

01

Trocando de usuários

Transcrição

Aprendemos anteriormente como buscar por arquivos e programas no sistema operacional com o comando `locate`, Porém o Terminal retorna uma lista extensa de arquivos que possuem em seu nome, parte do texto que usamos para pesquisar. Mas, qual é o arquivo que será executado quando digitamos o comando `vi`, `gedit` ou mesmo o `firefox`?

Para saber onde os programas estão instalados usamos o comando `which`, seguido do nome do programa que desejamos localizar:

```
guilherme@ubuntu:~$ which firefox
/usr/bin/firefox
guilherme@ubuntu:~$ which vi
/usr/bin/vi
guilherme@ubuntu:~$ which gedit
/usr/bin/gedit
guilherme@ubuntu:~$ █
```

Os programas que executamos antes estão todos dentro de um mesmo diretório chamado `/usr/bin`. Podemos listar todos os programas dentro desta pasta usando o comando `ls` da seguinte forma: `ls /usr/bin`, nos será listado uma lista enorme de programas do Linux.

Já que os programas ficam instalados nesse diretório, vamos colocar aquele arquivo/programa que criamos em aulas passadas, o `dorme`, dentro desse diretório. Isso com a ajuda do comando `mv`:

```
mv dorme /usr/bin
```

Teremos um erro:

```
guilherme@ubuntu:~$ mv dorme /usr/bin
mv: cannot move 'dorme' to '/usr/bin/dorme': Permission denied
```

O erro informa que não há permissão para mover o arquivo `dorme` para dentro do diretório `/usr/bin`. De fato, se verificarmos as permissões para o diretório `usr`, verificaremos que, somente o usuário `root` tem permissão de escrita para o mesmo.

```
guilherme@ubuntu:~$ ls -l /usr
total 100
drwxr-xr-x    2 root root 40960 Jun 11 18:42 bin
```

Perceba que para o diretório "bin", quem é dono tem permissão de leitura, escrita e execução, quem estiver no mesmo grupo tem de leitura e execução, e apenas execução para outros usuários. Então o problema é: como não somos "`root`" (o principal usuário da máquina), não podemos escrever dentro do diretório.

É preciso um cuidado especial ao realizar tarefas como `root`. Este usuário por ser o principal no sistema operacional, tem permissões para todas as operações, inclusive destrutivas, que farão o sistema parar de funcionar completamente.

Por ser um diretório muito importante, faz sentido não termos permissão de escrita para ele. Mas, mesmo assim, vamos fazer com que nosso programa `dorme` seja movido para dentro dele, uma vez que queremos que o programa seja executado por todos os usuários.

Para isso precisaremos mover o programa para o diretório como usuário `root`, ou seja, como se nosso usuário padrão fosse o usuário `root`, para isso, antes do comando usamos um comando auxiliar, o `sudo`:

```
sudo mv dorme /usr/bin
```

O Terminal pedirá a senha de usuário. Depois disso conseguimos executar esse comando como se fôssemos o usuário principal. Para termos certeza de que conseguimos mover o `dorme` para dentro do diretório `bin`, podemos executar:

```
ls /usr/bin | grep dorme
```

Ou até mesmo usar o comando `which`:

```
which dorme
```

No primeiro caso, se o programa `dorme` estiver dentro da `bin` ele será listado, e no segundo caso, o caminho `/usr/bin/dorme` será exibido. Agora podemos executar o `dorme` sem adicionar o `sh` ou o `./`.

Gerenciando senhas de usuários

Vamos mudar a senha do nosso usuário `guilherme`, para isso usamos o comando `passwd`. O Terminal irá pedir a senha atual, se houver, e depois duas vezes a nova senha.

Esta senha, será para o usuário padrão, para o usuário `guilherme`, caso queiramos trocar ou colocar uma senha para o usuário `root` fazemos uso do comando `sudo` da seguinte forma: `sudo passwd`.

```
guilherme@ubuntu:~$ passwd
Changing password for guilherme.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
guilherme@ubuntu:~$ sudo passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
guilherme@ubuntu:~$ █
```

Podemos nos logar como outro usuário, para isso usamos o comando `su` seguido do nome do usuário com qual queremos logar:

```
su root
```

Assim podemos nos logar como o usuário *root*, basta que digitemos a senha do usuário, podemos descobrir também com qual usuário estamos logado usando o comando `whoami`. E podemos sair da sessão de um usuário usando o comando `exit`.