Software Construction
Basic IO: Sockets

Dhammika Elkaduwe

Department of Computer Engineering
Faculty of Engineering

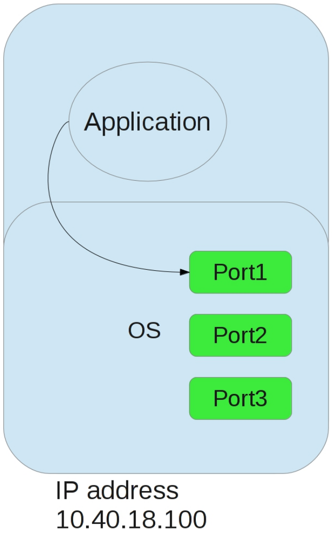
University of Peradeniya

ILO: what to look for

Look at the following concepts using an example:

- Use of sockets (in Java)
- OOP concepts
- Threads Vs. Code (how to use threads)
- IO handling
- Data exchanges

Sockets: basics



- Servers have ports to which one can establish connections
  - Clients make connection requests, server accepts
- Some ports are used for specific purposes (example: HTTP:80, DHCP: 67, SMTP:25 etc)and some are for general purpose communications
- Applications, via the OS can set-up connections via ports

Java TCP sockets

Java provides two types of sockets: server sockets and clients

```
import java.net.Socket;
// For clients
// Which provides
Sockets(String host, int port);
// for 2way communications
InputStream getInOutStream(); // for input
OutputStream getOutPutStream();// for output
```

```
import java.net.ServerSocket;
// For servers
// Which provides
serverSocket = new ServerSocket(socket);
Socket socket = serverSocket.accept();
```

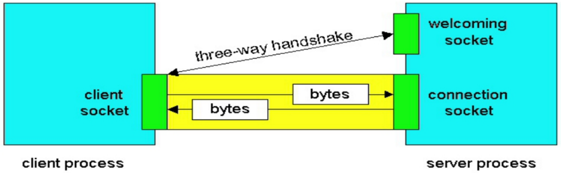
Java TCP sockets

Java provides two types of sockets: server sockets and clients

```
import java.net.Socket;
// For clients
// Which provides
Sockets(String host, int port);
// for 2way communications
InputStream getInOutStream(); // for input
OutputStream getOutPutStream();// for output
```

```
import java.net.ServerSocket;
// For servers
// Which provides
serverSocket = new ServerSocket(socket);
Socket socket = serverSocket.accept();
```

Server side

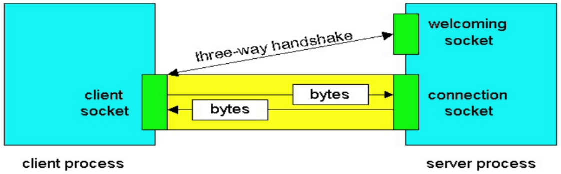


Server:

- Creates a socket (the welcoming socket)
- Waits for connections (using the *accept()* method)
- Once a request is received, opens a different socket for communications (returned by the accept method)

```
// create the welcome socket
serverSocket = new ServerSocket(socket);
Socket socket = serverSocket.accept();
// connection established.
// send and receive data via the new socket
```

Server side ...



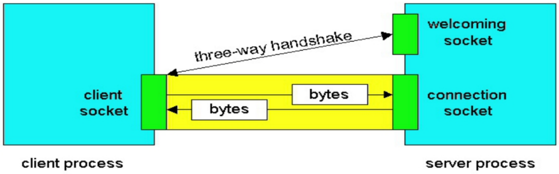
Server:

- A socket has two streams for input and output
- These streams can be used for receiving and sending
- These streams can be wrapped in buffered readers to improve performance.

```
InputStramReader inStream = new
    InputStreamReader(socket.getInputStream);
BufferedReader in = new BufferedReader(inStream);
```

see Server1.java

Connecting to the server



To connect as a client:

- You can write your own client code or
- use an existing tool like Linux tool *nc*

```
$
$ nc localhost 1250
```

see Server1.java

Question

Can more than one concurrent clients connect to this server?

```
while(true) {
  Socket socket = serverSocket.accept();
  handle(socket);
}

private void handle(Socket socket) throws IOException {

  ....
  for(line = in.readLine();
  line != null && !line.equals("quit");
  line = in.readLine()) {
    ...
  }
}
```

see Server1.java

Question

Can more than one concurrent clients connect to this server?
NO. Server socket is free, but the thread is doing something else.

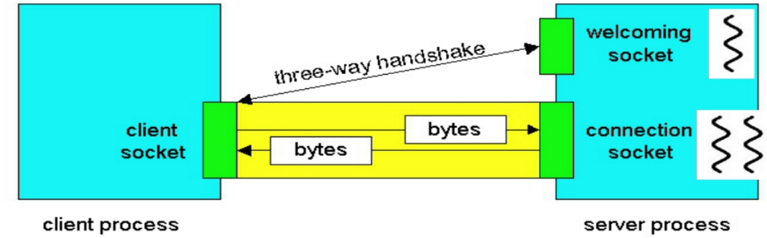
```
while(true) {
  Socket socket = serverSocket.accept();
  handle(socket);
}

private void handle(Socket socket) throws IOException {

  ....
  for(line = in.readLine();
  line != null && !line.equals("quit");
  line = in.readLine()) {
    ...
  }
}
```

see Server1.java

Task 1

- We cannot have concurrent connections since there is only one thread (two sockets)
- **Task 1** Make the server multi-threaded
  - Main thread handles the server part
  - Once a connection is established hand it over to another thread.
- Should we *extend* thread or **implement** runnable?

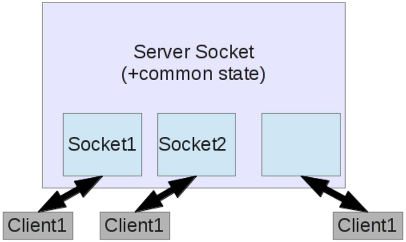
Task 1: Basic idea



Model of the problem:

- Single welcome socket (and associated state) common to all the communication sockets
- Welcome socket and communication sockets are different entities (communicating via interface)

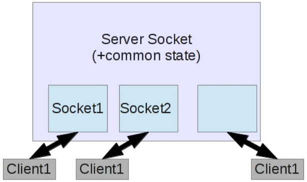
Option 1: Shared server socket



- Not the best of design; ideally developed in different classes
- Simple idea

```
private static ServerSocket serverSocket;
private static int socketNumber;
private Socket connectionSocket;
public Server2(int socket) throws IOException {/*create server
    socket*/ }
// overloaded constructor
public Server2(Socket socket) { // connection socket
  this.connectionSocket = socket;
}
```

Option 1: Shared server socket

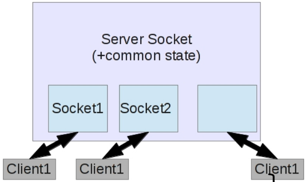


- Not the best of design; ideally developed in different classes
- Simple idea

```
public class Server2 implements Runnable {
  private static ServerSocket serverSocket;
  private static int socketNumber;
  private Socket connectionSocket;
  public Server2(int socket) throws IOException {/*server
      socket*/ }
  // overloaded constructor
  public Server2(Socket socket) { // connection socket
    this.connectionSocket = socket;
  }
```

see Server2.java

Option 1: Shared server socket



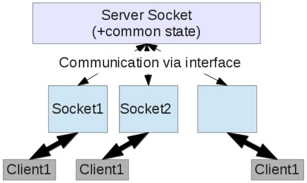
```
public void server_loop() {
  while(true) {
    Socket socket =
        serverSocket.accept();
    Thread worker = new Thread(new
        Server2(socket));
    worker.start();
  }
}
```

```
public void run() {
  try {
    // Handle the
        connection
```

see Server2.java

- Note that the *void run()* cannot be decleared to throw an exception
- So use try-catch

Option 2: without shared state



```
    public class Server3 {
      // Server class
      // you also implement an interface
           for communication
      public void server_loop() {
        while(true) {
          Socket socket =
              serverSocket.accept();
          CommServer worker = new
              CommServer(socket);
          worker.start();
    }

  class CommServer extends Thread {
    // handle connection
    public void run() {
      try {
        // Handle the connection
```

Make this a bit secure

Experience: the minute you open this up, people connect and say all kinds of things :)
Make it secure by:

- client needs to provide their name first before they can post anything else
  - Protocol: rules that governs how two entities would communicate with one another
- client name should be in the class list of CO225
  - Protocol: rules that governs how two entities would communicate with one another

Make this a bit secure...

Two issues:

- Need some data source which can given a name can say whether or not that name is in CO225 list.
    - Hash/set which has all the names
    - Need to create this using the class list (an excel sheet)
- The connection server need to act in two ways

initially	assume the string provided by the client to be the name (protocol requires the client to do this)
authenticate	check the if the name is in the class list. if not try the next string.
authenticated	post the text sent by the client with his/her name to the screen and echo back.

Make this a bit secure...

Two issues:

- Need some data source which can given a name can say whether or not that name is in CO225 list.
  - Hash/set which has all the names
  - Need to create this using the class list (an excel sheet)
- The connection server need to act in two ways

initially	assume the string provided by the client to be the name (protocol requires the client to do this)
authenticate	check the if the name is in the class list. if not try the next string.
authenticated	post the text sent by the client with his/her name to the screen and echo back.

Make this a bit secure...

Two issues:

- Need some data source which can given a name can say whether or not that name is in CO225 list.
  - Hash/set which has all the names
  - Need to create this using the class list (an excel sheet)
- The connection server need to act in two ways

initially	assume the string provided by the client to be the name (protocol requires the client to do this)
authenticate	check the if the name is in the class list. if not try the next string.
authenticated	post the text sent by the client with his/her name to the screen and echo back.

Protocol

Protocol:

- Once the connection is established, the client is expected send the registration number (Ex: E/11/111)
- If this registration number is in the CO225 class list the server should send "clear to send" string to client.
- If registration number sent by client is in the class list he/she will be allowed to post. After that posts will be tagged with his/her name.
- If the message is posted reply string should be "posted"
- If the string sent by the client is not a valid registration number, send "not registered" string to client and wait for next string and check if it is valid.

Interface: for authentication

```
  interface Atuh {
    /* given E-number return true if the name is
    * is in the class list
    */
    public boolean isRegistered(String);
    public String getName(String); // given E-number return name
}
```

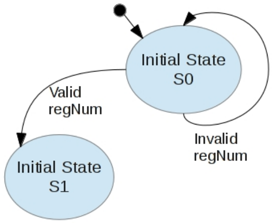
see Auth.java

Software as a state-machine

```
  interface Atuh {
    /* given E-number return true if the name is
    * is in the class list
    */
    public boolean isRegistered(String);
    public String getName(String); // given E-number return name
}
```

see Auth.java

Software as a *State Machine*



```
public void run() {
  for(line = in.readLine();
  line != null && !line.equals("quit");
  line = in.readLine()) {
    if(state == s0) {
      if(authInterface.isRegistered(line)) {
        state = s1;
        line = authOK;
        name = authInterface.getName(line);
      }
      else line = authFailed;
    }
    if(state == s1) {
      line = name +": " + line;
      System.out.println(line); line = poste
    }
    out.print(line + "\n"); out.flush();
  } // for loop
}// try
```

Comments about the design

- *CommServer* gets an *Auth* object which provides the required functionality for it to operate.
  - Suppose we wanted to allow all users to post. We just need a different *Auth* implementation.
  - Interface guarantees that functionality is there
- Creating a suitable object to authenticate was included in the main server (which can be taken out from there as well)

Data exchange formats

Specific problem:

- The AR office gives this information in a spreadsheet
- How to extract the data out form this?

Generic issue:

Question	How to exchange data between applications?
Solution	Standard formats to exchange data
Examples	CSV (Comma Separated Values), XML (eXtensible Markup Language), JSON (JavaScript Object Notation) ...

Data exchange formats

Specific problem:

- The AR office gives this information in a spreadsheet
- How to extract the data out form this?

Generic issue:

Question	How to exchange data between applications?
Solution	Standard formats to exchange data
Examples	CSV (Comma Separated Values), XML (eXtensible Markup Language), JSON (JavaScript Object Notation) ...

CSV

In CSV files:

- Each value is separated by a comma
- Each line contains one record
- First line gives the names of the fields

Reg No	Name	Field	Remark
E/99/001	Abeykoon A. B	CC	
E/99/002	Abeysinghe B	CC	

CSV file:

```
Reg No,Name,Field,Remark
E/99/001, Abeykoon A.B, CC,
E/99/002, Abeysinghe B, CC,
```

CSV to hashMap

```
class StudentDB implements Auth {
  private Map<String, String> classList;
  private String [] fields;
  public StudentDB(String cvsFile, String key, String val) {
    // read the file
    // figure out the index for the given key and val
    // (How: read the first line and keep in fields array)
    // read each line and put to hashMap
    tokens = line.split(","); // break on comma
    classList.put(tokens[keyIndex], tokens[valIndex]);
  }
  public boolean isRegistered(String eNum) {
    return classList.get(eNum) != null;
  }
  public String getName(String eNum) {
    return classList.get(eNum);
  }
```

see StudentBD.java

Next step

- As the next step we want to log everything which was posted.
- The logging can be to a file or database or something.
- **How to implement this?**

ILO: Revisited

Look at the following concepts using an example:

- Use of sockets (in Java)
- OOP concepts
- Threads Vs. Code (how to use threads)
- IO handling
- Data exchanges