EXPERIMENT: 33

IMPLEMENTING THE APPLICATIONS USING TCP FILE TRANSFER IN C

Aim: To implement the applications using TCP file transfer in java.

Algorithm:

1. Start the program.

2. Declare the variables and structures required.

3. A socket is created and the connect function is executed.

4. The file is opened.

5. The data from the file is read and sent to the server.

6. The socket is closed.

7. The program is stopped.

Steps to implement a TCP file transfer:

1. Set up the TCP client-server connection:

- Create a server socket using the `socket()` function.

- Bind the server socket to a specific IP address and port using the `bind()` function.

- Listen for incoming client connections using the `listen()` function.

- Create a client socket using the `socket()` function.

- Connect the client socket to the server using the `connect()` function.

2. Server-side implementation:

- Accept the client connection using the `accept()` function on the server side.

- Open the file to be transferred in binary mode using `fopen()`.

- Read the contents of the file in chunks and send them over the TCP connection using the `send()`

function.

- Close the file using `fclose()`.

3. Client-side implementation:

- Receive the file data from the server using the `recv()` function on the client side.

- Write the received data to a file on the client side using `fwrite()`.

- Continue receiving and writing data until the entire file is received.

- Close the file using `fclose()`.

Example content:

```
Hello, this is a test file.
Sent using TCP File Transfer in C!
```

Running the server:

Server Output:

```
Server listening on port 9092...
Client connected, sending file...
File sent successfully!
```

Client output:

```
Connected to server. Receiving file...
File received successfully!
```

Result: Thus the applications using TCP file transfer in java is completed successfully