Assignments

Free/Libre and Open Source Software (FLOSS)

194.114
TISS
2022W
TU Wien

Issued:

24. September 2022

This document specifies the homework (HX), teamwork (TX) and project (PX) of the course Free/Libre and Open Source Software (FLOSS). References to meetings are done via MXX. All deadlines are summarized in a separate document called "schedule.pdf", available in TUWEL.

1 General Hints

- This document contains all assignments of the course 194.114. While it might look much at first sight, students appreciate that they know everything they need to do for the whole course in advance.
- Please always feel free to ask questions. Start asking questions early, long before the deadline. At following places we will give answers:

  TUWEL forum: for any questions related to the course

  issue tracker: for any questions related to Elektra. The lecturer's account is `@markus2330`, the tutor's account is `@flo91`. `https://issues.libelektra.org`

  email: lecturer markus.raab@complang.tuwien.ac.at and tutor: florian.lindner@student.tuwien.ac.at
- Pull-requests (PRs) for home- and teamwork are created in the public Elektra repository `https://git.libelektra.org`.
- As usual in FLOSS, and also in Elektra, some parts are already mature, others are experimental. It can happen, that you find a problem that hinders you in your originally planned task. If this happens, please report the problem and work on this hindering problem instead. You will not have any disadvantage because of this. Instead it is assumed that you worked carefully and get extra points for reporting previously unknown issues.

  Up to 3 extra points.
- Following programming languages can be used for home- or teamwork: C, C++, Java, Python, Rust (Kotlin, Lua, Ruby, Go).
  Note: For the languages given in brackets we cannot give as much help, we assume that you are proficient in the language yourself.
- The title of every issue and PR must start with "[FLOSS XY]", where X is the initial letter and Y is the number of the project, team- or homework.
- To be fully exposed to the FLOSS world, please install some FLOSS operating system like GNU/Linux directly on your computer. This can be done in parallel to another operating system and only requires some disk space. If you have troubles installing GNU/Linux, immediately ask for help in the TUWEL forum.
- Make sure that you get yourself a comfortable way to develop with Elektra early. If you have troubles installing Elektra, immediately ask for help on `https://issues.libelektra.org`.

- Please always describe who in your team did what in the PR description or submission.
- Recommended team size is two, three is allowed if not possible otherwise. The teamwork must be done in a team. The project can be done alone or in a team. The team for teamwork and project does not have to be the same.
- Quality is preferred over quantity. You can get full points even if the assignment is only partially, e.g., missing features, but properly fulfilled. I.e., always complete PRs fully according to Definition of Done below, so that they can be merged and no work gets lost.
- There are up to three deadlines:

  The main deadline for your submission — PRs must fulfill the Definition of Done below — then

  a review deadline for reviewing the submissions of others, and

  a correction deadline for correction of your submission based on the reviews you received. Here again Definition of Done must be fulfilled, e.g., the continuous integration must pass.

Definition of Done

The PR template of the libelektra repository defines what needs to be done for every PR in the public repository, and as such for most of the assignments. The template will be automatically shown when you create a PR. As reference the text is repeated below:

- Short descriptions of your changes are in the release notes (added as entry in `doc/news/_preparation_next_release.md` which contains `_(my name)_`).
- Details of what you changed are in commit messages (first line should have `module: short statement` syntax).
- References to issues, e.g. `close #X` are in the commit messages.
- The PR is rebased with current master.
- The continuous integration passes the build. If not, fix in this order:
  - add a line in `doc/news/_preparation_next_release.md`
  - reformat the code with `scripts/dev/reformat-all`
  - make all unit tests pass
  - fix all memleaks

If you have any troubles fulfilling these criteria, please write about the troubles as comment in the PR. We will help you.

Furthermore sometimes following points need to be fulfilled:

- You added unit tests for your code.

- You fully described what your PR does in the documentation (and not in the PR description).
- You fixed all affected documentation.
- Code is conforming to our Coding Guidelines. E.g., you added code comments, logging, and assertions as appropriate.
- You updated all meta data, e.g. README.md of plugins.
- You mentioned every code not directly written by you in reuse.
- Documentation is introductory, concise, good to read and describes everything what the PR does.
- Examples are well chosen and understandable.
- You added the "work in progress" label if you do not want the PR to be reviewed yet.
- You added the "ready to merge" label if the Definition of Done is fulfilled and no further pushes are planned by you.

Project Variants

For the project, you can choose one of the following variants:

New Initiative: Create a new FLOSS initiative. Please only take the option if you have an innovative new idea which can be realized within one term. Otherwise prefer joining an existing initiative. You will be fully responsible for where the issues, the source code etc. live. This is probably the most time-consuming variant.

Existing Initiative: Continue or start maintaining an initiative or a module of some initiative. It is fine, if the FLOSS (module) is already written or maintained by you. You are encouraged to improve badly maintained software, e.g., fix something which is not working for you. It depends on the initiative in which ways you can help.

The only requirements are that the FLOSS must:

- be publicly available, and
- be fully licensed using FLOSS licenses.

For example, following initiatives have many plugins suitable to be worked on as your project:

- `https://www.jenkins.io` in Java.
- `https://www.nextcloud.com` in PHP.

More (specific) examples will be given in meetings and TUWEL.

Elektra: Continue working on Elektra, where you also do your home- and teamwork. This is probably the easiest variant, as you do not need to acquaint yourself with several initiatives. You can help Elektra by fixing an issue that is too big for home- or teamwork. Examples for such issues are tagged with `project2022W`.

H0 Registration

Deadline:
12.10.2022, 23:59

Estimated Effort:
1h

Points:
1

Tasks:
- Registration in TISS
- TUWEL Forum Entries
- Upload StudentID
- Create and Close Issue

Topic:
M00: 05.10.2022, 14:00

- Registration in TISS.
- Write a post in the TUWEL discussion forum, including:
  - your preference of programming languages to be used: C, C++, Java, Python, Rust (Kotlin, Lua, Ruby, Go).
  - which variant of the project you consider.
  - your GitHub account for `https://issues.libelektra.org`.
- Reply to at least one other post in the TUWEL forum to find a team partner for teamwork and project.
- Once you found a team partner, reply to your own post stating who your team partner for both teamwork and project is. Different partners for teamwork and project are allowed.
- Upload the StudentID in TUWEL.
- Create a new issue in `https://issues.libelektra.org`. For example, this can be a question or a vision about how a configuration system should be. The title of the issue must start with "[FLOSS H0]".
- Close the issue once you got an answer.

If you complete the above points correctly, you are successfully registered in the course and you will receive your first point in TUWEL. This means you will also get a certificate at the end of the term.

H1 Bug Triage

Deadline:
19.10.2022, 23:59

Estimated Effort:
4h

Points:
6

Tasks:
- Create an issue
- Triage for 5 issues

Topic:
M01: 12.10.2022, 14:00

Start to install and use Elektra. Create a new issue in `https://issues.libelektra.org` about your first impression and steps with Elektra. Write about what you understood and what is unclear. Answer the question: Which information was missing for you and where did you look for it? This issue will be later used in T3.

Then for at least five issues, do a bug triage:

- Assign the issue to you.
- Change the title so that it starts with "[FLOSS H1]".
- Reproduce the issue and write about your outcome.
- If the issue actually describes several problems, create new issues accordingly.
- Use more appropriate tags.
- Give more detailed instructions of how to reproduce the issue.
- Write in the issue where in the source code the problem is.

If anything is unclear, in particular about the last three points, please ask in the issue. It is very likely that someone can help you.

Suitable issues are labelled as `floss2022W`.

H2 Pull Requests: Freestyle

Create five PRs using your fork with five branches, each of them with an added line in the release notes `doc/news/_preparation_next_release.md`, stating what the intend of your change was. Like always, write "[FLOSS H2]" in the title of the PRs, e.g. "[FLOSS H2] Fix typo".
For each of the PRs either:

- Fix TODO markers in the source code.
- Improve the readability of some sentence you found hard to understand.
- Add code comments, assertions, log statements or inline comments, see `doc/CODING.md`.
- Write a new example of how to use Elektra in `examples` or in the appropriate bindings folder, e.g. `src/bindings/swig/ruby/examples`.

Create three reviews of other H2 pull-requests. In the review make sure the items of the PR template are fulfilled and everything you did in the review is written in the main comment of your review.

To get all points, you need to rebase the PRs on merge conflicts, the continuous integration must pass and the reviews must be accounted for.

Note: Five PRs are needed if the PRs are small (e.g. fix only a few TODO or spellings). Less PRs might be enough if they solve non-trivial problems. Quality is preferred over quantity.

Hint: Start asking questions early, long before the deadline. Ask questions in new issues or directly in the PR if they are related to the code.

Deadline:
09.11.2022, 23:59

Reviews until:
16.11.2022, 23:59

Corrections until:
23.11.2022, 23:59

Estimated Effort:
5

Points:
10

Tasks:
- Create 5 PRs
- Review 3 PRs

Topics:
- M02: 19.10.2022, 14:00
- M03: 09.11.2022, 14:00

For corrections:
- M04: 23.11.2022, 14:00

H3 Pull Requests: Fix Issues

- Submit PRs to fix each issue you are assigned to.
- Write "[FLOSS H3]" in the title of the PRs.

Hint: Create a PR as early as possible to get feedback. Create new issues and new PRs as needed.

Furthermore, create at least three reviews of other's H3 PRs.
For the reviews, make sure that:

- The review is helpful for the author of the PR.
- The items of the PR template are fulfilled.
- The PR actually solves the underlying issue.
- The CI executes the tests.
- Everything you checked for (even if it was okay) is written in the main comment of your review.

Hint: Please always feel free to ask questions in the issue (if it is related to the issue), in the PR (if it is related to the code). Start asking questions early, long before the deadlines.

Deadline:
30.11.2022, 23:59

Reviews until:
07.12.2022, 23:59

Corrections until:
14.12.2022, 23:59

Estimated Effort:
5h

Points:
13

Tasks:
- Create 5 PRs
- Review 3 PRs

Topics:
- M01: 12.10.2022, 14:00
- M02: 19.10.2022, 14:00
- M03: 09.11.2022, 14:00
- M04: 23.11.2022, 14:00

T1 Continuous Integration

Create at least two PRs in which your team:

1. Writes new tests, e.g., in `tests/ctest` or in the appropriate bindings folder, which demonstrate that an, ideally new, bug is now fixed. To find new bugs, you can either:
   - Check documentation and tests for consistency.
   - Run fuzzy tester, e.g., AFL.
   - Look at output of sanitizers, e.g., ASAN.
   - Test code that had no coverage before, see `https://doc.libelektra.org/coverage/master/debian-bullseye-full/`.
2. Fixes and includes disabled tests, either as reported in issues, marked as `DISABLED_` or as documented in `doc/todo/TESTING`.

Write “[FLOSS T1]” in the title of the PRs.
Document in the PR description:

- How you found the bugs.
- On which CI systems the tests run now.
- Who did which parts of this task.

Furthermore, create at least two reviews of T1 PRs.
For the reviews, make sure that:

- The review is helpful for the author of the PR.
- The items of the PR template are fulfilled.
- The PR actually solves the underlying issue.
- The CI executes the tests.
- Everything you checked for (even if it was okay) is written in the main comment of your review.

Deadline:
23.11.2022, 23:59

Reviews until:
30.11.2022, 23:59

Corrections until:
07.12.2022, 23:59

Estimated Effort:
5h

Points:
10

Tasks:
- Create 2 PRs
- Review 2 PRs

Topic:
M04: 23.11.2022, 14:00

T2 Documentation

Create at least three PRs in which your team:

- Automatizes the validation of one tutorial.
- Enhances or creates a tutorial in `doc/tutorials` which uses automatized validation.
- Improves the generated documentation of the man pages or API `https://doc.libelektra.org/api`.

Write “[FLOSS T2]” in the title of the PRs.
Document in the PR description:

- On which CI systems the tests run.
- Who did which parts of this task.

Furthermore, create at least three reviews of T2 PRs.
For the reviews, make sure that:

- The documentation is clear and understandable for you.
- The CI executes the validation of tutorials.
- Everything you checked for (even if it was okay) is written in the main comment of your review.

Deadline:
07.12.2022, 23:59

Reviews until:
14.12.2022, 23:59

Corrections until:
21.12.2022, 23:59

Estimated Effort:
5h

Points:
10

Tasks:
- Create 3 PRs
- Review 3 PRs

Topic:
M05: 30.11.2022, 14:00

T3 Remove Entry Barriers

Create at least two PRs, in which your team removes at least two entry barriers. These can be either an entry barrier

- as reported in H1,
- you stumbled across later, or
- someone else reported.

Write “[FLOSS T3]” in the title of the PRs.
Furthermore, create at least two reviews of T3 PRs.
For the reviews, make sure that:

- The PR reduces an entry barrier.
- Everything you checked for (even if it was okay) is written in the main comment of your review.

Deadline:
14.12.2022, 23:59

Reviews until:
21.12.2022, 23:59

Corrections until:
11.01.2023, 23:59

Estimated Effort:
5h

Points:
10

Tasks:
- Create 2 PRs
- Review 2 PRs

Topic:
M06: 07.12.2022, 14:00

P0 Start

Choose a topic that you find yourself interesting. E.g. for Elektra:

- Something you are missing and you want for yourself.
- A missing feature reported in the issue tracker, e.g., tagged with `project2022W`.
- Something you find interesting in `doc/todo`.

Submit a short — not more than three pages long — document in the P0 workshop in TUWEL that includes:

- A proposal what you plan to do as a project:
  - What problem you want to solve.
  - Sketch how you want to implement it, e.g., architecture and which language you want to use.
- Which source code management tool/issue tracker you want to use.
- A research of similar or alternative initiatives/modules/plugins.
- Who will do which part of the project.

The document should be exclusively about your own project.

If you work on Elektra, create a PR with user-facing documentation. For example, if you create a new tool, start by writing the man page of that tool.

If you do not work on Elektra, announce your project in the TUWEL forum. Create user-facing documentation in the source code repository you use.

Link to the forum entry, issue and/or PR in the document.

Until the feedback deadline, discussions about scope take place, either in the PR or TUWEL forum.

Furthermore, create at least two reviews in TUWEL and, if the reviewed person improves Elektra, the accompanied PR.

Hint: It makes sense to already start critical parts of the implementation. If you do so, please include that in the PR, even if it does not compile.

Deadline:

02.11.2022, 23:59

Reviews until:

09.11.2022, 23:59

Feedback:

16.11.2022, 23:59

Estimated Effort:

3h

Points:

5

Tasks:

- Submit PR or
- TUWEL Forum Entry
- Submit Document
- Review PRs/Documents

Topic:

M01: 12.10.2022, 14:00

P1 Midterm

The goal of P1 is to have a working, but probably minimal, implementation of the proposal submitted in P0. The code should be of high quality with unit tests and documentation.

Furthermore, explain in the extended document, which is not longer than five pages in total:

- The content as requested in P0 — shortened and improved.
- How you document your project.
- How you keep the entry barriers low and how you attract new developers in your project. Give an explanation of how new developers can join. Maybe one of your reviewers gets interested.
- Who did which parts of the project and document.
- Include FLOSS literature research with at least five citations.

Submit this document and again do reviews of two other documents via the P1 workshop in TUWEL.

Deadline:
21.12.2022, 23:59

Reviews until:
11.01.2023, 23:59

Estimated Effort:
8h

Points:
15

Tasks:
- Submit PR
- Submit Document
- Review PRs/Documents

Topics:
- M05: 30.11.2022, 14:00
- M06: 07.12.2022, 14:00
- M07: 14.12.2022, 14:00

P2 Presentation

Workflow:

- Submit your presentation before 11.01.2023, 23:59 in TUWEL.
- Before 18.01.2023, 23:59, make sure that you correct and upload the latest slides.
- The presentation can be on any week but must be at latest on 25.01.2023. If your presentation is earlier, make sure that you upload your slides earlier accordingly.

Content of the presentation:

- The presentation can be about anything related to FLOSS, e.g.:
  - some FLOSS tool you played around with
  - FLOSS development you did
  - about the project
- The presentation must be about your experience.
- Team presentations are allowed.
- The duration must be less than 10 minutes per person.
- Like done in the lecture, you can choose to produce a video of your presentation upfront.
- You get extra points for creative, unconventional presentations, e.g., when encouraging active participation.

Deadline:
11.01.2023, 23:59

Corrections until:
18.01.2023, 23:59

Presentations:
25.01.2023, 14:00

Estimated Effort:
2h

Points:
0

Tasks:
- Submit Presentation
- Hold Presentation

Up to 3 extra points.

P3 Finish

Deadline:

18.01.2023, 23:59

Estimated Effort:

7h

Points:

20

Tasks:

- Finish Implementation
- Testing
- Finish Documentation
- Get all PRs merged
- Submit decision PR
- Submit Document

Topics:

- M08: 21.12.2022, 14:00
- M09: 11.01.2023, 14:00

The main goal here is to finish the project and get all left-over PRs finalized and merged. The work done for unmerged PRs will be lost, which would be a pity we definitely want to avoid.

Write or improve at least one architectural decision as introduced in M09 (11.01.2023, 14:00). If you are working on Elektra, use the template in `doc/decisions/template.md` and submit an extra PR for the decision.

Finalize and submit the document — but it is still in total not more than five pages — in a subsumed style:

- The content of P0 and P1 — shortened and improved — with extended details about which software architecture you actually used.
- Describe which tests should be done for a release. It is expected that the code passed these tests.
- How the maintenance or your project works (e.g., releases, packages).
- Describe the directory structure you use.
- Who did which parts of the project and document.

Link to the respective parts in the repository if (further) explanations are available there. E.g., if you described the directory structure in a README.md, it is enough to link to the file, no replication of the content is needed.

You get extra points for helping other people to get their PRs merged. These points also substitute points missing in P3.

Up to 10 extra points.

This time no review is necessary.