

Easy Software-Installation on Linux, Solaris, NetBSD, etc. using pkgsrc



Problems

Installation of Open Source software on Unix and Unix-like systems has a number of problems:

- Many programs and lots of version changes
- Compilation costs time
- Software often is not written with portability in mind (but we don't want to give a coding lesson here...)
- Installation is not trivial



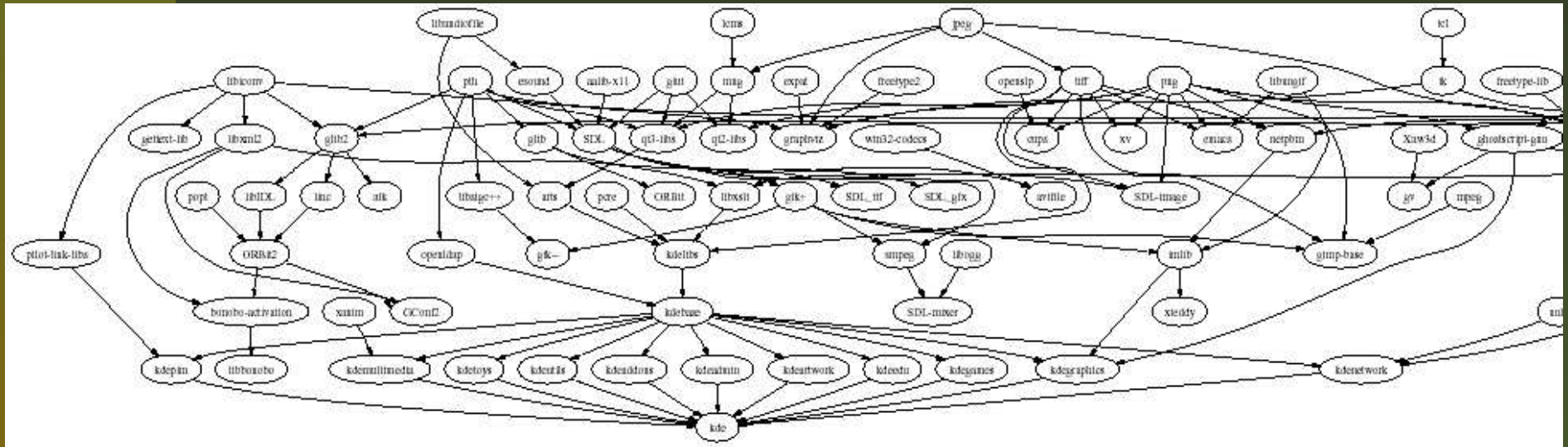
Problems (cont'd)

- Installation is not trivial:
 - Some basic knowledge about tools is necessary
 - Various ways to configure things (GNU autoconf, Imake, ...)
 - Side effects (depending on other packages, compiler, ...)
 - Many inter-dependent packages
 - Troubleshooting requires expert knowledge



Problems (cont'd)

Illustration of complexity of inter-depending packages:

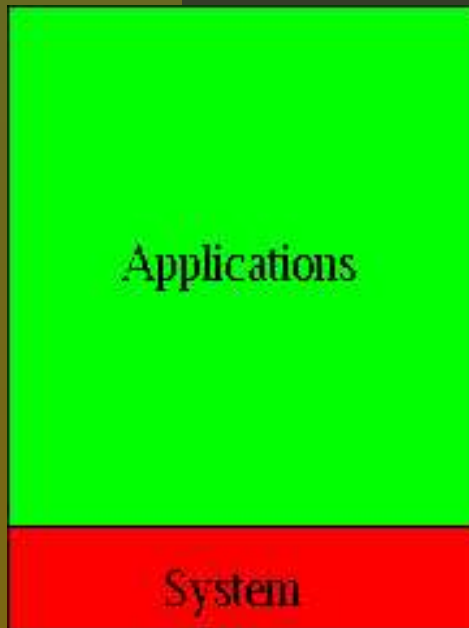


(created from a pkgsrc system running NetBSD, using pkgdepgraph and dot/graphviz)



Solution: It depends! (1/2)

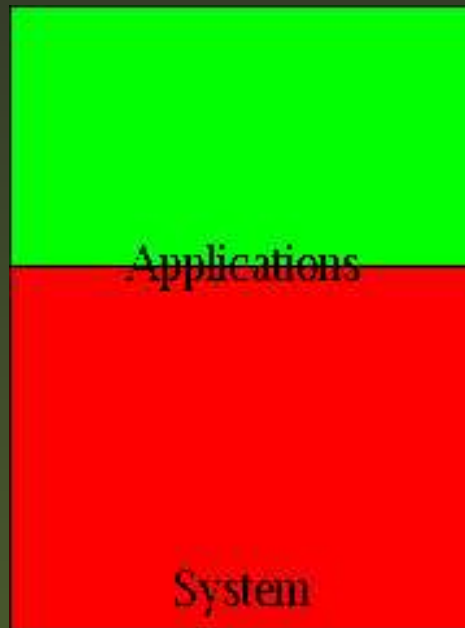
Classic, flexible
software management:



- difficult to install
+ easy to maintain

E.g. Solaris, Irix,
Linux From Scratch

Hybrid systems:



+ easy to install
+ easy to maintain

E.g. NetBSD, FreeBSD,
Debian & Gentoo Linux

Complete integration
of applications and system:



+ easy to install
- difficult to maintain

E.g. SuSE, RedHat,
Mandrake Linux



Solution: It depends! (2/2)

Where do you want to go today?

- **Easy Installation:** choose this if your software doesn't change often. Use ready-to-user binary distribution. E.g. for desktop systems install Windows or SuSE Linux from CD/DVD.
- **Easy Maintenance:** choose this if you have few packages that change a lot. Take a stable base operating system, and install important packages on your own, e.g. compile on your own on a webserver with Solaris, Apache and PHP.
- **Both:** Welcome to pkgsrc!



A Cross-Platform Solution: pkgsrc



Introducing pkgsrc

- System for easy installation and updating of packages
- Source-based package management system
- Uses original source code for compiling
- Creation and installation of binary packages is possible
- Components: Management tools & packages collection (pkgsrc)
- Automatic handling of dependencies (of course!?!)



Introducing pkgsrc (cont'd)

- Originally ported from FreeBSD to NetBSD
- Primary development platform of pkgsrc: NetBSD
- Ported to: AIX, BSD/OS, Darwin, FreeBSD, Irix, Linux, NetBSD, OpenBSD, Solaris, Windows w/ “Interix”
- Linux Distributions: SuSE 9.0, Debian, ROOT Linux, Slackware, RedHat 8.1/9, Mandrake 9.2, Bluewall, ...



pkgsrc in Detail



How to get going

- Grab pkgsrc
- Install bootstrap kit (binary, or compile via pkgsrc/bootstrap)
- `cd pkgsrc/www/mozilla`
- `bmake install`



Grabbing pkgsrc

```
% cd $HOME/OS
% env CVS_RSH=ssh \
  cvs -d \
    anoncvs@anoncvs.NetBSD.org:/cvsroot \
    co pkgsrc
U pkgsrc/Makefile
U pkgsrc/Packages.txt
U pkgsrc/README
...
```

Alternative: ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-current/tar_files/pkgsrc.tar.gz



Bootstrap Kit: Binaries

- Grab a precompiled binary or compile on your own
- Precompiled binary kits are available on <http://www.pkgsrc.org/> for:

Darwin 7.0/powerpc OpenBSD 3.2/i386

Darwin 6.6/powerpc Slackware 8.1/i386

Debian Linux/i386 Slackware 9/i386

FreeBSD 4.7/i386 Solaris 8/sparc

FreeBSD 5.1/i386 Solaris 8/i386

IRIX 6.5/mips Solaris 9/sparc

IRIX64 6.5/mips Solaris 9/i386



Bootstrap Kit: Compiling (1/2)

```
% cd pkgsrc/bootstrap
% setenv MY_HOME $HOME/OS/OS-`uname -s`
% setenv LOCALBASE ${MY_HOME}/pkg
% setenv PKG_DBDIR ${MY_HOME}/db/pkg
% setenv PKGSRCDIR ${MY_HOME}/pkgsrc
% ./bootstrap \
?     --prefix=${LOCALBASE} \
?     --pkgdbdir=${PKG_DBDIR} \
?     --pkgsrcdir=${PKGSRCDIR} \
?     --ignore-user-check
===> bootstrap command: ./bootstrap --prefix=/home/feyrer/OS/OS-Linux/pkg -
===> bootstrap started: Sun Mar 28 21:19:05 CEST 2004
Working directory is: work
===> building as unprivileged user feyrer/bedienst
===> Creating mk.conf.example in work
===> running: (/bin/sh ./files/install-sh -d -o feyrer -g bedienst -m 755 /
===> running: (/bin/sh ./files/install-sh -d -o feyrer -g bedienst -m 755 /
===> running: (/bin/sh ./files/install-sh -d -o feyrer -g bedienst -m 755 /
===> running: (/bin/sh ./files/install-sh -d -o feyrer -g bedienst -m 755 /
.....
```



Bootstrap Kit: Compiling (2/2)

....

```
/usr/bin/install -c -m 444 linkfarm.cat1 /home/feyrer/OS/OS-Linux/pkg/man/c
/usr/bin/install -c -m 444 pkg_view.1 /home/feyrer/OS/OS-Linux/pkg/man/man1
/usr/bin/install -c -m 444 pkg_view.cat1 /home/feyrer/OS/OS-Linux/pkg/man/c
==> Installing packages(7) man page
==> running: /bin/sh ./install-sh -c -m 444 packages.cat7 /home/feyrer/OS/
```

Please remember to add /home/feyrer/OS/OS-Linux/pkg/bin to your PATH environment variable, and /home/feyrer/OS/OS-Linux/pkg/man to your MANPATH environment variable,

An example mk.conf file "work/mk.conf.example" with the settings you provided to "bootstrap" has been created for you. Please copy work/mk.conf.example to /etc/mk.conf.

You can find extensive documentation of the NetBSD Packages Collection in /home/feyrer/OS/OS-Linux/pkgsrc/Packages.txt and packages(7).

Hopefully everything is now complete.

Thank you

```
==> bootstrap started: Sun Mar 28 21:19:05 CEST 2004
```

```
==> bootstrap ended: Sun Mar 28 21:28:35 CEST 2004
```

```
%
```



Bootstrap Kit: Adjust \$PATH etc.

```
% cd $HOME/OS/OS-`uname -s`/pkg
% set path=( `pwd`/bin `pwd`/sbin $path )
% rehash
% setenv PKG_DBDIR $HOME/OS/OS-`uname -s`/db/pkg
%
% pkg_info
digest-20021220      Message digest wrapper utility
```



Installed Commands

The binaries installed by the bootstrap procedure provide the core functionality of the pkgsrc system:

```
% cd OS/OS-`uname -s`/pkg/
```

```
% ls bin sbin
```

```
bin:
```

```
bmake          cpio          ftp  
digest         pax          tar
```

```
sbin:
```

```
linkfarm      pkg_add      pkg_create   pkg_info  
mtree         pkg_admin   pkg_delete   pkg_view
```



Compiling Packages - Overview

Beware! Make sure that instead of “make” the BSD-compatible “bmake” is being used!

```
% setenv MAKECONF `pwd`/pkgsrc_env_no-root # see below
%
% cd $HOME/OS/pkgsrc
% cd misc/figlet
% bmake
% bmake install
...
%
% pkg_info
digest-20021220      Message digest wrapper utility
figlet-2.2.1nb1     Print text banners in fancy ASCII art ch
```



Compiling Packages - Details (1/2)

```
% bmake
==> *** No /home/feyrer/OS/OS-Linux/./distfiles/pkg-vulner
==> *** skipping vulnerability checks. To fix, install
==> *** the pkgsrc/security/audit-packages package and run
==> *** ' /home/feyrer/OS/OS-Linux/pkg/sbin/download-vulnera
=> Checksum OK for figlet221.tar.gz.
work.i386 -> /home/feyrer/OS/OS-Linux/tmp/misc/figlet/work.i
==> Extracting for figlet-2.2.1nb2
==> Patching for figlet-2.2.1nb2
==> Applying pkgsrc patches for figlet-2.2.1nb2
==> Overriding tools for figlet-2.2.1nb2
==> Configuring for figlet-2.2.1nb2
==> Building for figlet-2.2.1nb2
gcc -O2 -DDEFAULTFONTDIR=\"/home/feyrer/OS/OS-Linux/pkg/shar
chmod a+x figlet
gcc -O2 -o chkfont chkfont.c
%
```



Compiling Packages - Details (2/2)

```
% bmake install
==> Installing for figlet-2.2.1nb2
==> Becoming root@rfhinf032 to install figlet.
Warning: not superuser, can't run mtree.
Become root and try again to ensure correct permissions.
install -d -o feyrer -g bedienst -m 755 /home/feyrer/OS/OS-Linux
mkdir -p /home/feyrer/OS/OS-Linux/pkg/share/figlet
cp figlet /home/feyrer/OS/OS-Linux/pkg/bin
cp chkfont /home/feyrer/OS/OS-Linux/pkg/bin
chmod 555 figlist showfigfonts
cp figlist /home/feyrer/OS/OS-Linux/pkg/bin
cp showfigfonts /home/feyrer/OS/OS-Linux/pkg/bin
cp fonts/*.flf /home/feyrer/OS/OS-Linux/pkg/share/figlet
cp fonts/*.flc /home/feyrer/OS/OS-Linux/pkg/share/figlet
cp figlet.6 /home/feyrer/OS/OS-Linux/pkg/man/man6
==> Registering installation for figlet-2.2.1nb2
%
```



Compiling as Non-root

To use pkgsrc without root privileges, put the following into \$MAKECONF (shortened!):

```
MY_NAME!=          whoami
MY_GROUP!=         groups | sed 's/ .*$$//'
MY_HOME=           ${HOME}/OS
BINOWN=            ${MY_NAME}
BINGRP=            ${MY_GROUP}
WRKOBJDIR=         ${MY_HOME}/tmp
X11PREFIX=         ${MY_HOME}/pkg # X needs xpkgwedge installed!
LOCALBASE=         ${MY_HOME}/pkg
VARBASE=           ${MY_HOME}/var
OBJMACHINE=        1
SU_CMD=            /bin/sh -c
CHOWN=             true
CHGRP=             true
BINMODE=           755                # for Solaris strip(1)
```

Complete: http://www.feyrer.de/OS/pkgsrc_env_no-root!



Behind the Scenes

1. `make fetch`: Download sources
2. `make checksum`: Ensure integrity
3. `make install-depends`: Install required packages
4. `make extract`: Unpack
5. `make patch`: Apply patches
6. `make configure`: Configure
7. `make build`: Compile
8. `make install`: Install and register package (for `pkg_info(1)`, `pkg_delete()`, etc.)



Other Interesting Targets

- `make package`: Create binary package for `pkg_add(8)`
- `make clean`: Remove working directory
- `make deinstall`: Deinstall package
- `make replace`: Replace installed package with new version
- `make update`: Rebuild package and all dependencies



What packages are there: Categories

```
% cd .../pkgsrc/
% ls
CVS                cross              japanese          parallel
Makefile          crypto            lang              pkglocate
Packages.txt      databases        licenses         pkgtools
README            devel            mail              plan9
archivers         distfiles        math              print
athena            doc              mbone             security
audio             editors           meta-pkgs        shells
benchmarks        emulators        misc              sysutils
biology           finance          mk                templates
cad               fonts            nessus-libraries textproc
chat              games            nessus-plugins   time
comms             graphics         net               wm
converters        ham              news              www
corba             inputmethod     packages          x11
```



Example:d WWW Category

```
% cd .../pkgsrc
% ls www
CVS                libwww             php4-sablot
Makefile          links             pkg
Mosaic            links-gui         privoxy
MozillaFirebird  lynx             privoxy-user
adzap             lynx-current     py-HTMLgen
amaya             mMosaic          py-curl
analog           make_album       py-pcgi
ap-DBI           mknmz-wwwoffle  py-zpublisher
ap-Embperl       moz-bin-plugger  qDecoder
ap-access-referer moz-linux-plugger quanta
ap-aolserver     mozilla          quanta-docs
ap-auth-cookie   mozilla-bin      quanta3
ap-auth-ldap     mozilla-bin-nightly ruby-borges
ap-auth-pam      mozilla-flash-bin ruby-htmlsplit
ap-auth-pgsql    mozilla-flashplugin ruby-tag
ap-auth-postgresql mozilla-linux     ruby-uri
ap-auth-script   mozilla-stable   ruby-webrick
...
```



Number of Available Packages

```
% date
Sun Mar 28 22:07:02 MEST 2004
%
% cd .../pkgsrc/
% ls */*/Makefile | wc -l
    5348                <- total
% ls wip/*/Makefile | wc -l
    792                 <- SourceForge's pkgsrc-wip
% expr 5348 - 792
4556                   <- NetBSD's pkgsrc
```



Internals



Makefile: Construction Manual

```
% cat x11/xteddy/Makefile
# $NetBSD: Makefile,v 1.10 2002/08/25 21:52:57 jlam Exp $

DISTNAME=          xteddy-1.1
CATEGORIES=        x11 games
MASTER_SITES=      http://www.ITN.LiU.SE/~stegu/xteddy/

MAINTAINER=        johnam@mail.kemper.org
HOMEPAGE=          http://www.ITN.LiU.SE/~stegu/xteddy
COMMENT=           Xteddy is a cuddly teddy bear for your X Windows desktop

USE_BUILDLINK2=    YES
USE_X11=           YES
GNU_CONFIGURE=     YES

pre-install:
    ${INSTALL_DATA_DIR} ${PREFIX}/share/xteddy
    ${INSTALL_DATA_DIR} ${PREFIX}/share/xteddy/pixmaps

.include " ../../graphics/xpm/buildlink2.mk "

.include " ../../mk/bsd.pkg.mk "
```



Dependencies

Various ways:

- Compile-time only: `BUILD_DEPENDS`
- Compile- and runtime: `DEPENDS`
- Compile- and runtime: `buildlink[23].mk`



Dependencies: *DEPENDS

```
% cd ../pkgsrc/  
% grep ^DEPEND meta-pkgs/kde3/Makefile  
DEPENDS+=      kdeartwork-3.1.4:../../misc/kdeartwork3  
DEPENDS+=      kdeaddons-3.1.4:../../misc/kdeaddons3  
...
```

The variable `DEPENDS` is assigned pairs of “Name-Version:Directory”. “Name-Version” is name and version of the required package, “Directory” is the path relative to this pkg’s directory where the package can be found if it’s not installed and needs to be built from source.



Dependencies: buildlink[23].mk

These files contain variables which say ...

- which header-files to use
- which libraries to use
- which name+version of this package should be expected
- in which pkgsrc directory to look if the package needs to be installed
- if there are additional CPP flags to use
- if this package needs further packages installed



Example: tiff/buildlink3.mk

```
% cat graphics/tiff/buildlink3.mk
# $NetBSD: buildlink3.mk,v 1.7 2004/03/18 09:12:12 jlam Exp $

BUILDLINK_DEPTH:=          ${BUILDLINK_DEPTH}+
TIFF_BUILDLINK3_MK:=       ${TIFF_BUILDLINK3_MK}+

.if !empty(BUILDLINK_DEPTH:M+)
BUILDLINK_DEPENDS+=        tiff
.endif

BUILDLINK_PACKAGES:=       ${BUILDLINK_PACKAGES:Ntiff}
BUILDLINK_PACKAGES+=       tiff

.if !empty(TIFF_BUILDLINK3_MK:M+)
BUILDLINK_DEPENDS.tiff+=    tiff>=3.6.1
BUILDLINK_PKGSRCDIR.tiff?=  ../../graphics/tiff
.endif # TIFF_BUILDLINK3_MK

.include "../../devel/zlib/buildlink3.mk"
.include "../../graphics/jpeg/buildlink3.mk"

BUILDLINK_DEPTH:=          ${BUILDLINK_DEPTH:S/+$/+//}
```



Questions? Answers!

<http://www.pkgsrc.org/>

<http://www.NetBSD.org/packages/>

info@pkgsrc.org/

