

# University of St Andrews School of Computer Science

CS2003 — Internet and the Web — 2020/21

Assignment: Remote server

Deadline: 2020-09-23 Credits: 10% of coursework mark

MMS is the definitive source for deadline and credit details

---

**You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.**

## Aim / Learning objectives

The aim of this assignment is twofold. First, it is to ensure that you are competent at remote working as this is essential given the current situation, and the skills used in this practical will be required for future practicals in the module. Second, it is to gain experience around client/server programming.

## Requirements

You are to build a pair of client and server programs in Java that must:

- meet the specification outlined below (both protocol and arguments)
- be tested thoroughly on at least two host servers

## Specification

### Command-line arguments

Your programs must be named `SimpleClient` and `SimpleServer`. `SimpleClient` must take two arguments and be able to be run as follows:

```
java SimpleClient <IPaddress> <port>
```

where `<IPaddress>` is the IP address of a server, and `<port>` is the port number to connect to.

`SimpleServer` must take one argument and be able to be run as follows:

```
java SimpleServer <port>
```

where <port> is the port number to connect to.

Your programs must be tested and should raise appropriate errors if incorrect numbers of arguments are passed.

## Protocol

The protocol for your client and server is text-based. A signal-sequence diagram for a successful session is shown below in Figure 1.

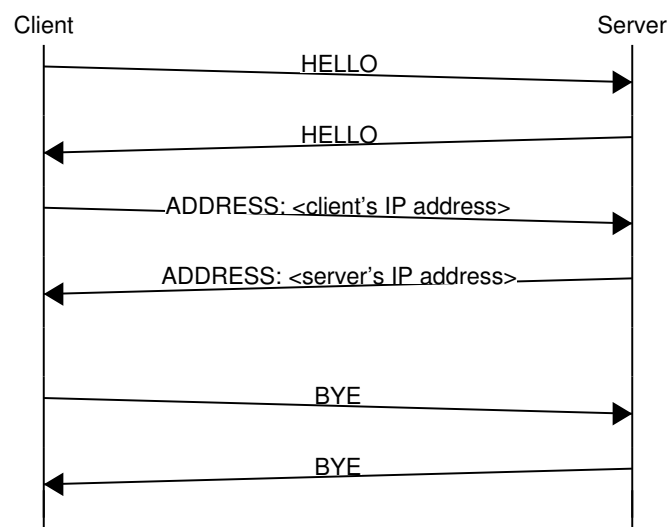


Figure 1: A successful run of the protocol.

Client and server both use TCP and send ASCII strings. The client initiates communication to the server. When the server receives one of the following strings, it will respond as shown in Table 1.

<i>Received</i>	<i>Response</i>
HELLO	Respond with HELLO until receiving ADDRESS: IPaddress
ADDRESS: IPaddress	Respond with ADDRESS: IPaddress until receiving BYE
BYE	Respond with BYE and close connection

Table 1: Server responses

## Server program

Your server program must obey the protocol as described. It will sit and wait for TCP connections, and respond with strings as appropriate. If an unexpected string is received, it should respond with the string listed in the protocol (i.e., if it is waiting for ADDRESS but does not receive this, it should respond with HELLO).

## Client program

Your client program will send text according to the protocol. It will initiate the protocol, i.e., the first thing it will do is send HELLO. It will then continue until a BYE is received and then terminate.

## Testing

You will develop two programs that obey the protocol correctly. This means that they will not be very interesting, since they will complete a transaction as shown in Figure 1 and terminate. That said, you must demonstrate successful operation of this.

You should also test that your programs are robust against unexpected input or protocol violations. To do this, you should use the program `nc` which is installed on the CS Linux machines and will also be used in the Week 1 exercise class. You should also test other aspects of your programs as you deem necessary, and with recall of testing strategies covered in first-level.

## Remote access

You will find the CS System Wiki page on working remotely useful: [https://systems.wiki.cs.st-andrews.ac.uk/index.php/Working\\_remotely](https://systems.wiki.cs.st-andrews.ac.uk/index.php/Working_remotely) You should also review the Orientation Week material on remote access, the FAQ in the Lab Team, and attempt the Week 1 exercise tasks on running commands on the host servers.

## Host servers

The School has four host servers (see [https://systems.wiki.cs.st-andrews.ac.uk/index.php/Linux\\_Host\\_service](https://systems.wiki.cs.st-andrews.ac.uk/index.php/Linux_Host_service)). Their hostnames and current IP addresses are shown in Table 2.

<i>Hostname</i>	<i>IP address</i>
klovia.cs.st-andrews.ac.uk	138.251.22.78
lyrane.cs.st-andrews.ac.uk	138.251.22.77
palain.cs.st-andrews.ac.uk	138.251.22.76
trenco.cs.st-andrews.ac.uk	138.251.22.79

Table 2: Host servers

When you login to your personal host server, you will see that this is a DNS alias for one of the above servers. You can check which server is yours by one of the following:

- looking at the command-line prompt after you have logged into your host server
- running the command `hostname` on your host server
- using the DNS lookup tool `dig` as follows:  
`dig +short username.host.cs.st-andrews.ac.uk`  
 (we will look at DNS in more detail later in the module)

You must test that one of your programs can run on your host server and the other on a different host server from the list above.

## Report

Your report should show evidence of testing, describe any problems you have had, and also show evidence of some successful runs of the protocol.

## Hints

- Working out a machine's IP address is surprisingly complicated! Remember that an IP address identifies an *interface* and not a *host*. A host may have multiple interfaces. One solution is to identify the interface used to connect to a particular server, for instance by opening a connection (i.e., a socket) and then looking at the relevant address (`getLocalAddress()`)
- Make judicious use of the `nc` tool. Look at the Week 1 exercise tasks and also check the documentation, i.e., `man nc`.
- The server will need to keep track of its current *state* to work out what to send next.

## Submission

A single file containing your code and report in ZIP format must be submitted electronically via MMS by the deadline. Submissions in any other format will be rejected.

The standard mark descriptors apply to this practical. Note that there are no extensions; to achieve excellent marks you should meet all of the requirements with exceptional clarity and demonstrated insight.

## Policies and Guidelines

### Marking

See the standard mark descriptors in the School Student Handbook:  
[https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark\\_Descriptors](https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptors)

### Lateness penalty

The standard penalty for late submission applies (Scheme B: 1 mark per 8 hour period, or part thereof):

<https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#lateness-penalties>

### Good academic practice

The University policy on Good Academic Practice applies:

<https://www.st-andrews.ac.uk/students/rules/academicpractice/>