

Nobody expects the Finnish Inquisition



or: confessions of a Debian package torturer

Lars Wirzenius <liw@iki.fi>

February 25, 2006 @ FOSDEM

In space no one can hear you scream

- Debian's goal: the best operating system, ever.
- Little automatic testing. Quality assured by massive user base. This doesn't actually work well for low-popularity packages.
 - Does an obscure package uninstall properly? Will its maintainer and four real users ever uninstall it? Does anyone test this kind of thing? What happens when someone tries the package and decides they don't want it?
 - Even popular packages have problems.

Your mission, should you decide to accept it...

- Systematic testing of all packages in Debian requires automation or an army of drudges. And I have better use for an army...
- Therefore **piuparts** was born:
 - Package Installation, UPgrading And Removal Testing Suite
 - Builds a minimal chroot, installs a package, removes it, and checks that the chroot is back to what it was before the installation.
 - Also upgrades to new package version, or between Debian releases.

Too much butter on those trays

- `piuparts *.deb`
 - Creates a **chroot** automatically
 - Uses your `/etc/apt/sources.list` to find the mirror
 - **Installs**, then **removes and purges** a package. Then installs via `apt-get`, **upgrades**, `removes`, and `purges`.
 - Reports **new**, **removed**, or **modified** files. Some changes are ignored by default, because they're OK.
 - Lots of output. Errors, if any, are at the end, but sometimes there's useful stuff earlier on that helps debugging.

That's not a knife. THAT's is a knife.

```
0m0.0s DEBUG: Setting up minimal chroot for sid at /tmp/tmpfWhyUZ.
0m0.0s DEBUG: Starting command: debootstrap --resolve-deps sid
/tmp/tmpfWhyUZ http://liw.iki.fi/debian/
0m0.1s DUMP:    I: Retrieving Release
...
1m1.9s DUMP:    I: Base system installed successfully.
...
1m2.2s DEBUG: Created policy-rc.d and chmodded it.
1m2.2s DEBUG: NOT minimizing chroot because of dpkg bug
...
1m3.4s DEBUG: Starting command: chroot /tmp/tmpfWhyUZ dpkg -i tmp/liwc_1.20-
2_i386.deb
...
1m3.5s DEBUG: Starting command: chroot /tmp/tmpfWhyUZ apt-get -yf --no-
remove install
...
1m3.9s DEBUG: Starting command: chroot /tmp/tmpfWhyUZ dpkg --remove liwc
...
1m3.9s DEBUG: Starting command: chroot /tmp/tmpfWhyUZ dpkg --remove --
pending
...
1m4.0s DEBUG: Starting command: chroot /tmp/tmpfWhyUZ dpkg --purge liwc
...
1m4.7s INFO: PASS: Installation and purging test.
...
1m6.5s INFO: PASS: Installation, upgrade and purging tests.
```

You think you can catch Keyser Soze?

```
0m52.9s ERROR: Command failed (status=25600): 'chroot /tmp/tmpM1bBtd  
apt-get -y install aspell-lt'
```

```
...
```

```
Setting up aspell-lt (1.1-4) ...
```

```
dpkg: error processing aspell-lt (--configure):
```

```
subprocess post-installation script returned error exit status 1
```

```
Errors were encountered while processing:
```

```
aspell-lt
```

```
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

My God, it's full of stars.

```
0m7.8s ERROR: Package purging left files on system:  
/usr/lib/aspell  
  owned by: libaspell15, aspell-no  
/usr/lib/aspell/no.dat  
/usr/lib/aspell/no.multi  
/usr/lib/aspell/no_phonet.dat
```

We're no longer called Sonic Death Monkey.

1) `postinst` creates file (config, log, ...), `postrm` does not remove it

2) Bad handling of alternatives

- different names in `postinst` / `postrm`

3) Unconditionally using `ucf` in `postrm` during purge

- may **rely** on essential packages only

4) `>/dev/null 2>&1`

5) Not using `invoke-rc.d` if available

I feel the need, the need for speed.

- Building a chroot takes quite a while.
- You build your packages with pbuilder, don't you?
 - `piuparts -p`
- Or you can have piuparts remember its own chroot
 - `piuparts -s sid.tar.gz`
 - `piuparts -b sid.tar.gz`
- Either way, it'll only take seconds now.
- A local mirror is luvrly, too!

That's all he does! You can't stop him.

- Systematic testing of etch/sid since August
 - `piuparts-master / piuparts-slave`
 - install/remove/purge within sid
 - upgrades from sarge via etch to sid
 - logs of failed tests analyzed manually, bugs reported when confirmed by analysis (some failures due to piuparts, test environment)
- About 260 bugs reported, 110 fixed by uploads
 - 40 % isn't all that good, actually...

Whatever life holds in store for me, I will never forget these words

- Mount `/proc`, check for processes
- Automate analysis of failed tests
 - no automated bug reporting, ever, though
- Run on multiple architectures
- Re-test all packages after changes in piuparts
- Deal with packages that replace parts of the minimal chroot
- Various other improvements

Do you think you might agree **not** to marry me?

- Help process failed logs
 - no idea yet how this should be set up, though
 - put all failed logs into a version control system?
- Test your own packages before uploading
- Suggest enhancements, possibly with patches

I GIGGLE EVERY TIME I SEE A
PACKAGE FAIL MY TESTING!

DO YOU *WANT* ME TO GIGGLE
AT YOU?

BE GIGGLE SAFE! RUN PIUPARTS
BEFORE UPLOADING!