03

Removendo extensão e salvando em outro diretório

Transcrição

Criamos um script capaz de fazer a varredura em todos os arquivos `.jpg` que existiam dentro de `imagens-livros`. Vimos que o teste funcionou parcialmente, pois a conversão do arquivo `.jpg` para `.png`, não teria ".jpg" no nome. Por exemplo: "algortimos.jpg.png". Queremos manter apenas o nome do arquivo original.

Faremos alguns testes no terminal e depois acessaremos o script.

Acessaremos a pasta `imagens-livros` e listaremos o conteúdo do diretório com o comando `ls`.

```
$ cd ~/Downloads/imagens-livros/
$ ls
```

Podemos também listar um arquivo acrescentando seu nome após o `ls`:

```
$ ls algoritmos.jpg
```

Foram feitas algumas pesquisas e vimos que alguns comandos conseguem **manipular** textos. Um deles é o `awk`!

É necessário redirecionar a saída, para que o `awk` possa tratá-la. Para redirecionar o comando para a saída, usamos o `|` (lê-se *pipe*). Especificamos para o `awk` qual será o **campo delimitador** e qual será o local onde faremos o corte na mensagem exibida.

Onde seria esse corte? Justamente no ***ponto*** da mensagem, em `algoritmos.jpg`, faremos o corte entre a palavra "algoritmos" e a extensão `jpg`.

```
$ ls algoritmos.jpg | awk -F. '{ print $1 }'
```

O `awk` irá separar a mensagem em duas partes. O comando `-F` define o local de corte. E com o `'{ print $1 }'` conseguimos imprimir a *primeira* parte da mensagem.

Como o comando funcionou no terminal, ele **deve** funcionar no script. Copiaremos o comando testado e o colaremos dentro do script. Mas, antes, vamos acessá-lo:

```
$ cd ~/Scripts/
$ nano conversao-jpg-png.sh
```

Dentro do laço `for` pegaremos o conteúdo da `imagem`, que está com a extensão `.jpg`, e usaremos o comando que criamos para usar somente o resultado sem a extensão.

```
#!/bin/bash

cd ~/Downloads/imagens-livros
```

```
for imagem in *.jpg
do
    ls algoritmos.jpg | awk -F. '{ print $1 }'
    convert $imagem $imagem.png
done
```

Todavia, pegaremos o **conteúdo** que virá de `imagem`. Vamos aproveitar para *armazenar* somente o resultado em uma outra variável, que chamaremos de `imagem_sem_extensao`.

O armazenamento ocorre quando colocamos `$()` no comando.

```
#!/bin/bash

cd ~/Downloads/imagens-livros

for imagem in *.jpg
do
    imagem_sem_extensao=$(ls $imagem | awk -F. '{ print $1 }')
    convert $imagem_sem_extensao.jpg $imagem_sem_extensao.png
done
```

Salvamos o script e antes de executá-lo removeremos todo o conteúdo `.png` que está no diretório `imagens-livros`.

```
$ cd ~/Downloads/imagens-livros/
$ rm *.png
```

Nesse ponto temos somente o conteúdo `.jpg`. Vamos executar o nosso script.

```
$ ~/Scripts/
$ bash conversao-jpg-png.sh
```

Certo! Vamos acessar a pasta para ver o resultado. Veremos que foram mantidos somente o nome dos arquivos originais e a eles foi acrescentado a extensão `.png`

Os diretores da *Multillidae*, nos pediram algumas coisa a mais. A primeira delas é para que guardemos os arquivos convertidos em um diretório próprio chamado de `png`. Eles também pediram que seja mostrado uma mensagem de **sucesso** ou de **falha** uma vez que o script for executado.

Guardaremos as conversões no diretório `png`:

```
$ nano conversao-jpg-png.sh
```

Será que dentro do diretório `imagens-livros`, existe o diretório `png`? Vamos usar o `if` para fazer a verificação.

```
#!/bin/bash

cd ~/Downloads/imagens-livros
if [ -d png ]
```

```
for imagem in *.jpg
do
    imagem_sem_extensao=$(ls $imagem | awk -F. '{ print $1 }')
    convert $imagem_sem_extensao.jpg $imagem_sem_extensao.png
done
```

Essa verificação `if [ -d png ]` verifica somente se o diretório existe. Se o diretório não existe, ele vai ser criado. O símbolo de negação é `!`.

```
if [ ! -d png ]
```

Isso significa que, se não existir o diretório `png`, então ele será criado!

```
if [ ! -d png ]
then
    mkdir png
```

Uma vez que já passamos a condição, temos que finalizar o comando `if` com `fi`:

```
if [ ! -d png ]
then
    mkdir png
fi
```

tudo o que fizemos salvamos com "Ctrl + x" e "yes".

Antes de executar o script, vamos ao diretório `imagens-livros` e removemos todos os arquivos `.png`, pois eles serão salvos dentro do diretório `png`.

```
$ cd ~/Downloads/imagens-livros/
$ rm *.png
$ ls
```

Voltemos ao script.

```
$ bash conversao-jpg-png.sh
```

Vamos ver o resultado. Dentro de `imagens-livros` temos o diretório `png`, onde estão armazenado todos os arquivos convertidos nesse formato.

Tivemos um avanço, mas nosso trabalho não para por aqui. Assim que o script for executado, temos que saber se a conversão ocorreu com sucesso ou se houveram falhas.