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

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

SCALE FOR PROJECT MATT-DAEMON (/PROJECTS/42CURSUS-MATT-DAEMON)

You should evaluate 2 students in this team

★

Git repository

```
git@vogsphere-v2.1337.ma:vogsphere/intra-uuid-871e8c1e-07d6
```

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-funcional 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.

- Evaluation must be made with a virtual machine running Linux with a core > 3.14 with root rights.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/60873/en.subject.pdf)

Matt_daemon source

Foreword

Preleminary instructions

Before evaluation starts, make sure that:

- the project is in C++,
- the project uses the functions fork, signal and chdir,
- the LOCK_SH flag is not visible in the project,
- the LOCK_EX flags as well as LOCK_UN are present in the project,
- a Tintin_reporter class is present,
- classes have the form of Coplien,
- a Makefile containing the usual rules is present.

Make sure that the project doesn't use any library that could impair the eductational value of the project. If it does, check "Cheat" on top of the grading scale.

If one of the conditions has not been met, evaluation stops.

✓ Yes ✕ No

Using Matt_daemon

Running daemon

Before running the program, run the following command: `rm

-f /var/lock/matt_daemon.lock ; rm -rf /var/log/matt_daemon/`.

Run the program as the root user.

The "netstat -tulpn" command will allow you to check that the 4242 port
is in LISTEN mode.

The /var/lock/matt_daemon.lock and /var/log/matt_daemon/matt_daemon.log
files are available.

✓ Yes ✕ No

Daemon quitting

Look for the daemon's PID with the `ps aux | grep matt_daemon` command,
and make a kill -15 (daemon's PID).

The /var/lock/daemon.lock file must not be available.

If one of the conditions has not been met, evaluation stops.

✓ Yes ✕ No

Using Matt_daemon

Using Matt_daemon

Before running the program, run the following command: `rm

-f /var/lock/matt_daemon.lock ; rm -rf /var/log/matt_daemon/`.

Run the program with the root rights.

Use the netcat command to connect to the daemon ("nc localhost 4242").

Type a word in the command prompt and enter (don't write "quit").

End the session typing "quit" in the command prompt.

Program must quit correctly and the /var/lock/matt_daemon.lock file must
not be present anymore.

✓ Yes ✕ No

Matt_daemon log

Matt_daemon simple log

The /var/log/matt_daemon/matt_daemon.log file is present. The informations are clearly displayed. Launch and quit of the daemon also appear clearly. Instruction typed in the command prompt appear clearly. Timestamp appears in the form requested by the subject. A message indicates the call when the daemon quits with the quit command.

✓ Yes ✕ No

Matt_daemon advanced logging

Run the daemon again and kill it with the signal you like.
If a signal is not accurately managed, the evaluated student must be able to justify it. The daemon quits well with the deleting of the /var/lock/matt_daemon.lock file. The signal is visible in our log /var/log/matt_daemon/matt_daemon.log file.

✓ Yes ✕ No

Matt_daemon error management

Matt_daemon error basique

Before running the program, launch the following command `rm -f /var/lock/matt_daemon.lock ; rm -rf /var/log/matt_daemon/` .
Try to run the daemon in root twice. An explicit error message appears and stops the second daemon from running. A proper message is visible in the /var/log/matt_daemon/matt_daemon.log file with an explanation of the problem.

✓ Yes ✕ No

Matt_daemon error avancée

Before running the program, launch the following command `rm -f /var/lock/matt_daemon.lock ; rm -rf /var/log/matt_daemon/` .
Run the daemon in root. Assessor can try to crash the program (sans SIGKILL).
If the program quits unexpectedly, the daemon must quit properly and a message must be saved in the log file.

✓ Yes ✕ No

Matt_daemon socket

Matt_daemon

Run the daemon in root. Try to connect more than 3 clients simultaneously.
An error message must appear to show the problem.

✓ Yes ✗ No

Matt_daemon

Run the daemon in root. Try to connect more than 3 clients simultaneously.
Quit the client and run again. The client must be able to connect simply,
without any error. Properly close all clients as well as the daemon.

✓ Yes ✗ No

Matt_daemon bonus

Bonus will be taken into account only if ALL the previous points are valid.

Bonus specific to this project

For this project, you can take into account up to 5 different bonuses.
They should be at least slightly useful and implemented correctly.
Here is a list of acceptable examples:

- Graphic client.
- Utilitary functions.
- Logs archival.
- Mail sending according to filters.
- Authentification system.
- Using advanced encryption system.
- ...

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

Rules of procedure (https://profile.intra.42.fr/legal/terms/4)
Declaration on the use of cookies (https://profile.intra.42.fr/legal/terms/2)
Privacy policy (https://profile.intra.42.fr/legal/terms/5)
General term of use of the site (https://profile.intra.42.fr/legal/terms/6)
Terms of use for video surveillance (https://profile.intra.42.fr/legal/terms/1)
Legal notices (https://profile.intra.42.fr/legal/terms/3)