

**Instituto de Matemática e Estatística
da Universidade de São Paulo**

MAC 0499

Trabalho de Formatura Supervisionado

Proposta de TCC

Um estudo de caso do uso de virtualização de
dispositivos para testes no subsistema IIO do
kernel Linux

Lucas Pires Stankus

Orientador: Prof. Paulo Meirelles

Coorientador: B.Sc. Marcelo Schmitt

26 de maio de 2021

1 Introdução

O kernel Linux é um dos maiores projetos do mundo, possuindo milhões de linhas de código e mais milhares de novas adicionadas todo mês. Assegurar que um projeto desta escala está funcionando como esperado e que contribuições não vão gerar regressões é um incrível desafio que os mantenedores dos subsistemas enfrentam. Para facilitar este trabalho, muitos desenvolvedores estão voltando para ferramentas de testes automatizados e integração contínua (CI).

Uma grande quantidade de iniciativas visando assegurar a qualidade e confiabilidade do sistema surgiram no kernel Linux nos últimos anos. Projetos como KernelCI¹, Kselftest² e KUnit³ são exemplos mais recentes de iniciativas que surgiram buscando melhorar o estado de testes automatizados e CI no desenvolvimento do kernel. Todavia, mesmo com grande progresso, os diversos subsistemas do kernel Linux apresentam muitas diferenças entre si e, por isso, possuem demandas específicas, muitas vezes não atendidas por estes projetos mais abrangentes. Para solucionar esse problema, alguns utilizam suas próprias ferramentas de testes, como o mmtests⁴ do subsistema de gerenciamento de memória (MM). Entretanto, existem outros que também não são propriamente atendidos pelo instrumental existente de testes e não possuem ferramentas próprias.

Um exemplo de subsistema carente de testes automatizados é o de entrada e saída industrial (IIO), responsável por drivers de diversos dispositivos conversores digital-analógicos (DAC) e conversores analógico-digitais (ADC) que não se encaixam nos subsistemas `input` e `hwmon`. Testar código pertencente ao IIO, frequentemente, representa um desafio para os desenvolvedores visto que, para realizar testes efetivos, seria necessário ter o hardware específico em mãos ou desenvolver algum tipo de dispositivo virtual, tornando em certo ponto inacessível para a maioria dos contribuidores.

Nessa situação, o trabalho de identificar regressões em contribuições submetidas reside, majoritariamente, nos mantenedores e contribuidores do subsistema, desperdiçando o tempo destes ao procurar por erros que poderiam ter sido identificados com ferramentas de testes ou no CI. A categoria de aparelhos acolhidos pelo IIO variam, desde acelerômetros e giroscópios até sensores de pressão e sensores químicos, atendendo diversos setores, como internet das coisas e setores industriais. Com isso, o funcionamento correto destes drivers pode ser essencial para partes críticas da produção, por

¹<https://kernelci.org/>

²<https://www.kernel.org/doc/Documentation/kselftest.txt>

³https://kunit.dev/third_party/kernel/docs/

⁴<https://github.com/gormanm/mmtests>

exemplo.

A partir disso, este trabalho busca realizar um estudo sobre o estado atual das ferramentas de testes do kernel Linux para uso nos drivers do IIO. Para isso, serão exploradas desde técnicas de virtualização de dispositivos utilizando o QEMU até as ferramentas com mais adoção pelo kernel, para discutirmos as abordagens de uso dessas ferramentas quando consideradas as demandas do subsistema IIO. Este estudo tem intersecção com a pesquisa do mestrandoo Marcelo Schmitt, sob orientação dos professores Paulo Meirelles e Fabio Kon. Com isso, será possível unir esforços durante a fase de avaliação.

2 Objetivos

Este trabalho de conclusão de curso tem o objetivo de realizar um estudo sobre o desenvolvimento de drivers de dispositivos no subsistema IIO do kernel Linux. O fim deste estudo é analisar possíveis formas de se testar estes drivers, analisando o uso de virtualização de dispositivos e validando o ferramental atual presente na comunidade do kernel Linux para testes subsistema IIO.

3 Metodologia

O desenvolvimento de drivers para o kernel Linux será feito seguindo as práticas comuns do subsistema IIO, aderindo às preferências e ritmo dos mantenedores. A partir da experiência adquirida com o desenvolvimento, realizaremos um estudo de caso, onde será analisado o instrumental atual de testes e CI do kernel, buscando evidenciar os acertos e deficiências das ferramentas.

4 Cronograma de atividades

A Tabela 1 resume as atividades previstas para este trabalho e contém a previsão da linha do tempo destas. Durante os meses de junho, julho e agosto, o aluno participará do programa Google Summer of Code (GSoC), um programa feito pela Google para estimular alunos do mundo inteiro a participarem de projetos de software livre. Nesse programa, o aluno propõe um projeto e, se for aceito, recebe mentoria de membros mais experientes da comunidade para conseguir completá-lo dentro do cronograma do programa. No caso deste trabalho, o programa servirá para aprofundar os conhecimentos do aluno sobre desenvolvimento de drivers para o kernel Linux. Nesse

período, será possível analisar também a virtualização de dispositivos usando QEMU, assim avaliando suas vantagens e desvantagens comparado ao ciclo normal de desenvolvimento com hardware em mãos. Os meses de setembro e outubro serão para evoluir o desenvolvimento do driver que será objeto de estudo, bem como, um exploração da literatura relacionada e uma investigação complementar sobre o instrumental existente na comunidade do kernel Linux para testar os drivers pertencentes ao subsistema do IIO.

Tabela 1: Cronograma de atividades por mês de desenvolvimento

Atividade / Mês	Mai/Jun	Jul/Ago	Set/Out	Nov/Dez
Desenvolvimento de driver para o kernel Linux	X	X	X	
Exploração de virtualização de dispositivos com QEMU	X	X		
Análise exploratória da literatura relacionada		X	X	
Avaliação de instrumental existente para testar drivers presentes no IIO			X	
Escrita da monografia		X	X	X