

Caracteres coringas no bash

Transcrição

Aprendemos a criar um diretório com o comando `mkdir`, mas como podemos apagar um diretório? E um arquivo? Removemos um diretório usando o comando `rmdir` e para remover um arquivo utilizamos o comando `rm`, dentro de `workspace` por exemplo podemos fazer:

```
rmdir projetos-java
rm arquivo3.txt
```

Removemos/Apagamos o diretório `projetos-java` e o arquivo `arquivo3.txt`. Uma observação interessante é que o comando para remover diretórios só irá funcionar para aqueles diretórios que estiverem vazios. Ele não apaga os arquivos dentro do diretório automaticamente.

Dentro do diretório `workspace` atualmente há três arquivos e um diretório (*use o comando `ls` para listar os arquivos*):

```
arquivo10.txt arquivo1.txt arquivo2.txt projetos-php
```

No `arquivo10.txt` temos a frase "Bem vindo" e nos outros dois a frase "meu primeiro teste". Para que o terminal imprima os textos de todos os arquivos com um determinado nome, por exemplo, "`arquivo...txt`" podemos fazer:

```
cat arquivo?.txt
```

O resultado será:

```
meu primeiro teste
meu primeiro teste
```

Também podemos usar o caractere `*`:

```
cat arquivo*.txt
```

Mas o resultado será um pouco diferente do primeiro caso:

```
Bem vindo
meu primeiro teste
meu primeiro teste
```

Esses caracteres de `?` e `*` são utilizados para que sejam buscados mais de um arquivo que tenham o mesmo nome base, porém existe uma pequena diferença: O `?` só encontra os arquivos que tenham apenas UM caractere diferente do nome base, enquanto o `*` busca quaisquer números de caracteres.

Por isso que o arquivo `arquivo10.txt` é encontrado no segundo exemplo, mas não no primeiro. O `*` consegue lidar com o `10` depois do trecho `arquivo` no nome do arquivo, enquanto o `?` não consegue, pois são dois caracteres em `10`.

Atenção: se buscarmos o arquivo como `"*.txt"`, ou seja, com aspas, o terminal interpreta o asterisco como se o mesmo fosse parte do nome do arquivo:

```
cat "*.txt"
```

O comando acima nos trará uma mensagem de erro como a seguinte:

```
cat: *.txt: No such file or directory
```

Comentamos anteriormente que não podíamos apagar um diretório que possuía outros subdiretórios ou arquivos. Na verdade podemos, com o comando `rm -r` isto é possível. Ele remove **Recursivamente**, ou seja, apaga o diretório e tudo que está dentro dele. Podemos fazer por exemplo:

```
rm -r workspace/
```

Dessa forma apagamos todo o diretório `workspace` e tudo que havia dentro dele. Muito cuidado na hora de utilizar este comando para não apagar diretórios importantes.