



**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE**

---

# **Introdução a Sistemas Abertos**

## **Manipulação de Usuários e Permissões**



# Conteúdo Programático

- Criação de Contas
  - Comando de Criação
  - Arquivos afetados
  - Comando de troca de senha
- Criação de Grupos
  - Comando de Criação
  - Arquivo afetado
  - Comando de Vinculação
- Permissões
  - Identificação de Dono
  - Identificação de Grupo
  - Identificação de Outros Usuários
  - Tipos de Permissão
  - Comandos
    - `chmod`
    - `chown`



# Manipulação de Contas

---

## ■ **adduser**

- Adiciona um usuário ou grupo no sistema. Por padrão, quando um novo usuário é adicionado, é criado um grupo com o mesmo nome do usuário.
- A identificação do usuário (UID) escolhida será a primeira disponível no sistema especificada de acordo com a faixa de UIDS de usuários permitidas no arquivo de configuração `/etc/adduser.conf`



# Manipulação de Contas

## ■ adduser

- adduser [opções] [usuário/grupo]
- Ex.: adduser mitnik

## ■ Criação de contas de Sistema

- Para criar uma conta de sistema é preciso dizer para o comando adduser que ele deve encaixar a UID do usuário dentro das reservadas para usuários de sistema
- Ex: adduser --system proxy



# Manipulação de Contas

---

- **passwd**
  - Troca a senha de um usuário
  - Ex: passwd kelven
  
- **userdel**
  - Remove um usuário
  - Ex.: userdel mitnik



# Manipulação de Contas

---

- Arquivos Afetados
  - O programa passwd lê as definições do arquivo `/etc/passwd.conf` e grava os dados da nova conta nos arquivos:
    - `/etc/passwd`
    - `/etc/shadow`
    - `/etc/group`



# Manipulação de Contas

- /etc/passwd
  - Este arquivo contém as informações das contas dos usuários.
  - Exemplo:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
sales:x:1000:1000:Sales Filho,,,:/home/sales:/bin/bash
identd:x:100:65534::/var/run/identd:/bin/false
sshd:x:101:65534::/var/run/sshd:/bin/false
```



# Manipulação de Contas

- /etc/shadow

- Este arquivo contém as senhas criptografadas dos usuários. Contém também informações como data de expiração, quando a senha foi alterada pela última vez...

- Exemplo:

```
root:$1$ID9zHTEO$bQ2UiWzn4mvRtDDQXU81E1:13324:0:99999:7:::  
daemon:*:13324:0:99999:7:::  
bin:*:13324:0:99999:7:::  
sys:*:13324:0:99999:7:::  
sync:*:13324:0:99999:7:::  
nobody:*:13324:0:99999:7:::  
sales:$1$nYN2ISh8$rcYLUO.hTERQX/sGJMB5z1:13324:0:99999:7:::  
identd:!:13324:0:99999:7:::  
sshd:!:13324:0:99999:7:::
```



# Manipulação de Contas

## ■ /etc/group

- Por padrão, será criado um grupo para cada novo usuário com o mesmo nome do login
- O usuários possuem um grupo primário, mas podem pertencer a mais de um grupo
- Exemplo:

```
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
sales:x:1000:  
ssh:x:103:
```



# Manipulação de Grupos

---

## ■ **groupadd**

- Adiciona um novo grupo
- Exemplo: `groupadd novogrupo`

## ■ **addgroup**

- Adiciona um usuário a um grupo
- Exemplo: `addgroup sor admin`

## ■ **groupdel**

- Remove um grupo
- Ex.: `groupdel admin`



# Atividade

---

- Verificar o conteúdo dos arquivos:
  - /etc/passwd
  - /etc/shadow
  - /etc/group
- Criar um usuário chamado aula
- Criar um grupo chamado ifrn
- Alterar a senha de aula para C@s@!0
- Colocar o usuário aula no grupo ifrn
- Rever os arquivos do primeiro tópico



# Permissões

---

- A permissão de acesso protege o sistema de arquivos Linux do acesso indevido ou não autorizados.
- A permissão de acesso do GNU/Linux impede que um programa mal intencionado, por exemplo, apague um arquivo
- O sistema GNU/Linux, como qualquer outro sistema seguro e confiável impede que usuários iniciantes (ou mal intencionados) instalem programas enviados por terceiros sem saber para que eles realmente servem e causem danos irreversíveis em seus arquivos, seu micro ou sua empresa.



# Donos, Grupos e outros usuários

- O princípio da segurança no sistema de arquivos GNU/Linux é definir o acesso aos arquivos por donos, grupos e outros usuários
- **Dono**
  - É a pessoa que criou o arquivo ou o diretório
- **Grupo**
  - Para permitir que vários usuários diferentes tivessem acesso a um mesmo arquivo
- **Outros**
  - É a categoria de usuários que não são donos ou não pertencem ao grupo do arquivo



# Tipos de Permissões de acesso

- Quanto aos tipos de permissões que se aplicam ao dono, grupo e outros usuários, temos 3 permissões básicas:
  - **r** - Permissão de leitura para arquivos. Caso for um diretório, permite listar seu conteúdo
  - **w** - Permissão de gravação para arquivos. Caso for um diretório, permite a gravação de arquivos ou outros diretórios dentro dele
  - **x** - Permite executar um arquivo (caso seja um programa executável). Caso seja um diretório, permite que seja acessado através do comando **cd**

# Tipos de Permissões de acesso

- As permissões de acesso a um arquivo/diretório podem ser visualizadas com o uso do comando **ls -l**

```

sor@labsor:~$ ls -l
total 12
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 2 sor sor 4096 Abr  4 20:03 dir_002
sor@labsor:~$
  
```

Dono	Grupo	Outros
<b>rwX</b>	<b>r-x</b>	<b>r-x</b>

Dono: **sor**  
Grupo: **sor**

# Etapas para acesso a um arquivo/ diretório

---

- O acesso a um arquivo/diretório é feito verificando primeiro se o usuário que acessará o arquivo é o dono, se for as permissões de dono são aplicadas
- Caso não seja o dono do arquivo/diretório, é verificado se ele pertence ao grupo correspondente do arquivo se for as permissões de grupo são aplicadas
- Caso não pertença ao grupo, são verificadas as permissões de acesso para os outros usuários



# Alterando Permissões

---

- `chmod`
  - Muda a permissão de acesso a um arquivo ou diretório
  - Uso: `chmod [opções] [permissões] [diretório/arquivo]`
- O Linux entende com as indicações de usuário, grupo e outros pelas letras “u”, “g” e “o” respectivamente e os símbolos “+” e “-” para adicionar ou remover permissões

# Alterando Permissões

- Exemplos de utilização

```
sor@labsor:~$ ls -l arquivo1.txt  
-rw-r--r-- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

Adicionando permissão de execução para o dono

```
sor@labsor:~$ chmod u+x arquivo1.txt  
sor@labsor:~$ ls -l arquivo1.txt  
-rwxr--r-- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```



# Alterando Permissões

- Exemplos de utilização

```
sor@labsor:~$ ls -l arquivo1.txt  
-rwxr--r-- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

Adicionando permissão de escrita para o grupo e para os outros

```
sor@labsor:~$ chmod go+w arquivo1.txt  
sor@labsor:~$ ls -l arquivo1.txt  
-rwxrw-rw- 1 sor sor 38 Abr  4 19:18 arquivo1.txt  
sor@labsor:~$ _
```



# Alterando Permissões

- Exemplos de utilização

```
sor@labsor:~$ ls -l arquivo1.txt  
-rwxrw-rw- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

Removendo permissão de escrita do grupo e de leitura de outros

```
sor@labsor:~$ chmod g-w,o-r arquivo1.txt  
sor@labsor:~$ ls -l arquivo1.txt  
-rwxr---w- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```



# Modo Permissões Octal

- Ao invés de utilizar os modos de permissão +r, -r, etc, pode ser usado o modo octal para se alterar a permissão de acesso a um arquivo. O modo octal é um conjunto de oito números onde cada número define um tipo de acesso diferente

**0** - Nenhuma permissão de acesso. Equivalente a -rwx.  
**1** - Permissão de execução (x).  
**2** - Permissão de gravação (w).  
**3** - Permissão de gravação e execução (wx).  
**4** - Permissão de leitura (r).  
**5** - Permissão de leitura e execução (rx).  
**6** - Permissão de leitura e gravação (rw).  
**7** - Permissão de leitura, gravação e execução. Equivalente a +rwx.



# Alterando Permissões em Octal

- Exemplos de utilização

```
sor@labsor:~$ ls -l arquivo1.txt  
-rwxr---w- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

Removendo todas as permissões

```
sor@labsor:~$ chmod 000 arquivo1.txt  
sor@labsor:~$ ls -l arquivo1.txt  
----- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

# Alterando Permissões em Octal

- Exemplos de utilização

```
sor@labsor:~$ ls -l arquivo1.txt  
----- 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```

Adicionando permissões rwx para o dono e rx para os demais

```
sor@labsor:~$ chmod 755 arquivo1.txt  
sor@labsor:~$ ls -l arquivo1.txt  
-rwxr-xr-x 1 sor sor 38 Abr  4 19:18 arquivo1.txt
```



# Alterando dono de Arquivo

## ■ chown

- Muda dono de um arquivo/diretório.  
Opcionalmente pode também ser usado para mudar o grupo
- Uso: `chown [opções] [dono.grupo] [diretório/arquivo]`

```
root@debian:/home/sor# ls -l
-rw-r--r-- 1 aluno aluno 9 Dez 18 09:26
arquivo1.txt
root@debian:/home/sor# chown sor.sor arquivo1.txt
root@debian:/home/sor# ls -l
-rw-r--r-- 1 sor sor 9 Dez 18 09:26 arquivo1.txt
root@debian:/home/sor#
```