

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

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the bene fits of the entire community.

SCALE FOR PROJECT FAMINE (/PROJECTS/42CURSUS-FAMINE)

You should evaluate 1 student in this team

Git repository



Introduction

For the smooth running of this evaluation, please respect the following rules:


- Remain polite, kind, respectful and constructive whatever happens during this conversation. It's a matter of confidence between you and the 42 community.
- Highlight the potential problems you 've had with the work you're presented to the person or the group you're grading, and take the time to talk about and discuss those issues.
- Accept the fact that the exam subject or required functions might lead to different interpretations. Listen to your discussion partner's perspective with an open mind (are they right or wrong ?) and grade them as fairly as possible.
42's teaching methods can make sense only if peer-evaluation is taken seriously.

Guidelines

- You must only evaluate what you will find in the student's or group's GiT repository.
- Take the time to check that the GiT repository matches the student or group and the project.
- Double check that no malicious alias was used to mislead you and make you grade something different from the official repository content.

- If a script supposed to help evaluate the exam is supplied by either side, the other side will have to strictly check it to avoid nasty surprises.
- If the evaluating student has not yet taken this project, they will have to read the exam subject in its entirety before starting the evaluation.
- Use the flags available on this grading system to signal an empty or non- functional project, a norm flaw, cheating, etc. In that case, evaluation stops and final grade is 0 (or -42 if it's a cheating problem). However, if it's not a cheating problem, you are invited to keep talking about the work that has been done (or not done, as a matter of fact) in order to identify the issues that lead to this stalemate and avoid it next time.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/13234/en.subject.pdf>)

Famine source

Foreword

Foreword

Before evaluation begins, make sure:

- The project is in C or assembler.
- A Makefile containing the usual rules is included.
- A VM running on the OS chosen by the assessed student is available for the evaluation.
- Root rights for this machine are available.

Also make sure the project doesn't use libraries that might impair the educational value of the project. In other words, the library(ies) shouldn't be able to make coffee by itself (or themselves).

If one condition is not observed, evaluation stops.

 Yes

 No

Using Famine

Famine - Compilation and execution

Launch a make and execute the virus once. No error should be visible.

If one condition is not observed, evaluation stops.

☐ Yes

☐ No

Attempting to use Famine

Create two temporary folders named test and test 2 in the temporary folder of the VM. Compile the sample.c program provided in the subject in 64 bits as indicated in the subject. Sample file works correctly without displaying errors at launch.

Copy the sample binary in the folder you've just created. Now, simply launch the virus. No error should be visible.

If one condition is not observed, evaluation stops.

☐ Yes

☐ No

Virus infection

Launch a simple 'strings' command on the sample located in the folder you have just created. It must show a signature containing both logins. The sample file works correctly without displaying errors at launch. If the login of one of the group members is not clearly visible, evaluation stops.

Launch the virus again and look through your 'strings' command to see if your sample doesn't contain the login you previously found again. If it does, evaluation stops.

Sample file works properly without displaying any error at launch. If one condition is not observed, evaluation stops.

☐ Yes

☐ No

Virus infec-ception

Now, copy a 64 bits binary of your choice in the test2 folder located in the temporary folder of the OS.

Then, launch an infected sample in the test folder of the temporary folder of the OS. No error should be visible.

Use your 'strings' command to make sure your binary located in the test2 folder contains the login you have found previously. If it does not, the evaluation stops.

Finally, launch the infected binary located in the test2 folder and make sure it executes without any error. It must behave EXACTLY like it behaved before infection.

If one condition is not observed, evaluation stops.

☐ Yes

☐ No

Crash dat virus!

Basic checking

Try to use the virus while the read/write permissions are not available in the temporary folder.

Try to use the virus on files you don't own the read/write permissions of (put them in the test/test2 folders).

The virus should never display any error message.

☐ Yes

☐ No

Free checking

Assessor is free to try to make the virus crash.

In case of unexpected quitting, the virus should quit properly without displaying any error message on the computer. No error message should be displayed on the computer.

☐ Yes

☐ No

Virus bonus

Bonus will be taken into account only if the mandatory part is PERFECT. PERFECT meaning it is completed, that its behavior cannot be faulted, even because of the slightest mistake, improper use, etc... Practically, it means that if the mandatory part is not validated, none of the bonus will be taken in consideration.

Specific bonuses

You can record up to 5 different bonuses. A bonus should be at least slightly useful and implemented properly.

Here are bonus exemples you can grade:

- Infection on 32 bits binary.
- Useful functions. - Packing use.

- Infection optimisation.
- Infection on a non-binary file.
- ...




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

 No author file


 Invalid compilation

 Norme

 Cheat

 Crash

 Incomplete group

 Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

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

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

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

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

Rules of procedure (<https://signin.intra.42.fr/legal/terms/4>)

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