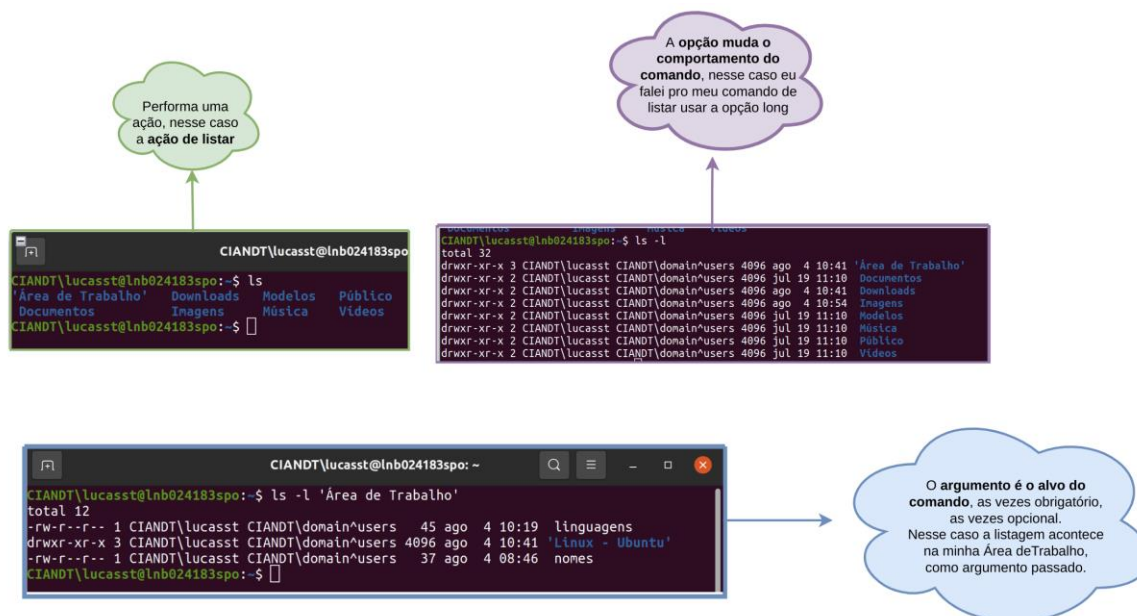




Sintaxe dos comandos – Linux

Todo **comando** do linux tem uma **sintaxe bem definida**, basicamente um comando pode ser composto de:

[comando] [opção/opções] [argumento(s)]

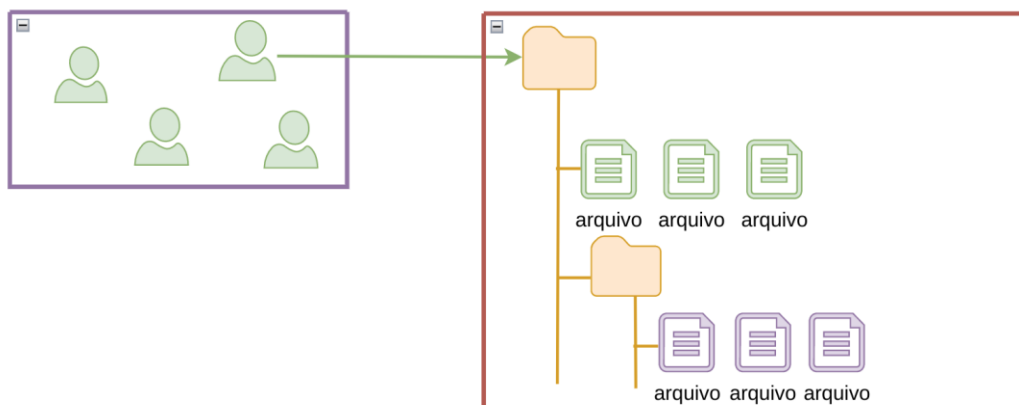


Alguns **comandos** aceitam **múltiplas opções** numa **mesma sentença**: "**comando -xyz**"



Permissões de arquivos – Linux

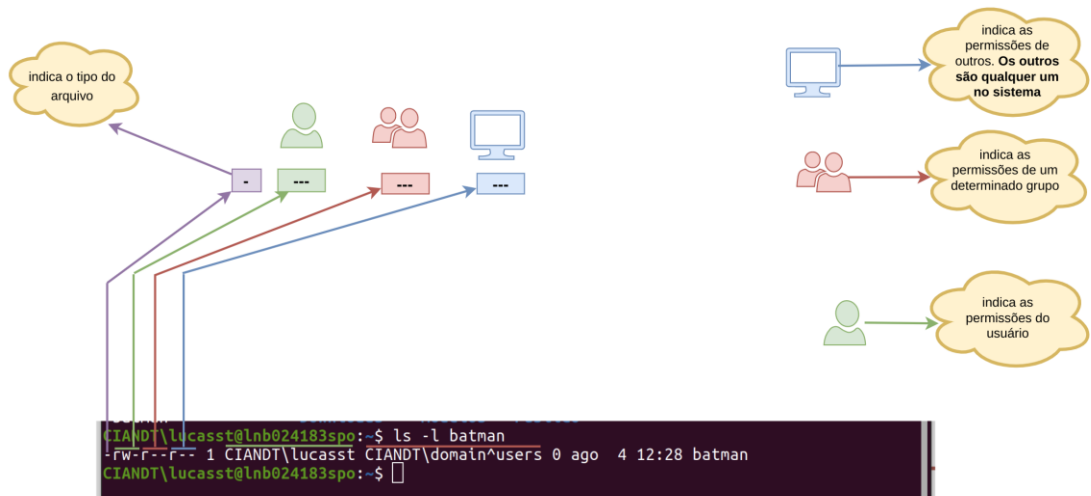
Sistemas **UNIX** são **multi usuários**, significa que eu posso ter vários **usuários** em um **mesmo sistema**. Como usuário eu posso controlar o acesso aos meus **arquivos/diretórios** restringindo as **permissões**



a **eles**:

Existem **3 tipos de permissões**, quando você lista as permissões de um arquivo e vê aquele monte de letra na frente dele, são as permissões referentes aos 3 tipos existentes.

Cada conjunto de permissão é representando por 3 caracteres que em ordem representam cada uma das seguintes permissões: **leitura, escrita e executável** (caso o arquivo seja um executável):



A **mudança de permissões** pode ser feita através do **comando chmod** com a **opção de permissão** (letra que representa o tipo e se vai adicionar ou remover) e o **argumento onde o comando vai agir** (o arquivo/diretório alvo):

```
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
-rw-r--r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$ chmod g+w batman
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
-rw-rw-r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$
```

Adicionando permissão de escrita para o grupo

```
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
-rw-rw-r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$ chmod a-r batman
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
--w-rw-r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$
```

Removendo permissão de leitura para TODOS

```
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
--w-rw-r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$ chmod u+wr batman
CIANDT\lucasst@lnb024183spo:~$ ls -l batman
-rw-rw-r-- 1 CIANDT\lucasst CIANDT\domain\users 0 ago 4 13:06 batman
CIANDT\lucasst@lnb024183spo:~$
```

Adicionando permissão de leitura e escrita para o usuário

Diretórios podem ter a permissão de executável (*terceiro caractere em cada tipo*) como **"x"**, isso significa que **eles podem ser acessados** e obviamente o comando chmod pode mudar essa permissão.

Permissões também podem ser passadas com **números**, basicamente falando **temos uma tabela que vai de 0 a 7**, cada número representa **uma permissão ou um conjunto delas**:

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--X
2	Write	-W-
3	Execute + Write	-WX
4	Read	r--
5	Read + Execute	r-X
6	Read + Write	rw-
7	Read + Write + Execute	rwX

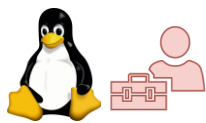
Quando usamos o comando **"chmod xxx file"** cada **digito** vai se referir a um **tipo de permissão**, primeiro digito é pra user, segundo digito pra group e terceiro digito para other:

```
CIANDT\lucasst@lnb024183spo: ~  
CIANDT\lucasst@lnb024183spo:~$ ls -l batman  
-rw--w---- 1 CIANDT\lucasst CIANDT\domain\users 0 ago  4 13:06 batman  
CIANDT\lucasst@lnb024183spo:~$ chmod 000 batman  
CIANDT\lucasst@lnb024183spo:~$ ls -l batman  
-----1 CIANDT\lucasst CIANDT\domain\users 0 ago  4 13:06 batman
```

Removendo todas as permissões para todos os grupos

```
CIANDT\lucasst@lnb024183spo: ~  
CIANDT\lucasst@lnb024183spo:~$ ls -l batman  
-----1 CIANDT\lucasst CIANDT\domain\users 0 ago  4 13:06 batman  
CIANDT\lucasst@lnb024183spo:~$ chmod 007 batman  
CIANDT\lucasst@lnb024183spo:~$ ls -l batman  
-----rwx 1 CIANDT\lucasst CIANDT\domain\users 0 ago  4 13:06 batman  
CIANDT\lucasst@lnb024183spo:~$
```

Dando todas as permissões para o tipo "outros"



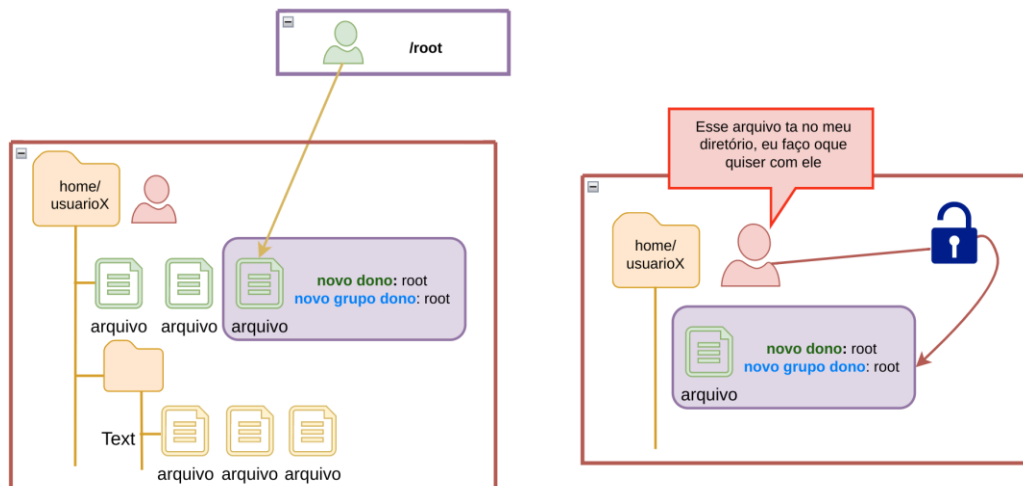
Donos de arquivos – Linux

Arquivos/diretórios possuem um owner(dono). Se por algum motivo você quiser **mudar o dono de um determinado arquivo**, primeiro você precisa subir sua permissão para o **"/root"** (a raiz de todo o sistema) com **"su -"**.

Então pode fazer um **"chown [novoDono] [arquivo]"** para mudar o **usuário dono do arquivo**. O mesmo pode ser feito para alterar o **grupo dono** do arquivo **"chgrp [novoDono] [arquivo]"**.

O ponto principal aqui é entender que **arquivos associados a determinados donos têm que ficar no level onde um terceiro não possa alterar**.

Por exemplo, se eu **mudar os donos de um arquivo** que está dentro da pasta de **usuário X** para **"root"**, ainda sim o **usuário X** poderia deletar, pois esse arquivo **estaria dentro do diretório desse usuário** que por sua vez possui **totais permissões sobre ele**:



Então a ideia seria trazer o **arquivo que não pode ser modificado** por ninguém para o **nível de root**, dessa maneira nenhum usuário poderia rescrever qualquer tipo de acesso e permissão, a não ser o

próprio root:

