

Aula 19 - arquivos .deb

Paulo Tozzo - paulotp@al.insper.edu.br

Fevereiro 2018

Hello.deb

Comece instalando o hello.deb usando o seguinte comando:

```
sudo apt install ./hello.deb
```

you can find the file in “/usr/local/bin” and run it using: `~ python hello ~`

Esse guia ensinará a criar esse programa e transformá-lo em um arquivo .deb # Criando a base de um arquivo .deb

Criar um arquivo .deb é bem fácil, primeiramente vamos criar uma pasta nova para conter o programa que será transformado e um .deb crie a pasta e chame ela de hello (pode ser outra coisa também mas todos os comandos desse guia assume que a pasta chama hello)

Pense que essa nova pasta é o root do sistema e o programa precisa estar no caminho desejado a partir desse root, portanto se quisermos instalar em “usr/local/bin” o programa precisa estar em “hello/usr/local/bin” assim para facilitar achar esse programa futuramente crie esse caminho

Todo .deb tem uma pasta especial chamado ‘Debian’ que será usada para guardar as configurações do .deb ele deve estar diretamente no root, ou seja hello/DEBIAN deve existir, crie ele agora.

Dentro da pasta Debian deve existir um arquivo chamado control, ele é um “sumário” do pacote um exemplo que pode ser usado é o abaixo:

```
Package: hello
Version: 1.0
Section: base
Priority: optional
Architecture: all
Depends: python (> 2.7)
Maintainer: Your Name <you@email.com>
Description: Hello World
    Super programa de teste
```

Note o espaço antes do Super programa... ele é necessário. Essas são somente algumas das opções que control pode ter se tiver interessado você pode ver o resto delas nesse link: <https://manpages.debian.org/wheezy/dpkg-dev/deb-control.5.en.html> Vamos criar um programa python chamado world para ser instalado dentro de hello/usr/local/bin :

Agora adicione algum código de python nele, pode ser um simples print. nota: se você usar alguma biblioteca não comum do python que precisa ser instalada você precisa adicionar ela nas dependências do control (ou pelo menos deveria)

preinst, postrm e config

control não é o único arquivo especial do debian também existe o preinst, postinst, prerm, postrm eles rodão antes ou depois da instalação ou remoção, esses arquivos precisam ter linhas de comando em formato bash script já que o terminal irá rodar esse arquivo na hora certa. Crie um arquivo chamado postinst na mesma pasta do controle com um código simples (pode ser um echo). Também crie um arquivo postrm com outro

código simples. esses arquivos devem ter permissão de serem executados portanto altera a permissão deles com o seguinte comando:

```
chmod +x ~/file
```

Também temos o `conffiles` é um arquivo com o caminho de todos as pastas ou arquivos de configuração do projeto, um arquivo de configuração não é apagado ao usar o comando `apt remove` e caso o programa tenha um update o `dpkg` pergunta se você quer o novo config ou quer continuar com o velho, ele é usado para guardar as configurações específicas do usuário. Com isso dito vamos fazer um config simples, crie um arquivo chamado `conffiles` na mesma pasta que o `control`, agora crie uma pasta chamada `etc` em “hello/” e coloque alguma coisa dentro dele pode ser um `.txt` sem, nada agora no arquivo `conffiles` adicione o caminho desse txt dessa maneira: `/etc/nome_do_arquivo.txt`

Build e Remove

Agora é bem simples é só fazer o build da pasta usando o seguinte comando:

```
dpkg-deb --build hello
```

Se tudo tiver dado certo deve ter aparecido um arquivo `.deb`, vamos instalar ele agora:

```
sudo apt install ./hello.deb
```

esse arquivo foi instalado em “`/usr/local/bin`” dê uma olhada nele (ubuntu tem o shortcut `ctrl+l` super útil), e o arquivo config deve estar em “`/etc`” junto com todos outros arquivos de configuração (é mais fácil usar “`ls /etc`” no terminal). Agora vamos remover o programa:

```
sudo apt remove hello
```

e é isso... mas o arquivo de config ainda existe, para removê-lo pode ser uso `purge` em vez de `remove`, `purge` remove também os arquivos de config e ele pode ser usado mesmo que o programa já tenha sido removido com `remove`.

DPKG e apt

usamos `apt` para instalar o programa certo? Na verdade o `apt` usa o `dpkg` para instalar um programa, portanto podemos instalar um programa usando diretamente o `dpkg` usando a seguinte linha: `~ sudo dpkg -i /caminho/para/o/.deb` ~ tudo certo, certo? Na verdade sim o programa esta instalado direitinho... ou menos se você tiver todas as dependências, um dos principais motivos para usar o `apt` é que esse instala as dependências enquanto o `dpkg` não, tanto que existe o comando `apt install -f` que olha as dependências faltando e conserta elas. Também quando você usa o `apt` instalar o `apt` olha em varias listas internas e vê se existe um repositório com esse pacote, já baixa o arquivo `.deb` e usa o `dpkg` para instalar ele, alem de olhar a dependências e instalar elas também. Se estiver interessado é possível olhar essas listas que se encontram em `/var/lib/apt/lists` elas são bem grandes, a minha ocupa 202.5 MB de espaço.

Agora acabou, so não esqueça de apagar o programa `hello` do computador usando `apt purge`.