

Introdução ao GNU/Linux

Comandos e Scripts

Felipe Dias de Oliveira



Escola Técnica Estadual
Governador Eduardo Campos

fdoprof@gmail.com

5 de novembro de 2021

1 Introdução

- O que é Software Livre?
- O que é Software Livre?
- Um Breve histórico...
- O que são Distribuições?
- Algumas Distribuições...

2 Primeiros Passos

- Sistema de Arquivos e Diretórios
- Manipulando arquivos e diretórios
- Coringas
- Comandos Úteis

Lista o conteúdo do diretório atual:

ls

O comando **ls** possui muitos parâmetros, veremos aqui as opções mais utilizadas.

Lista o conteúdo do diretório atual:

ls

O comando **ls** possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o **-l** que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

`d` – indica que se trata de um diretório

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

`d` – indica que se trata de um diretório

`l` – indica que se trata de um link (como se fosse um atalho)

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

`d` – indica que se trata de um diretório

`l` – indica que se trata de um link (como se fosse um atalho)

`-` – hífen, indica que se trata de um arquivo

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

`d` – indica que se trata de um diretório

`l` – indica que se trata de um link (como se fosse um atalho)

`-` – hífen, indica que se trata de um arquivo

`c` – indica dispositivo de caractere

Lista o conteúdo do diretório atual:

```
# ls
```

O comando `ls` possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o `-l` que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

`d` – indica que se trata de um diretório

`l` – indica que se trata de um link (como se fosse um atalho)

`-` – hífen, indica que se trata de um arquivo

`c` – indica dispositivo de caractere

`b` – indica dispositivo de bloco

Lista o conteúdo do diretório atual:

```
# ls
```

O comando **ls** possui muitos parâmetros, veremos aqui as opções mais utilizadas.

A primeira dela é o **-l** que lista os arquivos ou diretórios de uma forma bem detalhada (quem criou, data de criação, tamanho, dono e grupo a qual eles pertencem).

```
# ls -l /
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

d – indica que se trata de um diretório

l – indica que se trata de um link (como se fosse um atalho)

- – hífen, indica que se trata de um arquivo

c – indica dispositivo de caractere

b – indica dispositivo de bloco

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

O campo `rw-r-xr-x` lista as **permissões**, enquanto os campos `root` indicam quem é o **usuário** e **grupo** dono desse diretório que, no nosso caso, é o administrador do sistema, o `root`.

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

O campo `rwxr-xr-x` lista as permissões, enquanto os campos `root` indicam quem é o usuário e grupo dono desse diretório que, no nosso caso, é o administrador do sistema, o `root`.

O número antes do dono indica o número de `hard links`.

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

O campo `rwxr-xr-x` lista as permissões, enquanto os campos `root` indicam quem é o usuário e grupo dono desse diretório que, no nosso caso, é o administrador do sistema, o `root`.

O número antes do dono indica o número de hard links.

O campo `1024` indica o tamanho do arquivo, e o campo `2019-10-15 23:17` informa a data e hora em que o diretório foi criado.

```
drwxr-xr-x4 root root 1024 2019-10-15 23:17 boot
```

O campo **rwxr-xr-x** lista as permissões, enquanto os campos **root** indicam quem é o usuário e grupo dono desse diretório que, no nosso caso, é o administrador do sistema, o root.

O número antes do dono indica o número de hard links.

O campo **1024** indica o tamanho do arquivo, e o campo **2019-10-15 23:17** informa a data e hora em que o diretório foi criado.

Finalmente, no último campo temos o **nome do arquivo** ou diretório listado, que, no nosso exemplo, é o **boot**.

A opção **a** lista todos arquivos, inclusive os ocultos:

A opção `a` lista todos arquivos, inclusive os ocultos:

```
# ls -a /root
```

A opção `a` lista todos arquivos, inclusive os ocultos:

```
# ls -a /root
```

```
..aptitude.bashrc.profile .rnd.ssh.vmware
```

```
.. .bash_history .kde .qt root_161206 .viminfo .Xauthority
```

No Linux, arquivos e diretórios ocultos são iniciados por um “.” (ponto).

A opção `a` lista todos arquivos, inclusive os ocultos:

```
# ls -a /root
```

```
..aptitude.bashrc.profile .rnd.ssh.vmware
```

```
.. .bash_history .kde .qt root_161206 .viminfo .Xauthority
```

No Linux, arquivos e diretórios ocultos são iniciados por um “.” (ponto).

Lista arquivos de forma recursiva, ou seja, lista também os subdiretórios que estão dentro do diretório `/`:

A opção `a` lista todos arquivos, inclusive os ocultos:

```
# ls -a /root
..aptitude.bashrc.profile .rnd.ssh.vmware
.. .bash_history .kde .qt root_161206 .viminfo .Xauthority
```

No Linux, arquivos e diretórios ocultos são iniciados por um “.” (ponto).

Lista arquivos de forma recursiva, ou seja, lista também os subdiretórios que estão dentro do diretório `/`:

```
# ls -R /
```

A opção `a` lista todos arquivos, inclusive os ocultos:

```
# ls -a /root
.. aptitude.bashrc.profile .rnd.ssh.vmware
. .bash_history .kde .qt root_161206 .viminfo .Xauthority
```

No Linux, arquivos e diretórios ocultos são iniciados por um “.” (ponto).

Lista arquivos de forma recursiva, ou seja, lista também os subdiretórios que estão dentro do diretório `/`:

```
# ls -R /
```


Eles podem substituir uma palavra completa ou somente uma letra, seja para listar, copiar, apagar, etc. São usados três tipos de coringas no GNU/Linux:

Eles podem substituir uma palavra completa ou somente uma letra, seja para listar, copiar, apagar, etc. São usados três tipos de coringas no GNU/Linux:

- * - Utilizado para um nome completo ou restante de um arquivo/diretório;

Eles podem substituir uma palavra completa ou somente uma letra, seja para listar, copiar, apagar, etc. São usados três tipos de coringas no GNU/Linux:

- * - Utilizado para um nome completo ou restante de um arquivo/diretório;
- ? - Esse coringa pode substituir uma ou mais letras em determinada posição;

Eles podem substituir uma palavra completa ou somente uma letra, seja para listar, copiar, apagar, etc. São usados três tipos de coringas no GNU/Linux:

- * - Utilizado para um nome completo ou restante de um arquivo/diretório;
- ? - Esse coringa pode substituir uma ou mais letras em determinada posição;
- [padrão]** - É utilizado para referência a uma faixa de caracteres de um arquivo/diretório;

Eles podem substituir uma palavra completa ou somente uma letra, seja para listar, copiar, apagar, etc. São usados três tipos de coringas no GNU/Linux:

- * - Utilizado para um nome completo ou restante de um arquivo/diretório;
- ? - Esse coringa pode substituir uma ou mais letras em determinada posição;
- [padrão]** - É utilizado para referência a uma faixa de caracteres de um arquivo/diretório;

[a-z][0-9] - Usado para trabalhar com caracteres de a até z seguidos de um caractere de 0 até 9;

`[a-z][0-9]` - Usado para trabalhar com caracteres de a até z seguidos de um caractere de 0 até 9;

`[a,z][1,0]` - Usado para trabalhar com os caracteres a e z seguidos de um caractere 1 ou 0 naquela posição;

[a-z][0-9] - Usado para trabalhar com caracteres de a até z seguidos de um caractere de 0 até 9;

[a,z][1,0] - Usado para trabalhar com os caracteres a e z seguidos de um caractere 1 ou 0 naquela posição;

[a-z,1,0] - Faz referência do intervalo de caracteres de a até z ou 1 ou 0 naquela posição;

[a-z][0-9] - Usado para trabalhar com caracteres de a até z seguidos de um caractere de 0 até 9;

[a,z][1,0] - Usado para trabalhar com os caracteres a e z seguidos de um caractere 1 ou 0 naquela posição;

[a-z,1,0] - Faz referência do intervalo de caracteres de a até z ou 1 ou 0 naquela posição;

Lembrando que os 3 tipos de coringas mais utilizados (“*],[]”) podem ser usados juntos.

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls
```

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Vamos listar todos os arquivos do diretório /home/usuário. Podemos usar o coringa “*” para visualizar todos os arquivos do diretório:

Lembrando que os 3 tipos de coringas mais utilizados (“*],[]”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Vamos listar todos os arquivos do diretório /home/usuário. Podemos usar o coringa “*” para visualizar todos os arquivos do diretório:

```
# cd /home/usuário
```

Lembrando que os 3 tipos de coringas mais utilizados (“*],[”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Vamos listar todos os arquivos do diretório /home/usuário. Podemos usar o coringa “*” para visualizar todos os arquivos do diretório:

```
# cd /home/usuário  
# ls *
```

Lembrando que os 3 tipos de coringas mais utilizados (“*],[]”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Vamos listar todos os arquivos do diretório /home/usuário. Podemos usar o coringa “*” para visualizar todos os arquivos do diretório:

```
# cd /home/usuário  
# ls *  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Lembrando que os 3 tipos de coringas mais utilizados (“*],[]”) podem ser usados juntos.

Vejamos alguns exemplos:

Supondo que existam 5 arquivos no diretório /home/usuário. Podemos listá-los:

```
# ls  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Vamos listar todos os arquivos do diretório /home/usuário. Podemos usar o coringa “*” para visualizar todos os arquivos do diretório:

```
# cd /home/usuário  
# ls *  
arq1.txt arq2.txt arq3.txt arq4.new arq5.new
```

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com `.txt`:

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

```
# ls *new*  
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com `.txt`:

```
# ls *.txt
```

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

```
# ls *new*  
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com `.txt`:

```
# ls *.txt
```

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório `/home/usuário` que tenham "new" no nome:

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com .txt:

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com .txt:

```
# ls *.txt
```

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com .txt:

```
# ls *.txt
```

Listar todos os arquivos que começam com o nome arq, tenham qualquer caractere no lugar do coringa, e terminem com .txt:

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com .txt:

```
# ls *.txt
```

Listar todos os arquivos que começam com o nome arq, tenham qualquer caractere no lugar do coringa, e terminem com .txt:

```
# ls arq?.txt
```

Manipulando arquivos e diretórios

Para listarmos todos os arquivos do diretório /home/usuário que tenham "new" no nome:

```
# ls *new*
```

```
arq4.new arq5.new
```

Listar todos os arquivos que começam com qualquer nome e terminam com .txt:

```
# ls *.txt
```

Listar todos os arquivos que começam com o nome arq, tenham qualquer caractere no lugar do coringa, e terminem com .txt:

```
# ls arq?.txt
```

Para listar todos os arquivos que começam com o nome arq, tenham qualquer caractere entre o número 1-3 no lugar da 4ª letra e terminem com .txt.

Para listar todos os arquivos que começam com o nome arq, tenham qualquer caractere entre o número 1-3 no lugar da 4ª letra e terminem com .txt.

Neste caso, se obtém uma filtragem mais exata, pois o coringa especifica qualquer caractere naquela posição e [] especifica números, letras ou intervalo que serão usados.

Para listar todos os arquivos que começam com o nome `arq`, tenham qualquer caractere entre o número 1-3 no lugar da 4ª letra e terminem com `.txt`.

Neste caso, se obtém uma filtragem mais exata, pois o coringa especifica qualquer caractere naquela posição e `[]` especifica números, letras ou intervalo que serão usados.

```
# ls arq[1-3].txt
```

Para listar todos os arquivos que começam com o nome `arq`, tenham qualquer caractere entre o número 1-3 no lugar da 4ª letra e terminem com `.txt`.

Neste caso, se obtém uma filtragem mais exata, pois o coringa especifica qualquer caractere naquela posição e `[]` especifica números, letras ou intervalo que serão usados.

```
# ls arq[1-3].txt
```

Manipulando arquivos e diretórios

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new
```

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new
```

```
# ls *new*
```

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new
```

```
# ls *new*
```

```
# ls arq?.new
```

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new
```

```
# ls *new*
```

```
# ls arq?.new
```

```
# ls arq[4,5].*
```

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new  
# ls *new*  
# ls arq?.new  
# ls arq[4,5].*  
# ls arq[4,5].new
```

Para listar somente `arq4.new` e `arq5.new` podemos usar os seguintes métodos:

```
# ls *.new  
# ls *new*  
# ls arq?.new  
# ls arq[4,5].*  
# ls arq[4,5].new
```