

# Relatório Técnico: Construção de um Sistema Linux From Scratch (LFS)

## 1. Introdução

Este relatório apresenta a construção de um sistema Linux From Scratch (LFS) no contexto da disciplina de Sistemas Operacionais II. O projeto teve como foco a criação de um sistema GNU/Linux funcional, inicializável via pendrive USB, compilado inteiramente a partir do código-fonte, proporcionando uma compreensão profunda da arquitetura interna do sistema operacional Linux.

## 2. Metodologia de Compilação de Software

A metodologia adotada envolveu a obtenção segura dos códigos-fonte, verificação de integridade, compilação controlada e otimização de desempenho, com uso intensivo de logs e paralelismo controlado via cgroups.

### 2.4. Otimização de Binários e Gerenciamento de Vídeo

Além das otimizações de compilação, o kernel foi configurado com suporte a *DRM FBDEV Emulation (CONFIG\_DRM\_FBDEV\_EMULATION)*. Essa camada de compatibilidade permite que o console de texto (fbcon) opere sobre drivers gráficos modernos (Intel/AMD), eliminando falhas de tela preta comuns em sistemas construídos do zero.

## 3. Gerenciamento de Dependências

O gerenciamento de dependências foi tratado de forma rigorosa, respeitando dependências obrigatórias, recomendadas e de tempo de execução, além da análise de dependências aninhadas.

### 3.3. Troubleshooting e Depuração

Durante o desenvolvimento, diversos problemas práticos foram identificados e resolvidos.

Destacam-se:

- Erro 127 no Initramfs, causado por bibliotecas compartilhadas ausentes no ambiente initrd.
- Substituição de identificadores /dev/sdaX por UUIDs, garantindo portabilidade e confiabilidade do boot em diferentes máquinas.

## 4. Configuração do Sistema Base

Esta etapa consolidou os binários compilados em um sistema funcional, incluindo configuração de shell, usuários, kernel e bootloader.

### 4.4. Bootloader GRUB (BIOS Legacy – USB)

O teste final do sistema foi realizado em modo BIOS Legacy (MBR), com inicialização via pendrive USB (/dev/sdb1). O GRUB foi configurado para localizar a partição raiz via UUID, aumentando a portabilidade:

```
menuentry "LFS 12.4 (USB Boot)" { search --no-floppy --fs-uuid --set=root  
74ce1171-b392-4fff-9115-f425645f7760 linux /boot/vmlinuz-6.18-lfs-12.4  
root=UUID=74ce1171-b392-4fff-9115-f425645f7760 ro rootdelay=10 initrd /boot/initrd.img-6.18 }
```

### 4.5. Initramfs e Boot via USB (Solução de Engenharia Customizada)

Diferente da construção padrão do LFS, este projeto implementou um Initramfs customizado. Essa decisão foi necessária para resolver a latência de detecção do barramento USB em hardware real, garantindo que os drivers *usb-storage* e *uas* fossem carregados antes da tentativa de montagem da partição raiz, evitando erros de *Kernel Panic*.

## 5. Considerações sobre Segurança e Otimização

Além do hardening básico e da gestão de certificados, o projeto considerou a distribuição eficiente

do sistema.

### **5.3. Distribuição e Otimização de Imagem**

Para distribuição, aplicou-se a técnica de *Zero-Fill* no espaço livre da partição antes da geração da imagem .img. Com isso, uma imagem de 16GB pôde ser compactada via Gzip para menos de 1GB, facilitando o compartilhamento e armazenamento em repositórios remotos como o Google Drive.

## **6. Especificações Técnicas do Sistema**

- Kernel Linux: 6.18.0
- GCC: 15.2.0
- Sistema inicializável via USB (BIOS Legacy)
- Initramfs customizado para suporte a USB

## **7. Conclusão**

O projeto LFS foi concluído com sucesso, resultando em um sistema Linux funcional, portátil e altamente customizado. A experiência consolidou conhecimentos avançados em compilação, boot, depuração e engenharia de sistemas operacionais, indo além do escopo básico proposto pelo LFS tradicional.