10.6 Network Communication with Streams

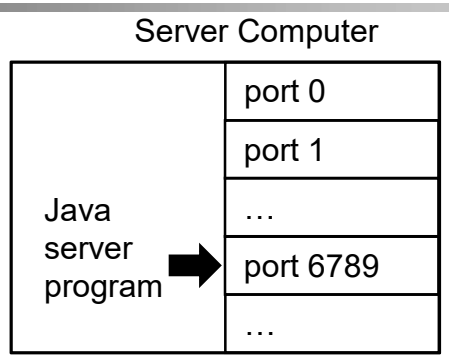
- Classes such as Scanner and PrintWriter
  - » (Scanner) Read data from a file or the Keyboard
  - » can be used with any data stream – such as communicating over a network using streams
  - » The code is nearly identical
    - – abstraction
    - – Polymorphism

- Communicate with one another over a network
  - » must speak the same language , protocol
    - – ex) TCP/IP Transmission Control Protocol and Internet Protocol
    - – Ex) UDT User Datagram Protocol
  - » TCP
    - – stream-based protocol
    - – Reliable protocol because it guarantees that data from the sender is received in the same order in which it was sent.
  - » server : the program that is waiting for a connection
  - » client : the program that initiates the connection

- Java uses sockets for network programming
  - » A socket
    - – describes one end of the connection between two programs over the network.
    - – consists of the address that identifies the remote computer and a port ranging from 0 to 65535
    - – Two applications may not bind to the same port
  - » Figure 10.7
    - – The process of communicating between a client and server

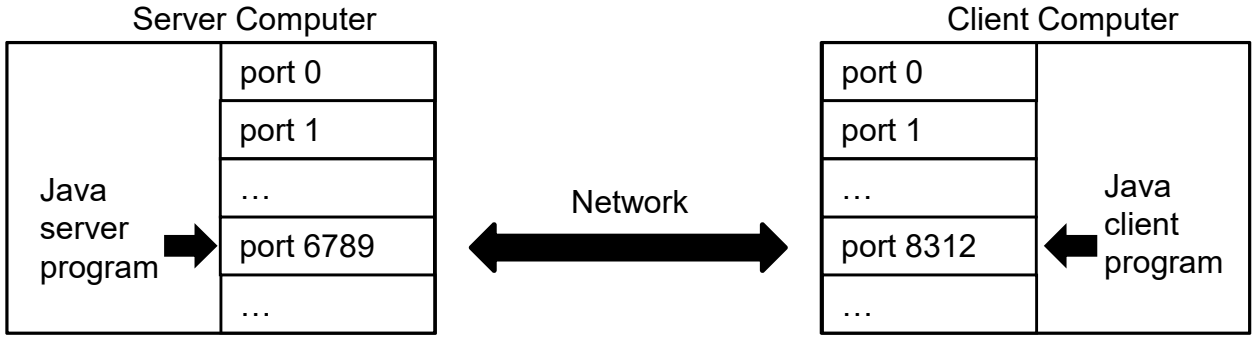
FIGURE 10.7 Client/Server Network Communication via Sockets

1. The Java server program listens and waits for a connection on port 6789. Different programs may be listening on other ports.



In Listing 10.13 the server and client are running on the same computer which is identified by the special address of "localhost".

2. The Java client program connects to the server on port 6789. It uses a local port that is assigned automatically, in this case, port 8312.



3. The Java server program can now communicate over a socket bound locally to port 6789 and remotely to the client's address at port 8312, while the client communicates over a socket bound locally to port 8312 and remotely to the server's address at port 6789.

Introduction to Sockets and Networking

- Server program
  - » Listen for a connection on a specified port; when one is made:
    - – Create a Scanner with an InputStreamReader based on the socket that the server will listen on; use this for input from a client
    - – Create a PrintWriter with the socket to send data to the client
  - » See Listing 10.12

Listing 10.12 Network Server Program

```
 import java.util.Scanner;
import java.io.InputStreamReader;
import java.io.DataOutputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.ServerSocket;

public class SocketServer
{
 public static void main(String[] args)
 {
  String s;
  Scanner inputStream = null;
  PrintWriter outputStream = null;
  ServerSocket serverSocket = null;
```

```
try
{
    // Wait for connection on port 6789
   System.out.println("Waiting for a connection.");
   serverSocket = new ServerSocket(6789);
   Socket socket = serverSocket.accept();

   // Connection made, set up streams
   inputStream = new Scanner(new InputStreamReader(socket.getInputStream()));
   outputStream = new PrintWriter(new DataOutputStream(socket.getOutputStream()));

   // Read a line from the client
   s = inputStream.nextLine();

   // Output text to the client
   outputStream.println("Well, ");
   outputStream.println( s + " is a fine programming language!“ );
   outputStream.flush();
   System.out.println("Closing connection from " + s);

   inputStream.close();
   outputStream.close();
}
catch (Exception e)
{
      // If any exception occurs, display it
      System.out.println("Error " + e);
}
}
}
```

```
public class ServerSocket
extends Object
implements Closeable
```

This class implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.

The actual work of the server socket is performed by an instance of the `SocketImpl` class. An application can change the socket factory that creates the socket implementation to configure itself to create sockets appropriate to the local firewall.

Since:

`JDK1.0`

See Also:

`SocketImpl`, `setSocketFactory(java.net.SocketImplFactory)`, `ServerSocketChannel`

Constructor Summary

Constructors

Constructor and Description
`ServerSocket``()` Creates an unbound server socket.
`ServerSocket``(int port)` Creates a server socket, bound to the specified port.
`ServerSocket``(int port, int backlog)` Creates a server socket and binds it to the specified local port number, with the specified backlog.
`ServerSocket``(int port, int backlog, InetAddress bindAddr)` Create a server with the specified port, listen backlog, and local IP address to bind to.

accept

```
public Socket accept()
              throws IOException
```

Listens for a connection to be made to this socket and accepts it. The method blocks until a connection is made.

A new Socket s is created and, if there is a security manager, the security manager's checkAccept method is called with s.getInetAddress().getHostAddress() and s.getPort() as its arguments to ensure the operation is allowed. This could result in a SecurityException.

Returns:

the new Socket

Throws:

IOException - if an I/O error occurs when waiting for a connection.

SecurityException - if a security manager exists and its checkAccept method doesn't allow the operation.

SocketTimeoutException - if a timeout was previously set with setSoTimeout and the timeout has been reached.

IllegalBlockingModeException - if this socket has an associated channel, the channel is in non-blocking mode, and there is no connection ready to be accepted

See Also:

SecurityManager.checkAccept(java.lang.String, int)

Socket Class

InputStream	getInputStream() Returns an input stream for this socket.
boolean	getKeepAlive() Tests if SO_KEEPALIVE is enabled.
InetAddress	getLocalAddress() Gets the local address to which the socket is bound.
int	getLocalPort() Returns the local port number to which this socket is bound.
SocketAddress	getLocalSocketAddress() Returns the address of the endpoint this socket is bound to.
boolean	getOOBInline() Tests if SO_OOBINLINE is enabled.
OutputStream	getOutputStream() Returns an output stream for this socket.
int	getPort() Returns the remote port number to which this socket is connected.

Introduction to Sockets and Networking

- Client program
  - » Initiate a connection to the server on a specified port
  - » Create a Scanner to read from the socket
  - » Create a PrintWriter to send to the socket
  - » Set Listing 10.13

Listing 10.13 Network Client Program

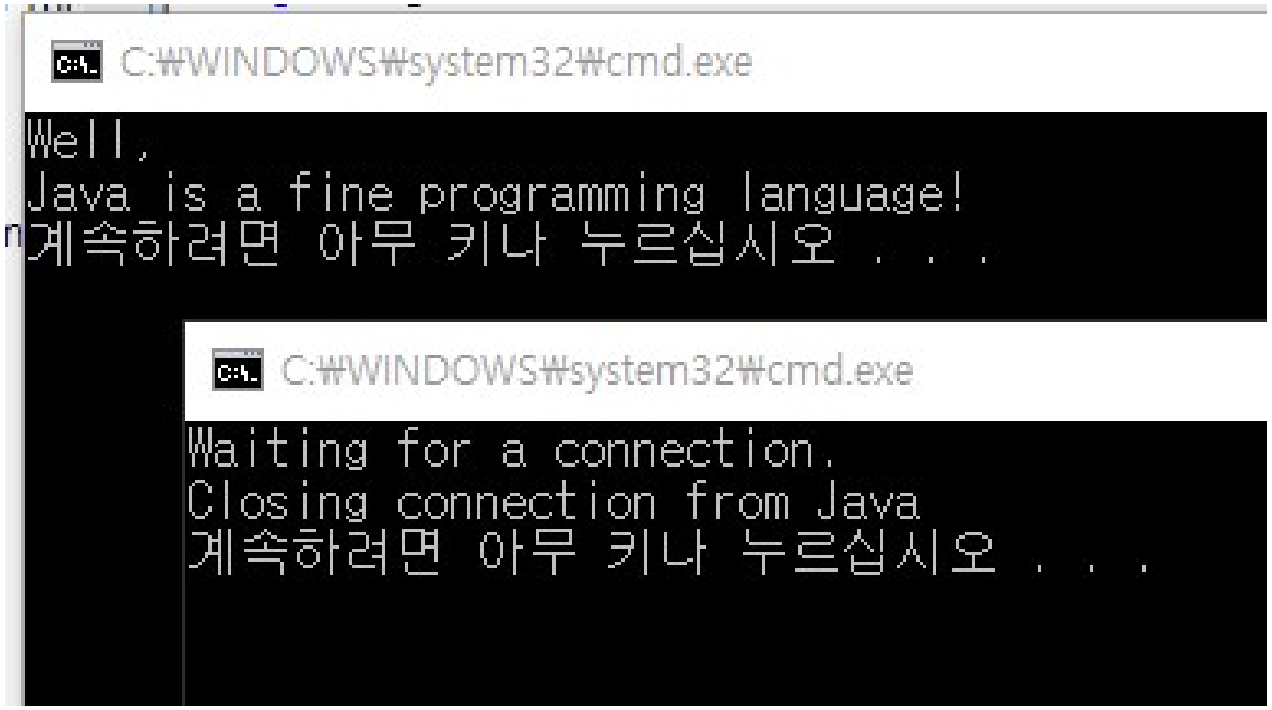
```
import java.util.Scanner;
import java.io.InputStreamReader;
import java.io.DataOutputStream;
import java.io.PrintWriter;
import java.net.Socket;

public class SocketClient
{
 public static void main(String[] args)
 {
  String s;
  Scanner inputStream = null;
  PrintWriter outputStream = null;
  try
  {
   // Connect to server on same machine, port 6789
   Socket clientSocket = new Socket("localhost",6789);
   // Set up streams to send/receive data
   inputStream = new Scanner(new
InputStreamReader(clientSocket.getInputStream()));
   outputStream = new PrintWriter(new
DataOutputStream(clientSocket.getOutputStream()));
```

```
// Send "Java" to the server
  outputStream.println("Java");
  outputStream.flush(); // Sends data to the stream

  // Read everything from the server until no more lines
  // and output it to the screen

  while (inputStream.hasNextLine())
  {
    s = inputStream.nextLine();
    System.out.println(s);
  }
  inputStream.close();
  outputStream.close();
 }
 catch (Exception e)
 {
  // If any exception occurs, display it
  System.out.println("Error " + e);
 }
}
}
```


C:₩WINDOWS₩system32₩cmd.exe
Well,
Java is a fine programming language!
계속하려면 아무 키나 누르십시오 . . .
C:₩WINDOWS₩system32₩cmd.exe
Waiting for a connection.
Closing connection from Java
계속하려면 아무 키나 누르십시오 . . .

Thread

- After a connection is made with the first client, the server will not respond if a second client wishes to connect
  - » Solution : use threads
  - » Threads allow a program to run concurrently

Example of flexibility of streams

- The URL class gives us a simple way to read from a webpage
  - » Thanks to polymorphism we can create a Scanner that is linked to a website
  - » The example outputs the text from wikipedia

```
URL website = new
    URL("http://www.wikipedia.org");
Scanner inputStream = new Scanner(
    new InputStreamReader(website.openStream()));

while (inputStream.hasNextLine())
{
  String s = inputStream.nextLine();
  System.out.println(s);
}
inputStream.close();
```

끝