Software Packages (not only) in Linux

Michal Hrušecký

openSUSE Team @ SUSE

May 13, 2013

Introduction

Who am I?

- used to maintain build of distribution from OpenEmbedded
- maintainer of MySQL packages in openSUSE/SLE
- Gentoo developer

⇒ package maintainer for past seven years for various distributions

What is the talk about?

- packaging and software packages - what, why and how

Software Packages - what and why

What is software package?

- archive with files to be installed
- metadata for package manager
- most common are `.rpm` and `.deb`

Usual metadata:

- description
- license
- dependencies
- extra installation instructions
- checksums

Why do we need them?

Convenience

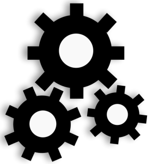
- easy installation
- easy update
- clean uninstall
- easy distribution of software

Security

- avoid development tools on production machines
- detect tempering

Life without packages

- Get a tarball
- Find out dependencies
- Build and install dependencies
- Build and install package itself
  - `./configure --prefix=/usr --enable-this \ --disable-that --with-lib=there`
  - `make`
  - fix what is broken
  - `make install`
  - try it
  - `make uninstall`
  - clean up left-overs



Life with packages

- *pkg-manager* `install pkg`
- *pkg-manager* `remove pkg`

RPM

Basic information

- one of the oldest in Linux world
- created in 1997 for RedHat
- used by various distributions
  - RedHat/Fedora and derivates
  - openSUSE/SLE, Mandriva/Mageia, Meego, ...
- several frontends to make operations easier
  - yum - used by RedHat/Fedora and derivates
  - zypper - used by openSUSE/SLE and Meego
  - urpmi - used by Mandriva/Mageia

File format

- 32 bytes lead (magic, rpm version, type of package, architecture)
- signature
- header
    - name, version and release
    - license
    - summary
    - description
    - changelog
    - requires and provides
    - file list with rights, md5s and more
- archive
    - cpio, compressed by gzip, bzip2, xz, . . .

`.spec` file

- source recipe for `rpm`
- contains multiple sections
  - information about package
  - `%prep` section
  - `%build` section
  - `%install` section
  - `%check` section
  - `%files` section
  - optionally more: `%pre`, `%post`, ...

Example .spec file (1/3)

```
Name:           nano
BuildRequires:  ncurses-devel
Url:            http://www.nano-editor.org/
License:        GPL v3 or later
Group:          Productivity/Editors/Other
Summary:        Pico Editor Clone with Enhancements
Version:        2.1.9
Release:        1
Source:         %{name}-%{version}.tar.bz2
BuildRoot:      %{_tmppath}/%{name}-%{version}-build

%description
GNU nano is a small and friendly text editor. It aims to emulate
Pico text editor while also offering a few enhancements.
...
```

Example .spec file (2/3)

```
%prep
%setup -q

%build
%configure --disable-rpath --enable-all
%{__make} %{?jobs:-j%jobs}

%install
%make_install
%find_lang %{name}

%check
make check
```

Example .spec file (2/3)

```
%files -f %{name}.lang
%defattr(-, root, root)
%doc AUTHORS BUGS COPYING ChangeLog INSTALL NEWS README THANKS TO
%{_mandir}/fr
%{_mandir}/fr/man?
%{_mandir}/man?/*.*
%{_mandir}/*/man?/*.*
%{_bindir}/*
%{_infodir}/*.gz
%{_datadir}/nano

%changelog
```

Portage

Basic information

- created for Gentoo distribution
- doesn't focus on binary packages
- used mainly in Gentoo Linux
- used by Gentoo/Alt project
  - Gentoo/FreeBSD
  - Gentoo prefix
    - Linux, MacOS, HP-UX, Solaris, Windows, ...
- usually only source packages to be compiled
- provides means for customisation
- supports inheritance

Supported functions

- `pkg_setup` - pre-build environment configuration and checks
- `src_unpack` - unpacking sources
- `src_prepare` - patching
- `src_compile` - compilation itself
- `src_test` - optional testing
- `src_install` - installation
- `pkg_preinst`, `pkg_postinst`, `pkg_prerm`, `pkg_postrm`

Example .ebuild file (1/2)

```
EAPI=4

inherit libtool

DESCRIPTION="GNU charset conversion library for libc which doesn'
HOMEPAGE="http://www.gnu.org/software/libiconv/"
SRC_URI="ftp://ftp.gnu.org/pub/gnu/libiconv/${P}.tar.gz"

LICENSE="LGPL-2.1"
SLOT="0"
KEYWORDS="~amd64 ~ppc ~sparc ~x86"
IUSE="nls"

DEPEND="!sys-libs/glibc"
```

Example .ebuild file (1/2)

```
src_prepare() {
   # Make sure that libtool support is updated to link "the linux
   # on FreeBSD.
   elibtoolize
}

src_configure() {
    econf $(use_enable nls)
}
```

Bitbake

Basic information

- created for OpenEmbedded (meta-distribution)
- supports multiple build and target operating systems
- builds whole distribution
- reuses staging directory
- focuse on binary packages
- supports inheritance
- inspired by portage
- Final package can be .rpm, .deb, .opkg, .tgz, . . .

Example BitBake recipe

```
DESCRIPTION = "GNU nano (Nano's ANOther editor, or
Not ANOther editor) is an enhanced clone of the
Pico text editor."
HOMEPAGE = "http://www.nano-editor.org/"
LICENSE = "GPLv2"
SECTION = "console/utils"
DEPENDS = "ncurses"
SRC_URI = "http://www.nano-editor.org/dist/v2.0/nano-${PV}.tar.gz
           file://glib.m4"
inherit autotools
EXTRA_OECONF = "--enable-all"

do_configure_prepend () {
           install -m 0644 ${WORKDIR}/glib.m4 m4/
}
```

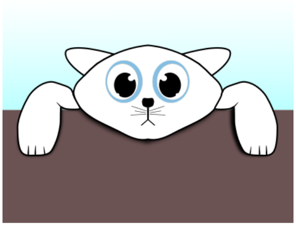
Another BitBake example

```
...

PACKAGES_DYNAMIC = "libpurple-protocol-*"
python populate_packages_prepend () {
       purple   = bb.data.expand('${libdir}/purple-2', d)

       do_split_packages(d, purple, '^lib(.*)\.so$',
              output_pattern='libpurple-protocol-%s',
              description='Libpurple protocol plugin for %s',
              prepend=True, extra_depends='')

}

...
```

Open Build Service

What users wants from developers

- Provide packages for as many distribution as he can
- Provide latest testing version/nightlies to the testers
- Maintain some stable versions of his software
- Make sure binaries work
    - recompile software whenever dependencies changes
    - compile against right versions of libraries



Friendly developer therefore needs

- Provide users with packages they want
- Let users help with packaging but have a last word
- Build packages quickly and automatically
- Make all building of packages as automatic as possible
- Get notification when something fails
- Make use of VCS you already have

Distribution needs

- Package many applications with complicated dependencies
- Rebuild them as needed
- Build everything fast and make use of multiple machines
- Let people easily contribute new packages
- Workflow for maintenance updates
- Know who maintains what
- Workflow for security updates
- Create DVDs and other images

What is Open Build Service?

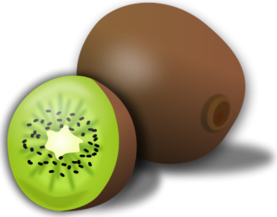
- builds packages
- open source server application
  - you can get your own instance
  - API between instances
- being used to develop openSUSE, SLE, Meego and others
- developed mainly by SUSE

What can it do?

- Distribute builds among build servers
- Follow dependencies
- Build packages for different distributions:
  - openSUSE/SLE
  - Fedora/RHEL/CentOS
  - Mandriva
  - Debian/Ubuntu
  - ...
- Create repositories for these distribution
- Build packages different architectures:
  - i586/x86_64
  - arm/PPC/PPC64
  - ...
- Push packages to the mirrors

Can it build more?

- Build packages for architectures you don't have
  - qemu-user
  - `binfmt`
- Build images - kiwi
  - image builder
  - self expandable images
  - easy to deploy
  - highly configurable
  - used by SUSE Studio



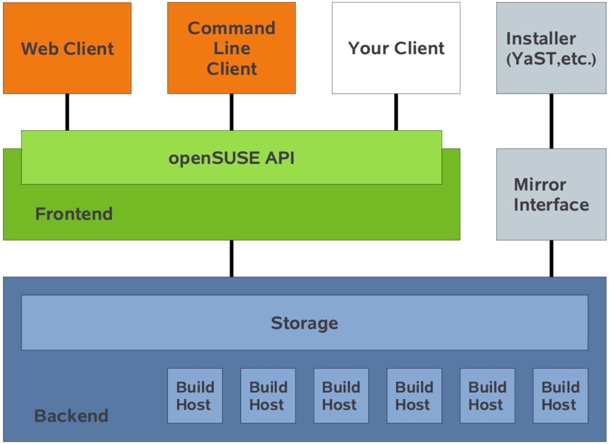
From what can it build?

- Behaves like VCS
  - `osc co`
  - `osc commit`
  - `osc log`
  - `osc diff`
- Your own VCS - git, subversion, ...
  - clone & create tarball
  - download tarball
  - extract and modify recipe
  - compile

What else can it do?

- Hidden projects
  - Usefull for security
  - Some patches can't be made public for some time
- Submit requests
  - User creates patched version
  - User sends it to another project
  - Maintainers review the request
  - Maintainers accept/decline request
- Maintenance requests
  - Special kind of submit request
  - One request for multiple projects

Architecture



How to get started?

- Try openSUSE Build service

  `http://build.opensuse.org`

- Get obs appliance from SUSE Studio

  `http://www.susestudio.com`

- Get code and news from obs website

  `http://www.buildservice.org`

Thank you! Questions?