I no longer do a "make install" on my servers.

Why I love apt-get install

Summary

- Speaker introduction
- Problem Space
- Recompiling a package for stable
- Meta-packages
- Private packages with one binary
- Private packages from vendor rpm
- Private packages from random software
- Conclusion

Speaker

- SysAdmin of Un*x and Linux since 1993
- Debian contributor
- Free Software programer on spare time: switchconf, at daemon, logcheck
- Debian Developer since February 2017

Problem Space

- Was responsible for some Un*x servers
- Needed more tools than where available on Un*x
- Several software installed using “make install”
- No easy way to list installed software and their versions
- No safe way to remove software
- Missing the power of apt-get
- Decided to not run a “make install” on a new machine

What is a Debian package?

- Is an archive that contains the files to install and a group of control files
- Dependencies declarations like depends, recommends, suggests or conflicts
- Scripts to install, upgrade or remove the package

Problem Space II

- Several PCs for “Laboratórios de Tecnologias de Informação”, 60 a 80 units, 3 or 4 variants.
- Dual boot Windows and Debian stable/testing
- One multi user server with the same Debian software for remote access by the students
- Several requests for software not available on Debian or other Distributions
- Sometimes requests for versions more recent than on Debian stable

How I started

- Learned how to recompile, in crude way, a testing or unstable package for stable
- Setup a private deb package repository, a mirror in http://deb-ppa-archive.calhariz.com
- Created several meta-packages to make easier the instalation of software by student classes like tp-200402-aed, tp-200402-ia

Recompile a Debian package

- Get the sources of the package, dget URL to .dsc file
- Install the build depends, sudo apt-get build-depends package
- Increment the version in the changelog, dch --bpo
- Recompile and rebuild the package, debuild -uc -us

tp-200402-aed – equivs file

- Section: taguspark
- Priority: optional
- Standards-Version: 3.5.10
-
- Package: tp-200402-aed
- Version: 0.3
- Maintainer: Jose Calhariz <jose.calhariz@tagus.ist.utl.pt>
- Depends: gdb, ddd, make, valgrind, kcachegrind, valgrind-callgrind, xemacs21, xemacs21, xemacs21-mule, xemacs21-gnome-mule, xemacs21-support, xemacs21-bin, xemacs21-mulesupport, xemacs21-basesupport, mysql-client-4.1, libmysqlclient14-dev,
- Architecture: all
- Description: Programas para a cadeira de Algoritmos e Estruturas de Dados no Taguspark, 2º Semestre 2004/2005.

Building private packages by vendor

- hp-dl360g4p52-rom
- hp-dl360g4pp54-rom
- hp-dl385a05-rom
- hp-ilo-rom
- hp-msa20-rom
- hp-sa6400-rom
- hp-sa6i-rom
- hp-template-scexe

hp-ilo-rom - Makefile

```
SCEXE=CP010428.scexe
URL=http://ftp.hp.com/pub/softlib2/software1/sc-linux-fw-ilo/p1980791503/v51876/$(SCEXE)

build:configure build-up-stamp
build-up-stamp:
	if [ ! -f $(SCEXE) ] ; then wget --progress=dot:mega $(URL) ; fi
	touch build-up-stamp

install:build install-up-stamp
install-up-stamp:
	install --directory $(DESTDIR)/$(LIBDIR)
	install --preserve-timestamps $(SCEXE) $(DESTDIR)/$(LIBDIR)/updater.scexe
	touch install-up-stamp
```

hp-ilo-rom - postinst

```
SCEXE=updater.scexe
case "$1" in
    configure)
        # Because it needs to use old sintax of coreutils, like tail +number
        export _POSIX2_VERSION="199209"
        # Because it returns an error when firmware is updated
        /usr/lib/hp-ilo-rom/updater.scexe || true
    ;;
    abort-upgrade|abort-remove|abort-deconfigure)
    ;;
    *)
        echo "postinst called with unknown argument \`$1'" >&2
        exit 1
    ;;
esac
```

Building private packages from vendor rpm

- hpacucli
- hpadu
- hpasm
- hpnicfwupg
- hponcfg
- hprsm
- hp-template-alien

hpacucli - Makefile

```
RPMFILE=$(PACKNAME)-$(RPMVER)-18.linux.rpm
URL=ftp://ftp.compaq.com/pub/softlib2/software1/pubsw-linux/p308169736/v41554/$(RPMFILE)

build-up-stamp:
	if [ ! -f $(RPMFILE) ] ; then wget --progress=dot:mega $(URL) ; fi
	fakeroot alien -g --single --scripts $(RPMFILE)
	cp $(BUILDDIR)/debian/postinst $(BUILDDIR)/debian/postrm $(BUILDDIR)/debian/preinst $(BUILDDIR)/debian/prerm $(BUILDDIR)/debian/copyright debian/
	if [ ! -d $(BUILDDIR) ] ; then echo "Builddir \"$(BUILDDIR)\" don't exist, something wrong happened" ; exit 1 ; fi
	# Apply patches
	dpatch apply-all
	touch build-up-stamp

install-up-stamp:
	find $(BUILDDIR) -maxdepth 1 -mindepth 1 -not -name debian -print0 | \
		xargs -0 -r -i cp -a {} $(DESTDIR)
	touch install-up-stamp
```

Random software for the classes

- debmake is your friend
- Sometimes is very easy to do a private package

edumips64 - edumips64

- #!/bin/bash
-
- JAR_DIR=/usr/share/java
- JAR_FILE=edumips64-0.5.2.jar
-
- java -jar ${JAR_DIR}/${JAR_FILE} &

edumips64 - Makefile

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

install-up-stamp:
	[ -d $(DESTDIR)$(BINDIR) ] || mkdir -p $(DESTDIR)$(BINDIR)
	[ -d $(DESTDIR)$(SHAREDIR) ] || mkdir -p $(DESTDIR)$(SHAREDIR)
	cp $(PACKNAME) $(DESTDIR)$(BINDIR)
	cp $(PACKNAME)-$(VER).jar $(DESTDIR)$(SHAREDIR)
	touch install-up-stamp
```

papi - Makefile

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

install-up-stamp:
	[ -d $(MANDIR) ] || mkdir -p $(MANDIR)
	cp $(PACKNAME).8 $(MANDIR)
	[ -d $(ETCDIR) ] || mkdir -p $(ETCDIR)
	[ -d $(ETCDIR)/before.d ] || mkdir -p $(ETCDIR)/before.d
	[ -d $(ETCDIR)/after.d ] || mkdir -p $(ETCDIR)/after.d
	[ -d $(LOGROTATEDIR)/ ] || mkdir -p $(LOGROTATEDIR)
	cp debian/logrotate $(LOGROTATEDIR)/$(PACKNAME)
	cp conf $(ETCDIR)
	cp plugins.before/*.sh $(ETCDIR)/before.d/
	cp plugins.after/*.sh $(ETCDIR)/after.d/
	[ -d $(BINDIR) ] || mkdir -p $(BINDIR)
	cp $(PACKNAME) $(BINDIR)
	touch install-up-stamp
```

papi - control

- Source: papi
- Section: electronics
- Priority: optional
- Maintainer: Jose Calhariz <jose.calhariz@tagus.ist.utl.pt>
- Build-Depends: debhelper (>= 5.0.0), gfortran, dpatch,
- Standards-Version: 3.6.1
-
- Package: papi
- Architecture: any
- Depends: ${shlibs:Depends}, ${misc:Depends}
- Description: <insert up to 60 chars description>
- <insert long description, indented with spaces>

storcli - Makefile

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

install-up-stamp:
	install --directory $(DESTDIR)/$(SBINDIR)
	install --preserve-timestamps storcli64 $(DESTDIR)/$(SBINDIR)/storcli
	touch install-up-stamp
```

winmips64 - Makefile

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

install-up-stamp:
	[ -d $(DESTDIR)$(SHAREDIR) ] || mkdir -p $(DESTDIR)/$(SHAREDIR)
	unzip winmips64.zip -d $(DESTDIR)/$(SHAREDIR)
	cp winmips64.pth $(DESTDIR)/$(SHAREDIR)
	[ -d $(DESTDIR)$(BINDIR) ] || mkdir -p $(DESTDIR)/$(BINDIR)
	cp winmips64-cmd $(DESTDIR)/$(BINDIR)/winmips64
	touch install-up-stamp
```

winmips64 – control

- Source: winmips64
- Section: taguspark
- Priority: optional
- Maintainer: Jose Calhariz <jose.calhariz@tagus.ist.utl.pt>
- Build-Depends: debhelper (>= 4.0.0), unzip
- Standards-Version: 3.6.1
-
- Package: winmips64
- Architecture: any
- Depends: wine, ${shlibs:Depends}, ${misc:Depends}
- Description: is an instruction set simulator
- WinMIPS64 is an instruction set simulator, and is designed as a
- replacement for the popular Microsoft Windows utility WinDLX. The
- classic text Computer Architecture - a Quantitative Approach, by
- Hennessy & Patterson, 4th edition from its 3rd edition has switched
- from the 32-bit DLX architecture, to the 64-bit MIPS
- architecture. Hence the need for a new teaching tool. WinDLX had a
- very nice friendly user interface, including a full graphical
- simulation of the 5-stage pipeline. Therefore it was decided to
- create a similar tool for the MIPS64 with a very similar interface.

yay - Makefile

```
OPTFLAGS   = -Wall
SDL_LIBS   = `sdl-config --static-libs`
SDL_CFLAGS = `sdl-config --cflags`
CFLAGS     = $(OPTFLAGS)  $(SDL_CFLAGS)
LDFLAGS    = $(SDL_LIBS)

CSRC       = yay.c
TARGET     = yay
OBJ        = $(CSRC:.c=.o)

default: $(TARGET)

%.o: %.c
	$(CC) $(CFLAGS)  -c -o $@ $<

$(TARGET): $(OBJ)
	$(CC) -o $@ $(OBJ) $(LDFLAGS)

clean:
	rm $(OBJ) $(TARGET)
```

yay - rules

```
#!/usr/bin/make -f
# -*- makefile -*-
# Sample debian/rules that uses debhelper.

# Uncomment this to turn on verbose mode.
#export DH_VERBOSE=1


PACKNAME=$(shell dh_listpackages)
SOURCENAME=$(shell dpkg-parsechangelog | grep "Source:" | cut -d ' ' -f 2)
VER=2.74a.1
DEBVER=$(shell dpkg-parsechangelog | grep "Version:" | cut -d ' ' -f 2)
VER=$(shell dpkg-parsechangelog | grep "Version:" | cut -d ' ' -f 2 | cut -f 1 -d -)
MDIST=lenny
DIST=$(shell dpkg-parsechangelog | grep "Distribution" | cut -d ' ' -f 2)
SECTION=main
CATEG=other_sources
REPOSTARGET=root@debian.tagus.ist.utl.pt:/var/www/debian/debian/dists/$(MDIST)/$(DIST)/$(SECTION)/
REPOSARCH=~/debian/$(SECTION)/$(CATEG)/$(SOURCENAME)

PACKAGE=$(shell dh_listpackages)
```

yay - rules

```
• configure: configure-stamp
• configure-stamp:
•         dh_testdir
•         # Add here commands to configure the package.
•
•         touch configure-stamp
•
• build: configure build-stamp
• build-stamp:
•         dh_testdir
•
•         # Add here commands to compile the package.
•         $(MAKE)
•         #docbook-to-man debian/$(PACKAGE).sgml > $(PACKAGE).1
•
•         touch build-stamp
```

yay - rules

- install: build
- dh_testdir
- dh_testroot
- dh_clean -k
- dh_installdirs
-
- # Add here commands to install the package into debian/$(PACKAGE).
- cp yay $(CURDIR)/debian/$(PACKAGE)/usr/bin/yay
-

yay - control

```
Source: yay
Section: graphics
Priority: extra
Maintainer: Jose Calhariz <jose.calhariz@tagus.ist.utl.pt>
Build-Depends: debhelper (>= 5.0.0), libsdl-dev
Standards-Version: 3.6.1

Package: yay
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description:  yay - yet another YUV viewer
 yay simply displays 4:2:0, 4:2:2, 4:4:4 YUV and Y-only pictures and
 sequences.
 .
 It tries to find a first match of geometry information like
 _<width>x<height> somewhere in the path/filename, e.g.
 .
    * mobile_352x288.yuv or
    * path/to/yuv/mobile_640x480x400frames/file.yuv
 .
 yay depends on SDL (simple directmedia layer), which is available for
 a lot of different platforms.
```

yay - control

- Source: yay
- Section: graphics
- Priority: extra
- Maintainer: Jose Calhariz <jose.calhariz@tagus.ist.utl.pt>
- Build-Depends: debhelper (>= 5.0.0), libsdl-dev
- Standards-Version: 3.6.1
-
- Package: yay
- Architecture: any
- Depends: ${shlibs:Depends}, ${misc:Depends}
- Description: yay - yet another YUV viewer
- yay simply displays 4:2:0, 4:2:2, 4:4:4 YUV and Y-only pictures and
- sequences.
- .
- It tries to find a first match of geometry information like
- _<width>x<height> somewhere in the path/filename, e.g.
- .
-    * mobile_352x288.yuv or
-    * path/to/yuv/mobile_640x480x400frames/file.yuv
- .
- yay depends on SDL (simple directmedia layer), which is available for
- a lot of different platforms.

Conclusion

- Sometimes is very easy to build a private package
- On a private package we can do many shortcuts
- Fill in the control file for documentation is very important
- Sometimes a simple debmake is enough

Thank you