

Como fazer um pacote .deb?

Gustavo Noronha Silva <dockov@zaz.com.br>

Nota de Copyright

Esse manual é livre e pode ser distribuído e utilizado sob os termos contidos na FDL da GNU.

© Gustavo Noronha Silva (KoV) 2000

Sumário

| | | |
|----------|---|----------|
| 1 | Antes de Começar | 1 |
| 1.1 | Por quê? | 1 |
| 1.2 | Do que preciso? | 1 |
| 2 | Começando | 3 |
| 2.1 | Preparando o pacote | 3 |
| 2.2 | Modificando o Makefile | 3 |
| 3 | Editando os arquivos de controle | 5 |
| 3.1 | Rodando o dh_make | 5 |
| 3.2 | changelog | 5 |
| 3.3 | control | 6 |
| 3.4 | copyright | 6 |
| 3.5 | dirs | 6 |
| 3.6 | docs | 6 |
| 3.7 | init.d.ex | 7 |
| 3.8 | manpage.1.ex | 7 |
| 3.9 | menu.ex | 7 |
| 3.10 | README.Debian | 7 |
| 3.11 | rules | 7 |
| 4 | Finalizando o empacotamento | 9 |
| 4.1 | Rodando o dpkg-buildpackage | 9 |
| 4.2 | Corrigindo erros | 9 |
| 4.3 | Atualizando o pacote | 9 |

5 Contato

11

Capítulo 1

Antes de Começar

1.1 Por quê?

Depois de inúmeros pedidos decidi fazer esse manual sobre como fazer um pacote .DEB, vou tentar fazer um resumo máximo aqui, já que o `maint-guide` que eu traduzi é bem resumido já.

1.2 Do que preciso?

Antes de empacotar, saiba que você deve ter uma assinatura GPG e pacotes próprios instalados. Pegue o `gnupg` (`apt-get install gnupg`) e leia sobre ele no manual que se encontra no CIPSGA (<http://www.cipsga.org.br>)

Não vou dar detalhes de pacotes nem nada...o objetivo é ser objetivo =)... caso algum comando dê errado ou não exista, leia o `maint-guide` ou use o `apt-get` para instalá-lo procurando o arquivo falho com `dpkg -S` ou usando o `auto-apt` (`auto-apt run comando`) para que ele auto-instale.

Usarei o programa `gklog`, de minha autoria como exemplo, é um programa simples e de simples instalação/utilização.

Capítulo 2

Começando

2.1 Preparando o pacote

Crie um diretório qualquer no lugar de sua preferência. Nesse diretório coloque o arquivo tar.gz do programa e descompacte-o assim:

```
tar zxpvf gklog-0.3.tar.gz
```

lembre-se de uma coisa...o programa gklog cria o diretorio gklog-0.3 quando é descompactado, isso é necessário e deve ser feito com seu programa preferido. *Deve-se* usar a convenção: *nome do prg-versão* para que corra tudo bem e o nome do prg deve ser compacto e não exceder, digamos, 20 letras.

```
Downloader for X -> d4x
```

2.2 Modificando o Makefile

O próximo passo é conferir o modo como o programa se instala. Isso é feito no Makefile do programa. É muito simples...basta que você faça com que esse programa se instale como se o diretório root fosse \$DESTDIR, uma variável que será setada pelo programa de empacotamento...vejamos um exemplo:

O Makefile do gklog estava assim no seu início e no método install (o que importa na maioria dos casos):

Início:

```
SHELL = /bin/bash  
CFLAGS = -O2
```

```
BIN = /usr/bin
```

Método Install:

```
install: all
    install -d /etc/init.d/
    install -d /etc/rc2.d/
    install -d /usr/bin/
    install gklog.sh /etc/init.d/
    ln -s /etc/init.d/gklog.sh $(DESTDIR)/etc/rc2.d/S90gklog
    install gklog.rules /etc/
    install gklog $(BIN)
```

A variável *BIN* estava setada para */usr/bin*. . . basta que mudemos isso assim:

Mude isso no começo

```
SHELL = /bin/bash
CFLAGS = -O2
# Edited for Debian GNU/Linux
DESTDIR =
```

```
BIN = $(DESTDIR)/usr/bin
```

E fica assim no install:

```
install: all
    install -d $(DESTDIR)/etc/init.d/
    install -d $(DESTDIR)/etc/rc2.d/
    install -d $(DESTDIR)/usr/bin/
    install gklog.sh $(DESTDIR)/etc/init.d/
    ln -s $(DESTDIR)/etc/init.d/gklog.sh $(DESTDIR)/etc/rc2.d/S90gklog
    install gklog.rules $(DESTDIR)/etc/
    install gklog $(BIN)
```

Isso possibilitará que os arquivos sejam instalados num diretório temporário (*tmp/*) dentro do diretório *debian* do seu programa possibilitando o empacotamento, mais tarde. Outra coisa a ser observada. . . um pacote *deb* **nunca** deve instalar nada em */usr/local* e sim em */usr* nem que você tenha que modificar o código fonte, garanta isso! =)

Capítulo 3

Editando os arquivos de controle

3.1 Rodando o `dh_make`

Bem... estamos prontos para começar a meter a mão na massa:

Rode isso:

```
dh_make -e seu@email.com.br -f ../gklog-0.3.tar.gz
```

É claro que você vai substituir as coisas com seus próprios dados. Ele pergunta o tipo de pacote... selecione 's' single binary, único binário. Isso porque o `gklog` só gera um binário e portanto um `.deb`, não é aconselhável criar multiple binary ou librarys sem ter lido o `maint-guide` e o `packaging manual`. Verifique os dados e confirme com ENTER

Note que um diretório `debian` foi criado. Nele estão os arquivos importantes do pacote Debian. Vamos a eles então.

3.2 changelog

Edite esse arquivo para colocar as mudanças que você vai fazendo no pacote a cada versão de revisão/lançamento do pacote ele deve ser atualizado.. para isso use o `dch -i` de dentro do diretório principal do seu pacote, ele irá colocar novos dados no arquivo e o abrirá com seu editor preferido. Não faça nada além disso nesse arquivo, ele tem formato específico e é usado como referência de versão entre outras informações.

No `gklog` eu vou colocar em baixo de * Initial release:

```
* Fixed Makefile problems
```

(arrumei problemas no makefile) edite o seu e coloque o que você fez.

3.3 control

Esse é um dos arquivos mais importantes do pacote. Nele você coloca o nome do pacote, versão e etc...ele tem um criador automático com isso...no gklog iremos trocar o unknown do campo Section por admin pois ele é um pacote para administração de sistema...talvez a maioria irá usar x11 aqui =) olhe seu ftp de Debian para saber os diretórios de seções existentes. Também colocaremos em Description uma pequena descrição (no máximo 60 caracteres) e nas linhas a seguir uma longa descrição (quantos espaços quiser...) essa descrição **tem que ter** 1 espaço em branco antes de cada linha e linhas em branco devem conter um único ' '.

Os do gklog eu coloquei:

```
Description: Log analyzer and line marker.  
GKLog is able to use rules defined by user to mark and log  
specified lines from logs with user-defined colors.
```

Note os espaços em branco antes de GKLog e specified.

3.4 copyright

É a licença do seu programa...coloque toda a licença aqui no local indicado ou fale que ela está localizada em /usr/share/common-licenses caso seja uma das contidas lá (GPL, LGPL (2 e 2.1), Artistic License, BSD) que todo Debian tem.

3.5 dirs

Aqui você coloca os diretórios que precisa criar mas que o Makefile não cria...Makefile do gklog toma conta disso (os diretórios aqui não devem conter o '/' no início)

3.6 docs

Nesse arquivo colocamos os arquivos que vão em /usr/doc no gklog eu pus:

```
README  
TODO  
AUTHORS  
gklog.sh
```

3.7 `init.d.ex`

Vou comentar esse arquivo porque ele é extremamente útil para programas que devem ser iniciados ao ligar o computador. .. edite-o (é um script shell) para que seu programa seja iniciado e terminado ao ligar/desligar. O `gklog` deveria ter um desses, mas acabei achando que o próprio admin deveria habilitar o `gklog` no seu sistema e então não editei esse arquivo, crie-o se necessário e tire o `.ex` do final.

3.8 `manpage.1.ex`

Caso seu programa tenha uma página de manual coloque-a nesse diretório com o nome do programa e extensão `?.?` onde o `?` é um número de seções a maior parte dos programas vai em `1..naum` entrarei em detalhes. Caso seu programa não tenha uma, edite esse arquivo respeitando os códigos para seu programa.

3.9 `menu.ex`

Esse eu acho o máximo!! Você já percebeu que sempre que instala um programa no seu Debian ele logo tem uma entrada de menu em **TODO** window manager? É graças ao sistema de menus do Debian. ... edite esse arquivo para que seu programa faça parte desse sistema. O do `gklog` eu fiz assim:

```
?package(gklog):needs=text section=Apps/System\  
title="gklog" command="/usr/bin/gklog"
```

substitua `System` da `section` para a seção do menu que deve levar seu programa e o `command` para onde irá se encontrar seu executável após a instalação. Leia o `maint-guide` para mais informações.

3.10 `README.Debian`

Coloque aqui qualquer discrepância entre a versão original e a *debianisada*. Se não houver o que colocar aqui, delete-o.

3.11 `rules`

`rules` é um outro Makefile, que automatizará a construção do pacote. Edite-o para suas necessidades, na seção `binary-arch`: é que se encontra algo de interesse para o `gklog`, ele não precisa de todos aqueles scripts `dh_*` (leia o `maint-guide` e suas `manpages` para entendê-los) ficou assim o meu:

```
# Build architecture-dependent files here.
binary-arch: build install
#     dh_testversion
#     dh_testdir
#     dh_testroot
#     dh_installdebconf
#     dh_installdocs
#     dh_installexamples
#     dh_installmenu
#     dh_installemacsen
#     dh_installpam
#     dh_installinit
#     dh_installicron
#     dh_installmanpages
#     dh_installinfo
#     dh_undocumented
#     dh_installchangelogs changelog
#     dh_link
#     dh_strip
#     dh_compress
#     dh_fixperms
#     # You may want to make some executables suid here.
#     dh_suidregister
#     dh_makeshlibs
#     dh_installdeb
#     dh_perl
#     dh_shlibdeps
#     dh_gencontrol
#     dh_md5sums
#     dh_builddeb
```

Lembre-se de que, o método `binary` (qualquer dos dois que seja) deve rodar o `make install` dessa forma:

```
$(MAKE) install DESTDIR='pwd'/debian/tmp
```

Observe também, que se você já está se aventurando no woody, essa regra já mudou... confira no início do seu arquivo `rules` qual o valor atribuído a `DH_COMPAT`... se for 1 é isso aí mesmo... se for 2, o `DESTDIR` deve ser `'pwd'/debian/nome_do_pacote`... fácil fácil =)

Capítulo 4

Finalizando o empacotamento

4.1 Rodando o `dpkg-buildpackage`

Bem... depois disso tudo, você deve estar pronto para criar o pacote. entre no diretório principal e digite:

```
dpkg-buildpackage -rfakeroot
```

Espera a compilação do pacote e pronto, no diretório pai do principal estará seu deb preparado. Basta instalá-lo com `dpkg -i`.

4.2 Corrigindo erros

Existe um programa chamado `lintian` use:

```
lintian -i gklog_0.3-1_i386.changes
```

para checar erros no pacote... se tudo correu bem pode começar a distribuir! Se um erro acontecer durante a construção do pacote, corrija o arquivo errado e execute o `dpkg-buildpackage -rfakeroot`, caso o fonte tenha sido totalmente compilado e você não queira passar por tudo aquilo de novo, use:

```
fakeroot debian/rules binary
```

4.3 Atualizando o pacote

Para atualizar o pacote use o `uupdate`, coloque o novo `tar.gz` no seu diretório de empacotamento entre no velho diretório do fonte e digite:

```
uupdate -u gklog-0.4.tar.gz
```

Repare que a versão usada no comando é a **nova**. =) That's all folks!!

Capítulo 5

Contato

Aqui estão informações para quem quiser contactar-me! Esse e outros documentos são parte do projeto Debian-BR (<http://debian-br.sourceforge.net/>). Você pode enviar um e-mail para <debian-user-portuguese@lists.debian.org> ou para mim para mais detalhes.

Kov's Page (<http://www.doctorkov.cjb.net>)

GPG Key (<http://www.linuxmagic.org/~kov/html/pgp.html>)

Gustavo Noronha Silva

By Kov!!