

Balíš, balím, balíme

Debian balíčkování pro (středně) pokročilé

Ondřej Surý • ondrej.sury@nic.cz • ondrej@debian.org • 15.- 10.
2015

Obsah přednášky

- Struktura binárního balíčku (.deb)
- Struktura zdrojového balíčku
- Základní toolchain
- Doporučený toolchain
- Balíčkování v gitu
- Čistota build prostředí
- Různé...
- Jak se stát Debian vývojářem?



Moje Debian portfolio (od roku 2000)

- Většina je spravovaná v týmech
- PHP + PECL (pkg-php)
 - Nyní v běhu koinstalovatelné verze
 - Nejdéle sloužící PHP maintainer v Debianu
- libjpeg-turbo
 - Přechod Debianu z IJ Jpeg na LJT
- Hromada DNS balíčků (pkg-dns)
 - NSD
 - getdns
 - libldns
 - OpenDNSSEC + SoftHSM
- Berkeley DB (pkg-db)
 - Sjednocení release na jednu verzi
 - A moje osobní mise nahrazení za LMDB

- CZ.NIC projekty
 - BIRD
 - Knot DNS
 - Datovka (Qt5)
- Cyrus SASL
- Cyrus IMAPD
- LibTIFF (nově)
- ...a další drobnosti

Starší projekty

- Balíčkování GTK/GNOME/Freedesktop
- Redmine/Ruby
 - Nikdy více...

Struktura binárního balíku

- **ar** archiv složený z:
 - debian-binary
 - Verze .deb formátu (2.0)
 - control.tar.gz
 - Informace o balíčku (control)
 - Maintainer skripty
 - {pre,post}{inst,rm}
 - A další (md5sum, conffiles)
 - data.tar.xz
 - Obsah balíčku
- **Nástroje pro práci s .deb soubory**
 - dpkg-deb
 - dpkg (wrapper okolo dpkg-deb)
 - ar, tar

```
$ ar xv knot_2.0.1-4_amd64.deb
x - debian-binary
x - control.tar.gz
x - data.tar.xz

$ dpkg-deb -X knot_2.0.1-4_amd64.deb output/
./
./etc/
[...]
./usr/sbin/knotd
[...]

$ dpkg-deb -e knot_2.0.1-4_amd64.deb DEBIAN/
$ ls DEBIAN/
conffiles control md5sums postinst postrm preinst
prerm

$ dpkg -I knot_2.0.1-4_amd64.deb
new debian package, version 2.0.
size 218134 bytes: control archive=2737 bytes.
Package: knot
Version: 2.0.1-4
Architecture: amd64
```

Struktura zdrojového balíčku

- Metadata zdrojového balíčku

`<pkg>_<dver>.dsc`

- Metadata zdrojového balíčku
- Maintainer + Uploaders
- Build závislosti
- Seznam binárních balíčků
- Kontrolní součty (sha256)
- a další...

- Upstream tarball

`<pkg>_<uver>.orig.?z`

- Ideálně původní vanilla
- Případně repack (např. dfsg)

- Debian změny oproti upstream

`<pkg>_<dver>.debian.tar.xz`

- Ideálně pouze adresář debian/

```
Format: 3.0 (quilt)
Source: knot
Binary: knot, [...], knot-doc
Architecture: any all
Version: 2.0.1-4
Maintainer: Debian DNS Packaging <pkg-dns-devel@l.a.d.o>
Uploaders: Ondřej Surý <ondrej@d.o>
Homepage: http://www.knot-dns.cz/
Standards-Version: 3.9.6
Vcs-Browser: https://anonscm.d.o/git/pkg-dns/knot.git
Vcs-Git: git://anonscm.d.o/pkg-dns/knot.git
Build-Depends: debhelper (>= 9), [...], pkg-config
Build-Depends-Indep: ghostscript, [...], python-sphinx
Package-List:
    knot deb net optional arch=any
    [...]
Checksums-Sha1: [...]
Checksums-Sha256:
    183b[...]6ce6 958560 knot_2.0.1.orig.tar.xz
    29b7[...]681c 17688 knot_2.0.1-4.debian.tar.xz
Files:
    96c1[...]22e5 958560 knot_2.0.1.orig.tar.xz
    643a[...]4d6d 17688 knot_2.0.1-4.debian.tar.xz
```

Verzování Debian balíčků

[epoch:]upstream_version[-debian_revision]

- **epoch** - [0-9]
- **upstream_version** - [0-9a-z.+~:~]
- **debian_version** - [0-9a-z.+~]
- Porovnávání verzí
 - Nejprve se numericky porovná **epoch**
 - Pak zleva doprava **upstream_version**
 - Rozdělí se na nenumerické a numerické části
 - Všechna písmena se řadí před čísla
 - ~ se řadí až za prázdný konec
 - A nakonec **debian_version**
 - Stejný algoritmus jako u **uver**

```
# předchozí verze byla úplně špatně
# např. nový balík se stejným jménem
# příklad: git (původně GNU Interactive Tools)
dpkg --compare-versions '1:2.1.4-2.1' gt '4.3.20-7'

# repackovaný upstream tarball, odstraněny nesvobodné
# části
dpkg --compare-versions '1.0.3-3' lt '1.0.3+dfsg-1'

# botched upload
# ale nestojí to za epoch bump
dpkg --compare-versions '2.1.0+really+2.0.0-1' gt \
    '2.1.0-1'

# finální verze je větší než release candidate
dpkg --compare-versions '2.1.0-1' gt '2.1.0~rc1-1'

# upload do Debian security nebo proposed-updates
dpkg --compare-versions '5.6.14-0+deb7u1' lt '5.6.14-1'

dpkg --compare-versions '0:0' gt '1'
dpkg --compare-versions '0~' lt '0'
dpkg --compare-versions '~' lt '~a'
dpkg --compare-versions '~' lt '~'
```

Zdrojové balíky – adresář debian/

- d/changelog
 - Seznam změn v balíčkování
 - aka Debian changelog
- d/control
 - Metadata zdrojového i binárních balíčků
- d/rules
 - Pravidla (Makefile) pro sestavení balíčků
- d/copyright
 - Seznam všech licencí + licence debian/*
- d/source/format
 - **3.0 (quilt)** nebo **3.0 (native)**
- d/patches/
 - Jednotlivé patche spravované pomocí **quilt**
- A mnoho dalších...

```
$ ls -ld debian/*
debian/changelog
debian/compat
debian/control
debian/copyright
debian/docs
debian/init-d-script
debian/knot-resolver.default
debian/knot-resolver.init
debian/knot-resolver.install
debian/knot-resolver.lintian-overrides
debian/knot-resolvers.dirs
debian/knot-resolver.service
debian/knot-resolver.tmpfile
debian/kresd.conf
debian/patches/
debian/rules
debian/source/
```

Základní toolchain

- **d/rules** je Makefile
- **d/rules** **<target>**, kde **<target>** je:
 - clean
 - Uvede balíček do původního stavu
 - build (build-arch, build-indep)
 - Kompilace upstream zdrojů
 - install (install-indep, install-arch)
 - Instalace do dočasné cesty
 - binary (binary-arch, binary-indep)
 - Sestavení finálního balíčku
- Minimální **d/rules** →
 - Ale takhle prosím fakt ne :)

```
#!/usr/bin/make -f
# -*- makefile -*-

build:
    gcc -o helloworld helloworld.c

clean:
    rm -f helloworld

install: build
    mkdir debian/tmp/usr/bin
    cp -a helloworld debian/tmp/usr/bin/

binary-indep: build

binary-arch: build
    mkdir debian/tmp/DEBIAN/
    dpkg-gencontrol
    dpkg-deb -b debian/tmp/ <package>_<dver>.deb

binary: binary-arch binary-indep

.PHONY: build clean install binary binary-arch \
        binary-indep
```


dh_make – aneb jak jednoduše začít

dh_make

- Vytvoří základní kostru balíčku v debian/
- Potřeba pokaždé upravit pro potřeby konkrétního projektu
- Typy balíčků
 - Single (--single)
 - Pouze jeden binární balíček
 - Multi
 - Více binárních balíčků
 - Library
 - Knihovna (libfoo0 + libfoo-dev)
 - ...a další

dh_make_perl, gem2deb

- Jednoduché zabalení CPAN modulu, Gemu

```
$ tar -xJf knot-resolver-1.0-beta.tar.xz
$ cd knot-resolver-1.0-beta/
knot-resolver-1.0-beta$ dh_make -s -f \
                        ../knot-resolver-1.0-beta.tar.xz
Maintainer name   : Ondřej Surý
Email-Address     : ondrej@debian.org
Date              : Fri, 09 Oct 2015 16:08:12 +0200
Package Name      : knot-resolver
Version           : 1.0-beta
License           : blank
Type of Package   : Single
Hit <enter> to confirm:
Done. Please edit the files in the debian/ subdirectory
now. You should also
check that the knot-resolver Makefiles install into
$DESTDIR and not in / .
```

Doporučený toolchain – debhelper

- Obsáhlá sada `dh_*` příkazů pro sestavení balíčku
 - `dh_auto_configure`, `dh_auto_build`, `dh_auto_install`, `dh_auto_test`
 - Automaticky rozpozná build system a spustí (např.) `./configure`, `make` a `make install` se správnými parametry
 - `dh_install`, `dh_strip`, `dh_installcron`, ...
 - Úrovně podpory (viz Compatibility levels)
 - Poslední: v9 (d/compat)
 - Multi-arch podpora
 - Hardening (`dpkg-buildflags`)
- Rozšiřitelný framework (ehm... v Perlu)
- `dh7` vylepšil toolchain o příkaz `dh` →
 - Spouští sekvenci `dh_*` příkazů

```
#!/usr/bin/make -f
# See debhelper(7) (uncomment to enable)
# output every command that modifies files on the build
system.
#DH_VERBOSE = 1

# see EXAMPLES in dpkg-buildflags(1) and read
/usr/share/dpkg/*
DPKG_EXPORT_BUILDFLAGS = 1
include /usr/share/dpkg/default.mk

# see FEATURE AREAS in dpkg-buildflags(1)
export DEB_BUILD_MAINT_OPTIONS = hardening=+all

# see ENVIRONMENT in dpkg-buildflags(1)
# package maintainers to append CFLAGS
export DEB_CFLAGS_MAINT_APPEND = -Wall -pedantic
# package maintainers to append LDFLAGS
export DEB_LDFLAGS_MAINT_APPEND = -Wl,--as-needed

# main packaging script based on dh7 syntax
%:
    dh $@ #DH7_ADDON#
```

Standardní debhelper sekvence

```
$ dh binary --no-act
dh_testdir
dh_auto_configure
dh_auto_build
dh_auto_test
dh_testroot
dh_prep
dh_installdirs
dh_auto_install
dh_install
dh_installdocs
dh_installchangelogs
dh_installexamples
dh_installman
dh_installdatalogs
dh_installdcron
dh_installdebconf
dh_installemacsen
dh_installifupdown
dh_installinfo
dh_installinit
dh_installmenu
dh_installemime
dh_installemules
dh_installogcheck
```

```
dh_installogrotate
dh_installpam
dh_installppp
dh_instaludev
dh_installdwm
dh_installdsettings
dh_bugfiles
dh_ucf
dh_lintian
dh_gconf
dh_icons
dh_perl
dh_usrlocal
dh_link
dh_installdfonts
dh_compress
dh_fixperms
dh_strip
dh_makeshlibs
dh_shlibdeps
dh_installddeb
dh_gencontrol
dh_md5sums
dh_buildddeb
```

Instalace souborů do balíčků

dh_installdirs

- Pomocný skript na vytvoření adresářů popsaných v `d/<package>.dirs`

dh_install

- Pomocný skript na rozdělení `d/tmp/` do `d/<package>/` adresářů
- Načítá `d/<package>.install`
- Neumí přejmenovávat soubory!
- Od **compat 9** mohou být `.install` skripty i spustitelné (`#!/usr/bin/dh-exec`)
 - Umí přejmenovávat soubory
 - Proměnné (`env + dpkg-architecture`)

```
$ cat debian/knot-resolvers.dirs
/etc/knot-resolver
/var/lib/knot-resolver
```

```
$ cat debian/libknot1.install
usr/lib/*/libknot.so.*
```

```
$ cat debian/libknot-dev.install
usr/include/
usr/lib/*/*.a
usr/lib/*/*.so
usr/lib/*/pkgconfig/*
```

```
$ cat debian/knot-host.install
usr/bin/khost
usr/share/man/man1/khost.1
```

```
#!/usr/bin/dh-exec
src/libfoo-*.so.* debian/foo-
plugins/usr/lib/foo/${DEB_HOST_MULTIARCH}/
etc/example.conf => debian/foo/etc/foo/foo.conf
[linux-any kfreebsd-any] arch-specific /usr/lib/foo/
```

Instalace dokumentace

dh_installdocs

- Pomocný program, který instaluje soubory vypsané v d/<package>.docs + d/copyright, d/README.Debian (+ registrace doc_base) do d/<package>/usr/share/doc/<package>
- Speciální volba --link-doc, která způsobí symlinkování dokumentace mezi balíky
 - (Teoreticky) šetří místo
 - !!!POZOR!!! Nemíchat linky mezi arch:all a arch:any balíčky (rozbíjí binNMU)
 - Moje doporučení: **NEPOUŽÍVAT**

```
# příklad ze staršího balíčku
override_dh_installdocs:
    dh_installdocs -p cyrus-common -p cyrus-doc-2.4
    dh_installdocs -p cyrus-doc \
        --link-doc=cyrus-doc-2.4
    dh_installdocs --remaining-packages \
        --link-doc=cyrus-common
```

Úpravy standardního chování

- **dh <target> --no-act**
 - Vypíše řetěz příkazů, které se spouští
- Každý příkaz lze přepsat v **d/rules** pomocí
 - **override_<target>**
- Některé příkazy berou env proměnnou
- Některým je potřeba předat parametry ručně
- Běžná použití:
 - **dh_strip** – umístění dbg symbolů do <package>-dbg
 - **dh_install** – parametr **--{list,warn}-install** pro kontrolu instalace všech souborů
 - **dh_auto_configure** – předání **./configure** parametrů navíc

```
$ dh build --no-act
dh_testdir
dh_auto_configure
dh_auto_build
debian/rules override_dh_auto_test

$ cat debian/rules
#!/usr/bin/make -f
[...]
%:
    dh $@ #DH7_ADDON#

# run tests, but don't fail the build on failure
override_dh_auto_test:
    -make check
```

Užitečná rozšíření debhelperu

- Rozšíření pro
 - Další buildsystémy
 - PHP (PEAR, PECL)
 - Další init systémy
 - systemd
 - Různé jazyky
 - PHP, python, Ruby
 - Další
 - autotools, dh-buildinfo

```
#!/usr/bin/make -f
%:
    dh $@ \
        --with systemd \
        --with autotools-dev \
        --with autoreconf \
        --with apache2 \
        --with python2 \
        --with python3 \
        ...
```

dh-systemd

- Novější **dh_installinit** už systemd umí sám
- Nainstaluje a zaktivuje systemd unit a tmpfile soubory
 - d/<package>.service
 - d/<package>.tmpfile
- Přidává dva příkazy:
 - **dh_systemd_enable**
 - Aktivuje unit file při instalaci
 - Deaktivuje unit file při odstranění b.
 - **dh_systemd_start**
 - Zodpovídá za start/stop/restart při instalaci/upgrade balíčku

```
#!/usr/bin/make -f

%:
    dh $@ --with systemd

# zrestartuj službu až po kompletní instalaci
# normálně se služba zastaví už před upgrade balíčku
override_dh_systemd_enable:
    dh_systemd_enable --restart-after-upgrade
```


dh-autoreconf a autotools-dev

- Některé upstream tarbally byly generovány již před časem, nebo na starších systémech
- **dh_autoreconf**
 - Spustí aktuální verzi **autoreconf -fi**
- **dh_autotools-dev_{update,restore}config**
 - Zaktualizuje všechny výskyty config.sub a config.guess souborů
 - Často důležité pro podporu novějších architektur (např. arm64)
- Oba příkazy si pamatují změny a v **clean** cíli obnoví změněné soubory

```
#!/usr/bin/make -f
%:
    dh $@ --with autoreconf --with autotools_dev

override_dh_autoreconf:
    dh_autoreconf --as-needed
```

Debian Hardening

- Při kompilaci se dá zapnout několik voleb kompilátoru, které zkalí výslednou kompilaci
 - Format security, Fortify source, Stack protector, PIE, relro, bindnow
- Více v Debian Wiki:
<https://wiki.debian.org/Hardening>
<https://wiki.debian.org/HardeningWalkthrough>

```
#!/usr/bin/make -f

DPKG_EXPORT_BUILDFLAGS = 1
include /usr/share/dpkg/default.mk

export DEB_BUILD_MAINT_OPTIONS = hardening=+all

%:
    dh $@
```

git-buildpackage

gbp [clone|import-dsc|import-org|buildpackage]

- Sada nástrojů na udržování balíčkování v git repozitáři
- Umožňuje importovat stávající balíky
- Udržuje upstream zdrojáky ve stejném gitu
- A Debian balíčkování ve vlastní větvi
- Standardně:
 - **upstream** – Větev se zdrojáky
 - **master** – upstream + debian/
 - **pristine-tar** – udržuje přesný stav orig.tar.?z
- Ale taky:
 - **{master-,}wheezy, jessie** – Větve pro starší verze...

```
$ cd knot
$ git describe
v2.0.1-86-gd04fab6
$ git archive \
  --output=/tmp/knot-2.0.1-86-gd04fab6.tar.xz \
  --prefix=knot-2.0.1-86-gd04fab6 HEAD
$ cd /tmp
$ gbp clone git://anonscm.debian.org/pkg-dns/knot.git
$ cd knot
$ gbp import-orig ../knot-2.0.1-86-gd04fab6.tar.xz
What is the upstream version? [86-gd04fab6] 2.0.1+86-gd04fab6
gbp:info: Importing '../knot-2.0.1-86-gd04fab6.tar.xz' to \
  branch 'upstream'...
gbp:info: Source package is knot
gbp:info: Upstream version is 2.0.1+86-gd04fab6
pristine-tar: committed knot_2.0.1+86-gd04fab6.orig.tar.xz.
delta to \
  branch pristine-tar
gbp:info: Merging to 'master'
gbp:info: Successfully imported version 2.0.1+86-gd04fab6 of \
  ../knot-2.0.1-86-gd04fab6.tar.xz
```

gbp pq – patch management v gitu

gbp pq [import|export|rebase|drop]

- Umí pracovat s patchi v **d/patches/** v samostatné git větvi **patch-queue/<*>**
- Výsledné patche jsou git kompatibilní
- Jednoduše udržuje informaci o patchi a autorovi

```
$ gbp pq import
gbp:info: Trying to apply patches at \
'a4becbda0045432ef6c7de8cefa03f256b7542e1'
gbp:info: Patches listed in 'debian/patches/series' \
imported on 'patch-queue/master'
$ git log --format=oneline master..patch-queue/master
01720dce daemons go to /usr/sbin
ed218c65 Install modules to /usr/lib/knot-resolver
8f69d5be Install into /usr and honour $DESTDIR
$ emacs config.mk
$ git commit -a -m 'Use gcc instead of cc'
$ gbp pq export
gbp:info: On 'patch-queue/master', switching to 'master'
gbp:info: Generating patches from git (master..patch-
queue/master)
On branch master
Changes not staged for commit:
  (use "git add <file>.." to update what will be committed)
  (use "git checkout -- <file>.." to discard changes in...)
        modified:   debian/patches/series
Untracked files:
  (use "git add <file>.." to include in what will be...)
        debian/patches/0004-Use-gcc-instead-of-cc.patch
no changes added to commit (use "git add" and/or "git commit -
a")
$ gbp pq drop
gbp:info: Dropped branch 'patch-queue/master'.
```

Čisté build prostředí – git-pbuilder (+cowbuilder)

- Balíčky standardně vytváříte pro **unstable**
- Váš pracovní počítač asi nebude přesně to ono
- git-pbuilder je wrapper okolo pdebuild používající cowbuilder
- pdebuild z balíku pbuilder vytváří “čisté” build prostředí
 - pbuilder samotný pracuje s tar.gz
 - cowbuilder je copy-on-write varianta, která je mnohem rychlejší (nědelá se tar xz)
 - qemu builder umí použít qemu (např. arm)
- Proměnné **DIST** a **ARCH** ovlivňují, které prostředí se použije

```
# nastavíme git-buildpackage
$ cat ~/.gbp.conf
[DEFAULT]
builder = git-pbuilder
cleaner = true

# i386 funguje i na amd64 hostiteli
$ DIST=jessie ARCH=i386 git-pbuilder create
[...chroupy...debootstrap...chroupy...]

# vytvoříme balíček v prostředí Debian 8, arch i386
$ DIST=jessie ARCH=i386 gbp buildpackage

$ ls -ld /var/cache/pbuilder/base-*
[...]
/var/cache/pbuilder/base-wheezy-amd64.cow
/var/cache/pbuilder/base-wheezy-armhf.cow
/var/cache/pbuilder/base-wheezy-backports-amd64.cow
/var/cache/pbuilder/base-wheezy-backports-armhf.cow
/var/cache/pbuilder/base-wheezy-backports-i386.cow
/var/cache/pbuilder/base-wheezy-i386.cow
[...]
$ ls -ld /var/cache/pbuilder/base-* | wc -l
38
```

Konfigurační soubory (conffile) v balíčcích

- Speciální zacházení
 - Nová verze se nainstaluje, jen pokud se stará nezměnila
 - Automaticky se neodstraňuje

dpkg-maintscript-helper

- Pomocník, který umí
 - **mv_conffile** – Přejmenovat **conffile**
 - **rm_conffile** – Smazat starý **conffile**
- A navíc věci, co neumí dpkg
 - **symlink_to_dir** – změnit symlink na adresář
 - **dir_to_symlink** – vice versa
 - Nejčastěji kvůli `dh_installdocs --link-doc=...`
- `d/<package>.maintscript` přidá volání do maintainer skriptů (`{pre,post}{inst,rm}`)

```
$ cat debian/knot.maintscript:
rm_conffile /etc/init/knot.conf 2.0.0-1~

$ cat debian/cyrus-common.maintscript
mv_conffile \
  /etc/logcheck/violations.ignore.d/cyrus-common-2_4 \
  /etc/logcheck/violations.ignore.d/cyrus-common \
  2.4.17+caldav~beta10-3~ # prior version

$ cat debian/cyrus-common-2.4.maintscript
symlink_to_dir \
  /usr/share/doc/cyrus-common-2.4 \ # pathname
  cyrus-common \ # old-target
  2.4.17+caldav~beta10-11~ # prior version

$ cat debian/nettle-dbg.maintscript
dir_to_symlink \
  /usr/share/doc/nettle-dbg \ # pathname
  libnettle6 \ # new-target
  2.7.1-5~ \ # prior version
  nettle-dbg # package (volitelně)
```

deb-triggers

- Instalace balíku může vyžadovat akci v jiném balíku
 - Můžeme akci zavolat ručně, ale...
 - Volání nemusí být aktuální
 - Chceme akci spustit jen jednou, typicky restart služby (webserver)

deb-triggers

- Deklarace v `d/<package>.triggers`
- **interest** *trigger-name*
 - *trigger-name* může být adresář
 - Deklarujeme zájem
- **activate** *trigger-name*
 - Deklarujeme aktivaci

```
$ cat debian/libapache2-mod-php5.triggers
interest-noawait /etc/php5/apache2/conf.d
```

```
$ cat debian/php5-fpm.triggers
interest-noawait /etc/php5/fpm/conf.d
```

```
$ cat debian/php5-module.triggers
activate-noawait /etc/php5/apache2/conf.d
activate-noawait /etc/php5/fpm/conf.d
```

Symbole ve sdílených knihovnách

- Sdílené knihovny deklarují podporované API pomocí SOVERSION, které se musí změnit, když dojde k odebrání symbolů
 - <https://autotools.io/libtool/version.html>
- Balíčky s knihovnami mohou přidat **d/<package>.symbols**, kde deklarují aktuální symboly
- Dá se nastavit striktní kontrola
- Pro C++ je potřeba symboly dekodovat pomocí **c++filt**
 - Lepší je použít pomocníky z <https://pkg-kde.alioth.debian.org/symbolfiles.html>

```
$ cat debian/libgetdns1.symbols
libgetdns.so.1 libgetdns1 #MINVER#
getdns_address@Base 0.1.5
getdns_address_sync@Base 0.1.5
[...]

$ cat debian/rules
#!/usr/bin/make -f
# -*- makefile -*-

# build selže, pokud se symboly změnily
# dh_makeshlibs volá dpkg-gensymbols(1)
export DPKG_GENSYMBOLS_CHECK_LEVEL=4

$ head debian/libbotan-1.10-1.symbols
# SymbolsHelper-Confirmed: 1.10.10 alpha amd64 arm64 armel
armhf hurd-i386 i386 kfreebsd-amd64 kfreebsd-i386 mips mipse1
powerpc ppc64el s390x sparc64
libbotan-1.10.so.1 libbotan-1.10-1 #MINVER#
(c++)"Botan::AES_128::clear()@Base" 1.10.10
(c++)"Botan::AES_128::clone() const@Base" 1.10.10
(c++|arch=!alpha !amd64 !arm64 !kfreebsd-amd64 !ppc64el !
s390x !sparc64)"Botan::AES_128::decrypt_n" 1.10.10
(c++|arch=amd64 kfreebsd-amd64)"Botan::AES_128::decrypt_n
(unsigned char const*, unsigned char*, unsigned long)
const@Base" 1.10.10
```


Přesunujeme soubory mezi balíčky

- Dva balíčky nemohou instalovat stejný soubor
 - Při instalaci není automaticky zaručeno správné pořadí instalace balíčků
 - Po přesunu souboru jen tak nejspíš dojde k chybě, pokud soubor přesuneme např. z balíčku foo do balíčku foo-data
- Musíme deklarovat, že nový balíček foo-data=1.2-3 přepisuje soubory ve starém balíčku foo=1.2-2 (a starších) → Breaks+Replaces
- Přejmenování balíčku je jen speciální případ přesunutí všech souborů

```
Source: ruby-augeas
[...]

Package: ruby-augeas
Architecture: any
XB-Ruby-Versions: ${ruby:Versions}
Depends: ${misc:Depends}, ${shlibs:Depends}
Replaces: libaugeas-ruby (<< 0.5.0-1~),
          libaugeas-ruby1.8 (<< 0.5.0-1~),
          libaugeas-ruby1.9.1 (<< 0.5.0-1~)
Provides: libaugeas-ruby, libaugeas-ruby1.8, libaugeas-
ruby1.9.1
Breaks: libaugeas-ruby (<< 0.5.0-1~),
        libaugeas-ruby1.8 (<< 0.5.0-1~),
        libaugeas-ruby1.9.1 (<< 0.5.0-1~)
Description: Augeas bindings for the Ruby language
[...]

Package: libaugeas-ruby
Section: oldlibs
Priority: extra
Architecture: all
Depends: ruby-augeas, ${misc:Depends}
Description: Transitional package for ruby-augeas
```

Kontrola výsledných balíků

lintian <package>.changes

- Zkontroluje výsledek pomocí sady pravidel, která se stále vyvíjí
- Doporučuji používat minimálně verzi ze stable-backports
- Kategorie
 - Chyby – některé vedou rovnou k odmítnutí
 - Varování
 - Drobnosti (lintian --pedantic)
- V `d/<package>.lintian-overrides` můžete některé chyby ignorovat, když jste si jistí, že je to tak správně

```
$ lintian --pedantic php5_5.6.14+dfsg-1_amd64.changes
P: php5 source: no-dep5-copyright
P: php5 source: debian-watch-may-check-gpg-signature
W: php5-common: binary-without-manpage usr/sbin/php5dismod
W: php5-common: binary-without-manpage usr/sbin/php5enmod
W: php5-common: binary-without-manpage usr/sbin/php5query
W: php5-common: script-not-executable usr/share/php5/php5-
  helper
W: php-pear: binary-without-manpage usr/bin/pear
W: php-pear: binary-without-manpage usr/bin/peardev
W: php-pear: binary-without-manpage usr/bin/pecl
E: libapache2-mod-php5filter: apache2-module-depends-on-real-
  apache2-package apache2-bin
E: libapache2-mod-php5: apache2-module-depends-on-real-
  apache2-package apache2-bin
N: 33 tags overridden (12 errors, 7 warnings, 14 info)
```

Kontrola chování balíčků – piuparts

- I formálně (lintian) správný balík může obsahovat různé chyby při instalaci/odinstalaci/smazání
- **piuparts** vytváří speciální prostředí, ve kterém balík nainstaluje (--install), odinstaluje (--remove) a promaže (--purge)
- Pokud např. balík po sobě nechá nějaké soubory nebo změněné soubory, je to chyba
- Výsledky jsou dostupné na <https://piuparts.debian.org/> nebo je možné piuparts použít lokálně
 - Kontroluje se i upgrade mezi Debian vydáními (wheezy2jessie, jessie2stretch)

```
$ sudo /usr/sbin/piuparts knot-host_2.0.1-4_amd64.deb
[...]  
0m0.0s DEBUG: Setting up minimal chroot for sid at  
/tmp/tmpdL7gks.  
[...]  
1m21.1s DEBUG: Starting command: ['chroot',  
'/tmp/tmpdL7gks', 'eatmydata', 'dpkg', '-i', 'tmp/knot-  
host_2.0.1-4_amd64.deb']  
[...]  
1m31.3s INFO: PASS: All tests.  
1m31.3s INFO: piuparts run ends.
```

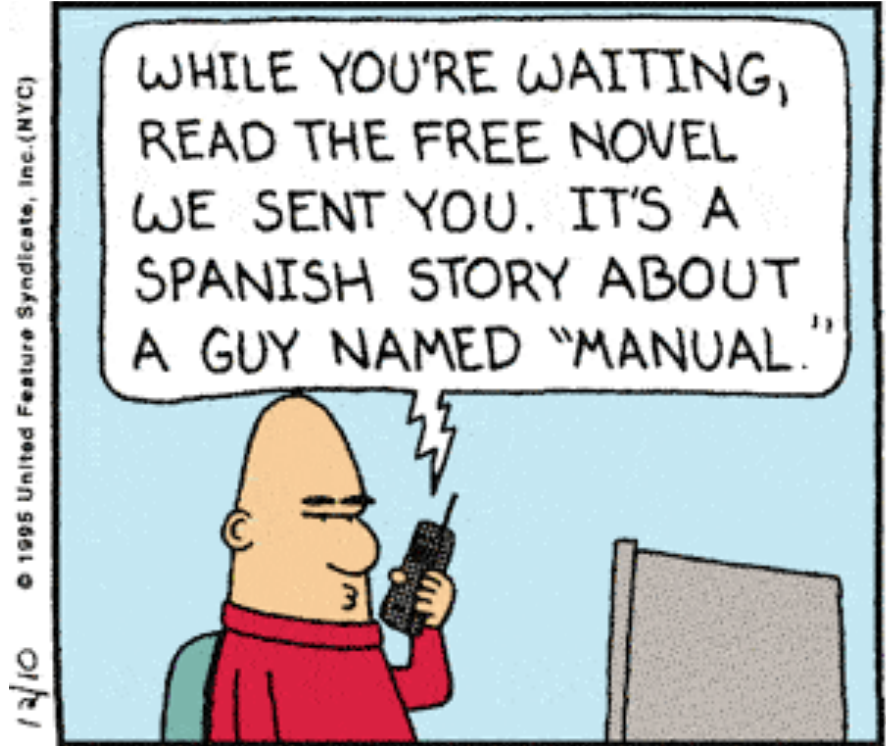
Jak se stát Debian vývojářem?

- Opravte chybu v balíčku, něco zabalte, přidejte se do existujícího týmu, napište dokumentaci, něco přeložte, ...
 - <https://www.debian.org/devel/join/newmaint>
 - <http://mentors.debian.net/>
- Začněte se starat o nějaký balíček
 - A staňte se Debian správcem (Debian Maintainer)
 - <https://wiki.debian.org/DebianMaintainer>
- Až budete mít nějakou práci za sebou
 - Staňte se Debian vývojářem (Debian Developer)

```
pub 4096R/0x0C99B70EF4FCBB07 2010-11-14 [expires: 2017-05-08]
Key fingerprint = 30B9 33D8 0FCE 3D98 1A2D 38FB 0C99 B70E F4FC BB07
uid [ultimate] Ondřej Surý <ondrej@sury.org>
uid [ultimate] [jpeg image of size 3934]
uid [ultimate] Ondřej Surý <ondrej@debian.org>
uid [ultimate] Ondřej Surý <ondrej.sury@nic.cz>
sub 4096R/0xC8BE659746582F7F 2010-11-14 [expires: 2017-05-25]
Key fingerprint = 24C8 D117 D44D 9709 38E0 0AC7 C8BE 6597 4658 2F7F
sub 4096R/0x763748032B5E9DCD 2015-06-16 [expires: 2016-06-15]
Key fingerprint = 2F7E 2BC7 04A9 2D69 51E0 7176 7637 4803 2B5E 9DCD
```

Knihovnička

- Debian Policy Manual
 - <https://www.debian.org/doc/debian-policy/>
- Debian Developer's Reference
 - <https://www.debian.org/doc/manuals/developers-reference/>
- Debian New Maintainers' Guide
 - <https://www.debian.org/doc/manuals/maint-guide/>
- Preferred debian packaging setup for dkg
 - https://wiki.debian.org/DanielKahnGillmor/preferred_packaging
- Balíčkování specifických věcí
 - Python, Perl, PHP, Ruby, Kernel, ...



Otázky a ukázky

DebConf 2017

- Usilujeme o pořádání DebConf 2017
 - <https://wiki.debconf.org/wiki/DebConf17/Plans/Prague>
- Hledáme:
 - Dobrovolníky
 - Místní tým
 - Wiki
 - Sponzory
- Hlaste se!

Díky za pozornost