

# Softwareverwaltung mit Pkgsrc

Ein Softwareverwaltungssystem  
für \*BSD, Solaris, AIX, HP/UX, Irix, OSF/1, Tru64, Linux ...  
und viele andere

**Stefan Schumacher**

**[www.net-tex.de](http://www.net-tex.de)**

`<stefan@net-tex.de>`

Id: pkgsrc-vortrag.tex,v 1.11 2006/11/28 14:16:59 stefan Exp

Magdeburg, November 28, 2006



# Inhaltsverzeichnis

- 1 Probleme der Softwareverwaltung
- 2 Pkgsrc
  - Fähigkeiten
  - Installation
  - Konfiguration mit mk.conf



# Probleme

- Viele verschiedene Programme und Versionen sowie Änderungen
- Kompilieren kostet Zeit und Geld
- Portabilitätsprobleme der Anwendungssoftware



# Probleme

- Installation von Hand ist nicht trivial:
  - Kenntnisse der Toolchain nötig
  - Diverse Konfigurationswerkzeuge (GNU autoconf, Imake, ...)
  - Abhängigkeiten und Abhängigkeiten und Abhängigkeiten ...
  - Passende Compiler und Toolchain nötig, problematisch bei verschiedenen Architekturen und Betriebssystemen



# Probleme

- Betriebssystemaktualisierungen ./.  
Anwendungsaktualisierungen
- Pakete in Basis und als Anwendung (Sendmail, Postfix, OpenSSH, OpenSSL, XFree/X.org)
- Trennung der Konfigurationsdateien, rc.d-Skripte, Binaries



## Lösungen: Was will ich haben

- System- und Architekturunabhängigkeit
- Installation von Quellcode *und* Binaries
- Paketdatenbank («*Was ist in welcher Version wo installiert?*«)
- Sichten für verschiedene Paketversionen



# Arten der Softwareverwaltung

- Binärinstallationen
  - MS Windows, SuSE ...
  - installieren Binaries von CD/DVD
  - schlechtes Versionsmanagement  $\rightsquigarrow$  Updates schwierig
- Handarbeit
  - stabiles Betriebssystem und Paketinstallation von Hand
  - manuelles Versionsmanagement bei häufigen Updates
  - flexibel aber aufwändig zu handhaben



# Arten der Softwareverwaltung

- Binärinstallationen
  - MS Windows, SuSE ...
  - installieren Binaries von CD/DVD
  - schlechtes Versionsmanagement  $\rightsquigarrow$  Updates schwierig
- Handarbeit
  - stabiles Betriebssystem und Paketinstallation von Hand
  - manuelles Versionsmanagement bei häufigen Updates
  - flexibel aber aufwändig zu handhaben
- einfache Versionsverwaltung und (Binär-)Installation
  - guter Kompromiss aus beiden Varianten
  - automatisiertes Versionsmanagement
  - manuelle Eingriffe möglich
  - flexibel aber weniger aufwändig, da automatisiert



# Arten der Softwareverwaltung

- Binärinstallationen
  - MS Windows, SuSE ...
  - installieren Binaries von CD/DVD
  - schlechtes Versionsmanagement  $\rightsquigarrow$  Updates schwierig
- Handarbeit
  - stabiles Betriebssystem und Paketinstallation von Hand
  - manuelles Versionsmanagement bei häufigen Updates
  - flexibel aber aufwändig zu handhaben
- einfache Versionsverwaltung und (Binär-)Installation
  - guter Kompromiss aus beiden Varianten
  - automatisiertes Versionsmanagement
  - manuelle Eingriffe möglich
  - flexibel aber weniger aufwändig, da automatisiert



# Geschichte

- 1997 Portierung der FreeBSD-Paketverwaltung auf NetBSD (Hubert Feyrer, Alistair G. Crooks)
- 1999 wilder Hack der Toolchain auf Solaris («Zoularis»)
- 1999 Linux
- 2001 Darwin / MacOS X
- 2002 FreeBSD, OpenBSD, IRIX
- 2003 AIX, BSD/OS
- 2004 DragonFlyBSD, Interix, OSF/1



- AIX
- BSD/OS
- Bluewall Linux
- Darwin PowerPC (MacOS X)
- Debian GNU Linux/i386
- DragonflyBSD/i386
- Fedora Linux/i386
- FreeBSD/i386
- IRIX (n32 bit/64 bit)
- Interix
- OSF/1
- OpenBSD/i386
- PC-BSD
- ROOT Linux
- RedHat Linux
- Slackware Linux/i386
- Solaris sparc & i386
- SuSE Linux



## Was es kann

- Lizenzen prüfen
- Paketquellcode herunterladen (Bulk, Shellskript, BAT-Datei)
- Patches einspielen
- Abhängigkeiten überprüfen und auflösen
- Integritätsprüfung des Codes und der Binaries (Prüfsummen)
- prüfen auf Sicherheitslücken und veraltete Pakete
- unprivilegierter Kompilationsprozess
- Aktualisierung des Paketbaumes oder eines Paketes und seiner Abhängigkeiten
- natürlich alles sowohl aus den Quellen als auch mit Binaries
- einfache und *zentrale* Konfiguration
- stabile und Entwickler-Zweige (Codefreeze)



# installieren und anwenden



# Installation

CVS oder FTP:

---

```
1 $ env CVS_RSH=ssh \  
2   cvs -d anoncvs@anoncvs.NetBSD.org:/cvsroot \  
3   co pkgsrc  
4 U pkgsrc/Makefile  
5 U pkgsrc/Packages.txt  
6 U pkgsrc/README  
7 ...
```

---

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



# Paketkategorien

```
1 $ cd ../pkgsrc/  
2 $ ls  
3 CVS                comms                games                misc                security  
4 Makefile           converters           geography            mk                  shells  
5 Packages.txt       cross                graphics             multimedia          sysutils  
6 README             databases            ham                  net                  templates  
7 archivers          devel                inputmethod         news                 textproc  
8 audio              distfiles            lang                 packages            time  
9 benchmarks         doc                  licenses             parallel            wm  
10 biology            editors              mail                  pkglocate           www  
11 bootstrap          emulators            math                  pkgtools            x11  
12 cad                 finance              mbone                print  
13 chat                fonts                meta-pkgs            regress
```

# WWW-Kategorie

```
1 $ cd ../pkgsrc
2 $ ls www
3 CVS                cadaver             jakarta-servletap  p5-Apache-Test
4 Makefile           calamaris           jakarta-tomcat     p5-Apache-ePerl
5 Mosaic             cgic                jakarta-tomcat4    p5-CGI
6 SpeedyCGI          cgicc              jsdk20             p5-CGI-Applicatio
7 adzap              cgilib             jssi               p5-CGI-FastTempla
8 amaya              checkbot           kannel             p5-CGI-FormBuilde
9 analog            chimera            kdewebdev3         p5-CGI-Kwiki
10 ap-Embperl         clearsilver        kimagemapeditor   p5-CGI-Minimal
11 ap-access-referer cocoon              lhs                p5-CGI-Session
12 ap-aolserver       communicator        libghttp           p5-CGI_Lite
13 ap-auth-cookie     cronolog           libgtkhtml         p5-ExtUtils-XSBui
14 ap-auth-ldap       curl               libwww             p5-FCGI
15 ap-auth-mysql      cvsweb             liferea            p5-HTML-Clean
16 ap-auth-pam        dillo              links              p5-HTML-FillInFor
17 ap-auth-pgsql      drivel            links-gui           p5-HTML-FixEntiti
18 ap-auth-postgresq elinks             lynx               p5-HTML-Format
19 ap-auth-script     elinks04           mMosaic            p5-HTML-Mason
20 ap-bandwidth       emacs-w3m          make_album         p5-HTML-Parser
21 ...
```



## www/ap-perl

```
1  $ cd www/ap-perl/
2  $ ls
3  CVS          MESSAGE      PLIST        distinfo
4  DESCР        Makefile     buildlink3.mk patches
5
6  $ cat DESCР
7  The Apache/Perl integration project brings together the full power of the
8  Perl programming language and the Apache HTTP server. [...]
9
10 $ cat PLIST
11 @comment $NetBSD: PLIST,v 1.1 2001/11/01 02:15:49 zuntum Exp $
12 lib/httpd/mod_perl.so
13 share/httpd/htdocs/manual/mod/mod_perl.html
14
15 $ cat distinfo
16 $NetBSD: distinfo,v 1.10 2005/06/24 06:58:26 jlam Exp $
17
18 SHA1 (mod_perl-1.29.tar.gz) = 6cdc30b5bc6db26997a18da02570bd2822cdf956
19 RMD160 (mod_perl-1.29.tar.gz) = 4a6eb2223f0bde9df1608bac2dbba543475fed42b
20 Size (mod_perl-1.29.tar.gz) = 378877 bytes
21 SHA1 (patch-aa) = dfdc20ee2ce2ffdc5a5e87b02f4946e92b43bc6d
22 SHA1 (patch-ab) = 18b02747ae4b9ab17ce252cf003bf9fd7e4b2560
```

# Make-Ziele

- 1 `make fetch`: Quellen herunterladen
- 2 `make checksum`: Integritätsprüfung mit Prüfsummen
- 3 `make install-depends`: Installiert Abhängigkeiten
- 4 `make extract`: entpacken
- 5 `make patch`: Patches einspielen
- 6 `make configure`: Konfigurieren
- 7 `make build`: Kompilieren
- 8 `make install`: Installieren und registrieren



## Andere Ziele

- `make package`: Binärpaket für `pkg_add(8)` erzeugen
- `make clean`: Kompilationsverzeichnis säubern
- `make clean-depends`: Kompilationsverzeichnisse der Abhängigkeiten säubern
- `make deinstall`: Deinstallieren
- `make replace`: Neue Version drüberinstallieren (Wahrschau!)
- `make update`: Paket inkl. Abhängigkeiten aktualisieren
- `make fetch-list`: Shellskript zum Quellen-Herunterladen erzeugen
- `make show-options`: Paketoptionen zeigen, sofern vorhanden



## www/ap-perl installieren

mod-perl installieren:

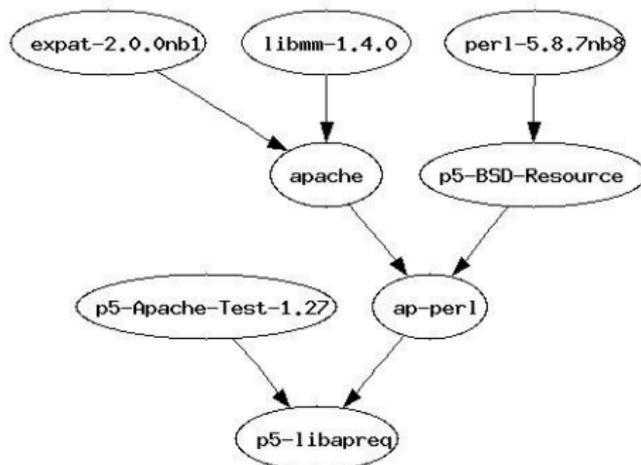
- 
- 1 `cd /usr/pkgsrc/www/ap-perl/`
  - 2 `make install clean clean-depends`
- 



# www/ap-perl installieren

mod-perl installieren:

- 1 `cd /usr/pkgsrc/www/ap-perl/`
- 2 `make install clean clean-depends`



installed packages subgraph (ap-perl-1.29nb5), generated by pkgdepgraph v2.8, on Tue Nov 28 08



# Helferlein

- `pkg_admin` Administriert Paketdatenbank
- `pkg_add`, `pkg_delete` Binärpakete installieren/löschen
- `pkg_chk` überprüft und aktualisiert Pakete
- `pkg_tarup` installiertes Paket in Binärpaket umwandeln
- `pkgfind` Paket nach Namen in pkgsrc suchen
- `pkg_view` Pakete in Chroots installieren



# Pakete aktualisieren

---

```
1 $ cd ../pkgsrc/graphics/gimp
2 $ make update
```

---

Problem: Abhängigkeit baut nicht  $\rightsquigarrow$  Gimp ist weg



# Pakete aktualisieren

---

```
1 $ cd ../pkgsrc/graphics/gimp
2 $ make update
```

---

Problem: Abhängigkeit baut nicht  $\rightsquigarrow$  Gimp ist weg

Lösung: Bauen in einem Chroot mit `/usr/pkgsrc/mk/bulk/`



# Pakete aktualisieren

---

```
1 $ cd ../pkgsrc/graphics/gimp
2 $ make update
```

---

Problem: Abhängigkeit baut nicht  $\rightsquigarrow$  Gimp ist weg  
Lösung: Bauen in einem Chroot mit `/usr/pkgsrc/mk/bulk/`

---

```
1 # mksandbox --pkgsrc=/usr/pkgsrc/ --src=/usr/src/ --xsrc=/usr/xsrc/ /usr/sandbox/
2 # chroot -u root /usr/sandbox
3 chroot# cd /usr/pkgsrc/graphics/gimp; make package clean; logout
4 # pkg_delete -r gimp ; pkg_install /usr/pkgsrc/packages/All/gimp.tgz
```

---

# Pakete aktualisieren

---

```
1 $ cd ../pkgsrc/graphics/gimp
2 $ make update
```

---

Problem: Abhängigkeit baut nicht  $\rightsquigarrow$  Gimp ist weg

Lösung: Bauen in einem Chroot mit `/usr/pkgsrc/mk/bulk/`

---

```
1 # mksandbox --pkgsrc=/usr/pkgsrc/ --src=/usr/src/ --xsrc=/usr/xsrc/ /usr/sandbox/
2 # chroot -u root /usr/sandbox
3 chroot# cd /usr/pkgsrc/graphics/gimp; make package clean; logout
4 # pkg_delete -r gimp ; pkg_install /usr/pkgsrc/packages/All/gimp.tgz
```

---



# Konfiguration



## Konfiguration mit /etc/mk.conf

- zentrale Konfiguration in /etc/mk.conf
- setzt lokale und globale Variablen der Makefiles
- pkgsrc/mk/defaults/mk.conf für globale Optionen
- `make show-options` für Paketoptionen



# make show-options

```
1 $ cd pkgsrc/sysutils/bacula
2 $ make show-options
3 Any of the following general options may be selected:
4 Exactly one of the following database options is required:
5     catalog-mysql      Store Bacula catalog in MySQL database.
6     catalog-pgsql      Store Bacula catalog in PostgreSQL database.
7     catalog-sqlite     Store Bacula catalog in SQLite database.
8     catalog-sqlite3
9
10 These options are enabled by default: catalog-sqlite
11 These options are currently enabled: catalog-pgsql
```

## /etc/mk.conf

```
# Bacula mit PostgreSQL statt SQLite
PKG_OPTIONS.bacula = -catalog-sqlite catalog-pgsql
# postfix-current mit PCRE, SASL und TLS
PKG_OPTIONS.postfix-current = pcre sasl tls

# Adobe- und Opera-Lizenzen akzeptieren
ACCEPTABLE_LICENSES+=adobe-acrobat-license
ACCEPTABLE_LICENSES+=opera-850-license

# X-Version
X11_TYPE=XFree86

# neue Shells in /etc/shells registrieren
PKG_REGISTER_SHELLS= YES
```



## /etc/mk.conf

```
# Bacula mit PostgreSQL statt SQLite
PKG_OPTIONS.bacula = -catalog-sqlite catalog-pgsql
# postfix-current mit PCRE, SASL und TLS
PKG_OPTIONS.postfix-current = pcre sasl tls

# Adobe- und Opera-Lizenzen akzeptieren
ACCEPTABLE_LICENSES+=adobe-acrobat-license
ACCEPTABLE_LICENSES+=opera-850-license

# X-Version
X11_TYPE=XFree86

# neue Shells in /etc/shells registrieren
PKG_REGISTER_SHELLS= YES
```



# /etc/mk.conf

```
# Bacula mit PostgreSQL statt SQLite
PKG_OPTIONS.bacula = -catalog-sqlite catalog-pgsql
# postfix-current mit PCRE, SASL und TLS
PKG_OPTIONS.postfix-current = pcre sasl tls

# Adobe- und Opera-Lizenzen akzeptieren
ACCEPTABLE_LICENSES+=adobe-acrobat-license
ACCEPTABLE_LICENSES+=opera-850-license

# X-Version
X11_TYPE=XFree86

# neue Shells in /etc/shells registrieren
PKG_REGISTER_SHELLS= YES
```



# /etc/mk.conf

```
# Bacula mit PostgreSQL statt SQLite
PKG_OPTIONS.bacula = -catalog-sqlite catalog-pgsql
# postfix-current mit PCRE, SASL und TLS
PKG_OPTIONS.postfix-current = pcre sasl tls

# Adobe- und Opera-Lizenzen akzeptieren
ACCEPTABLE_LICENSES+=adobe-acrobat-license
ACCEPTABLE_LICENSES+=opera-850-license

# X-Version
X11_TYPE=XFree86

# neue Shells in /etc/shells registrieren
PKG_REGISTER_SHELLS= YES
```



## /etc/mk.conf

```
# Lokale Wurzel für Paketinstallationen
LOCALBASE?= /usr/pkg
# Lokale Wurzel für Crosskompilierte Pakete
CROSSBASE?= ${LOCALBASE}/cross

# Verzeichnis für Paket-Quellen
DISTDIR?=/mnt/nfs2/pkgsrc/distfiles

# überprüft Quellen mit MD5 und lädt bei Fehler erneut
herunter
FAILOVER_FETCH=yes

# Sparc-Zielarchitektur
SPARC_TARGET_ARCH= sparcv7
```



# /etc/mk.conf

```
# Lokale Wurzel für Paketinstallationen
LOCALBASE?= /usr/pkg
# Lokale Wurzel für Crosskompilierte Pakete
CROSSBASE?= ${LOCALBASE}/cross

# Verzeichnis für Paket-Quellen
DISTDIR?=/mnt/nfs2/pkgsrc/distfiles

# überprüft Quellen mit MD5 und lädt bei Fehler erneut
herunter
FAILOVER_FETCH=yes

# Sparc-Zielarchitektur
SPARC_TARGET_ARCH= sparcv7
```



# /etc/mk.conf

```
# Lokale Wurzel für Paketinstallationen
LOCALBASE?= /usr/pkg

# Lokale Wurzel für Crosskompilierte Pakete
CROSSBASE?= ${LOCALBASE}/cross

# Verzeichnis für Paket-Quellen
DISTDIR?=/mnt/nfs2/pkgsrc/distfiles

# überprüft Quellen mit MD5 und lädt bei Fehler erneut
# herunter
FAILOVER_FETCH=yes

# Sparc-Zielarchitektur
SPARC_TARGET_ARCH= sparcv7
```



# /etc/mk.conf

```
# Lokale Wurzel für Paketinstallationen
LOCALBASE?= /usr/pkg
# Lokale Wurzel für Crosskompilierte Pakete
CROSSBASE?= ${LOCALBASE}/cross

# Verzeichnis für Paket-Quellen
DISTDIR?=/mnt/nfs2/pkgsrc/distfiles

# überprüft Quellen mit MD5 und lädt bei Fehler erneut
herunter
FAILOVER_FETCH=yes

# Sparc-Zielarchitektur
SPARC_TARGET_ARCH= sparcv7
```



# Sicherheit

- Prüfsummen und OpenPGP-Signaturen für Quellcode und Patches
- `audit-packages`: vergleicht installierte Pakete mit Verwundbarkeitsliste
- `$ pkg_admin check` vergleicht installierte Dateien mit Paketliste
- `pkg_chk -u -n` zeigt »veraltete« Pakete an



# Sichten

- Problem: verschiedene Paketversionen auf einem Rechner  $\rightsquigarrow$  Versionskonflikte
- Lösung:
  - manuelles chroot
  - pkg\_view: automatisiertes chroot



# Sichten

- Problem: verschiedene Paketversionen auf einem Rechner  $\rightsquigarrow$  Versionskonflikte
- Lösung:
  - manuelles chroot
  - pkg\_view: automatisiertes chroot



# pkg\_select

- curses-basierte TUI
- Frontend für Pkgsrc
- Demo



## Literatur und Quellen:

- The pkgsrc guide:  
[www.netbsd.org/Documentation/pkgsrc/pkgsrc.pdf](http://www.netbsd.org/Documentation/pkgsrc/pkgsrc.pdf)
- Hubert Feyrer: <http://www.feyrer.de/NetBSD/clt6-pkgsrc.pdf>
- Jan Schaumann:
  - Pkgsrc  
<http://www.netbsd.org/gallery/presentations/jschauma/pkgsrc.pdf>
  - NetBSD/Desktop: Scalable Workstation Solutions:  
[http://2004.eurobsdcon.org/uploads/media/EBSD04\\_slides\\_19.pdf](http://2004.eurobsdcon.org/uploads/media/EBSD04_slides_19.pdf)  
[http://2004.eurobsdcon.org/uploads/media/EBSD04\\_19.pdf](http://2004.eurobsdcon.org/uploads/media/EBSD04_19.pdf)
- Rainer Orth: Portable Software-Installation mit pkgsrc  
[www.techfak.uni-bielefeld.de/~ro/talks/pkgsrc.pdf](http://www.techfak.uni-bielefeld.de/~ro/talks/pkgsrc.pdf)



## Noch Fragen?

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

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

<http://pkgsrc-wip.sourceforge.net/>

<http://net-tex.de>

