

01

## sh e chmod

### Transcrição

Até agora nós usamos os programas através do Terminal, programas como o *Firefox*, o *VI* e o *gedit*. Digitamos seus nomes e eles abrem. Vamos aprender nessa aula como *criar* nosso próprio programa, O **bash** suporta uma linguagem que nos permite criar nossos próprios programas, nossos próprios *scripts*. Vamos primeiramente criar um arquivo de *gedit*, já em *background*, com o nome `dorme` :

```
gedit dorme &
```

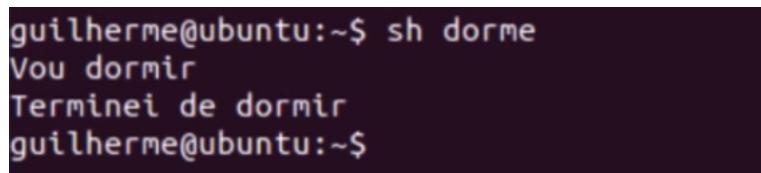
Dentro do editor, escrevemos o pequeno *script* que exibirá uma mensagem de `vou dormir` e depois de 5 segundos, exibe a mensagem `Terminei de dormir`, assim teremos:

```
echo "Vou dormir"
sleep 5
echo "Terminei de dormir"
```

Podemos executar nosso programa digitando o comando `dorme` , ou seja, o nome do programa, porém, caso façamos isso, teremos uma mensagem de erro, algo como: `dorme: command not found` . Isso por que o bash não conseguiu encontrar nosso programa, podemos realizar a execução direta do programa pedindo para o *bash* executá-lo da seguinte forma com comando `sh` :

```
sh dorme
```

Da forma que escrevemos o programa, no Terminal irá aparecer a primeira mensagem `Vou dormir` e, depois de 5 segundos, a segunda mensagem `Terminei de dormir` e o programa encerra-se.



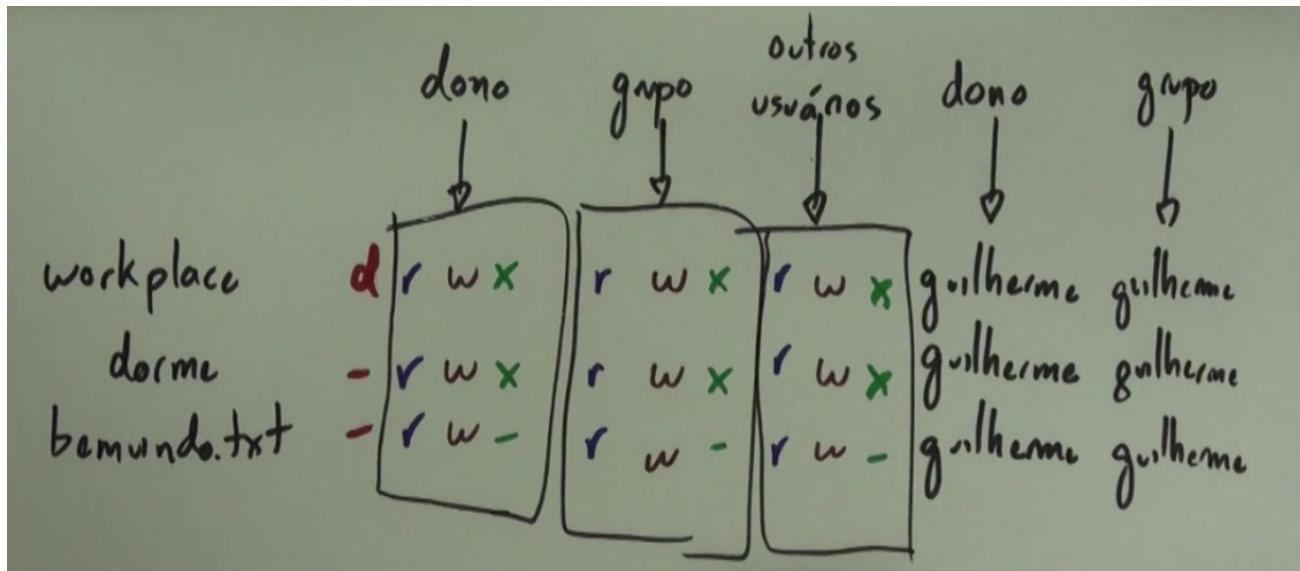
A terminal window showing the execution of a script named 'dorme'. The output shows the text 'Vou dormir' followed by 'Terminei de dormir', indicating the script's functionality.

```
guilherme@ubuntu:~$ sh dorme
Vou dormir
Terminei de dormir
guilherme@ubuntu:~$
```

Perceba que para programas como o *Firefox* ou o *gedit* não é necessário digitar o comando `sh` antes, mesmo sendo eles scripts ou programas binários, veremos como fazer isso, porém, precisamos falar um pouco sobre *permissões*.

### Permissões

Os arquivos no Linux podem ter permissões para **leitura** (r), **escrita** (w) e **execução** (x). Essas permissões são distribuídas para o dono do arquivo, ao grupo de usuários e também para outros usuários. Os **diretórios** (d) sofrem das mesmas regras. A imagem abaixo ilustra bem as permissões e suas distribuições.



Com o comando `ls -l`, podemos verificar a listagem de arquivos com todas as informações de permissões discutidas. Para o arquivo `dorme` por exemplo, temos as seguintes:

```
-rw-rw-r-- 1 guilherme guilherme 52 Jun 12 10:48 dorme
```

O primeiro dígito `-` informa que este item não é um diretório, os três seguintes indicam que o usuário dono do arquivo tem as permissões de leitura e escrita, o mesmo se repete para o grupo e por último, vemos que publicamente, ou seja, para outros usuários só há a permissão de leitura. Depois das informações de permissão temos o nome do usuário dono do arquivo e o grupo ao qual o usuário pertence.

Se quisermos que o arquivo `dorme` possa ser executado, ele precisará dessa permissão. Então vamos adicionar a permissão de execução para o *script* `dorme` para todos os usuários com a ajuda do comando `chmod`:

```
chmod +x dorme
```

Dessa maneira adicionamos a capacidade de execução ao arquivo `dorme` (para retirá-la usamos `-x`), utilizando o comando `ls -l` novamente podemos verificar se a mudança realmente foi aplicada:

```
-rwxrwxr-x 1 guilherme guilherme 52 Jun 12 10:48 dorme
```

Perceba que a permissão de execução foi aplicada tanto para dono e grupo, quanto para outros usuários. Ainda não podemos simplesmente digitar o nome do *script* por questões de localização, então, para executá-lo teremos que usar `./`, para indicar que o *script* está salvo no diretório atual:

```
./dorme
```

Assim como temos o `+x` para adicionar permissão de execução, temos também o `+r` e o `+w`. Podemos também combinar essas permissões da seguinte forma: `chmod +rwx`.