42 Evals

Back to all evaluation sheets

Points earned
0

minitalk

You should evaluate 1 student in this team

Introduction

Please follow the rules below:

Remain polite, courteous, respectful, and constructive throughout the evaluation process. The well-being of the community depends on it.

Identify with the student or group whose work is being evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.

You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only if the peer-evaluation is done seriously.

Guidelines

Please follow the guidelines below:

Only grade the work that was turned in to the Git repository of the evaluated student or group.

Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that 'git clone' is used in an empty folder.

Check carefully that no malicious aliases were u
evaluating something that is not the content of the of

To avoid any surprises and if applicable, review t
facilitate the grading (scripts for testing or automatio

Points earned
0

If you have not completed the assignment you are going to evaluate, you must read the entire subject prior to starting the evaluation process.

Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth. In these cases, the evaluation process ends and the final grade is 0, or -42 in the case of cheating. However, except for cheating, students are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

Remember that for the duration of the defense, no segfaults or other unexpected, premature, or uncontrolled terminations of the program will be tolerated, else the final grade is 0. Use the appropriate flag.

You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explain the reasons with the evaluated student and make sure both of you are okay with this.

You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed before the end of execution.

You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Attachments

Please download the attachments below:

subject.pdf

Points earned
0

Mandatory Part

Preliminary tests

If cheating is suspected, the evalvation stops here, Use the Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Prerequisites

Defense can only happen if the evaluated student or group is present. This way everybody learns by sharing knowledge.

If no work has been submitted (or wrong files, wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.

No empty repository (= nothing in Git repository).

No Norm error.

Cheating (= -42).

No compilation error. Also, the Makefile must not re-link.

If all of these requirements are passed, check Yes and go on.

Otherwise, use the appropriate flag at the end of the scale!

Yes No

General instructions

General instructions

- The Makefile compiles both executables → 1 p
- The server name is 'server' and it prints his PI
- The client name is 'client' and is run as follows
STRING_TO_PASS' → 2 points

Points earned
0

Rate it from 0 (failed) through 5 (excellent)

0 1 2 3 4 5

Mandatory Part

This project is an introduction t O signals. Check the code anda ensure the signals are used only for the communication between the server and the client.

Message transmission

It's possible to pass on a message of any size.

Received messages must be displayed by the server, and must be obviously corrects!

The server should never get stuck or print wrong characters.

Yes No

Simple setup

Simple setup

The server can receive multiple strings without needing to be restarted. → 1 point

Only one global variable per program is allowe
explanations. → 1 point

The communication is done only using the sig
SIGUSR2. → 3 points

Points earned
0

Received messages must be displayed by the server, and must be obviously correctsl

Rate it from 0 (failed) through 5 (excellent)

0 1 2 3 4 5

Bonus Part

Bonus Part

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or bad usage. In case all the mandatory points were not passed during the defense, bonus points must be ignored.

Unicode characters are supported both by the client and the server.

Yes No

Acknowledgement

Acknowledgement

The server acknowledges every message receive
to the client.

Yes No

Points earned
0

Ratings

OK Outstanding Empty Work

Incomplete Work Invalid Compilation

Norme Cheat Crash

Concerning Situations Leaks

Forbidden Functions