SCALE FOR PROJECT IRC (/PROJECTS/IRC)

Guidelines

You now know the principle of peer-correcting.
If it is not yet the case, find the guidelines on the state
of mind of peer-correcting in the dedicated document on e-learning.

Check the following at the beginning of the defense:

- There is a rendering (in the git repository)
- Valid author file
- The Makefile is present and compiles the server and
client executables well
- No standard fault, the Norminette is authentic
- No cheating (unauthorized functions, students must be
able to explain their code, ...)

If an element does not conform to the subject, the notation stops
. You are encouraged to continue discussing the project, but the
scale is not applied.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/5229/ft_irc_20-02-2015.pdf)

Subject (https://cdn.intra.42.fr/pdf/pdf/5233/ft_irc.en.pdf) bircd (/uploads/document/document/86/bircd.tar.gz)

preliminaries

Prerequisites

Are eliminative for this project:
- use of fcntl ()
- read (2) / write (2) blockers (that is not preceded by a check by select (2) that they will not be blocking)
- threads
- fork

If one of these points is true, the project is off-topic, the defense stops.
All is well ?

✓ Yes ✕ No.

Network Part

FD_ZERO ()

There is a FD_ZERO () before the call to select (2) on all FD_SET used?

✓ Yes ✕ No.

fd_set in writing

For writing, only the file descriptors on which one has something to write are put in the fd_set in writing?

✓ Yes ✕ No.

timeout

The select (2) timeout is intelligently managed (NULL timeout so that select (2) only releases when something happens on a socket or when a socket is closed)?

✓ Yes ✕ No.

read

Only one read must be done by file descriptor, after the select.

✓ Yes ✗ No.

write

Only one write must be done by file descriptor, after the select.

✓ Yes ✗ No.

Management of the read

We must have a correct management of the number of bytes actually read + a disconnection if read (2) returns 0.

✓ Yes ✗ No.

Receipt of orders

One must have a merge of read packets to form a command. Test with nc by sending a command interrupted by Ctrl + D to flush the buffers. The order must be interpreted normally, no unknown order or other.

✓ Yes ✗ No.

Rotating buffer

We have a revolving buffer in reading?

✓ Yes ✗ No.

Rotating buffer ++

We have a buffer turning in writing?

✓ Yes ✗ No.

The orders

The messages

Test with the client of the project, do the messages go well when you have two connected clients?

✓ Yes ✗ No.

More customers

Do messages go well when we have more than two connected clients?

✓ Yes ✗ No.

Nickname

The Nickname command is handled? Test in the client:

```
/ nick _nickname_
```

✓ Yes ✗ No.

Connection to the server

It manages the connection to the server from the command line? Test:

```
$> ./client
```

✓ Yes ✗ No.

Login in the client

/ connect _host _ [: port] works?

✓ Yes ✗ No.

Repeated connection in the client

/ connect _host _ [: port], while we are already connected works?

✓ Yes ✗ No.

User messages

Can we send messages from user to user? Test in the client:

/ msg

✓ Yes ✕ No.

Channel Management

It manages the channels?
/ join, / leave, and the messages are distinct on each channel?

✓ Yes ✕ No.

Who

The who command to know who is logged on the channel?
Test in the client:

/ who

✓ Yes ✕ No.

bonus

IPv6 support

The project supports both IPv4 and IPv6.

✓ Yes ✕ No.

Respect of the RFC

The project complies with the IRC protocol defined by RFC (rfc 1459 or rfc 2812).

✓ Yes ✕ No.

Graphic client

The project manages a client in GTK / qt / ncurse, etc ... that works
(not just a poor window that does not do anything!).

✓ Yes ✗ No.

Other bonuses

If there are other bonuses, count them here - you can
count up to 5 bonuses. Bonuses must be 100%
functional.

Example of valid bonuses:
- management of a login / password to connect to the server
- management of a key to enter a channel
- management of a prompt
- completion during a / join
- completion during a / msg
- ...

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

○

ratings

Do not forget to check the flag

✔ Okay ★ Outstanding project

Empty work Incomplete work No author file Invalid compilation

Standard cheat Crash ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Preview !!!

General term of use of the website
(https://signin.intra.42.fr/legal/terms/6)

Privacy policy
(https://signin.intra.42.fr/legal/terms/5)

Legal notices
(https://signin.intra.42.fr/legal/terms/3)

Declaration on the use of cookies
(https://signin.intra.42.fr/legal/terms/2)

Terms of use for video monitoring
(https://signin.intra.42.fr/legal/terms/1)

I
(https://sigi