SCALE FOR PROJECT LITTLE-PENGUIN-1 (/PROJECTS/LITTLE-PENGUIN-1)

Introduction

To ensure this evaluation goes smoothly, please respect the following set of rules :

- Please remain courteous, polite, respectful and constructive at all times during this exchange. The trust bond between the school's community and yourself depends on it.

- Should you notice any malfunctions within the submitted project, make sure you take the time to discuss those with the student (or group of students) being graded.

- Keep in mind that some subjects can be interpreted differently. If you come accross a situation where the student you're grading has interpreted the subject differently than you, try and judge fairly whether their interpretation is acceptable or not, and grade them accordingly. Our peer-evaluation system can only work if you both take it seriously.

Guidelines

- You may only evaluate whatever is in the GiT submission directory of the student you are grading.

- Make sure to check wether the GiT submission directory belongs to the student (or group) you're grading, and that it's the right project.

- Make sure no mischievous aliases have been used to trick you into correcting something that is not actually in the official submitted directory.

- Any script created to make this evaluation session easier - whether it was produced by you or the student being graded - must be checked rigorously in order to avoid bad surprises.

- If the student who is grading this project hasn't done the project him/herself yet, he/she must read the whole topic before starting the evaluation session.

- Use the flags available to you on this scale in order to report a submission directory that is empty, non-functional, that contains a norm errors or a case of cheating, etc...
In this case, the evaluation session ends and the final grade is 0 (or -42, in case of cheating). However, unless the student has cheated, we advise you to go through the project together in order for the two (or more) of you to identify the problems that may have led for this project to fail, and avoid repeating those mistakes for future projects.

- This evaluation must be made in the student's home-made linux distribution.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/3956/little_penguin_1.en.pdf)

Assignment 00

Kernel Origin

Is the kernel from git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git ?
Check the log file too see if it match.

Yes No

Configuration

Check the CONFIG_LOCALVERSION_AUTO in the config file. It must be enabled.

Yes No

Assignment 01

Hard stuff

Compile, insert and remove the module. Check the kernel log too see if the module is correct. (cf Subject)

☑ Yes ☒ No

Assignment 02

Hard stuff

Compile, insert and remove the module. Check the kernel log too see if the module is correct. (cf Subject)

☑ Yes ☒ No

Assignment 03

Thor, god of thunder

Check the log for the version line. Is thor_kernel in ?

☑ Yes ☒ No

We love patches

Check the patch in the student's git. Is it standard & working ? (cf Subject)
If you don't know the Linux standards, ask the student. He'll have to say 'My life is so much better now' and 'Windoz suckx' in his speech.

☑ Yes ☒ No

Assignment 04

Love pedantic

Check the coding style of the file. (There is a script for that.)
/usr/src/linux-4.x/scripts/checkpath.pl file.c -no-tree -file --strict
Check the debugs level too (info must remain info, debug must remain debug.)

☑ Yes ☒ No

Assignment 05

A keyboard ? What for ?

Ask the student to explain his rule_file, how it's working. The student must do a demo of his module.
Is it working great ?

☑ Yes ☒ No

Assignment 06

A real kernel developer

Is the misc_device /dev/fortytwo created on insertion and deleted when the module is removed ?

☑ Yes ☒ No

readFTW

Do the following tests:
- cat /dev/fortytwo must return the student id
- echo 123 > /dev/fortytwo must return an error 'Invalid argument'
- echo student_id > /dev/fortytwo must work.

☑ Yes ☒ No

Assignment 07

Terminator

Is the kernel a linux-next one ? Ask the student what's the difference between the mainline and a linux-next kernel. It's not that hard. At all.

☑ Yes ☒ No

Assignment 08

Creation and Deletion

Is the directory /sys/kernel/debug/fortytwo created on insertion and deleted on removal ?
Are the files foo, id and jiffies in it ?

☑ Yes ☒ No

Id

The id file should have the same behavior as Assignment 05. Test it !

☑ Yes ☒ No

Jiffies

Test the write on the file, should come with an error.
Test the read, should return the numbers of tick since the system booted.

☑ Yes ☒ No

Fly, you foo

Test the foo file:
- A non-root user should not write on it

- A root user should write on it, and the value must be safed.
- When read by any user, the value is returned.
- Test the locking. (What happens when you read from it while someone's writing to it ?)

Yes No

Assignment 09

Coding style, again ?!

Check the coding style of the student code.
Same as ex03.

Yes No

Indiana Jones Theme

Check the behavior of the student code.
The misc file should read a value, and the reversed value should be returned on read.
(123 > 321)

Yes No

Assignment 10

Almost real

Check the behavior of student module.
- The module create a proc entry (/proc/mymounts)
- The module list mount name with mount point on read.

Yes No

Ratings

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

⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Preview!!!