



Trabalho Final

SISTEMAS OPERACIONAIS

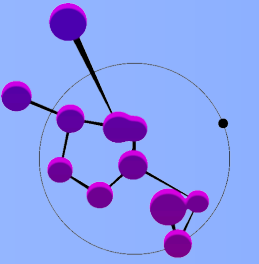


Luccas H. Vieira

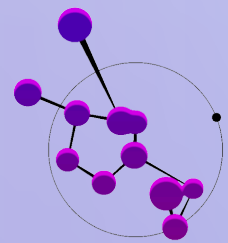


Matheus G. Sampaio

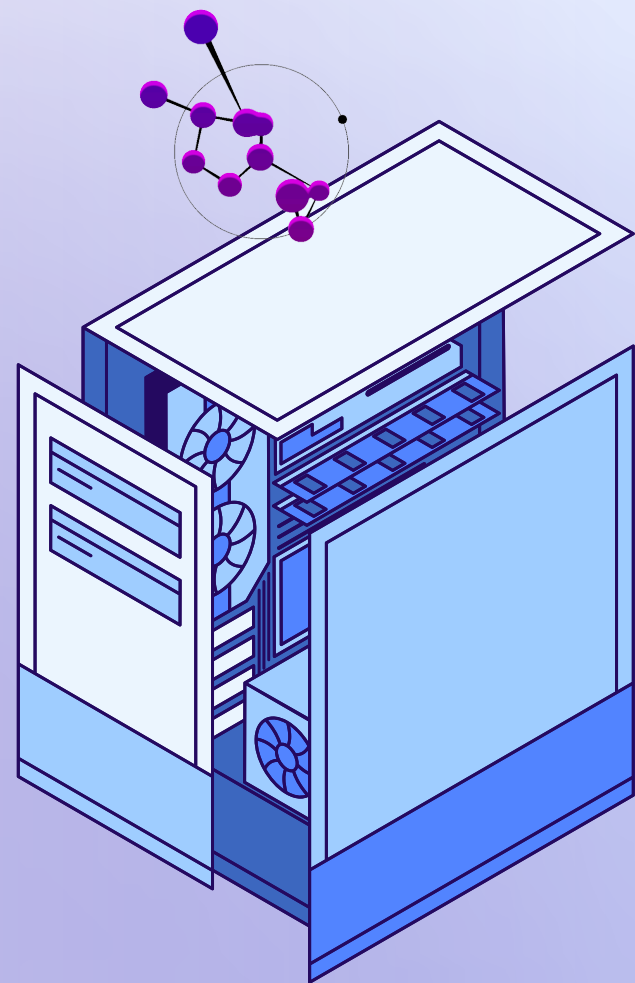




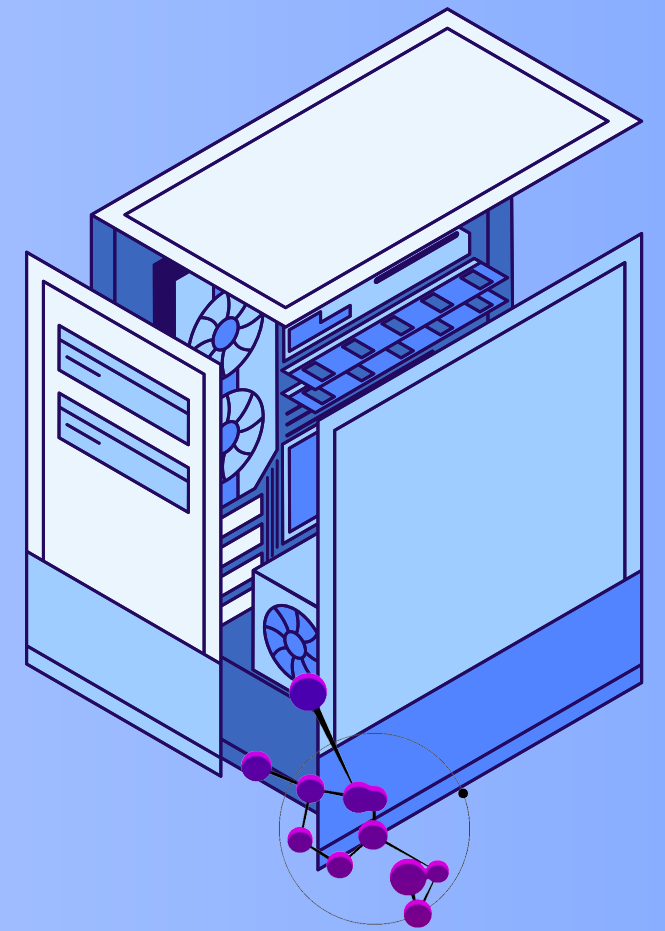
SIMULAÇÃO E TESTE DE DRIVER DE CARACTERE (LKM) PARA LINUX COM USO DE PENDRIVE BOOTÁVEL E EMULADORES



QUAL É O PROBLEMA?



Imagine um cenário onde uma equipe de técnicos precisa coletar informações personalizadas de sensores conectados a um equipamento industrial via interface serial. Para essa tarefa, foi desenvolvido um driver de caractere personalizado que coleta e interpreta os dados serializados recebidos por um dispositivo específico (simulado no projeto). Este driver precisa ser carregado de forma dinâmica em diversos ambientes Linux portáteis, diretamente de um pendrive bootável.



NOSSE

OBJETIVO

- Desenvolver um Driver de Caractere (LKM) que simule um sensor virtual para a coleta de dados.
- Criar um ambiente de teste portátil e persistente em um pendrive bootável.
- Demonstrar a viabilidade de um fluxo de trabalho completo — desde a compilação até o teste — em um ambiente dinâmico e controlado.

METODOLOGIA

FERRAMENTAS



Ferramentas Principais: Usamos um pendrive bootável, a ferramenta Ventoy para gerenciar a imagem ISO e uma distribuição Ubuntu Server como sistema operacional.

Preparação do Ambiente: A partir de uma máquina com Windows, configuramos o pendrive com o Ventoy para rodar o Ubuntu.

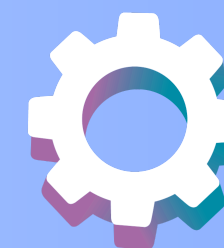
Instalação de Ferramentas: Dentro do Ubuntu, instalamos os pacotes essenciais, como gcc, make e os headers do kernel, para poder compilar nosso driver.

Ambiente de Desenvolvimento: O resultado é um ambiente Linux completo e autônomo, pronto para o desenvolvimento e teste de módulos de kernel de forma persistente.

DRIVE

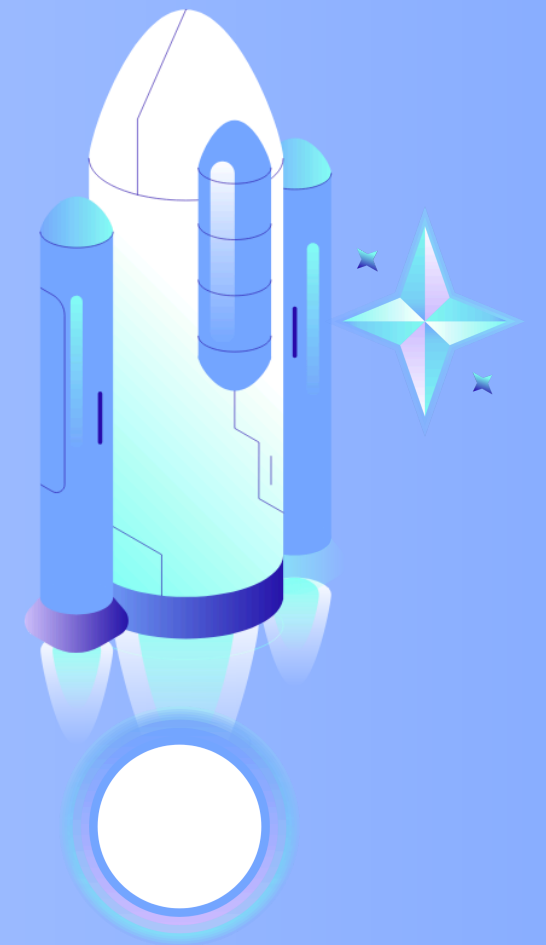
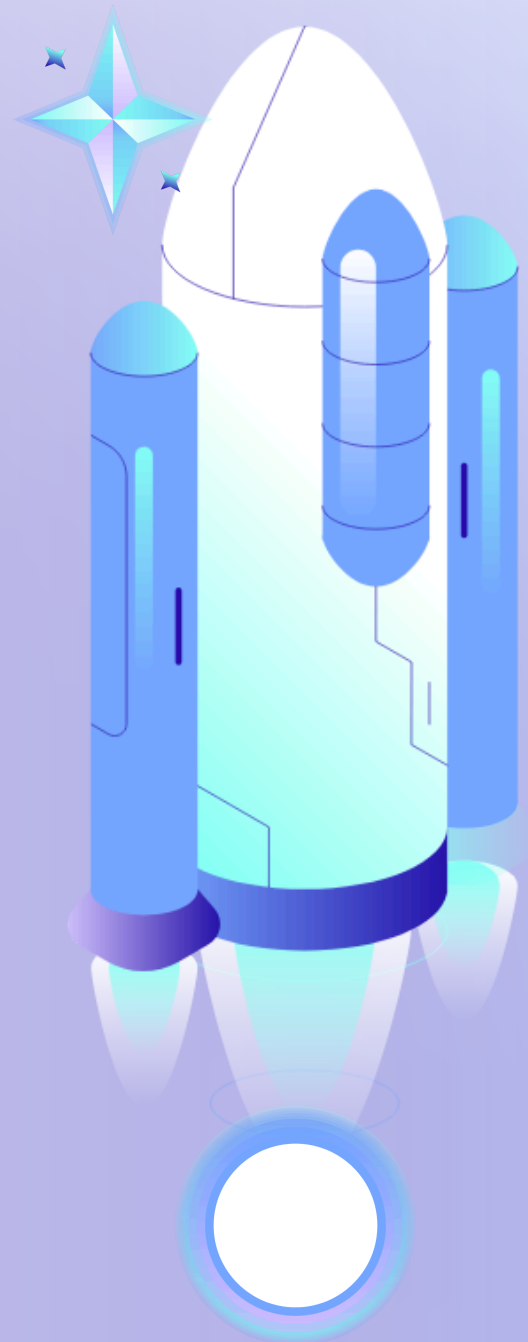


DEMONSTRAÇÃO PRÁTICA



CONCLUSÃO

- O projeto demonstrou a viabilidade de desenvolver e testar drivers de kernel em um ambiente completamente portátil.
- Conseguimos criar um driver de caractere funcional a partir de um pendrive bootável.
- A metodologia provou ser uma solução robusta e eficiente para tarefas de programação de baixo nível.
- A principal contribuição é a criação de um fluxo de trabalho autônomo e persistente, superando as limitações de ambientes de desenvolvimento tradicionais.



OBRIGADO!