

CS 580 Client-Server Programming
Fall Semester, 2002
Doc 12 Types of Servers
Contents

Types of Servers	2
Iterative verses Concurrent.....	3
Stateless verses Stateful Servers	8
Modes of Operation	9

References

Internetworking with TCP/IP, BSD Socket Version Vol. 3,
Comer, Stevens, Prentice-Hall, 1993

Copyright ©, All rights reserved. 2002 SDSU & Roger Whitney, 5500 Campanile Drive,
San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>)
license defines the copyright on this document.

Types of Servers

- Connectionless(UDP) verse Connection-Oriented (TCP)
- Iterative verses Concurrent
- Stateless verse stateful

Iterative verses Concurrent

Iterative

Single process

Handles requests one at a time

Good for low volume & requests that are answered quickly

Iterative Example

```
import java.net.Socket;
import java.net.ServerSocket;
import java.io.*;
import java.util.Date;

class SimpleDateServer {

    public static void main(String[] args) throws IOException {
        ServerSocket acceptor = new ServerSocket(4567);

        while (true) {
            Socket client = acceptor.accept();
            processRequest(
                client.getInputStream(),
                client.getOutputStream());
            client.close();
        }
    }

    static void processRequest(InputStream in,OutputStream out)
        throws IOException {

        BufferedReader parsedInput =
            new BufferedReader(new InputStreamReader(in));
        PrintWriter parsedOutput = new PrintWriter(out,true);
        parsedOutput.println(now.toString());
    }
}
```

Concurrent

Handle multiple requests concurrently

Normally uses thread/processes

Needed for high volume & complex requests

Harder to implement than iterative

Must deal with currency

Concurrent Server Example

```
| server |
server := SocketAccessor newTCPserverAtPort: 9009.
server listenFor: 5.

[ | acceptedSocket |
  "wait for a new connection"
  acceptedSocket := server accept.

  "fork off processing of the new stream socket"
  [ | stream char |
    stream := acceptedSocket readAppendStream.
    stream lineEndTransparent.
    [ (char := stream next) isNil ] whileFalse: [
      stream nextPut: char; commit ].
    stream close.
  ] forkAt: Processor userSchedulingPriority -1.
] repeat.
```

Single Process/Thread Concurrent Server

One can implement a concurrent server using one thread/process

```
while (true) {  
    check if any new connects (non-block accept)  
    if new connection accept  
    process a little on each current request  
}
```

Stateless verses Stateful Servers

State information

- Information maintained by server about ongoing interactions with clients

Stateless server

- Server that does not maintain state information

Stateful server

- Server that does maintain state information

State information cause problems

- Consumes resources
- How long does one maintain the state?

Modes of Operation

Stateful servers sometimes have different modes of operation

Each mode has a set of legal commands

In Login mode only the commands password & username are acceptable

After successful login client-server connection in transaction mode

In transaction mode command X, Y Z are legal

These modes are also called server states or just states