Homework
Project Specifications - June

Computer Systems & Programming
Academic Year 2023-2024

Overview

The project shall be carried out ***individually***: it must constitute ***an original creation*** , therefore it is not possible to share parts of the code with other students/groups, or copy contents deriving from other sources.

The project shall be submitted:

- via email to the teacher's official email
- at least by midnight of D-6
- not earlier than one week before the deadline (see previous bullet)

- in a `tar` file including:
  - the source file(s)
  - a Makefile (better) or a script for compilation
  - a README file to guide the installation, compilation and **test**

Project Specifications - I

- **Objective: implement a "remote login server", i.e. a server that lets clients connect to a shell running on a remote server**
- Connection should be stateful: the client connects to the server and let the user interact with the remote shell until they explicitely close the connection
- The server should be able to manage an unlimited number of clients, with minimal delay to establish a connection (hint: use `fork()` to create a dedicated process server for each client)
- The login server shall authenticate connections against the users defined on the operating system's server
- The login server should wait for input on a well-known port (defined in /etc/services).

Project Specifications - II

- The client executable should accept the following option on the command line:
  - IP address or DNS name of the server
- The client should connect to the server on a well-known port.
- The client should provide functionality and syntax similar to the `rlogin` command (see `man rlogin`) except:
  - no remote command execution (only interactive connection to the remote shell)
  - support the following options only: -l, -p
- A configuration file should provide to the client the default values for the option parameters

Project Specifications - III

- The server should correctly manage user rights, i.e.:
  - require client authentication with user and password, as defined in the server
    - password encryption not required
  - root access should not be allowed
  - the user should login directly to their “home” directory on the server
  - the user should have exactly the same rights as if he logged directly to the remote server

Project Specifications - IV

- The server should shutdown when it receives a user-selected signal or a `quit` command from the command line (not from a client!).
- The server executable should accept the following options on the command line (default values should be stored in a configuration file on the server):
  - Maximum number of simultaneously supported connections
  - Port to listen from (default value in */etc/services/*)
- **Extra bonus (I)**: support the execution of remote command at invocation time (like `rlogin`)
- **Extra bonus (II)**: support the transmission of Ctrl-C to the remote shell
- **Extra bonus (III)**: provide automated tests (see slide on correctness)

Implementation & Discussion

The project consists of a C language program that satisfies the specified requirements, using the library calls ***that are part of the course program***. The use of other calls is not accepted. If in doubt, ask the teacher.

The project code must correctly compile and execute in the required software environment (compiler version, kernel version, clib version)

Once the project has been submitted and a sufficient grade has been obtained, it is mandatory to show up at the immediately following exam session to take the exam.

The oral exam consists of a discussion on design choices and software implementation of the project. Taking inspiration from the project work, questions may be asked on any topics that are part of the course program. Specific questions will be asked on any set of calls that have not been used in the project (e.g., shared memory, semaphores)

Correctness

The project shall successfully pass (at least) **ALL of the following test cases**:

1. Server startup (start the server only)
2. Client correctly connect and authenticate to the server (after (1))
3. Client execute any shell command or set of commands (after (1))
4. Second client connect and execute any shell command (after (3), start another client that connects to the server)
5. Client shutdown cleanly (after (4))
6. Server shutdown cleanly (after (5)).

Evaluation Criteria

Prerequisite: the code must correcly compile, link and execute the tests (see previous slide) on Ubuntu 22.04.1 (gcc version 11.4.0). If it does not, **you cannot take the oral exam**.

Correctness of the code: main evaluation element that determines (alone!) the passing of the exam.

Error management: that's an integral part of the correctness of the code!

Modularity and readability of the code: division into functions, comments, function and variable names (sic!), etc...

Quality of documentation: user manual, software architecture, README file, project report.