Learning goals

Before the next day, you should have achieved the following learning goals:

- Understand what a client–server architecture is.
- Launch a server and leave it waiting and listening on one machine.
- Launch clients that connect to a server on the same machine (localhost).
- Launch clients that connect to a server on another machine.

The exercises for today are examples of simple network services that you can implement in a client-server way. For each of the exercises, you will need to define an interface, implement such interface through a server object, and then implement a client that uses the services implemented by the server.

You will also need to write a security policy file based on the example in the notes.

1 echo

This simple service receives a String and returns the same String. (Hint: This is the example we have seen in the notes.)

This service has traditionally listened at port 7.

2 date

This service answers requests for the time and date with the current time and date. You can access this at the server by using classes Calendar and Data (in package java.util.*).

A similar service has traditionally listened at port 37.

3 Calculator

This service exports simple methods for addition, subtraction, multiplication, and division. (In the real world, this kind of service usually provides services like complicated financial calculations).

4 Remote file access

This service provides the state of a file on the server machine. The client must provice a route and file name, which may or may not exist. If it exists, the server will return the content of the file as a String. Note: For simplicity, focus only on text files.

5 Finger (*)

This service says who is connected on the machine. You will simulate this by having two types of clients: *login* clients and *finger* clients. You will need two servers, or one server that can serve both services.

Login clients one will "connect" users to the server by sending pairs *(user, password)*. These pairs may or may not be correct. The user will be considered connected if and only if they are correct.

Finger clients will query about which users are connected, receiving both a set of logins and information about when those users did connect to the server.

A similar service has traditionally listened at port 79.