Relatório da Entrega 6 de Estudos Avançados em Sistemas de Software (05/06/2025)

Nome: David Tadokoro
NUSP: 10300507
Orientador: Paulo Meirelles

1) Contexto

No primeiro semestre de 2024, houve o oferecimento da disciplina de *Desenvolvimento em Software Livre* (MAC0470/5856), que envolveu mentorar os alunos no ecossistema do Linux, que inclui o próprio projeto do kernel Linux, ferramentas que suportam o desenvolvimento, distribuições GNU/Linux, entre outras. O programa da disciplina foi dividido em três fases, onde os alunos imergiram de forma prática em cada uma das camadas do ecossistema.

O projeto proposto para a disciplina envolve a escrita de um artigo científico sobre como treinar novos contribuidores de uma forma eficiente e prática com intuito de prepará-los com as habilidades necessárias para se tornarem reais desenvolvedores do kernel, baseado nas experiências da disciplina. Vale notar que o objetivo deste treinamento é de munir novos contribuidores (com pouca ou nenhuma experiência prévia em software livre) com a base essencial de habilidades e conhecimentos para irem além de serem os chamados *one-time contributors*.

Nesta sexta entrega, o objetivo era fazer uma primeira versão das seções de discussão e conclusão que, assim como os outros entregáveis relacionados a trechos do artigo, certamente irão passar por refinamentos.

Tanto o arquivo fonte em Markdown (`entrega-6/relatorio.md`) deste relatório, quanto os arquivos em Latex puro das seções de discussão e conclusão (`entrega-6/{discussion,conclusion}.tex`) se encontram no **meu repositório de artefatos para a disciplina**.

Disponibilizamos o **documento Latex no overleaf para consulta** e ele deve estar visível sem necessidade de liberação de acesso. Ele está um pouco mais organizado, mas ainda como um "dump" dos artefatos para a escrita do artigo.

2) Seção de Discussão

Discussion

After describing the results of this research work, we present our analysis of them, as well as their implications and limitations.

Key Findings

First of all, having a group of students that, in its majority, had little to no experience in Free Software development - even though most were familiar with the concept - allows us to consider the experience on the course as central (at least impactful) to the results collected at the end of the course.

RQ1 asked about the effective teaching techniques and resources to introduce newcomers to Free Software projects. The results answer RQ1 by showing that most students agreed that close mentorship throughout the course, along with in-loco workshops with tutorials devised by veterans, were essential in their development as Free Software contributors. Some individual reports even showed that, beyond the support of teaching assistants, the help of peers (i.e., fellow students of the course) by pairing or working in groups was effective in learning and contributing, enforcing, even more, the importance of the in-loco workshops.

Results show that starting the experience with contributions to the Linux kernel project was daunting to students. Technical difficulties inherent in the workflows of the Linux kernel, especially the environment setup part, and vague or even misleading tutorial instructions were major hurdles. However, student reports (corroborated by the observations of the teaching assistants and the professor) show that this rough start elevated the students' hard and soft skills, making them continuously more independent as the course progressed.

Beyond that, presenting the diverse Free Software development models from the Linux kernel, an email-based model with higher technical requirements, to Web-based models, which are considered friendlier to new contributors, gave a comprehensive and complete view of the Free Software ecosystem. The course experience made students considerably more confident in furthering their contributions to Free Software projects.

Regarding RQ2, which covered the hard and soft skills necessary for successfully contributing to Free Software projects, the results highlight the importance of the git version control system. The ability to manage a Linux system and use CLI tools, like the ones from the GNU coreutils, is essential for executing tasks related to the workflows of Free Software projects and troubleshooting problems. Students also pointed out hard skills like C programming language or device driver development that are specific to the Linux project, but we argue that every software project, not only Free Software ones, has intricate specifics that compose hard skills requirements. From students' reports, soft skills like good communication are paramount for success in Free Software ecosystems.

Implications

Introducing a group of developers to Free Software projects and communities is a difficult task that demands preparation, dedication, and coordination. Beyond introducing newcomers, equipping them with the necessary hard and soft skills for impactful contributions to these projects requires a more immersive approach.

Through this study, we presented a successful approach to training long-lasting contributors to Free Software projects, especially (but not limited to) the Linux kernel project.

This approach can be consolidated as a four-month course composed of:

1. In-loco workshops with tutorials devised by veteran developers of Free Software projects;
2. Hands-on experience with Free Software projects by sending contributions and participating in the review process;
3. Exposition of projects with different models of development for a comprehensive view of the complete Free Software ecosystem;
4. Monitoring of the progress in activities;
5. Close mentoring with teaching assistants.

Previous knowledge and experience, although desirable, are not requisites for this approach, as hard and soft skills are naturally developed through the course.

Limitations

This study relies on aspects that can limit the triumphant applicability of the presented approach.

The proposed approach was implemented in a university course that included undergraduate and graduate students. Due to this, the objects of study, i.e., the students, had an influenced motivation in terms of engaging in the course experience. On the one hand, students could have opted to use the minimal effort required to complete the course, which negatively impacts the results. On the other hand, students aspiring for good grades could have extra motivation and engagement in the course activities, which would positively impact the results. In any case, we argue that motivation and dedication are crucial for developers aiming to enter any Free Software project.

Another limitation to consider is the number of students who completed the course, which was 24. With more developers, the approach would certainly need more teaching assistants, but it is impossible to affirm to what extent the approach can escalate. With fewer developers, say, five, the approach could be overkill or even introduce overheads in the experience, as the teaching techniques described were envisioned for a group of considerable size.

3) Seção de Conclusão

Conclusion

This study demonstrated that with structured mentorship, hands-on practice, and exposure to real-world workflows, it is possible to effectively prepare newcomers for meaningful contributions to complex Free Software projects like the Linux kernel. The approach adopted - a university course combining tutorials by experienced developers, in-loco workshops, and personalized guidance - successfully demystifies FLOSS development, fostering hard and soft skill growth and building contributor confidence. While the results are promising, further validation is needed to assess scalability beyond academic environments. Nonetheless, this work offers a replicable model to sustainably onboard long-lasting contributors to Free Software ecosystems.

4) Conclusão da Entrega 6

Com esta entrega, as principais seções do artigo já têm uma base e podemos dedicar o resto para refinar o texto. Ainda assim, muito trabalho deverá ser feito em decorrência das necessidades apontadas nas outras entregas (embasamento teórico, fluidez do texto, escopo dos dados/resultados, etc.). Desta forma, como planejado, a entrega de uma primeira versão do artigo completo para a entrega 7 está no rumo certo e de acordo.