

Relatório Técnico: Desenvolvimento de Sistema Operacional Linux Minimalista Embarcado

1. Objetivo do Projeto

Desenvolver um Sistema Operacional (SO) Linux minimalista e monolítico, com foco em gerenciamento de arquivos mono-usuário. O SO foi configurado para inicializar via **BIOS Legacy (MBR)** a partir de um pendrive, garantindo máxima portabilidade e desempenho mínimo.

2. Arquitetura e Componentes

Componente	Versão	Decisão de Design
Kernel	Linux 6.6.30 (LTS)	Compilado estaticamente com drivers essenciais embutidos (Built-in).
Userland	BusyBox 1.36.1	Linkagem Estática (<code>CONFIG_STATIC=y</code>) para eliminar dependências de bibliotecas (<code>glibc</code>).
Bootloader	GRUB2	Instalado na MBR, com configuração de compatibilidade blindada.
Filesystem	Ext4	Padrão robusto para a partição raiz (<code>/dev/sdb1</code>).
Ambiente Host	Ubuntu	Utilizado para <i>cross-compilation</i> e <i>sandbox</i> via QEMU/Loopback.

3. Metodologia de Otimização (BusyBox)

A *userland* foi submetida a um regime de **máximo minimalismo** para reduzir o binário final e a pegada de memória.

- **Linkagem Estática:** Compilado com `CONFIG_STATIC=y`, resultando em um binário compacto (~1.6 MB) que não requer bibliotecas externas em `/lib`.
- **Limpeza Agressiva:** Desabilitação completa de subsistemas de rede, autenticação (`login` , `passwd`), e editores complexos (`vi` , `awk`), mantendo apenas os comandos básicos de I/O (`ls` , `cp` , `mount`).
- **Inicialização:** Configurado com `/etc/init.d/rcS` para montar `/proc` e `/sys` e iniciar o `shell` interativo.

4. Kernel: Blindagem de Hardware e Portabilidade

A estabilidade e o sucesso no hardware físico foram atingidos apenas após a correção dos *drivers* essenciais e dos parâmetros de boot no **kernel** (que causavam falhas no hardware físico, apesar do sucesso no QEMU).

4.1. Drivers Críticos (Built-in [*])

Para evitar falhas como "Kernel Panic: VFS: Unable to mount root fs", os seguintes *drivers* foram obrigatoriamente compilados de forma **estática/embedida** no Kernel:

- **Disco USB/SATA:** USB Mass Storage support , AHCI SATA support , e Generic ATA/PATA support .
- **Barramento USB:** xHCI HCD (USB 3.0) e EHCI HCD (USB 2.0).
- **Vídeo (Anti-Tela Preta):** VESA VGA graphics support e Framebuffer support .

4.2. Correção de Boot (GRUB) - Solução Definitiva

A solução para a **tela preta** e a **falha de montagem** (devido à latência USB e conflito de nome de disco) foi injetada diretamente na linha de comando do Kernel via GRUB.

Parâmetro	Finalidade
root=/dev/sdb1	Endereço Definitivo: Força o Kernel a montar a partição raiz no segundo disco (/dev/sdb), a posição provável do pendrive em um PC com HD interno (/dev/sda).
rootwait	Latência USB: Instrução para o Kernel aguardar a inicialização do barramento USB antes de tentar montar a partição raiz.
nomodeset	Estabilidade de Vídeo: Impede que o Kernel carregue drivers gráficos complexos, forçando o uso do modo de vídeo básico VESA.
vga=normal console=tty0	Visualização: Força o modo de texto clássico e envia o <i>output</i> do console para o monitor físico.

4.3. Configuração Final de Boot

O arquivo `grub.cfg` final, responsável pelo boot bem-sucedido:

```
menuentry "LFS" {
    linux /boot/vmlinuz-lfs-minimal root=/dev/sdb1 rw rootwait nomodeset vga=normal consc
}
```

5. Conclusão

O projeto demonstrou a capacidade de construir um sistema operacional do zero, aplicando princípios rigorosos de engenharia de sistemas embarcados para contornar falhas críticas de *hardware* (driver de disco/vídeo e latência USB) e *firmware* (conflito de nomenclatura de disco). O sistema final é funcional, compacto e cumpre os requisitos de ser um *boot* minimalista e monolítico em arquitetura x86 64.