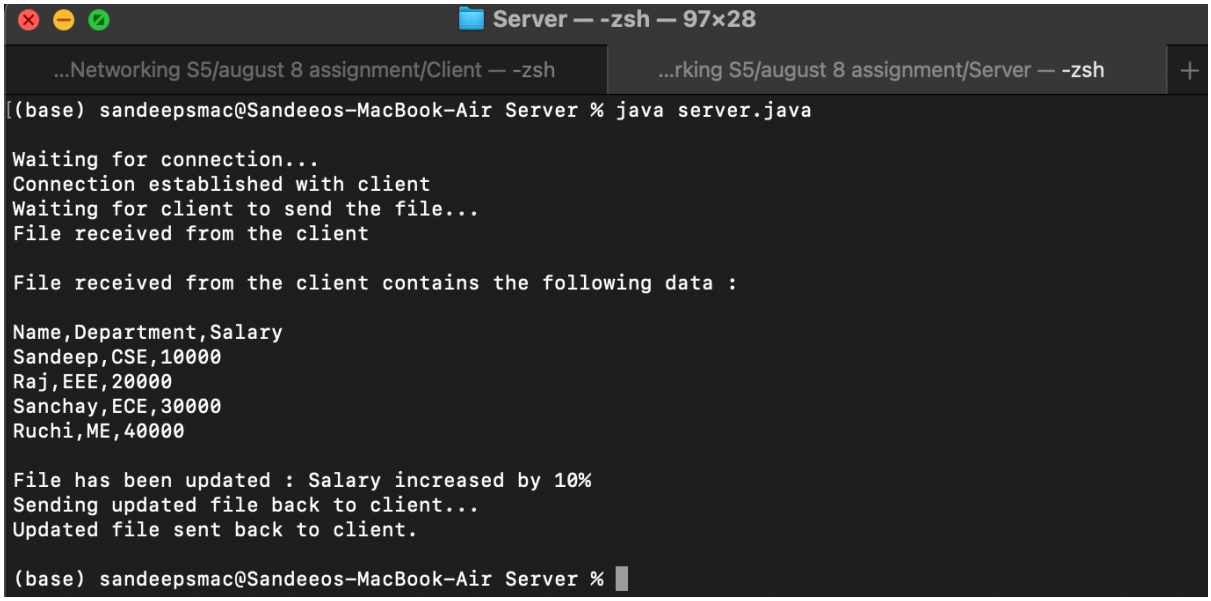
OUTPUT

CLIENT



SERVER SIDE OUTPUT IN NEXT PAGE

SERVER


Server — -zsh — 97×28
...Networking S5/august 8 assignment/Client — -zsh
...rking S5/august 8 assignment/Server — -zsh
(base) sandeepsmac@Sandeeos-MacBook-Air Server % java server.java
Waiting for connection...
Connection established with client
Waiting for client to send the file...
File received from the client
File received from the client contains the following data :
Name,Department,Salary
Sandeep,CSE,10000
Raj,EEE,20000
Sanchay,ECE,30000
Ruchi,ME,40000
File has been updated : Salary increased by 10%
Sending updated file back to client...
Updated file sent back to client.
(base) sandeepsmac@Sandeeos-MacBook-Air Server %