

# Gentoo - Der neue Stern am Himmel

Johannes Bauer

3. Erlanger Linuxtage

16. Januar 2005

# Wie kommt es zu dem Namen?

2/31

- ▶ „Gentoo-Pinguin“ (*Pygoscelis papua*)
- ▶ Kleine, robuste Pinguine, 150m Tauchtiefe

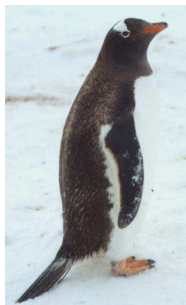


Foto von Bill Little, CA

# Wer hat Gentoo erfunden?

3/31

- ▶ **Daniel Robbins** (Debian, Stampede)
- ▶ „*Enoch*“ erstellt, umbenannt in *Gentoo*
- ▶ Zwischenzeitlich *FreeBSD* (Arbeit am Ports System)
- ▶ Rückportierung von Ports auf *Gentoo*
- ▶ Seit April 2004 kein aktiver Entwickler mehr

# Wozu noch eine Distribution?

4/31

- ▶ Zentraler Aspekt: Flexibilität
- ▶ Einführen einer „Meta-Distribution“
- ▶ Bau der Pakete vom Sourcecode

# Installation - Überblick

5/31

- ▶ Kein Installer, nur LiveCD verfügbar (minimal/universal)
- ▶ Drei Installationsstages:
  - ▶ Stage1: Bootstrappen des Systems (glibc, binutils, gcc)
  - ▶ Stage2: Compilieren der essentiellen Systemtools (ls, tar, gzip)
  - ▶ Stage3: Compilieren von Zusatzprogrammen (z.B. ssh, gnome)
- ▶ Erste Entscheidung: Welchen Stage wähle ich?

# Installation - Verlauf

6/31

- ▶ Herunterladen der entsprechenden Stage-Datei und des aktuellen Portage-Snapshot
- ▶ Booten via LiveCD, Knoppix, eines bereits vorhandenen Systems...
- ▶ Erstellen einer Partition (fdisk, mkfs.ext3/xfs/reiser, mkswap)
- ▶ Mounten der Partition

# Installation - Verlauf

7/31

- ▶ Entpacken des Stage-Archivs und des Portage-Snapshot in den neu erstellten Mountpoint
- ▶ Konfiguration des Netzwerks (/etc/gentoo/resolv.conf)
- ▶ „chroot /mnt/gentoo“
- ▶ ...und jetzt beginnt die eigentliche Installation!

# Installation - Stage1

8/31

- ▶ `cd /usr/portage`
- ▶ `scripts/bootstrap.sh`
- ▶ ...und dann aufgeregtes Warten (ca. 4 Stunden)



# Installation - Stage2

9/31

- ▶ emerge system
- ▶ ...und dann immernoch aufgeregtes Warten (ca. 6 Stunden)

# Installation - Stage3

10/31

- ▶ Jetzt hat man ein fertiges, aber noch ziemlich „blankes“ System
  - ▶ emerge ssh cups
  - ▶ emerge gnome gdm
  - ▶ emerge gimp mozilla-firefox mozilla-thunderbird
  - ▶ emerge openoffice
- ▶ ...und man hat ein schon gut benutzbares System (nach ca. weiteren 14 Stunden)

# Das Portage-System

## Allgemein

11/31

- ▶ Rückportiert von *FreeBSD*
- ▶ „ebuilds“: Skripten, die den Installationsablauf regeln
- ▶ Portage Tree: Baum aller „ebuilds“
- ▶ Synchronisation mittels rsync

# Das Portage-System

## Die USE-Flags

12/31

- ▶ Angabe persönlicher Präferenzen
- ▶ Benutzerspezifische Compilierung der Pakete
- ▶ Einheitlicher Aufbau

# Das Portage-System

## Beispiel mit wenigen Optionen

13/31

```
joe [~]: emerge -p -v apache
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild R ] net-www/apache-2.0.52-r1 +berkdb -debug  
-doc +gdbm -ipv6 +ldap +ssl -static -threads 0 kB
```

Total size of downloads: 0 kB

# Das Portage-System

14/31

## Beispiel mit mehr Optionen

```
[ebuild R ] dev-php/php-5.0.2-r1  -adabas -bcmath +berkdb
  -birdstep -bzlib -calendar -cdb -cpdflib +crypt -ctype
  -curl -curlwrappers -db2 +dba -dbase -dbm -dbmaker -dbx
  -debug -dio -empress -empress-bcs -esoob -exif +fam -
fdftk -filepro -flatfile -frontbase -ftp +gd -gd-external
  +gdbm -gmp -hyperwave-api -iconv -imap -informix -ingres
  -inifile -interbase -iodbc +jpeg +ldap -libedit -mcve
  -memlimit -mhash -mime -ming -mnogetsearch -msession -
mysql -mssql +mysql -mysqli +ncurses -nis +nls -oci8 -odbc
  -oracle7 -ovrimos -pcntl -pcre -pfpro +png -posix -postgre
  -qdbm +readline -recode -sapdb -sasl -session -shared
  -sharedmem -simplexml -snmp -soap -sockets -solid +spell
  -spl -sqlite +ssl -sybase -sybase-ct -sysvipc -tidy +
tiff -tokenizer +truetype -wddx +xml2 -xmlrpc -xpm -xsl
```

# Das Portage-System

## Portage in Aktion

15/31

```
joe [~]: emerge -p -v dev-games/simgear
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild N      ] dev-db/metakit-2.4.9.3-r2  +python +  
      tcltk 873 kB
```

```
[ebuild N      ] media-libs/plib-1.8.3  724 kB
```

```
[ebuild N      ] dev-games/simgear-0.3.6  927 kB
```

Total size of downloads: 2,525 kB

# Das Portage-System

## Versionsverwaltung der Pakete

16/31

- ▶ Unterteilung in „stabile“ und „instabile“ Versionen
- ▶ *Maskierung* instabiler Pakete
- ▶ Unterscheidung: Hard- und Softmaskierung
  - ▶ Softmaskierung: Neuere Paketversionen, „testing“, teilweise instabil
  - ▶ Hartmaskierung: Sicherheitsrelevante Bugs, für die noch kein Fix vorhanden ist



# Das Portage-System

## Softmaskierung

17/31

```
joe [~]: emerge -p latex-unicode
```

!!! All ebuilds that could satisfy "latex-unicode" have been masked.

!!! One of the following masked packages is required to complete your request:

- dev-tex/latex-unicode-20030605 (masked by: ~x86 keyword)
- dev-tex/latex-unicode-20040307 (masked by: ~x86 keyword)
- dev-tex/latex-unicode-20041017 (masked by: ~x86 keyword)

# Das Portage-System

## Hartmaskierung

18/31

```
joe [~]: emerge -p net-ftp/junkie
```

```
!!! All ebuilds that could satisfy "net-ftp/junkie" have  
    been masked.
```

```
!!! One of the following masked packages is required to  
    complete your request:
```

```
- net-ftp/junkie -0.3.1 (masked by: package.mask)
```

```
# <chriswhite@gentoo.org> (30 Dec 2004)
```

```
# Junkie being masked per security bug #74696
```

```
- net-ftp/junkie -0.3.1-r1 (masked by: package.mask)
```

# Das Runlevel-Control-System

19/31

- ▶ Was soll zum Bootzeitpunkt gestartet werden?
- ▶ ...und in welcher Reihenfolge?

# Das Runlevel-Control-System

20/31

## Erstellung des RC-Graphen

- ▶ Informationen über Dienste benötigt:
  - ▶ Was **braucht** der Dienst (need)?
  - ▶ Was **benutzt** der Dienst (use)?
  - ▶ Was **stellt** der Dienst zur Verfügung (provide)?
- ▶ Speicherung dieser Informationen in den Initskripten

# Das Runlevel-Control-System

21/31

## Erstellung des RC-Graphen - Beispiel

Auszug aus /etc/init.d/vixie-cron:

```
depend() {  
    use logger  
    need clock localmount  
    provide cron  
}
```

# Das Runlevel-Control-System

22/31

## Erstellung des RC-Graphen - Benutzung

Hinzufügen von „apache2“ zum „default“ Runlevel:

```
joe [~]: rc-update add apache2 default
* apache2 added to runlevel default
* Caching service dependencies...
* rc-update complete.
joe [~]:
```

# Das Runlevel-Control-System

23/31

## Erstellung eines eigenen Runlevels

- ▶ Anlegen eines neuen Unterverzeichnisses:
  - ▶ `mkdir /etc/runlevels/neuesrl`
- ▶ Hinzufügen der Dienste mittels `rc-update`:
  - ▶ `rc-update add net.eth0 neuesrl`
  - ▶ `rc-update add apache2 neuesrl`
  - ▶ `rc-update add cups neuesrl`

# Das Runlevel-Control-System

24/31

Erstellung eines eigenen Runlevels - Booten

- ▶ Automatisch in das neu erstellte Runlevel booten
- ▶ Eintrag z.B. im Grub abändern:  
title Kernel 2.6.10  
root (hd0,0)  
kernel /boot/bzImage-2.6.10 root=/dev/hda1 **softlevel=neuesrl**



# Wichtige Konfigurationsdateien

25/31

/etc/make.conf

- ▶ Selektion systemweiter USE-Flags
- ▶ Wahl des bevorzugten Paket-Mirrors
- ▶ Optimierungseinstellungen für den Compiler

```
#CFLAGS="-O2 -march=athlon-xp -fomit-frame-pointer"  
CFLAGS="-O2 -march=athlon-xp"  
CHOST="i686-pc-linux-gnu"  
USE="gtk2 gnome cdr dvd acpi fbcon avi crypt gpm X-qt -  
kde -mozilla usb"  
GENTOO_MIRRORS="http://ftp.uni-erlangen.de/mirrors/  
gentoo/"
```

# Wichtige Konfigurationsdateien

26/31

Softmaskierung: /etc/portage/package.keywords

- ▶ Wahl der Packetversionsgruppe mit Quantoren möglich
- ▶ Zweites Argument schaltet jeweiligen Tree frei, u.a.:
  - ▶ x86, ia64, amd64
  - ▶ ppc, ppc64, ppc-macos
  - ▶ alpha, arm, hppa, mips, s390, sh, sparc

=net-news/leafnode -2.0.0\_alpha20041227 ~x86

=dev-tex/latex-unicode-20041017 ~x86

# Wichtige Konfigurationsdateien

27/31

Hartmaskierung: `/etc/portage/package.unmask`

- ▶ Hartmaskierte Pakete haben oft sicherheitsrelevante Bugs  
→ sich der Gefahr bewusst sein
- ▶ Quantoren sind auch hier möglich, aber es ist strikt davon abzuraten, sie zu benutzen

```
=net-ftp/junkie -0.3.1-r1
```

# Wichtige Konfigurationsdateien

28/31

Eigenes Einspielen von Paketen: `/etc/portage/profile/package.provided`

- ▶ Teilt dem Portage-System mit, dass Paket von Hand installiert wurde
- ▶ Eventuelle Paketabhängigkeiten werden als aufgelöst betrachtet
- ▶ Quantorenlos, impliziertes „=“

`sys-kernel / development-sources - 2.6.9`

# Wichtige Konfigurationsdateien

29/31

Packetspezifische USE-Flags: `/etc/portage/package.use`

- ▶ Paketname und spezielle Änderungen an den USE-Flags
- ▶ Wird auch bei einem systemweiten Update wieder berücksichtigt

```
media-video/mplayer live dvdread -ogg  
media-audio/xmms -ipv6 oss -alsa
```

# Zusammenfassung

30/31

- ▶ Sehr konfigurierbares, schnelles System
- ▶ Einfache, logisch aufgebaute Paketverwaltung
- ▶ Sehr gute, schrittweise, mehrsprachige Dokumentation
- ▶ Lauffähig auf vielen verschiedenen Architekturen
- ▶ Gefühl eines **eigenen** Systems, keine Retorten-Distribution

# Abschluß

31/31

- ▶ Danke für die Aufmerksamkeit!