

6. Overrides para o lintian

6.1 Para binários

Crie o arquivo `debian/<pacote>.lintian-overrides`. Coloque as mensagens do lintian (sem E:, I:, P:, W: ou X:) dentro desse arq.

6.2 Para sources

Crie o arquivo `debian/sources/lintian-overrides`. Coloque as mensagens do lintian (sem E:, I:, P:, W: ou X:) dentro desse arq.

7. Patches com quilt

7.1 Criação de patch

- ▶ Estando no diretório do upstream, emita os comandos:

```
# quilt new nome_do_patch
# quilt add arquivo(s)_a_ser(em)_patchado(s)
```

- ▶ Edite normalmente o(s) arquivo(s) do upstream, envolvido(s) pelo patch. Depois, execute:

```
# quilt refresh
# quilt header -e
```

- ▶ Inserir como cabeçalho:

```
Description: descrever o que o patch faz.
Author: nome_do_autor_do_patch <e-mail>
Last-Update: YYYY-MM-DD
```

- ▶ Finalizar com:

```
# quilt pop -a
```

7.2 Testando um patch

- ▶ No diretório `debian`, aplique o patch com:

```
# quilt push nome_do_patch
```

- ▶ Verifique o resultado. Desaplique o patch com:

```
# quilt pop -a
```

7.3 Edição de patch preexistente

- ▶ No diretório `debian`:

```
# quilt push nome_do_patch
```

- ▶ Edite normalmente o(s) arquivo(s) envolvido(s) pelo patch. Depois, execute:

```
# quilt refresh
# quilt pop -a
```

8. DHs

- ▶ Executam ações especiais. Exemplo: o `dh_fixperms` corrige as permissões de arquivos e diretórios para que esses respeitem a política Debian. Também remove SUID e SGID.
- ▶ Relação completa de DHs (há uma manpage para cada um): # `apropos dh_`
- ▶ Para ver os DHs atuando durante a construção do pacote, descomente `export DH_VERBOSE=1`, em `debian/rules`.

9. Ignorar alterações em arquivo

Criar arq. `debian/source/options`. Podem ser utilizadas regexp. Ex:

```
extend-diff-ignore = "^test$"
```

10. Bibliografia essencial

- ▶ FHS: <http://pathname.com/fhs>
- ▶ Manual de política Debian: <http://debian.org/doc/debian-policy>
- ▶ Guia de referência para desenvolvedores: <http://debian.org/doc/manuals/developers-reference>
- ▶ Guia do novo mantenedor: <http://debian.org/doc/manuals/maint-guide>
- ▶ Canto dos desenvolvedores: <http://debian.org/devel>
- ▶ Debian FAQ (empacotamento): http://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html
- ▶ WNPP: <http://www.debian.org/devel/wnpp>
- ▶ Listas Debian: <http://lists.debian.org>

- ▶ Lista de pacotes virtuais: <http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt>
- ▶ Lista de pseudo-pacotes para bugs: <http://www.debian.org/Bugs/pseudo-packages>
- ▶ Times no Debian: <http://wiki.debian.org/Teams>

- ▶ Candidatura a mantenedor: <http://wiki.debian.org/DebianMaintainer>
- ▶ Candidatura a desenvolvedor: <https://nm.debian.org/>

EMPACOTAMENTO DE SOFTWARE NO DEBIAN GNU/LINUX

Versão 2.7 - 11 de maio de 2014



© 2007-2014 by João Eriberto Mota Filho

<http://debianet.com.br>
eriberto@eriberto.pro.br

2048R/C1CFC265: 1D75 E212 B34C F4BF A9E0 D0D8 DE6D E039 C1CF C265
4096R/04EBE9EF: 357D CB0E EC95 A01A EBA1 F0D2 DE63 B9C7 04EB E9EF

1. Construção da jaula Sid (unstable)

1.1 Procedimentos iniciais

```
# apt-get install debootstrap
# mkdir /jaula-sid
# debootstrap sid /jaula-sid http://ftp.br.debian.org/debian
# chroot /jaula-sid
# echo proc /proc proc defaults 0 0 >> /etc/fstab
# mount /proc
```

Obs: caso o sid (ou unstable) esteja quebrado e apresente problemas na instalação, utilize o stable e, depois, faça um dist-upgrade para sid.

1.2 Ajuste do /etc/bash.bashrc

Insira no final do arquivo /etc/bash.bashrc:

```
alias ls='ls --color=auto'
alias tree='tree -C'
alias debuildsa='dpkg-buildpackage -sa -ksua_chave_gpg'
alias uscan='uscan --verbose --report'
alias apt-cache='LANG=C apt-cache'
export DEBFULLNAME="seu_nome_completo_sem_acentos/cedilha"
export DEBEMAIL="seu_e-mail"
export EDITOR=mcedit
export QUILT_PATCHES=debian/patches
export QUILT_DIFF_ARGS="--no-timestamps --no-index -pab"
export QUILT_REFRESH_ARGS="--no-timestamps --no-index -pab"
export PS1='JAULA-SID-\u@\h:\w\$ '
mount /proc
```

Obs: txt pronto para uso em <http://bit.ly/bash-rc-txt>

A seguir, execute: # source /etc/bash.bashrc

1.3 Procedimentos complementares

```
# apt-get install blhc devscripts dh-make locales mc quilt
renameutils spell tree
# dpkg-reconfigure locales tzdata
# apt-get clean
```

1.4 Ajuste do /etc/lintianrc

No arquivo /etc/lintianrc, habilitando as seguintes linhas:

```
display-info = yes
pedantic = yes
display-experimental = yes
```

A jaula está pronta. Para sair da mesma, execute: # exit

1.5 Inserção das chaves GPG

Fora da jaula, exporte as suas chaves GPG (privada e pública):

```
$ gpg -a --export nr_da_chave > nr_da_chave.pub
$ gpg -a --export-secret-keys nr_da_chave > nr_da_chave.key
```

Mova os arquivos para dentro da jaula unstable, enjaule-se, importe as chaves e remova os arquivos. Para importar:

```
# gpg --import chave.key chave.pub
```

Edite o arquivo /etc/devscripts.conf e insira o número da sua chave GPG na linha DEBSIGN_KEYID, descomentando-a. Exemplo:

```
DEBSIGN_KEYID=C1CFC265
```

1.6 Backup da jaula para uso futuro

```
# tar -cvjf jaula-sid.tar.bz2 --exclude='*/proc/*'
--exclude='*/apt/lists/*dists*' /jaula-sid
```

2. Processo de empacotamento

- ▶ Verifique se já existe alguma menção ao futuro pacote no Debian com os comandos apt-cache search e wnpp-check.
- ▶ Verifique as licenças dos diversos arquivos visualmente e com o comando licensecheck.
- ▶ Baixe o código-fonte e certifique-se de que o mesmo esteja no formato nome_do_pacote-versão.tar.{bz2,gz,xz}.
- ▶ Debianização: # dh_make -f ../.tar.gz -c <licença>
- ▶ No diretório debian, apague os arquivos desnecessários.
- ▶ Caso necessário, faça um patch para o Makefile do upstream para que o mesmo use \$(DESTDIR) e hardening.
- ▶ Crie o pacote com o comando debuild e corrija os problemas mostrados pelo debuild / lintian.
- ▶ Caso o lintian aponte problemas difíceis de entender, execute # lintian -Ii contra o arq. .changes.
- ▶ Verifique com # tree debian/nome_do_pacote se todos os arquivos existem e se estão nos locais certos.
- ▶ No dir. debian, analise cada arq. e remova todo o lixo.
- ▶ Em debian, com # tail -n 1 *, veja se há linhas em branco no fim dos arq. e remova-as. (# dh_clean antes!)
- ▶ Em debian, com # grep -sr ' \$' *, veja se há espaços em branco no fim das linhas dos arquivos e remova-os.

- ▶ No diretório do upstream, execute # uscan. Isso irá verificar se o arquivo debian/watch está correto.
- ▶ Use a ferramenta spell para buscar erros de grafia nos arq. debian/{changelog, control, README}.
- ▶ Instale o pacote e teste o funcionamento do programa.
- ▶ Use o blhc contra o .build para verificações finais.

3. Verificação final

3.1 Construção do pacote (cowbuilder)

Fora da jaula, para criar uma jaula inicial, execute:

```
# apt-get install cowbuilder
# cowbuilder --create
```

Para verificar um pacote, execute:

```
# cowbuilder --build arquivo.dsc
```

Para atualizar a jaula cowbuilder, execute:

```
# cowbuilder --update
```

Importante: caso não consiga criar a jaula (Sid quebrado) execute:

```
# cowbuilder --distribution stable --create
```

Depois, edite o arq. /var/cache/pbuilder/base.cow/etc/apt/sources.\
list e altere o nome stable para unstable (ou sid). A seguir, execute:

```
# cowbuilder --update
```

3.2 Instalação do pacote (para SID, pode dar errado)

```
# apt-get install piuparts
# piuparts arquivo.deb
```

4. Revisão Debian (novo pacote)

Dentro do diretório debian, enjaulado, execute: # dch -i

5. Nova versão do programa

Dentro da jaula, faça o download (com wget, por exemplo) da nova versão do programa desejado. Depois, execute:

```
# apt-get source pacote
# cd diretório_do_pacote
# uupdate ../programa-nova-versão.tar.gz
```

Obs: pacote refere-se à versão antiga do código-fonte.