Homework
Project Specifications

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 one week before the exam date (i.e., by midnight of D-8)
- in a tar file including:
  - the source file
  - a Makefile or script for compilation
  - a README file to guide the installation, compilation and test

Project Specifications - I

- **Objective: implement a "log server", i.e. a server that writes to a log file the inputs it receives from clients**
- Connection should be stateful: the client connects to the server, sends a set of messages and closes the connection (hint: use stream sockets)
- 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)
- Messages in the log file should include a header by the log server with timestamp and info on the client (e.g., name, IP address)
- Log file is unique for all clients but messages from different clients cannot overwrite or intertwin (hint: beware of when and how you open the log file)

Project Specifications - II

- The client executable should accept the following options on the command line (default values should be stored in a configuration file):
  - IP address and listening port of the server
- When started, the client should automatically connect to the server
- The client should wait for a new message from its standard input and send it immediately to the server
- The client should loop forever (i.e., till it reads an EOF)
- When started, the server should open a new log file (without removing any old file)
- The server executable should wait for input on a well-known port
- The server should shutdown when it receives a user-selected signal or a quit command from the command line.

Project Specifications - III

- The server executable should accept the following options on the command line (default values should be stored in a configuration file):
  - listening port
  - directory to store the log file
- Extra bonus: when the log file size exceed a given threshold, the server should cancel the oldest log file in the log directory and create a new log file. In this case, the server should not create a new log file at start-up, but rather append to the most recent log file in the directory.

Implementation

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.

Correctness

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

1. Server startup (start the server only)
2. Client connect (after (1), record the connection on the log file)
3. Client message (after (1), let the client send a message and the server succesfully logs it)
4. Second client message (after (3), start another client that sends a new message to the server)
5. Client shutdown (after (4), let the client succesfully close the connection and the server log the closing)
6. Server shutdown (after (5), let the server succesfully stop, after recording the order of shutdown).

Evaluation Criteria

Prerequisite: the code must correcly compile, link and start 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 handling: it is 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.