

Redirecionamento e diretórios

Transcrição

Como vimos na aula passada, o comando `ls` nos lista os arquivos e diretórios no diretório atual.

```
guilherme@ubuntu:~$ ls
bemvindo.txt  Documents  examples.desktop  Pictures  Templates
Desktop       Downloads  Music             Public    Videos
guilherme@ubuntu:~$
guilherme@ubuntu:~$
```

Vimos também que o comando `pwd` nos informa o diretório atual, algo como `/home/guilherme`, mas e se quiséssemos mudar de diretório? Para mudarmos de diretório, podemos utilizar o comando `cd` - *Change Directory*:

```
cd Desktop
```

O comando `pwd` agora nos trará um resultado diferente. Algo como `/home/guilherme/Desktop`. Vamos exercitar a criação de arquivos novamente com o comando `echo`, lembra dele? Vamos criar um arquivo que lista algumas músicas. Por exemplo:

No Ubuntu, Fedora e talvez em outras distros Linux que suportam o nosso português (pt_BR), a tradicional pasta "Desktop" foi renomeada para "Área de Trabalho" e, como os iniciados sabem, ela precisa ser acessada pelo terminal com o uso de barras invertidas ("") para designar os espaços, ou seja:

```
cd Área\ de\ Trabalho/
```

Os iniciantes muitas vezes esquecem ou não sabem disso, sem falar que eles podem vir a executar scripts que ainda usam a designação tradicional para essa pasta, ou seja, "Desktop".

```
echo "Faithless" > musicas-favoritas.txt
```

Lembre-se de verificar a criação de arquivos e diretórios pelo explorador de arquivos. Só por questões de confirmação.

Criamos o arquivo `musicas-favoritas.txt` com o texto `Faithless` dentro do diretório `Desktop`. Vamos tentar colocar mais uma música dentro desse arquivo utilizando o comando `echo` da mesma maneira:

```
echo "REM" > musicas-favoritas.txt
```

Se utilizarmos o comando `cat` para ver o conteúdo do arquivo, teremos uma surpresa: Apenas a música `REM` estará listada no arquivo. Isso por que o comando `echo` escreve no arquivo, criando o arquivo caso não exista. Caso exista, o conteúdo do arquivo será **sobrescrito**. Não queremos sobrescrever, queremos adicionar um texto, concatená-lo com o que já está inserido no arquivo. Para tal usamos `>>` em vez de apenas um `>`:

```
echo "Faithless" >> musicas-favoritas.txt
```

Para confirmar a adição da segunda música, podemos ler o arquivo com o comando `cat`.

```
cat musicas-favoritas.txt
```

E teremos como resultado a saída:

```
REM  
Faithless
```

Até este ponto, criamos arquivos, listamos, entramos em um diretório, mas como voltamos para o diretório anterior? Para voltarmos para o diretório anterior usamos o comando `cd ..`. Existem também o diretório `.`, que referência o diretório atual.

Podemos sempre confirmar o diretório atual usando o comando `pwd`. Utilize-o para confirmar a troca de diretórios.

Já sabemos como navegar entre diretórios. Vamos aprender agora a criar um diretório, para isso utilizamos o comando `mkdir` seguido do nome do diretório que queremos criar:

```
mkdir workspace
```

Criamos o diretório `workspace` dentro do diretório `/home/guilherme`. Perceba que os mesmos comandos utilizados anteriormente para criação e leitura de arquivos e diretórios também funcionam em subdiretórios. Experimente criar outros arquivos e subdiretórios dentro do `workspace`, como por exemplo: `projetos-java` e `projetos-php`.

```
echo "meu primeiro teste" > arquivo1.txt  
echo "meu primeiro teste" > arquivo2.txt  
echo "meu primeiro teste" > arquivo3.txt
```

```
mkdir projetos-java  
mkdir projetos-php
```

Navegue por esses subdiretórios usando o comando `cd`. Use o atalho do **TAB** para não precisar digitar o nome completo dos diretórios na navegação, mas note que o TAB não irá completar o nome do diretório totalmente, por que depois do traço, os nomes são diferentes, então você precisa digitar manualmente pelo menos as primeiras letras depois de `projetos-` para poder usar o TAB novamente.

Depois de entrar no diretório `projetos-java`, o comando `pwd` nos trará o seguinte retorno:

```
/home/guilherme/workspaces/projetos-java
```

Agora, se estivéssemos dentro de um subdiretório que, por sua vez, estivesse dentro de outro (*como fizemos agora*), e quiséssemos ir para o diretório base, como faríamos? O comando `cd`, sozinho, sem pontos, fará isso. Não importa onde estejamos, com o comando `cd` sempre voltamos para o diretório base. Que neste caso é:

```
/home/guilherme/
```

O diretório base não é a raiz de tudo. Para visualizarmos a raiz do *HD* digitamos `ls /`. O `/` possui diversos outros diretórios úteis para o funcionamento do Linux. O diretório `home` por exemplo, é onde estão os diretórios de cada um dos usuários que utilizam a máquina.

Nesta aula aprendemos a criar diretórios, concatenar textos dentro de um arquivo e navegar entre diretórios.