

Livre Camp apresenta:

# Semana da Calourada Livre

# Olar :D

Samuel.Chenatti@gmail.com

## > O que vimos até agora?

- Introdução à Sistemas Operacionais (**Akari**)
- Comandos básicos (**Ju**)
  - **ls** (listar arquivos e diretórios)
  - **cd** (acessar diretórios)
  - **touch** (cria arquivos vazios)
  - **mkdir** (cria diretórios)
  - **cp** (copia arquivos e diretórios)
- Muita coisa pra decorar? :(
  - Não decore; use!

## > O que veremos hoje?

- Como navegar na estrutura de diretórios
- Permissões
- Comandos intermediários
  - **rm** (apagar arquivos e diretórios)
- Tips'n'tricks do terminal
- Comandos avançados (gerenciamento do sistema)
  - **sudo**
  - **du**
  - **htop**

# hoje vai ser **tenso**

mas aprenderemos coisas intermediárias

~\* hacker stuff \*~

## > Revisão rápida

- Na última aula vimos alguns comandos básicos
- Para usá-los, você os chama pelo nome e em seguida passa os parâmetros
- Parâmetros (ou opções) são separados por espaços e em alguns casos são apenas uma letra precedida de hífen
- Exemplo:

```
mkdir -p pasta1/pasta2/pasta3
```

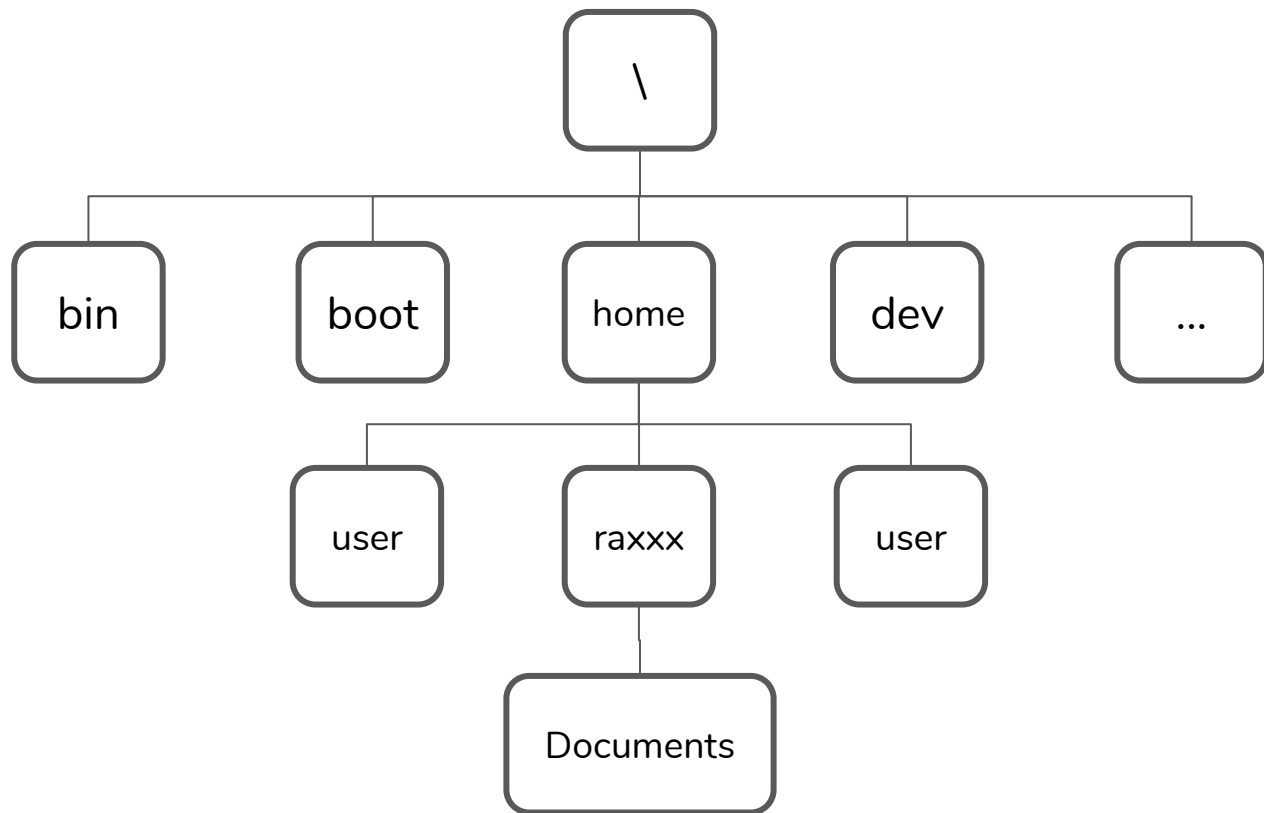
## > A estrutura de diretórios do GNU/Linux

- Permite visualizar os arquivos de maneira hierárquica (quais arquivos pertencem a quais pastas)
- Segue o padrão Unix-Like
- Parecida com a do Windows:
  - `C:/User/MeuUsuario/Documentos/...`
- **Interface gráfica só dificulta**

# **A estrutura de diretórios do GNU/Linux**



## > A estrutura de diretórios do GNU/Linux

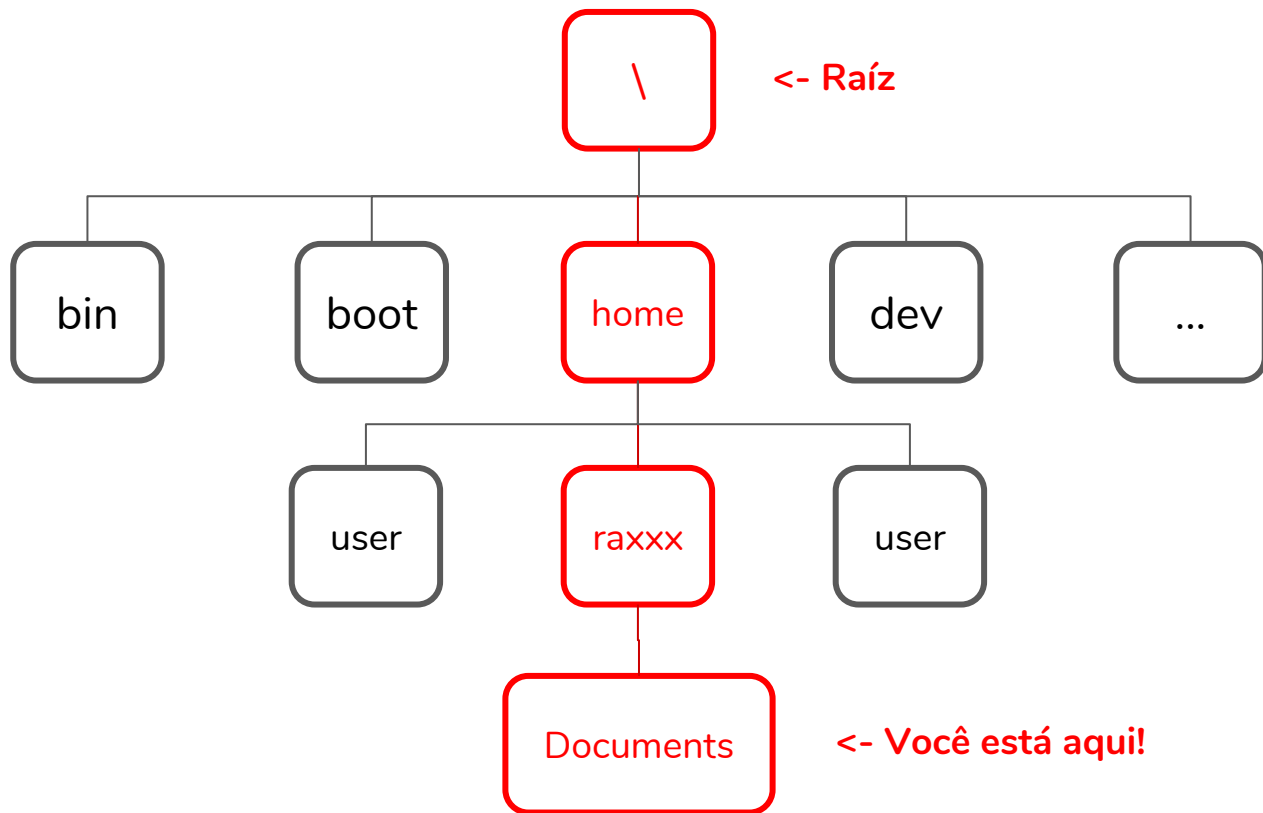


> Navegue até sua pasta documentos

```
cd ~
```

```
cd Documents
```

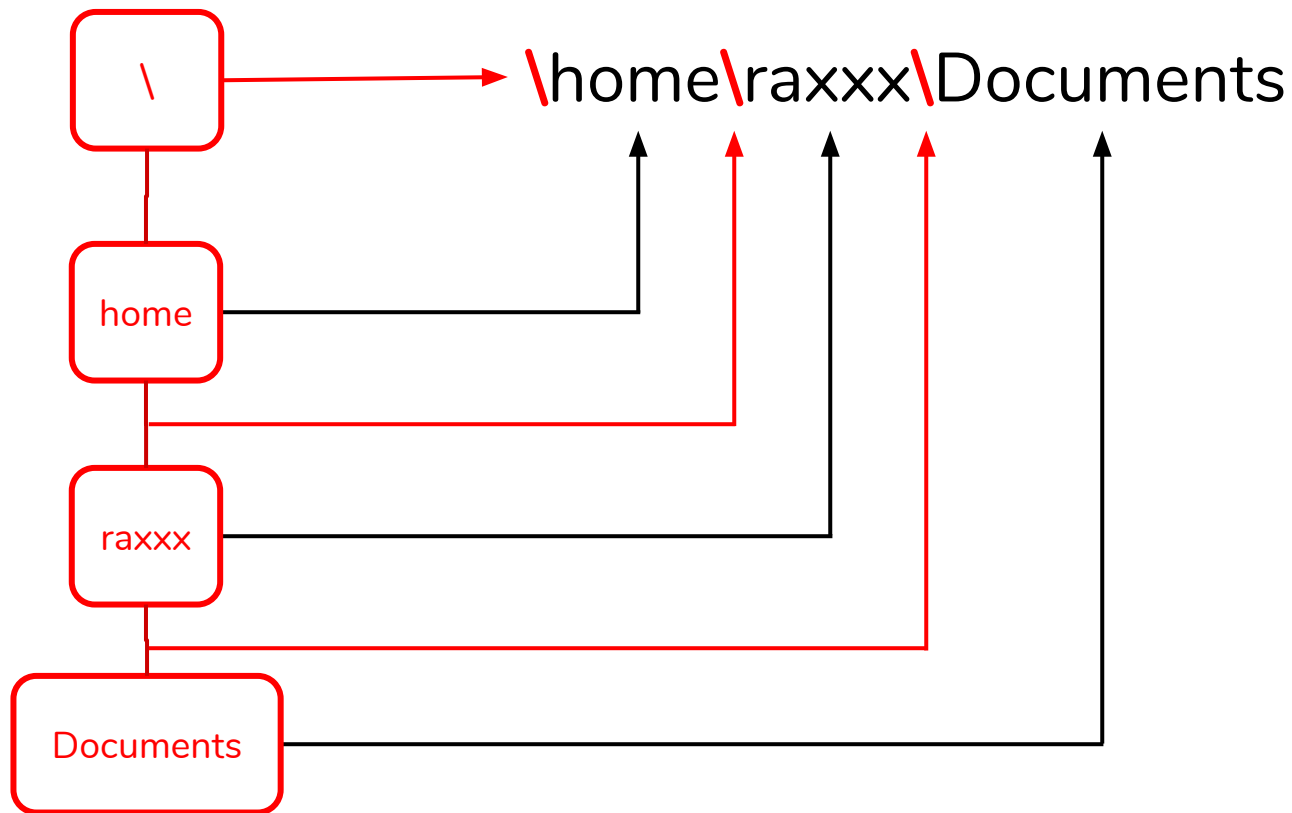
## > A estrutura de diretórios do GNU/Linux



## > A estrutura de diretórios do GNU/Linux

- Onde eu estou agora?
  - **pwd** (print working directory)
  - A saída mostra toda a hierarquia de diretórios até a pasta atual
  - Em algumas configurações do bash, este caminho aparece antes do nome do usuário

## > A estrutura de diretórios do GNU/Linux



## > A estrutura de diretórios do GNU/Linux

- Quando passamos um caminho como parâmetro para um comando, ele sempre parte do diretório atual
- Se estivermos em `/` e utilizarmos o **`mkdir MinhaPasta`** estaremos tentando criar uma pasta na raíz
- Se estivermos em `/` e utilizarmos **`mkdir ~/Documents/MinhaPasta`** estaremos criando uma pasta no nosso diretório de usuário

## > Exercício

- Qual a diferença entre os comandos

`mkdir MinhaPasta`

e

`mkdir /MinhaPasta`

?

## > Exercício

- Navegue até sua pasta de usuário e, a partir dela, crie uma pasta dentro de *Documents* chamada *MinhaPasta*
- ***Dica: o atalho para a sua pasta de usuário é ~***



## > A estrutura de diretórios do GNU/Linux

- Quando passamos um caminho como parâmetro para um comando, ele sempre parte do diretório atual
- Se estivermos em `/` e utilizarmos o **`mkdir MinhaPasta`** estaremos tentando criar uma pasta na raíz
- Se estivermos em `/` e utilizarmos **`mkdir ~/Documents/MinhaPasta`** estaremos criando uma pasta no nosso diretório de usuário

# **Permissões**

(visão rápida)

## > Permissões

- O Linux especifica as seguintes permissões **para cada arquivo:**
  - **r:** leitura
  - **w:** escrita
  - **x:** execução

## > Permissões

- O comando `ls -la` mostra as permissões dos arquivos no canto:

```
drwxr-xr-x  2 user group  4096 Nov 28 15:37 Arquivo
```

## > Permissões

- O formato é **d(rwx)(rwx)(rwx)**
- O primeiro caractere indica se é uma pasta (d) ou um arquivo comum (-)
- O primeiro grupo são as suas permissões de usuário
- O segundo, as permissões do grupo no qual você está
- No terceiro temos as permissões dos demais usuários
- O comando para alterar as permissões é o **chmod**

## > Permissões

- Em alguns casos somente o **superusuário (ou root)** pode executar alguns comandos ou alterar alguns arquivos
- **0 super usuário pode fazer qualquer coisa dentro do sistema (perigo!)**
- Para executar um comando como superusuário, utilizamos **sudo** antes de qualquer comando
- Isso só funciona se o administrador te der este poder

**Remover arquivos e pastas**

## > Remover arquivos e pastas

- Para remover um arquivo, usamos o comando rm:

**rm nomeDoArquivo**

- O rm é irreversível, então devemos usá-lo com prudência.



## > Exercício

- Navegue até a sua pasta de usuário
- Crie um arquivo chamado teste
- Remova o arquivo

## > Exercício

- Navegue até a sua pasta de usuário
- Crie um arquivo chamado teste
- Remova o arquivo

**cd ~**

**touch teste**

**rm teste**

## > Remover arquivos e pastas

- Para remover uma pasta, utilizamos a versão recursiva do rm:

```
rm -rf nomeDaPasta
```

## > Exercício

- Navegue até a sua pasta de usuário
- Crie a seguinte hierarquia: PastaA/PastaB/PastaC
- Apague a hierarquia toda de uma vez com `rm -r`

## > Exercício

- Navegue até a sua pasta de usuário
- Crie a seguinte hierarquia: PastaA/PastaB/PastaC
- Apague a hierarquia toda de uma vez com `rf -r`

**`cd ~`**

**`mkdir -p PastaA/PastaB/PastaC`**

**`rm -r PastaA`**

# **Tips'n'Tricks do terminal**

## > Tips'n'Tricks do terminal

- O objetivo do terminal é facilitar a sua vida
- Para isso, ele implementa uma série de funções muito úteis
- Ao apertar para cima e para baixo, **você pode navegar no seu histórico de comandos** (reescreve-los é chato)
- Usando tab, **o terminal completará o nome de arquivos, pastas ou comandos**

## > Exercício

- Crie um arquivo com um nome gigante; por exemplo, eu0deio0FrangoDoBandeiro
- Apague o arquivo utilizando rm, **mas ao invés de escrever o nome todo, escreva apenas as quatro primeiras letras e pressione tab**
- Utilize as setas para encontrar o comando que criou o arquivo e execute-o novamente



**Tamanho de arquivos e pastas**

## > Exercício

- Arquivos e pastas ocupam espaço no disco
- Para saber o quão grande é um arquivo, utilize o comando du da seguinte maneira:

**du -hs nomeDoArquivo**

## > Exercício

- Crie um arquivo com um nome qualquer
- Utilize o comando `du` para descobrir seu tamanho