Java Networks

Sockets

Inter Connection of computers is called a network. In a network, there may be several computers, some of them receiving the services and some of them providing the services to others. The computer which receives service is called a client and the computer which provides the service is called a server. Remember a client sometimes acts as a server and a server acts as a client. There are 3 requirements to establish a network:

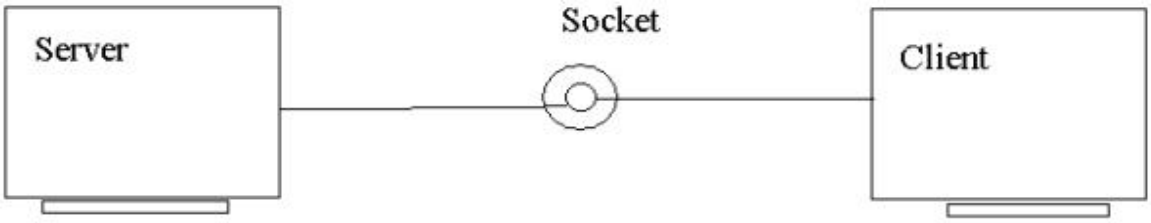
- ✧ Hardware: includes the computers, cables, modems, hubs etc.
- ✧ Software: includes programs to communicate between servers and clients.
- ✧ Protocol: set of rules and regulations that represents a way to establish connection and helps in sending and receiving data in a standard format. Popular protocol suit is TCP/IP. TCP is connection oriented and IP connection less potocols.

  e.g HTTP, FTP, SMTP, POP, UDP etc.

IP address: An IP address is a unique identification number alloted to every computer on a network or internet. IP address contains some bytes which identify the network and the actual computer inside the network.

DNS: Domain Naming Service is a service on internet that maps the IP addresses with corresponding website names.

Socket Programming: is a low level way to establish connection between server and a client with the help of a Socket



- ✧ Data will be sent or received through the Socket.
- ✧ Socket at server side is called Server Socket.
- ✧ Socket at client side is called Client Socket.
- ✧ The id number alloted to a Socket is called port number.

- ✧ When a new socket is created a new port number is alloted.
- ✧ When the service on the socket is changed we should change the port number.
- ✧ A Socket is a point of connection between a server and client.
- ✧ Server Socket is created using ServerSocket class.
- ✧ Client Socket is created using Socket class.
- ✧ ServerSocket and Socket classes are available in java.net package.

Program 1: Write a program to create a server for sending some strings to the client.

//A server that sends message to client

import java.net.*;

import java.io.*;

```
class Server1
{       public static void main(String args[]) throws IOException
        {       //Create Server side socket
                ServerSocket ss = new ServerSocket (777);
                //make this socket accept client connection
                Socket s = ss.accept ();
                System.out.println ("A connection established...");
                //attach OutputStream to socket
                OutputStream obj = s.getOutputStream ();
                //to send data to Socket
                PrintStream ps = new PrintStream (obj);
                //now send the data
                String str = "Hello Client";
                ps.println (str);
                ps.println ("Bye");
                //close connection
                s.close ();
                ss.close ();
                ps.close ();
        }
}
```

Program 2: Write a program to create a client which accepts all the strings sent by the server

```
// a client that accepts data from server
import java.util.*;
import java.io.*;
import java.net.*;
class Client1
{       public static void main(String args[]) throws IOException
        {       //Create client socket
                Socket s = new Socket ("localhost", 777);
                //attach input stream to Socket
                InputStream obj = s.getInputStream ();
                //to receive data from socket
                BufferedReader br = new BufferedReader (new InputStreamReader (obj));
                //read data coming from server
                String str;
                while ((str = br.readLine() ) != null )
                        System.out.println (str);
                //close connection

s.close ();
br.close ();
}
}
```

Applets and Java Web Start

An applet is a program that comes from server into a client and gets executed at client side and displays the result. An applet represents byte code embedded in a html page. (applet = bytecode+ html) and run with the help of Java enabled browsers such as Internet Explorer. An applet is a Java program that runs in a browser. Unlike Java applications applets do not have a main () method. To create applet we can use java.applet.Applet or javax.swing.JApplet class. All applets inherit the super class 'Applet'. An Applet class contains several methods that helps to control the execution of an applet.

Advantages:

- ✧ Applets provide dynamic nature for a webpage.
- ✧ Applets are used in developing games and animations

Creating an applet:

Let the Applet class extends Applet or JApplet class.

Overide the following methods of Applet class.

- **public void init ()**: This method is used for initializing variables, parameters to create components. This method is executed only once at the time of applet loaded into memory.
- **public void start ():** After init() method is executed, the start method is executed automatically. Start method is executed as long as applet gains focus. In this method code related to opening files and connecting to database and retrieving the data and processing the data is written.
- **public void stop ()**: This mehtod is executed when the applet loses focus. Code related to closing the files and database, stopping threads and performing clean up operations are written in this stop method.
- **public void destroy ()**: This method is exexuted only once when the applet is terminated from the memory. Executing above methods in that sequence is called applet life cycle. We can also use public void paint (Graphics g) in applets. After writing an applet, an applet is compiled in the same way as Java application but running of an applet is different. There are two ways to run an applet.
- Executing an applet within a Java compatible web browser.
- Executing an applet using 'appletviewer'. This executes the applet in a window.

Program 1: Write an applet program with a message and display the message in paint () method.

```
/*      <applet code="MyApplet.class" width = 600 height= 450>
        </applet>       */
import java.applet.Applet;
import java.awt.*;
public class MyApplet extends Applet
{       String msg="";
        public void init()
        {
                msg += "init";
        }
        public void start()
        {
                msg +=" start";
        }
        public void paint(Graphics g)
        {
                g.drawString(msg,10,100);
        }
        public void stop()
        {
                msg += " stop";
        }
        public void destroy()
        {
                msg+= " destroy";
        }
}
```