03

Convertendo imagem e montando script

Transcrição

Nós fomos contratados pela *Multillidae*, para ajudá-los em algumas tarefas que eles terão nas semanas seguintes.

A *Multillidae* está com um projeto de abrir uma **loja virtual** e um dos setores dessa loja será o setor de livros de tecnologia. A *Multillidae* comprou alguns livros da *Casa do Código* para colocar na plataforma online de vendas.

Entretanto, a plataforma da *Multillidae* só aceita os arquivos no formato `.png`. Mas, os arquivos que foram passados da *Casa do Código* estão na extensão `.jpg`.

A missão que nos foi dada é justamente **encontrar uma forma de converter esses arquivos** de extensão `.jpg` para `.png`.

Os diretores da *Multillidae* nos passaram o link (https://drive.google.com/open?id=0BzmYQVmw4W7nUW40M2dfQWxlTm8) para o download das imagens.

Muito bem! Após ter feito o download das imagens, vamos abrir o terminal e verificar o diretório, para ver se de fato, o arquivo está lá na pasta "Downloads".

Como foi visto no curso de Linux, (se você ainda não fez o curso, clique aqui (https://cursos.alura.com.br/course/linux-ubuntu-processos?preRequirementFrom=shellscripting)), para mudar de diretório basta utilizar o comando seguido da pasta `cd Downloads/` e depois utilizar o comando `ls` que **listará** o conteúdo do diretório em que estamos.

Perceba que assim acessamos o arquivo com as imagens em `.jpg`, mas ele está compactado em **zip**. Vamos descompactá-lo utilizando o comando `unzip imagens-livros.zip`

Vamos listar o diretório com o comando `ls`. O resultado será esse:

```
imagens-livros imagens-livros.zip
```

O primeiro diretório é a pasta que contém as imagens a serem convertidas e o segundo diretório é o arquivo compactado. Podemos removê-lo já que ele não é mais necessário. Utilizando o comando `rm imagens-livros.zip` conseguimos deixar somente o que nos interessa.

Legal, vamos entrar nesse diretório para ver o conteúdo dele:

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

```
rafael@rafael-VirtualBox:~/Downloads$ cd imagens-livros/
rafael@rafael-VirtualBox:~/Downloads/imagens-livros$ ls
algoritmos.jpg       codeigniter.jpg    java_ee.jpg              node.jpg                 scala.jpg          windows_server.jpg
amazon_aws.jpg       cordova.jpg        jenkins.jpg              nosql.jpg                scratch.jpg        xamarin_forms.jpg
arduino_pratico.jpg  dsl.jpg            jquery.jpg               orientacao_objetos.jpg   seguranca.jpg      zend.jpg
asp_net.jpg          elasticsearch.jpg  mantra_produtividade.jpg  postgres.jpg             turbine_css.jpg
big_data.jpg         es6.jpg            metricas_ageis.jpg       sass.jpg                 vue.jpg
rafael@rafael-VirtualBox:~/Downloads/imagens-livros$
```

Temos aqui todos esse livros que precisamos realizar a conversão.

Depois que soubemos de nossa missão dentro da *Multillidae*, começamos a fazer algumas pesquisas. Vimos que no próprio Ubuntu existe uma ferramenta **capaz** de fazer essa conversão: o ***ImageMagick***!!!

Para realizar essa conversão, basta utilizar o comando `convert` e dizer qual arquivo queremos converter. De início, daremos um foco maior no arquivo `algoritmos.jpg`, para ter a certeza de que realmente o ImageMagick irá realizar essa conversão. Feito isso, diremos o nome e a extensão do arquivo para o qual iremos converter:

```
$ convert algoritmos.jpg algoritmos.png
```

Na sequência, usamos o comando para listar, o `ls`. Vimos que o arquivo `algoritmos.png` se encontra nesse mesmo diretório. Mas, será verdade? Vamos confirmar se realmente esse arquivo tem a extensão `.png`.

Maravilha! Ao observar verificamos que o **ImageMagick** conseguiu converter esse arquivo! Ele parece ser a solução ideal para nós. Mas, repare que há cerca de 25 livros que precisam sofrer a conversão de extensão e utilizar o comando `convert livro.jpg livro.png` várias vezes para cada um dos livros não é uma solução muito elegante. Imagine que amanhã, os diretores tenham mais 100 imagens para converter, então, nesse caso, teríamos que colocar 100 vezes o mesmo comando para cada livro. Não seria nada prático, não é mesmo?

Justamente em situações como essas que o **Shell Scripting** consegue ajudar a *automatizar essas tarefas*.

SHELL SCRIPTING

Podemos interpretar o **Shell** como uma *interface* que nós, usuários, acessamos os recursos no Sistema Operacional. Já a palavra **Scripting**, significa *roteiro* e é uma lista de comandos que serão interpretados pelo Sistema Operacional.

Montaremos um script capaz de realizar essa conversão para nós, de uma forma mais eficiente do que o usuário simplesmente colocar o comando no terminal para cada imagem. Automatizaremos essa tarefa.

Para criar o arquivo, precisamos de um **editor de texto**. Fique a vontade para escolher o editor de seu gosto: Nano (https://www.nano-editor.org/), Vi/Vim (http://ex-vi.sourceforge.net/), gEdit (https://wiki.gnome.org/Apps/Gedit) ou algum outro de sua preferência. Ao longo do curso, usaremos o **Nano** e o **gEdit**.

Vamos voltar a nossa "home" com o comando `cd` para criar um diretório no qual vamos guardar todos os scripts feitos durante o curso.

```
$ cd
$ mkdir Scripts
$ cd Scripts/
```

Após ter mudado para dentro da nova pasta "Scripts", criaremos o primeiro Script para converter a imagem do formato `.jpg` para `.png`. Com o comando `nano nome_de_um_arquivo.sh` criaremos e editaremos esse arquivo utilizando o editor **Nano**. Como estamos trabalhando com Shell Scripting colocaremos a extensão `.sh`:

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

Com o editor aberto, temos que dizer como esses comandos serão interpretados. Na primeira linha, vamos dizer qual vai ser o interpretador do script. Para isso, diremos por meio do comando:

```
#!/bin/bash
```

Após especificar o interpretador dos comandos que serão digitados a seguir, nos resta somente digitá-los para checar se o script funcionará. Para realizar a conversão, utilizamos o comando `convert` e converteremos o arquivo `amazon_aws.jpg` para o formato `.png`:

```
#!/bin/bash

covert amazon_aws.jpg amazon_aws.png
```

Repare que o arquivo `amazon_aws.jpg` está em um diretório diferente do nosso script. As imagens estão no diretório "imagens-livros", que está dentro de "Downloads". Os scripts, por sua vez, estão no diretório "Scripts". Por essa razão, é necessário colocar o caminho de onde esses arquivos estão, para que possamos ter a referência de onde eles estão localizados.

Tanto o diretório "Scripts" quanto o diretório "Downloads" estão dentro da **home**. Com isso, podemos usar o `~` pois eles estão dentro da home!

```
#!/bin/bash

convert ~/Downloads/imagens-livros/amazon_aws.jpg ~/Downloads/imagens-livros/amazon_aws.png
```

Queremos salvar a conversão `amazon_aws.png` no mesmo diretório dos arquivos `.jpg`, para que tudo fique no mesmo lugar.

Se tudo deu certo, vamos salvar o nosso script com o comando `Ctrl + X` para sair e `Y` de (yes) para salvar as alterações.

Como falamos que o interpretador do script é o **bash**, para rodar esse script, colocamos `bash` e o nome do script e depois teclamos o "Enter":

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

Já que estamos no diretório "Scripts", vamos acessar o diretório "Downloads/imagens-livros":

```
$ bash conversao-jpg-png.sh
$ cd ~/Downloads/imagens-livros.
$ ls
```

Observe, agora temos o `algoritmos.png` e temos o `amazon_aws.png` , que é o resultado da conversão que nosso script fez. Mas perceba que o script só está fazendo a conversão do arquivo `amazon_aws` , ou seja, nós tiramos o comando do terminal e colocamos no script e isso fez com que voltassemos ao início do problema, precisando digitar uma linha de comando para cada imagem a ser convertida.

Vamos ver como melhorar esse script nas próximas aulas!