Relatório da Entrega 5 de Estudos Avançados em Sistemas de Software (22/05/2025)

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 quinta entrega, o objetivo era fazer uma primeira versão da seção dos resultados que, assim como os outros entregáveis relacionados a trechos do artigo, certamente irá passar por refinamentos.

Tanto o arquivo fonte em Markdown (`entrega-5/relatorio.md`) deste relatório, quanto o arquivo em Latex puro da seção de materiais e métodos (`entrega-5/results.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. De toda forma, a seção está a seguir na íntegra para consulta fácil (note que `\ref{}` não funciona neste caso).

2) Cronograma atualizado

Como aludido na última entrega (4), eu e meu orientador (Paulo) decidimos não incluir o oferecimento da disciplina neste ano de 2025, dado que acabaríamos tendo uma visão fragmentada da experiência muito diferente do que foi feito em 2024. Portanto, para não comprometer o plano de entregar um artigo coeso e

bem redigido como parte da disciplina, optamos por nos ater apenas à experiência de 2024.

Como consequência, adaptamos minimamente o cronograma das entregas para refletir a decisão tomada. Abaixo, temos uma tabela com as entregas 4 até a final, tanto originais quanto as adaptadas.

Entrega	**Original**	**Adaptada**
4 (08/05)	Seção de Materiais e Métodos + Seção de Resultados (2024)	Seção de Materiais e Métodos
5 (22/05)	Seção de Discussão + Seção de Conclusão	Seção de Resultados
6 (05/06)	Análise de Resultados (2025) + Atualizar Seção de Resultados	Seção de Discussão + Seção de Conclusão
7 (26/06)	Adaptar artigo a Resultados (2025)	Primeira Versão do artigo (rascunho)
Final (03/07)	Versão do artigo pronto para a submissão	Versão do artigo pronto para a submissão

3) Seção de Resultados

Results

The student survey and the teaching assistants' and professors' observations provide complementary insights into the students' development throughout the course. Beyond that, the students' blog posts offer fine-grained observations from their perspectives, which contribute to triangulating results that emerged from the other two data sources.

Sections~ to ~ show the results from the survey responses merged with the teaching assistants' and the professor's observations. Section~ brings results that cluster insights detected in students' blog posts.

Prior knowledge and shifting perceptions on Free Software

A significant majority of respondents (70%) indicated they were familiar with the concept of free software before the course. However, only 25% reported having ever contributed to a Free Software project before.

At the end of the course, despite this limited prior experience, 75% of the students agreed that they would be willing to use the experience gained to contribute to Free Software projects. Similarly, 75% felt comfortable contributing to Free Software by the end of the course. These results are consistent with the teaching assistants' observations that students began the course with a somewhat mystified and distant perception of Free Software, viewing it as complex and inaccessible. Over time, however, they came to understand Free Software development as

approachable and realistically within their reach, particularly after gaining hands-on experience.

Linux experience and confidence to contribute

At the start of the course, 70% of students had at least one year of experience using Linux, but only 25% had some experience with Linux from a development standpoint. By the end of the course, 65% agreed that the course had prepared them to contribute to a Linux-related project (a subsystem or sub-project of Linux). Nonetheless, only 40% expressed intent to do so voluntarily or professionally.

Teaching assistants noted that students initially struggled with fundamental command-line commands such as utilities from the *GNU coreutils*, like `cd`, `ls`, and `echo`, as well as overall familiarity with a *Command-Line Interface* (CLI) approach to do software development. They also found it challenging to troubleshoot issues arising during tutorial workshops, such as package discrepancies across distributions. From the perspective of the teaching assistants, these difficulties were, at large, surpassed as the course progressed, and students grew noticeably more independent and confident over time.

The importance of `git`

A core tool used throughout the whole course was `git`, and teaching assistants pointed out that the majority of students were unfamiliar with it, both regarding *Version Control System* (VCS) concepts as well as practical use of the tool and its commands.

Regarding the respondents' self-assessed skills, 85% reported having some (50%) or extensive (35%) experience using *git* locally prior to the course. Additionally, 70% had some (55%) or significant (15%) experience with web-based code hosting platforms like GitHub and GitLab; however, only 55% had some (35%) or substantial (20%) familiarity with pull request workflows on these platforms. Conversely, 80% had no prior experience with email-based contribution workflows, the primary method used in the Linux kernel project. Among 20 respondents, 90% agreed that the course experience improved their *git* skills, and the rest (10%) did not respond. By the course's end, 85% of respondents affirmed that *git* skills were among the most important for success in the course.

Role of mentorship and in-loco workshops

Mentorship and in-person workshops emerged as two of the course's most impactful elements. 90% of respondents agreed that mentorship helped complete assignments, and 85% considered it critical. Similarly, 85% and 80% of the respondents signaled that the class teaching assistants' support and workshops, respectively, were the biggest strengths of the course.

The teaching assistants confirmed that while the early stages of the course required intensive support, students gradually developed autonomy, with some independently finding and contributing to a Free Software project in the final phase with minimal guidance.

Student self-reported experiences in blog posts

The students' blog posts provide a multifaceted insight into the complexity and educational value of contributing to the Linux kernel, related projects, and Free Software in general. This section explores the main themes emerging from their reflections: initial challenges, learning outcomes, collaboration dynamics, and motivation for future contributions.

Navigating initial complexity and technical hurdles

Many students reported a steep learning curve associated with the initial setup and contributions to the Linux kernel. The compilation and deployment of a custom-built kernel and modules were frequently cited as significant technical challenges:

- Tooling and environment setup: Several students struggled to configure virtual machines, manage disk space, or resolve dependency issues during kernel compilation. Problems with outdated branches and unclear tutorial instructions also caused setbacks, as described by *student 2* and *student 4*;
- Debugging and troubleshooting: Errors such as missing macros or package build failures were common. Students emphasized that many problems stemmed from gaps in tutorial coverage or their inexperience, as *student 12* reflected on the need to learn about language and file management intricacies required for kernel compilation;
- Patience and persistence: The troubleshooting process demanded patience, with students often needing to restart tutorials, switch methods, or pair-program to overcome obstacles. The collective effort and peer support were crucial in overcoming frustrating moments, as highlighted by *student 1* and *student 9*.

Growth in skills and familiarity with Free Software workflows

Despite the aforementioned difficulties, the contributions significantly boosted the students' technical proficiency and understanding of open-source development workflows:

- Deepening knowledge of Linux and CLI: Multiple students gained valuable hands-on experience with bash scripting, `git` intermediate and advanced skills, including rebasing and patch management (fundamental to many Free Software workflows), and kernel internals. The exposure to managing large codebases and coordinating with maintainers provided a concrete

understanding of real-world software engineering practices as reported by `student 1`, *student 5*, and *student 6)*;

- Understanding Free Software workflows: The students learned not only the technical aspects but also the social and procedural dimensions of contributing to Free Software: formatting contributions according to code style rules, submitting code for review, incorporating feedback from maintainers, and appreciating the importance of documentation and community engagement as brought up by *student 3*, *student 6*, and *student 7*;
- Confidence and motivation: The successful submission of patches, even with the required iterations and suggested changes, generated a sense of achievement and confidence. Students felt more empowered to explore other Free Software projects and appreciated the accessibility of the ecosystem more than initially expected, as mentioned by *student 5*, *student 6*, and *student 15*.

Collaboration and peer support as key success factors

The reflections underscore the importance of collaborative learning and support networks:

- Learning with peers: In cases where individual progress stalled due to technical difficulties or setup issues, pairing with classmates or relying on group work proved to be effective strategies to continue learning and contributing, as reported by *student 4* and *student 9*;
- Support from teaching assistants and the professor: Guidance from professors and teaching assistants was highlighted as essential to navigating tricky points and clarifying misunderstandings, especially related to updating branches or interpreting error messages, as mentioned by *student 2*, *student 6*, and *student 11*;
- Free Software communities and ecosystems dynamics: Some students expanded their perspectives to consider broader aspects of Free Software projects - such as maintainers' responsiveness and project sustainability - recognizing the value of incremental contributions beyond feature development, as highlighted by *student 3*.

Tutorials improvements

Some blog posts point toward the need for improving the tutorials used as educational resources:

- More comprehensive tutorials: Students noted that while tutorials effectively outlined objectives and commands, they often lacked in-depth troubleshooting guidance and anticipation of common pitfalls. This led to confusion or repeated errors that might be avoidable with better documentation, as brought up by *student 2*, *student 8*, and *student 13*.

4) Conclusão

Apesar desta primeira versão seção de resultados necessitar de bons ajustes para melhorar, tanto a fluidez do texto, quanto a forma de se agrupar e apresentar os resultados (algumas partes estão redundantes, talvez), ela mostra que temos um boa quantidade de dados de boa qualidade e que a base desta seção está bem consolidada. Na próxima entrega (6), iremos finalizar as duas seções remanescentes (discussão e conclusão) e, como mencionado na entrega passada, restará trazer o embasamento teórico (principalmente na introdução e método) e refinar o texto para termos uma primeira versão do artigo na entrega 7.