

\$ Is-g Linux Study Group

Daniel Carlos Chaves Boll
Lucas Eduardo Fischer Mülling
Pablo Alessandro Hugen



Universidade Estadual do Oeste do Paraná – Unioeste



Curso de Bacharelado em Ciência da Computação



LambdaGroup

14 de junho de 2022

Introdução

O grupo de estudos surgiu a partir do interesse dos membros do laboratório LSC em distribuições Linux e a vontade de criar um grupo de estudo para o ano letivo de 2021.

Já realizamos outros grupos de estudos antes, com relativo sucesso.

- LambdaPET 1 – Grupo de estudos de programação e linguagens funcionais, realizado no ano letivo de 2019;
- LambdaPET 2 – Segunda edição, realizada no ano letivo de 2020;

Esperamos que este grupo forneça aos integrantes um conteúdo não muito visto no nosso curso e também sirva de motivação para trabalhos relacionados a distribuições Linux, e o próprio Kernel;

Quem somos

- Daniel Boll – A.K.A Ryan Gosling, literally me, acadêmico do terceiro ano, interessado em Rust e mais um milhão de outras linguagens, ricing, emojis (que não funcionam no \LaTeX) e big friend of copilot.
- Lucas Mulling – Acadêmico do quarto ano, interesses incluem: Tuna, programação funcional, LISP, hardware, embarcados, compilar Kernel as 2 da manha.
- Pablo Hugen – Ingresso no curso no ano de 2019, curioso, dentre muitas coisas, sobre sistemas UNIX, langs como python e go, sardinhas enlatadas e musica de baixa qualidade.

Sobre o grupo de estudos

O grupo de estudo aspira ser um ambiente aberto para a discussão dos tópicos planejados, não um ambiente de aula. Portanto, a participação dos membros é fundamental.

- Vamos usar bastante o repositório no github:
<https://github.com/LambdaGroup/lfs-g> [▶ Link](#)
- E também o servidor ssh `lfs@192.168.1.2` senha:windowssucks
- Tenham em mãos seu PC com GNU/Linux.
- Manifestem-se caso duvidas ou sugestões e mais importante, PARTICIPEM!

Cronograma

- Primeiro encontro: revisão de conteúdos que vamos precisar para o LFS.
- Encontros subsequentes: seguiremos os capítulos do LFS.
- Ao final vamos brincar um pouco com o Kernel e fazer os nossos próprios módulos e *drivers*.
- Vamos também fazer algumas atividades de CTF e outros.

Revisão básica da shell

- Vamos fazer uma revisão básica dos comandos da shell.
- A shell eh um programa que expõem as funcionalidades básicas do sistema operacional para o usuário.
- Podem ser tanto TUI (bash, mksh, zsh, bash) ou GUI (X, Wayland).
- Para o LFS vamos usar bash.

Comandos de navegação

- `cd` – mudar o diretório de trabalho atual.
- `pushd` – similar ao `cd`, esquema LIFO.
- `popd` – voltar ao diretório do topo da LIFO.
- `ls` – listar o conteúdo do diretório atual.
- `pwd` – caminho absoluto do diretório atual.
- `find` – buscar por arquivos e diretórios.
- `lsblk` – listar dispositivos de bloco (sistemas de arquivos).
- `mount` – montar sistemas de arquivos.

Comandos de criação de diretórios e links

- `mkdir` – cria um diretório.
- `ln` – cria link para arquivos e diretórios.

Para quem quiser se aprofundar:

<https://linuxjourney.com/lesson/the-shell> [▶ Link](#)

Comandos de manipulação de arquivos

- `cat` – concatenar arquivos e imprimir.
- `echo` – imprime texto.
- `vi` – editor de texto modal.
- `sed` – editor de *streams* de texto.
- `grep` – procura em arquivos por padrões de texto.

The Manual

`man` – Interface para os manuais do sistema. O `man` possui 9 secções diferentes:

- 1 – Comandos da *shell* ou executáveis.
- 2 – Chamadas de sistema.
- 3 – Chamadas de bibliotecas.
- 4 – Arquivos especiais (voltaremos a falar deles).
- 5 – Formatos de arquivos e convenções.
- 6 – Jogos.
- 7 – Diversos.
- 8 – Comandos de `root`.
- 9 – Rotinas de Kernel.

`tl;dr` – Too Long Didn't Read

FHS - Filesystem Hierarchy Standard

O Standard permite que:

- Software e usuários predigam a localização de arquivos e diretórios

Como isso é feito:

- Especificação de guias para cada área do sistema de arquivos.
- Mínimo de arquivos e diretórios.
- Exceções e áreas de conflito.

Quem usa:

- Desenvolvedores.

Mantido pela fundação Linux [▶ Link](#)

FHS - Filesystem Hierarchy Standard

O FHS define tipos de arquivos e diretórios, *Sharable*, *Unsharable*, *Static* e *Variable*:

- Arquivos compartilháveis podem ser compartilhados pelos usuários do sistema.
- Arquivos não compartilháveis não.
- Arquivos estáticos são aqueles que não mudam, e requerem privilégios para serem mudados.
- Arquivos variáveis podem ser mudados.

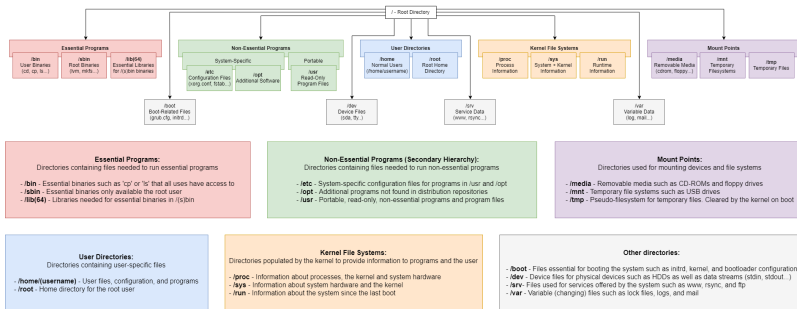
FHS - Filesystem Hierarchy Standard

	Sharable	Unsharable
Static	/usr /opt	/etc /boot
Variable	/var/mail /var/spool/news	/var/run /var/lock

Tabela 1: Exemplo de sistema de arquivos conforme ao padrão FHS

FHS - Filesystem Hierarchy Standard

The Filesystem Hierarchy Standard (FHS)



Created by Max Hösel and licensed under the Creative Commons **CC-BY 4.0** license. Last edit: 2018-05-20

Figura 1: FHS "padrão" da maioria das distribuições Linux

Usuários

- Para ver todos os usuários do sistema: `cat /etc/passwd`
- Dois tipos de usuários: de **sistema** e **regulares**.
 - Usuários de sistemas são usados para rodar *daemons* no sistema operacional.
 - Usuários de regulares são aqueles criados para login e para rodar processos interativos.
 - O Superusuário(*root*) e aqueles com permissão de sobrescrever qualquer(algumas restrições pontuais) permissão de arquivo no sistema. É possível fornecer permissões pontuais de superusuário a usuários comuns.
 - Para logar no superusuário do sistema: `su`.
 - Para fornecer permissões de superusuários a usuários comuns, geralmente é utilizado o comando `sudo`. Para configurar quais permissões são cedidas, o comando `sudoedit`.

Grupos

- Grupos são coleções de zero ou mais usuários.
- Por padrão, um usuário pertence a um grupo padrão(o mesmo nome do usuário)
- Grupos também possuem permissões. Por exemplo, todos os usuários no grupo `network` tem permissão de interagir com dispositivos de rede.
- Para listar todos os grupos do sistema: `cat /etc/group`

Grupos

- Grupos são coleções de zero ou mais usuários.
- Por padrão, um usuário pertence a um grupo padrão(o mesmo nome do usuário)
- Grupos também possuem permissões. Por exemplo, todos os usuários no grupo `network` tem permissão de interagir com dispositivos de rede.
- Para listar todos os grupos do sistema: `cat /etc/group`

Permissões

- No UNIX tudo é arquivo.
- Todo arquivo possui permissões, além de ser possuído por um usuário e um grupo.
- Para listar as permissões dos arquivos podemos usar o `ls -la`.
- Ok, mas qual o significado desses valores?

Permissões

- **Type**
 - -: Arquivos regulares
 - d: Diretórios
 - s: Arquivos especiais
- **Owner**: Permissões do usuário dono do arquivo.
- **Group**: Permissões do grupo dono do arquivo.
- **Other Users**: Permissões de outros usuários.

Permissões

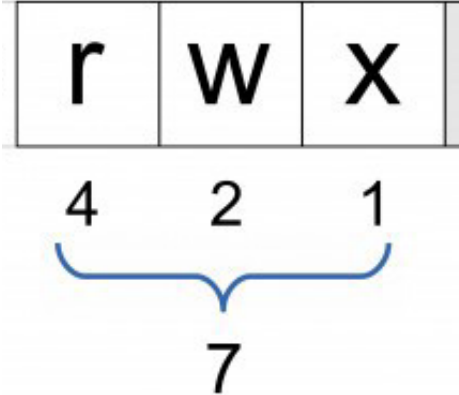


Figura 3: 3 bits para permissões

Permissões

- **r**: Permissões para leitura do arquivo.
- **w**: Permissões para escrita no arquivo.
- **x**: Permissões para a execução do arquivo.

Comando para gerenciar permissões

- **Grupos de um usuário:** `groups USER`
- **Adicionar um usuário a um grupo:** `usermod -a -G GROUP USER`
- **Criar um usuário a um grupo:** `useradd -s SHELL`
- **Mudar a senha de um usuário:** `passwd USER`
- **Mudar as permissões de um arquivo:** `chmod PERMIS FILE`
- **Mudar o grupo dono de um arquivo:** `chgrp GROUP FILE`
- **Mudar o usuário dono de um arquivo:** `chown USER:FILE FILE`

CTF - Bandit

- <https://overthewire.org/wargames/bandit/bandit0.html> [▶ Link](#)
- `ssh bandit0@bandit.labs.overthewire.org -p 2220`