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

SCALE FOR PROJECT FT_LS (/PROJECTS/FT_LS)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere-v2.hive.fi:vogsphere/intra-uuid-50c096f5-953c    ⎘
```

Introduction

In order to maintain high evaluation standards, you are expected to:

- Remain polite, courteous, respectful and constructive at every moment
of the discussion. Trust between you and our community depends on
your behaviour.

- Highlight the flaws and issues you uncover in the turned-in
work to the evaluated student or team, and take the time to discuss
every aspect extensively.

- Please take into account that discrepancies regarding the expected
work or functionalities definitions might occur. Keep an open
mind towards the opposite party (is he or she right or wrong?),
and grade as honestly as possible. 42's pedagogy only makes
sense if peer-evaluations are carried out seriously.

Guidelines

- You must grade only what exists in the GiT repository of the
student or team.

- Remember to check the GiT repository's ownership: is it the
student's or team's repository, and for the right project?

- Check thoroughly that no wicked aliases have been used to trick
you into grading something other than the genuine repository.

- Any script supposed to ease the evaluation provided by one party
must be thoroughly checked be the other party in order to avoid unpleasant
situations.

- If the evaluating student hasn't done the project yet, it is mandatory
that he or she reads it before starting the evaluation.

- Use the available flags on this scale to tag an empty work, a non
functional work, a coding style ("norm") error if applicable,
cheating, and so on. If a flag is set, the grade is 0 (or -42 in
case of cheating). However, cheating case excluded, you are encouraged
to carry on discussing what went wrong, why, and how to address it,
even if the grading itself is over.

Attachments

⬜ Subject (https://cdn.intra.42.fr/pdf/pdf/6604/ft_ls.en.pdf)

⬜ Get started (https://cdn.intra.42.fr/pdf/pdf/6605/get_started.en.pdf)

Preliminaries

Basic requirements

Verify the following points :

- The author file.
- Your project must respect the Norm.
- There's no forbidden functions in the reviewed source code.
- The Makefile must be present and must contain the following rules : all,
clean, fclean and re. They must be fully functionnal and correctly
implemented.
- No memory leaks.

If one of those points is not respected, this evaluation is over. Set the
correct flag, and finish the evaluation.

Nonetheless, you can continue it, for pedagogical purpose, but no points
should be counted.

⊘ Yes   ✕ No

Tests

Basic Tests

As explained in the subject, an identical display is expected,
except for ACL, extended attributes, and row formatting
display (considered as bonuses). Perform the following tests:

- ls [+1pts]
- ls -a [+1pts]
- ls -l [+2pts]
- Check that symbolic link display is accurate. [+1pts]

We will accept different padding for "-l" option but all
the relevant Information must be displayed. You must first
perform the test with no arguments, then with a file and
finally with a folder.

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

Basic Tests ++

Perform the following tests:
- ls -r [+1pts]
- ls -t [+1pts]
- test ls -r with several files/folders/... as parameter. [+1pts]
- test ls -t with several files/folders/... as parameter. [+1pts]
- test setuid/setgid/stickybit [+1pts]

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

Middle Tests

Perform the following tests:
- ls -R [+2pts]
- multiple option management: parsing and form ("-l -t" for en. as well as "-lt") [+2pts]
- multiple option management: accurate display. [+1pts]

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

Error Management

- nonexistant file/folder [+1pts]
- inaccessible file/folder [+1pts]
- unmanaged or nonexistant option [+3pts]

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

◯

Bonus

This part should be graded only if the mandatory part is complete and flawless.

ACL/Attributes bonus

Is it implemented ?

✓ Yes   ✕ No

Row formatting display bonus

Is it implemented ?

✓ Yes   ✕ No

Speed bonus

Assess the speed of your ls compared to the system's. Time
the result of ls -lR / using the command time.
$> time ls -lR /
- 0 : very slow
- 5 : as fast

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

◯

Options Bonus

Assess the number of additional options that are 100% functional
- 1 point for each additionnal option

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   🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

| |
| |

Finish evaluation