Università degli Studi di Bologna
Facoltà di Ingegneria

Principles, Models, and Applications for Distributed Systems M

Lab assignment 5
Connected Java sockets

Luca Foschini

Specification: Client

Develop a C/S application that manages the transfer of **all the files in a specific directory from the server to the client** (**multiple get**).

The client asks the user the **name of the directory**, relative to the working directory of the server, connects to the server (using **`java.net.Socket`**), builds an output stream to **send the request** and an input stream to **receive the files** or a **negative response**, if the directory does not exist.

Requested files are **saved in the working directory**, overwriting existing files having the same name.

Specification: Server

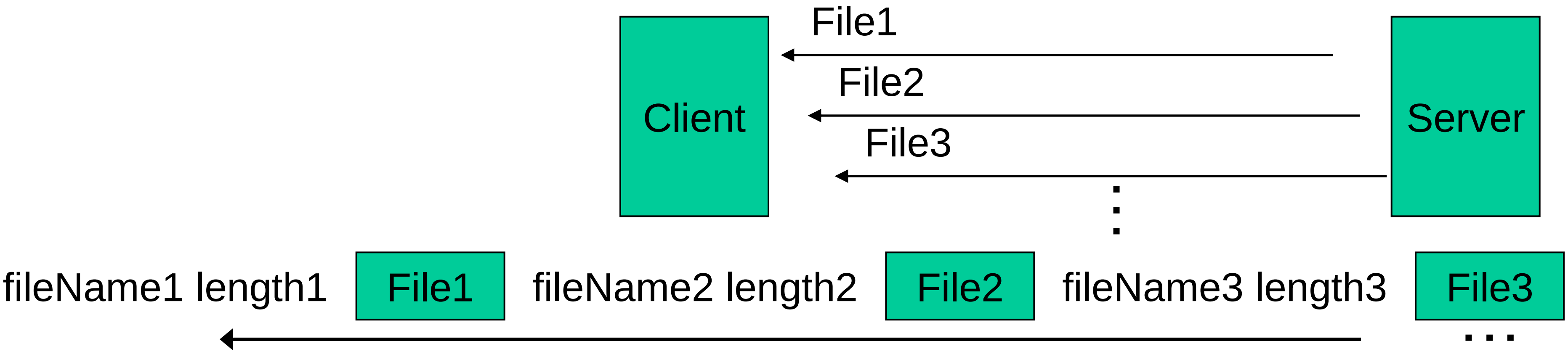
Server listens for client connection requests (using `java.net.ServerSocket`), then uses the resulting connected socket (`java.net.Socket`) to build an input stream to **receive a request** and an output stream to **send requested files** or a **negative response**, then closes the connection. **Server must use the same socket to send all the files.**

Server must be **concurrent and parallel**. For each request received by the father process, it must activate a child that manages the service completion.

Implementation Hints

To correctly send files, it is necessary to design a protocol to handle multiple file transfers. For example the server, **before sending the first file**, could send the **file name** and its **size in bytes**, then **sends the byte stream**. The server follows this protocol for each file and uses **the same socket** for all transfers.

All these steps are repeated until the server has sent all the files in the requested directory, then the server closes the connection to notify the end of transmission.


Client
File1
File2
File3
Server
fileName1 length1
File1
fileName2 length2
File2
fileName3 length3
File3
...