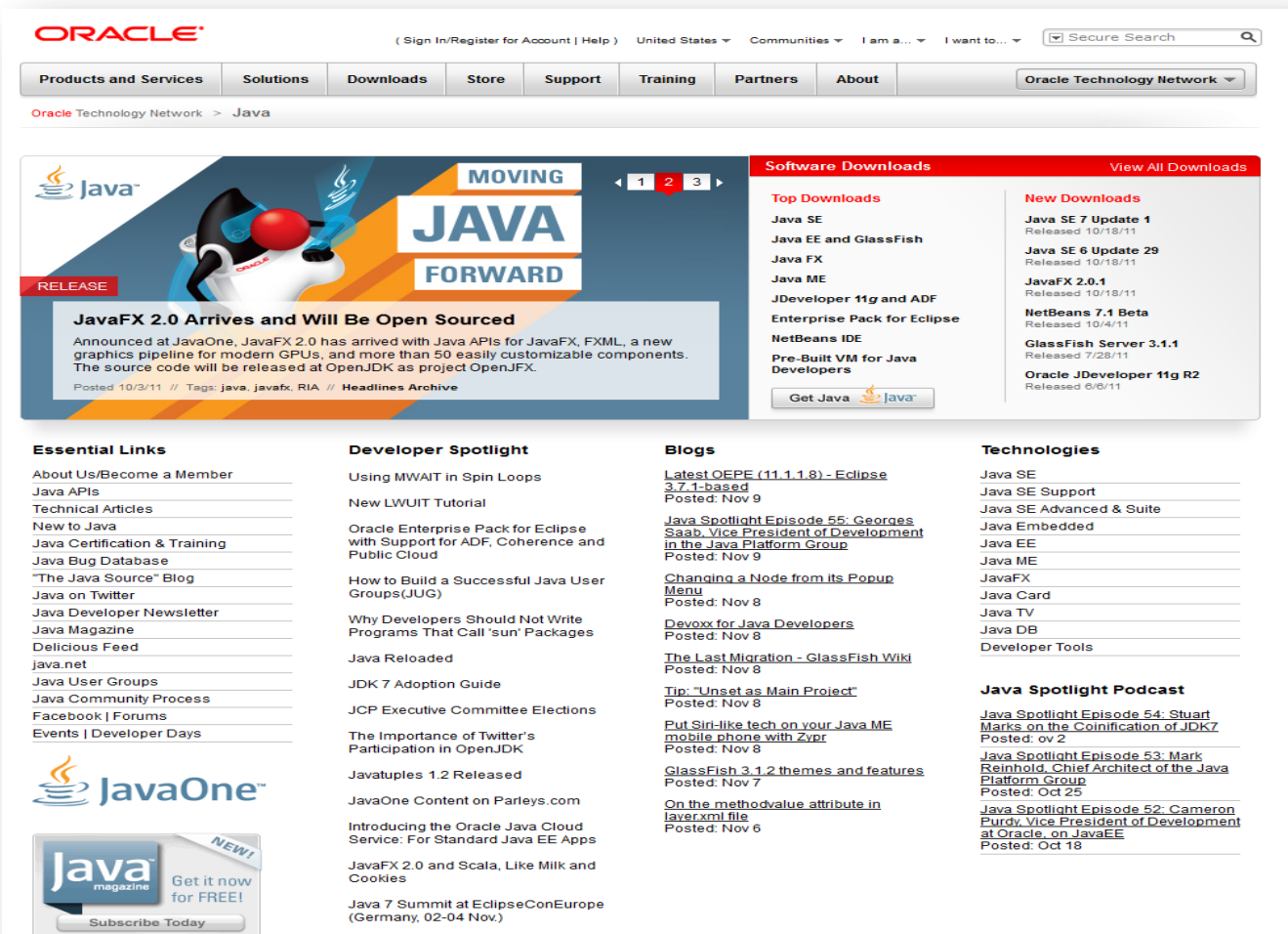


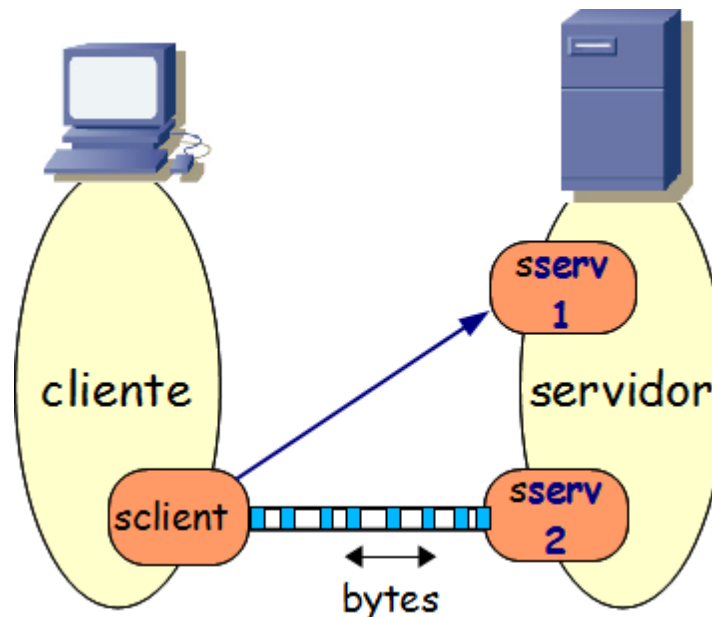
Tema 3 (2): Servidores TCP secuenciales



Bibliografía:

- ❑ [Kurose10] Apartados 2.1, 2.7, 2.8
- ❑ <http://www.oracle.com/technetwork/java/index.html>
- ❑ <http://java.com/es/about/>
- ❑ <http://download.oracle.com/javase/6/docs/api/index.html>

- Cliente:
 - Cuando crea un socket (`sclient`) establece la conexión con el servidor
- Servidor:
 - Debe haber creado un socket (`sserv1`) donde espera a los clientes que conectan con él
 - Cuando un cliente se conecta con un servidor:
 - El servidor crea un nuevo socket (`sserv2`) para que el proceso servidor se comuniquen con el cliente
 - De esta forma es posible que un servidor se comuniquen con varios clientes simultáneamente



- Clase `ServerSocket`

- Constructores

- `ServerSocket(int puerto) throws IOException`
 - Abre un socket en el puerto indicado en modo de escucha
 - Si `port=0`, entonces se elige cualquier puerto libre
- `ServerSocket(int puerto, int backlog) throws IOException`
 - Abre un socket en el `puerto` indicado en modo de escucha
 - `backlog` indica la longitud máxima de la cola de conexiones en espera
 - Cuando llega una solicitud de conexión y la cola está llena, se rechaza la conexión




- Clase **ServerSocket**
- Algunos métodos importantes
 - **Socket accept() throws IOException**
 - Acepta una conexión de un cliente y devuelve un socket asociado a ella
 - El proceso se bloquea hasta que se realiza una conexión
 - El diálogo con el cliente se hace por el nuevo socket
 - El ServerSocket puede atender nuevas conexiones
 - **close() throws IOException**
 - Cierra el socket servidor



```
import java.net.*;
import java.io.*;

class ServidorTCP {

public static void main(String args[]) {
    try{
        ServerSocket ss=new ServerSocket(7777);
        while(true){
            Socket s=ss.accept(); // espero a que llegue un cliente
            PrintWriter salida=new PrintWriter(s.getOutputStream(), true);
            salida.println("Bienvenido al servidor de prueba de Pepito");
            s.close();
        }
    } //try
    catch(IOException e) { System.out.println(e); }
} // main
} // ServidorTCP
```



El servidor espera un cliente.
Cuando éste se conecta, el
servidor le envía una cadena de
bienvenida y cierra la conexión

```
zoltar:~/java> nc localhost 7777
Bienvenido al servidor de prueba de Pepito
```



- **Servidor iterativo:** hasta que no termina la petición de un cliente no atiende al siguiente
 - A cada uno le envía una cadena con su número de cliente y después cierra el socket

```
import java.net.*;
import java.io.*;

class ServidorTCP2 {
public static void main(String args[]) {
    try{
        ServerSocket ss=new ServerSocket(7777);
        int cliente=1;
        while(true) {
            Socket s = ss.accept(); // espera una conexión de un cliente
            PrintWriter salida=new PrintWriter(s.getOutputStream(),true);
            salida.println("Usted es el cliente " + cliente);
            System.out.println("Atendiendo al cliente " + cliente++);
            s.close();
        }
        catch(IOException e) {
            System.out.println(e)
        } //catch
    } // main
}
```

