search...

SCALE FOR PROJECT GET_NEXT_LINE (/PROJECTS/GET_NEXT_LINE)

You should correct 1 student in this team

Git repository

Introduction

Please respect the following rules:

- Remain polite, courteous, respectful and constructivethroughout the correction process. The well-being of the community depends on it.

- Identify with the person (or the group) graded the eventualdysfunctions of the work. Take the time to discuss and debate the problems you have identified.

0 **rnugroho**

- You must consider that there might be some difference in how yourpeers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group'sGiT repository.

- Double-check that the GiT repository belongs to the studentor the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases was used to fool youand make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the correctingand the corrected students have reviewed the possible scripts used to facilitate the grading.

- If the correcting student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.

Attachments

☐ Subject (https://cdn.intra.42.fr/pdf/pdf/683/get_next_line.ro.pdf)

☐ Subject (https://cdn.intra.42.fr/pdf/pdf/661/get_next_line.fr.pdf)

☐ Subject (https://cdn.intra.42.fr/pdf/pdf/776/get_next_line.en.pdf)

Preliminaries

Basic conditions

The following conditions must be met

- Presence of the files get_next_line.c and get_next_line.h

- get_next_line.h contains the prototype of the get_next_line function and the macro that defines the number of characters read simultaneously on the filedescriptor. We will call it "BUFF_SIZE" (but you can call it as you wish).
If that's not the case, the grading with this scale stops.
You can however continue to discuss the project.

- No forbidden function/library.

- No memory leaks.

☐ Yes ☐ No

Tests

Basic tests

Set BUFF_SIZE to 8, and compile a test program that reads from the standard output using the get_next_line function.

Make at least the following tests

- Read and return a line of 8 characters ending by \n included from the standard output.

- Read and return two lines of 8 characters ending by \n included from the standard output.

- Read and return any number of lines of 8 characters ending by \n included from the standard output.

Add an open argv[1] in you main then

- Read and return a line of 8 characters ending by \n included from a file.

- Read and return two lines of 8 characters ending by \n included from a file.

- Read and return any number of lines of 8 characters ending by \n included from a file.

☐ Yes ☐ No

Middle Tests

- Read and return a line of 16 characters ending by \n included from a file.

- Read and return two lines of 16 characters ending by \n included from a file.

- Read and return any number of lines of 16 characters ending by \n included from a file.

- Read and return a line of 16 characters ending by \n included from the standard output.

- Read and return two lines of 16 characters ending by \n included from the standard output.

- Read and return any number of lines of 16 characters ending by \n included from the standard output.

☐ Yes ☐ No

Advanced Tests

- Read and return a line of 4 characters ending by \n included from a file.

- Read and return two lines of 4 characters ending by \n included from a file.

- Read and return any number of lines of 4 characters ending by \n included from a file.

- Read and return a line of 4 characters ending by \n included from the standard output.

- Read and return two lines of 4 characters ending by \n included from the standard output.

- Read and return any number of lines of 4 characters ending by \n included from the standard output.

- Read and return a line of 4 characters without \n included from a file.

- Read and return a line of 8 characters without \n included from a file.

- Read and return a line of 16 characters without \n included from a file.
(reminder, the end of a file must behave like the end of the line for your get_next_line).

- Read and return an empty line from a file.

☐ Yes ☐ No

Error management

Error management

Carry out AT LEAST the following tests to try to stress the error management

- Pass an arbitrary file descriptor to the get_next_line function on which it is not possible to read, for example 42. The function must return -1.

- Set BUFF_SIZE to 1, 32, 9999 then 10000000. This last value can't work (and can't be considered an error during this defence). Does one of you two know why?

☐ Yes ☐ No

Bonus part

Multi filedescriptor bonus

Only consider this bonus if you replied YES to all the preceding questions.
Carry out the tests to control that this bonus is indeed functional.

☐ Yes ☐ No

Other bonuses

Are there more bonuses? (Such as the use of just one static variable)
Your evaluation will depend on the number of available, useful and functional bonuses. (1 point per bonus).

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

Ratings

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

☐ Ok ☐ Outstanding project

☐ Empty work ☐ Incomplete work ☐ No author file

☐ Invalid compilation ☐ Norme ☐ Cheat ☐ Crash

☐ Leaks

Conclusion

Leave a comment on this evaluation

Finish evaluation