

≡ 🔎 (https://profile.intra.42.fr/searches)

mjusta

(https://
profile.intra.42.fr)

SCALE FOR PROJECT FT_IRC (/PROJECTS/FT_IRC)

You should evaluate 3 students in this team



Git repository

`git@vogsphere.42prague.com:vogsphere/intra-uuid-20e5cc6c-e016-4530-81`

Introduction

Please adhere to the following rules:

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.
- Identify the possible dysfunctions in the project of the student or group whose work is being evaluated. 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. Pedagogy is useful only if peer evaluation is done seriously.

Guidelines

- Only grade the work submitted in the Git repository of the evaluated student or group.
- Double-check that the Git repository belongs to the student or students. Ensure that the project is the expected one. Also, check that 'git clone' is used in an empty directory.
- Check carefully that no malicious aliases were used to deceive you and make you evaluate something that is not the content of the official repository.
- To avoid any surprises, and if applicable, review together any scripts used to facilitate grading (such as testing or automation scripts).
- If you have not completed the assignment you are going to evaluate, you must read the entire subject before starting the evaluation process.
- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, etc.
In these cases, the evaluation process ends, and the final grade is 0, or -42 in the case of cheating. However, except in cases of cheating, students are strongly encouraged to review the submitted work together to identify any mistakes that should not be repeated in the future.
- Remember that for the duration of the defence, no segfault, no other

unexpected, premature, uncontrolled or unexpected termination of the program, 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 explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify that there are no memory leaks. Any memory allocated on the heap must be properly freed before the program's execution ends.

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

Attachments

subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/193967/en.subject.pdf>)

bircd.tar.gz (<https://cdn.intra.42.fr/document/document/44796/bircd.tar.gz>)

Mandatory Part

README.md Compliance Check

Does the repository contain a README.md file at its root, and does it include all of the following?

- The first line is italicized and formatted exactly as: *This project has been created as part of the 42 curriculum by <login1>[, <login2>[, <login3>[...]]].*
- A "Description" section explaining the project's purpose and providing a brief overview.
- An "Instructions" section with relevant details about compilation, installation, and/or execution.
- A "Resources" section listing references (documentation, tutorials, etc.) and explaining how AI was used, specifying for which tasks and which parts of the project.

If any of the required elements is missing, the grade is 0.

Yes

No

Basic checks

- There is a Makefile, the project compiles correctly with the required options, is written in C++, and the executable is named as expected.
- Ask and check how many poll() (or equivalent) calls are present in the code. There must be only one.
- Verify that no call to accept, read/recv, write/send happens before a call to poll() (or equivalent). After these calls, errno should not be used to trigger specific actions (e.g., like reading again after errno == EAGAIN).
- Verify that each call to fcntl() is done as follows: fcntl(fd, F_SETFL, O_NONBLOCK); Any other use of fcntl() is forbidden.
- If any of these points is wrong, the evaluation ends immediately and the final mark is 0.

Yes

No

Networking

Check the following requirements:

- The server starts and listens on all network interfaces on the port given from the command line.
- Using the 'nc' tool, you can connect to the server, send commands, and the server responds.
- Ask the team what is their reference IRC client.
- Using this IRC client, you can connect to the server.

- The server can handle multiple connections simultaneously. The server should not block. It should be able to answer all demands. Conduct some tests with the IRC client and nc simultaneously.
- Join a channel using the appropriate command. Ensure that all messages from one client on that channel are sent to all other clients who have joined the channel.

 Yes No

Networking specials

Network communications can be disturbed by many unusual situations.

- As in the subject, using nc, try to send partial commands. Check that the server responds correctly. With a partial command sent, ensure that other connections continue to function properly.
- Unexpectedly terminate a client. Then check that the server remains operational for other connections and any new incoming client.
- Unexpectedly terminate an nc session with just half of a command sent. Check again that the server is not in an unusual state or blocked.
- Suspend a client (^Z) connected to a channel. Then flood the channel with another client. The server should not freeze. When the client is active again, all stored commands should be processed normally. Also, check for memory leaks during this operation.

 Yes No

Client Commands basic

- With both nc and the reference IRC client, check that you can authenticate, set a nickname, set a username, and join a channel. This should be fine (you should have already done this earlier).
- Verify that private messages (PRIVMSG) are fully functional with various parameters.

 Yes No

Client Commands channel operator

- With both nc and the reference IRC client, check that a regular user does not have the privileges to perform channel operator actions. Then test with an operator. All channel operation commands should be tested (deduct one point for each feature that is not working).

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



Bonus part

Evaluate the bonus part if, and only if, the mandatory part has been completed entirely and perfectly, and error management handles unexpected or improper usage. If all the mandatory points were not passed during the defense, bonus points must be completely ignored.

File transfer

File transfer should work with the reference IRC client.

 Yes No

A small bot

There is an IRC bot.

Yes

No

Ratings

Don't forget to check the flag corresponding to the defense

Ok

Outstanding project

Empty work

Incomplete work

Invalid compilation

Cheat

Crash

Incomplete group

Concerning situation

Leaks

Forbidden function

Can't support / explain code

Conclusion

Leave a comment on this evaluation (2048 chars max)

[Finish evaluation](#)

API General Terms of Use (https://profile.intra.42.fr/legal/terms/33)	Declaration on the use of cookies (https://profile.intra.42.fr/legal/terms/2)	Privacy policy (https://profile.intra.42.fr/legal/terms/5)	General term of use of the site (https://profile.intra.42.fr/legal/terms/6)	Rules of procedure (https://profile.intra.42.fr/legal/terms/4)	Terms of use for video surveillance (https://profile.intra.42.fr/legal/terms/1)	Legal notices (https://profile.intra.42.fr/legal/terms/3)
---	---	--	---	--	---	---