Gentoo Linux

Vortrag rund um Gentoo für die Linux User Group Augsburg.
Referent Christian Wazulek

Inhaltsverzeichnis

1 Geschichte **2**

1.1 Über Gentoo Linux 2

1.2 Besonderheiten von Gentoo? 2

1.3 Infos zu Gentoo 2

2 Installation **2**

2.1 Allgemeines zu den verschiedenen Installationsarten 2

2.1.1 Kleine Skizze des Installationsprozesses 3

2.2 Portagesystem 3

2.2.1 Überblick 3

2.2.2 Umgebungsabhängige Auflösung von Abhängigkeiten und Funktions Unterstützung 4

2.2.3 Profile 5

2.2.4 Aktualisierung des Portagetrees 5

2.3 Programme installieren mit emerge 6

2.3.1 "Unmergen" (Deinstallieren) von Paketen 8

2.3.2 System Update 9

2.3.3 World Update 10

2.3.4 System aufräumen 12

2.3.5 Pakete säubern 13

2.3.6 Den Portage-Tree durchsuchen 13

2.3.7 Hilfe erhalten 14

2.3.8 Nützliche Werkzeuge 14

2.3.9 Grafisches Frontend zu emerge 14

3 Runlevel **14**

3.0.10 Virtuelle Runlevel 15

3.0.11 Einrichten der Runlevels 15

4 Quellen **16**

Gentoo Linux

1 Geschichte

1.1 Über Gentoo Linux

Bei Gentoo handelt es sich um eine kleine flinke Pinguinrasse. Ausgesprochen "schen-too".

1.2 Besonderheiten von Gentoo?

Gentoo ist eine schnelle, moderne Distribution mit einem sauberen und flexiblen Design. Anders als die meisten Linux Distributionen, hat Gentoo ein Packetsystem, das an die "Ports" von BSD erinnert. Damit wird gewährleistet, daß immer die neuerste Version eines Programms installiert ist.

1.3 Infos zu Gentoo

Informationen zu Gentoo gib es unter folgenden Internetadressen: [1] [2] ausserdem gibt es ein hervorragendes Forum[3]

2 Installation

2.1 Allgemeines zu den verschiedenen Installationsarten

Die neue Boot CD startet von fast jedem modernen IDE CD-ROM Laufwerk, sowie von vielen SCSI CD-ROMs. Die CD-ROM unterstützt IDE Controller (eingebaut im Kernel) sowie alle SCSI Geräte (vorhanden als Module). Zusätzlich stellt sie Module für buchstäblich jede Netzwerkkarte zur Verfügung die Linux unterstützt, sowie Tools die es ermöglichen das Netzwerk zu konfigurieren und ausgehende ssh Verbindungen zum Dateidownload aufzubauen.

Um von der "Build" CD zu installieren, wird ein 486+ Prozessor benötigt und idealerweise mindestens 64 Megabyte RAM. (Gentoo Linux konnte erfolgreich mit 64 MB RAM + 64 MB Swap installiert werden, aber die Installation unter diesen Bedingungen war extrem langsam.) Um die Installation zu starten, kann man sich eines der CD ISO Images von [4] herunterladen. Im Augenblick gibt es zwei CDs. Auf der ersten CD ist gentoo-ix86-1.2.iso. Dieses Image ist recht klein (16MB) und enthält ein minimales "chrootfähiges" Image (stage1-ix86-1.2.tbz2) – alles was man benötigt um ein Gentoo Linux System von Grund auf aufzubauen. (Die Quelldateien werden dabei automatisch aus dem Internet geladen.)

[1] http://www.gentoo.org
[2] http://www.gentoo.de
[3] http://forums.gentoo.org
[4] http://www.ibiblio.org/gentoo/releases/build/

Die zweite CD ist gentoo-i686-1.2.iso. Dieses Image ist etwas größer (100+MB) – Es enthält das gleiche Material wie auf dem 16MB Image, plus einem vorkompiliertem i686 Basissystem (stage3-i686-1.2.tbz2) sowie ein teilweise kompiliertes i686 Basissystem (stage2-i686-1.2.tbz2) in TAR-Archiven. Die beiden neuen TAR-Archive benötigen einen Pentium Pro oder besser (auf einem K6-System und niedriger nicht einsetzbar). Bei Einsatz eines i686+ System indem nicht alles von Grund auf neu kompiliert werden soll, kann man damit die Installationsdauer von Gentoo Linux deutlich verkürzen.

2.1.1 Kleine Skizze des Installationsprozesses

stage	Installationsanforderungen
1	Einrichten von Partition und Dateisystem, emerge rsync, bootstrap, emerge system, emerge linux sources, abschliessende Konfiguration
2	Einrichten von Partition und Dateisystem, emerge rsync, emerge system, emerge linux sources, abschliessende Konfiguration
3	Einrichten von Partition und Dateisystem, emerge rsync (optional), abschliessende Konfiguration

2.2 Portagesystem

2.2.1 Überblick

Portage ist ein sehr mächtiges und fortgeschrittenes Paket-Management-System. Seine Flexibilität und Fähigkeit als einfaches Werkzeug zum Kompilieren von Software oder als Herzstück einer brandaktuellen Linux Distribution zu dienen ist nahezu einzigartig. Die Gentoo Linux Distribution wurde um Portage entwickelt.

Gentoo Linux wird oftmals als "Meta-Distribution" bezeichnet. Gentoo besteht aus Portage und über 2200 Anleitungen zum Kompilieren von Paketen, sogenannten ebuilds . Diese ebuilds geben Portage die Anweisungen wie ein bestimmtes Softwarepaket kompiliert und installiert werden soll. Durch die Benutzung von Profilen und dem Kommandozeilen Programm emerge können Benutzer und Entwickler Portage dazu nutzen um die Pakete zu installieren und zu pflegen, die die Grundlage des Betriebssystems und der darauf laufenden Anwendungen darstellen.

Ein Gentoo Linux wird "on-the-fly" kompiliert, d.h. direkt auf den entsprechenden Rechner angepasst und erstellt. Der Installationsprozess umfasst das Erstellen eines funktionierenden Compilers sowie einer minimalen Umgebung, in der Portage Quelltexte aus dem Internet laden kann, um den Rest des "Systemkerns" und etwaige Anwendungen zu installieren. Auch wenn Portage die Benutzung von vorkompilierten Anwendungen unterstützt, stellt diese nur einen Kompromiss dar und wird nur zur Installation auf langsamen Maschinen bzw. von Entwicklern die schnell ein bestimmtes Paket wiederherstellen wollen verwendet. Desweiteren gibt es einem die Møglichkeit, Pakete auf einer schnellen Maschine zu kompilieren um sie dann auf einem langsamen Rechner zu installieren.

Durch das Kompilieren "on-the-fly" und der Tatsache, dass Portage stark konfigurierbar ist sind nur sehr wenige Gentoo Installationen gleich. Im Grunde wird bei der Installation von Gentoo Linux eine angepasste Linux Distribution erstellt, die sich an den Optionen, wie sie in der Portage Konfiguration und den ebuilds selber definiert sind, orientiert.

Auf den ersten Blick mag die Idee hinter Portage ähnlich dem BSD Ports System sein. Beide kompilieren Pakete aus dem Source und erlauben es dem Benutzer Software sicher zu installieren bzw. zu deinstallieren. Beide lösen Abhängigkeiten automatisch auf. Viele Ideen von Portage wurden beim BSD Ports System ausgeliehen, jedoch handelt es sich bei Portage definitiv nicht um eine weitere "Ports-Kopie".

Das Portage System ist eine Verbindung aus einem auf Python basierenden Kern und Bash Script basierten ebuilds . Anstatt mit Makefiles und dem make Kommando zu hantieren, verbindet Portage die Möglichkeiten von Python und das einfach zu handhabende Shellscripting mit einigen objektbasierten Charakteren, um ein einzigartig mächtiges System zu erstellen. Dieses setzt Portage an die Spitze aller aktuellen Ports Systeme.

Einige der von Portage angebotenen erweiterten Funktionen sind die Möglichkeit verschiedene Versionen und Überarbeitungen des gleichen Pakets im Tree zu halten, Auflösung der entsprechenden Abhängigkeiten, feinstrukturiertes Paketmanagement, sichere Installation via Sandbox, Schutz von bestehenden Konfigurationsdateien, Profile und vieles mehr.

2.2.2 Umgebungsabhängige Auflösung von Abhängigkeiten und Funktions Unterstützung

Das Portage System ist im Hinblick der Flexibilität welches es dem User bietet einzigartig. Traditionelle BSD Ports Systeme tendieren dazu nur jeweils eine Version eines jeden Paketes im Tree zu unterstützen. Portage hat diese Begrenzungen nicht. Es können mehrere Versionen eines Paketes zur Installation zur Verfügung stehen.Paketabhängigkeiten (Pakete, die zum Kompilieren andere Pakete nötig sind) können entweder mit ihrem Namen oder mittels Namen mit zusätzlich angehängter Versionsnummer definiert werden. Das ermöglicht es mehrere Versionen eines Paketes zeitgleich im Tree zu pflegen.

Das Abhängigkeits-System unterstützt ebenfalls umgebungsabhängige Abhängigkeiten. Portage hat dafür ein leistungsstarkes Konzept, die sogenannten USE Settings . Durch das ändern einer Konfigurations-Variablen in einer Portage Konfiguration, ist es möglich bestimmte Unterstützung für Funktionen oder Bibliotheken während des Kompilierens zu aktivieren bzw. abzuschalten. Dies ist ein sehr flexibles und leistungsstarkes System, welchem wir uns im nächsten Kapitel widmen werden.

Zusätzlich unterstützt Portage Konzept der SLOTs . Während der Entwicklung von Gentoo Linux wurde klar, daß wir öfters mehrere Versionen bestimmter Pakete (wie z.B. Bibliotheken) benötigen, um die Ansprüche andere Pakete zu erfüllen. Der traditionelle Weg um dieses Problem zu lösen war es, verschiedene Versionen eines Paketes als unterschiedliche Pakete mit leicht abweichenden

Namen zu behandeln.

Anstatt verschiedene Versionen als eigene Pakete zu behandeln, brachten die Entwickler Portage bei, wie mehrere Versionen eines Paketes gleichzeitig durch Benutzung von SLOTs zu handhaben sind. Als Beispiel sei hier die gängige Freetype Bibliothek genannt. Die 1.x Reihe von Freetype ist mit der 2.x Reihe inkompatibel, jedoch sind oftmals beide Versionen nötig, um die Abhängigkeiten verschiedener Pakete zu erfüllen. Die meisten Distributionen und Ports Systeme tendieren dazu, ein "freetype" Paket für Freetype 1.x und ein "freetype2" Paket für Freetype 2.x zu führen. Dies betrachten wir als beinahe komplett beschädigtes Paketmanagement System. Wir ergänzen einfach die SLOT Nummer 1 zur ersten und Nummer 2 zur zweiten Version. Mit dieser Information ist Portage in der Lage beide Reihen zu pflegen und, falls nötig, beide auf höhere Versionen upzudaten.

2.2.3 Profile

Portage unterstützt ein weiteres Konzept, sogenannte Profile . Ein Profil enthält eine Liste von Paketnamen und Versionen mit Anweisungen und einigen Standardoptionen, welche von Portage benutzt werden sollen. Ein Profil enthält Informationen welche Pakete und welche Versionen jeweils erlaubt bzw. verboten oder als zwingend notwendig behandelt werden sollen. Der Benutzer kann zwischen verschiedenen Profilen wechseln in dem er einfach einen Symlink ändert (/etc/make.profile).

Der gesamte Aufwand, welcher bei der Entwicklung von Gentoo Linux betrieben wurde, resultiert in einer Sammlung von ebuilds und einem Profil. Dieses Profil beschreibt welche Pakete als Kernpakete für den Betrieb des Systems wichtig sind. Das Profil erlaubt darüber hinaus den Entwicklern bestimmte Pakete zu blockieren bzw. freizuschalten. Dies ist auch nützlich um defekte Pakete zwischenzeitlich zu deaktivieren. Die ebuild Dateien definieren einfach nur wie bestimmte Pakete von Portage kompiliert und installiert werden sollen, welche durch das Profil verlangt bzw. erlaubt werden.

2.2.4 Aktualisierung des Portagetrees

Der Portage Tree, welcher in /usr/portage liegt, enthält die Bibliothek der "Bauanleitungen" für verschiedene Pakete (sogenannte ebuilds). Darüber hinaus enthält der Tree auch noch Informationen aus dem Profil und der Datei package.mask, welche wichtig sind, um das System aktuell zu halten. Um immer die aktuellsten Versionen und neuesten Bugfixes zu haben, ist es wichtig den Tree regelmäßig mit dem offiziellen Tree abzugleichen. Sie können den Portage Tree mittels folgendem Kommando updaten: **emerge rsync**

Die Methode, welche von Portage genutzt wird, kann geändert werden. Schauen Sie unter Portage SYNC Einstellungen im Kapitel Portage konfigurieren nach, um weitere Informationen zu erhalten.

2.3 Programme installieren mit emerge

Der Vorgang des Kompilierens und Installierens eines Paketes durch Portage wird als mergen bezeichnet. Portage kompiliert Pakete und installiert diese kurzfristig in ein "Abbild-Verzeichnis", in dem es die zu installierenden Dateien aufzeichnet. Diese Dateien werden dann aus dem "Abbild-Verzeichnis" ins Root (/) Dateisystem integriert (gemerged).

Das emerge Kommando dient als Front-End des Portage Systems. Das Installieren und Entfernen von Paketen wird durch dieses Kommando und seine unzähligen Argumente kontrolliert.

Um die neueste, unmaskierte Version eines Paketes zu installieren, geben Sie einfach den Paketnamen, wie folgt ein

Beispiel 1
emerge galeon

Dieses Kommando wird zunächst alle benötigten Abhängigkeiten (bereits alle USE Einstellungen berücksichtigt) und dann die neueste und unmaskierte Version von Galeon kompilieren und installieren. Galeon könnte auch mit vollem qualifizierten Namen inklusive Kategorie angegeben werden: net-www/galeon

Das emerge Kommando akzeptiert auch die Angabe von direkten Ebuild Dateien. Dies erlaubt dem Nutzer auch ältere Versionen eines bestimmten Paketes oder Ebuilds einer dritten Partei zu installieren. Das Folgende ist ein Beispiel:

Beispiel 2
emerge /usr/portage/net-www/galeon/galeon-1.2.0-r3.ebuild

Zusätzlich zum angeben eines Paketnamens oder einer Ebuild Datei, unterstützt emerge verschiedene weitere Argumente. Eines dieser Argumente ist –pretend , vielleicht eines der nützlichsten. Durch dieses Argument wird das geplante Vorgehen nicht durchgeführt. Stattdessen gibt Portage eine Liste aller zu installierenden Pakete aus. Das Folgende zeigt eine Auflistung der Pakete, die während der Installation der neuesten Version des Kdevelop Paketes installiert würden:

```
            Beispiel 3

root@kodiak blocke # emerge --pretend kdevelop

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

Calculating dependencies ...done!
[ebuild  N   ] kde-base/kdelibs-2.2.2-r4 to /
[ebuild  N   ] dev-util/kdbg-1.2.2 to /
[ebuild   U  ] app-text/psutils-1.17 to /
[ebuild   U  ] app-text/a2ps-4.13b-r3 to /
[ebuild   U  ] app-text/jadetex-2.20 to /
```

```
[ebuild  N   ] app-text/sgmltools-lite-3.0.3-r2 to /
[ebuild  N   ] kde-base/kdoc-2.2.2-r1 to /
[ebuild  N   ] net-www/htdig-3.1.5-r2 to /
[ebuild  N   ] app-text/enscript-1.6.3-r1 to /
[ebuild  N   ] kde-base/kdebase-2.2.2-r2 to /
[ebuild  N   ] app-doc/qt-docs-2.3.1 to /
[ebuild  N   ] dev-util/kdevelop-2.0.2 to /
```

Mit N gekennzeichnete Pakete sind noch nicht auf ihrem Rechner installiert, würden aber durch die angegebene Aktion eingespielt werden. Pakete, die mit einem U gekennzeichnet sind, befinden sich bereits installiert auf Ihrem System (wobei es sich höchstwahrscheinlich um veraltete Pakete handelt) und werden durch diese Aktion aktualisiert.

Folgende Parameter sind ausserdem Verfügbar:

–fetchonly : Läd alle benötigten Dateien herunter, die für das Kompilieren notwendig sind, sowie alle Abhängigkeiten, die dadurch entstehen.

–emptytree : Diese Option lässt Portage vortäuschen, dass keine der Abhängigkeiten oder Pakete auf denen das zu installierende Paket beruht installiert sind. Dies lässt sich sehr gut Option –pretend verbinden, um eine komplette Liste der Abhängigkeiten für jedes einzelne Paket anzeigen zu lassen. Alle Abhängigkeiten mit Ausnahme von Glibc werden dargestellt.

–nodeps : Mit dieser Option versucht Portage nur die angegebene Pakete zu "mergen" und ignoriert sämtliche Abhängigkeiten. Bitte beachten Sie, dass diese Option zu Problemen führen kann, wenn Sie die Pakete, von welche das jeweilige Paket abhängig ist, nicht bereits installiert haben.

–onlydeps : Mit dieser Option ist es möglich, nur die Abhängigkeiten des jeweiligen Paketes zu "mergen", jedoch nicht das ausgewählte Paket selbst.

–noreplace : Wenn Sie Pakete zum "mergen" angeben, die bereits installiert sind, Sie jene aber nicht durch neue ersetzen wollen, hilft Ihnen diese Option weiter.

–usepkg : Anstatt das angegebene Paket zu kompilieren, versucht Portage mit dieser Option ein vorkompiliertes tbz2 Paket von einem angegebenen Platz zu installieren . Jener Platz ist in der PKGDIR Shellumgebungsvariable anzugeben.

–debug : Um eine noch detailliertere Ausgabe zu bekommen, was während der Aktion mit Portage passiert, benutzen Sie diese Option. Normalerweise werden Ausgaben "menschlich lesbarer" dargestellt. So haben Sie zum Beispiel als Entwickler die Möglichkeit, syntaktische Fehler in Bash Script basierten ebuild Dateien zu eruieren.

–autoclean : Zwingt emerge zum totalen Bereinigen von spezifischen Temporärverzeichnissen für Kompiliervorgänge, bevor das Paket kompiliert wird. Portage erledigt dies bei standard Konfiguration von selbst, dadurch ist diese Option nur für Entwickler interessant, die dieses Verhalten abgeschaltet haben.

–verbose : Sagt emerge , dass es im ausführlichen Modus laufen soll. Derzeit verursacht diese Option nur GNU Info Fehler-Meldungen. Diese können jedoch von Benutzer ohne Weiteres ignoriert werden.

2.3.1 "Unmergen" (Deinstallieren) von Paketen

Der Vorgang des "unmergens" ist, das die Dateien, die mit einem installierten Paket verbunden sind, gelöscht werden. Damit ist die Software vom System entfernt und kann nicht mehr benutzt werden, bis Sie jenes Paket wieder "mergen".

Pakete werden mittels des emerge Befehls und dem Parameter unmerge , gefolgt vom Namen des Paketes entfernt. Das folgende Beispiel beseitigt alle installierten Versionen vom ltrace Paket.

```
Beispiel 4
emerge unmerge ltrace
oder
emerge unmerge dev-util/ltrace
```

Portage erlaubt ausserdem das "unmergen" von ganz spezifischen Versionen eines Paketes. Reguläre Ausdrücke werden durch = (exakte Version), < (kleiner als), > (grösser als), <= (kleiner als oder gleich), und >= (grösser als oder gleich) dargestellt. Das folgende Beispiel würde alle Versionen, die gleich und älter des Paketes ltrace in der Version 0.3.15 sind, "unmergen".

```
Beispiel 5
emerge unmerge <=dev-utils/ltrace-0.3.15
```

Wenn Sie reguläre Ausdrücke für Pakete benutzen, so stellen Sie sicher, das Sie jeweils für die Zeichen > und < ein Backslash davorsetzen, so das Ihre Shell in diesem Fall dies nicht falsch interpretiert. Ausserdem ist es von Nöten, die Kategorie des Paketes, wie im Beispiel gezeigt, anzugeben. Für andere reguläre Ausdrucksbeispiele, führen Sie den Befehl emerge –help aus.

Warnung: Das "unmergen" von Paketen kann gefährlich sein. Wenn Sie ein Paket des Grundsystems entfernen, verliert Ihr System an Funktionalität und bei entfernten Bibliotheken droht funktionsuntüchtige Software. Portage warnt Sie nicht, wenn Sie Pakete des Grundsystems oder gar Abhängigkeiten anderer Pakete entfernen.

Wenn das zu entfernende Paket tatsächlich installiert ist, wird das Programm emerge exakt anzeigen, welche Pakete entfernt werden und wartet eine gewisse

Anzahl an Sekunden ab, um den Benutzer die Möglichkeit zu geben, den Vorgang mittels der Tastenkombination Control-C abzubrechen.

Beginnt erstmal der Vorgang des "unmergens", sehen Sie eine lange Liste von Dateinamen, die mit dem Paket verbunden sind. Manche dieser Dateinamen haben ein Merkmal (flag), das an der linken Seite der Datei angezeigt wird. Die Merkmale !mtime , !empty , und cfg verdeutlichen, weshalb einige Dateien nicht entfernt worden sind, als das Paket "unmerged" wurde. Dateien ohne jegliche Merkmale wurden erfolgreich vom System entfernt.

Das Merkmal !mtime sagt aus, dass die Datei geändert worden ist, nachdem das Paket installiert wurde. Das bedeutet, dass jemand nach dem "mergen" des Paketes diese Datei bearbeitet hat oder zu einem späteren Zeitpunkt andere Pakete es überschrieben haben. Dies erlaubt es, dass Pakete aktualisiert werden können, ohne die Gefahr, dass wichtige Dateien entfernt werden.

Das Merkmal !empty weist auf Verzeichnisse hin, welches Portage verbietet, diese zu entfernen, da sie nicht leer sind (Mehrere Pakete teilen sich oft das selbe Verzeichnis, welches das Paket, was "unmerged" wird, entweder selbst angehört oder ebenfalls benutzt). Der Konfigurationsdatei Schutz-Mechanismus tritt dann ein, wenn Sie das cfg Merkmal sehen. Das bedeutet, ein neueres Paket, was installiert werden wörde, übernimmt den Besitz jener Konfigurationsdateien und Portage verweigert die Entfernung dieser Dateien.

Warnung: Dateien werden immer dem letzten installierten Paket zugeordnet. Dies ist abhängig von der Reihenfolge der Installation und unabhängig von der aktuellen Versions- oder Revisionsnummer der Pakete, die installiert sind. Wenn ein Paket eine Datei besitzt, wird diese immer mit deinstalliert, auch wenn eine ältere Version eines Paketes diese Datei installiert hat, solange der Nutzer diese Datei nicht manuell geändert hat.

2.3.2 System Update

Portage unterstützt die Möglichkeit, alle installierten Pakete mit einem einzigen Befehl zu aktualisieren. Das System-Update-Feature ermöglicht es ihnen, die Kernpakete ihres Systems zu Versionen zu aktualisieren, die von den Gentoo-Entwicklern empfohlen werden und zum einwandfreien Betrieb von Gentoo Linux notwendig sind. Ein System-Update aktualisiert keine Pakete, die nicht als essenziell angesehen werden. Nur die Pakete, die im Portage Profil angegeben sind, werden als absolut wichtig für den Betrieb und die Aufrechterhaltung des Systems erachtet

Um ein System-Update zu starten, geben sie den folgenden Befehl ein:

Beispiel 6 emerge –update system

Portage wird nun, abhängig von den Versionen und Paketen, welche sie derzeit installiert haben, die Updates übersetzen und installieren, die vom aktuellen Portage-Profil empfohlen werden. Sie haben die Möglichkeit sich mit der Option –pretend eine Liste mit den Paketen die installiert bzw. aktualisiert werden, ausgeben zu lassen, wenn das oben gezeigte Beispiel ausgeführt würde.

Anmerkung: Wie sie aus der Gentoo-Installationsanleitung erfahren können, ist einer der ersten Schritte der Befehl emerge system , um das Grundsystem zu installieren. Mit emerge –update system werden diese Basispakete auf den aktuellsten Stand gebracht.

2.3.3 World Update

Portage unterstützt außerdem die Möglichkeit, alle nicht-essenziellen Pakete mit einem einzigen Befehl zu aktualisieren. Das Portage-System besitzt dafür einen gewissen Grad an "Intelligenz", die es ermöglicht, ein System mit verschiedene Versionen von Paketen, die in Konflikt zueinander stehen, sicher zu aktualisieren .

Das world-update-Feature von Portage überprüft das Systemprofil, die Liste der blockierten Pakete (package.mask), das world-Profil und die Abhängigkeiten (inkl. Versionkontrolle) von Paketen die im world-Profil stehen. Dadurch findet es heraus, welche Pakete aktualisiert werden müssen. Ein Paket wird nur aktualisiert, wenn es eine neue Version gibt und das Paket im world-Profil aufgeführt wird oder ein anderes Paket, das im world-Profil steht von ihm abhängt. Selbstverständlich darf das Paket oder eine spezielle Version dessen nicht durch das Systemprofil oder die package.mask blockiert sein.

Portage versucht nun alle Pakete, die im world-Profil aufgeführt sind auf die neuste verfügbare und unblockierte Version zu aktualisieren. Desweiteren überprüft Portage auch die Abhängigkeiten von jedem Paket im world-Profil und wird versuchen diese auf die neuste Version zu aktualisieren. Dabei wird eine Versionkontrolle durchgeführt, so dass die Versionshierachie gewahrt bleibt. Außerdem dürfen diese Pakte weder durch das Systemprofil noch durch package.mask blockiert sein. Schließlich werden auch noch die SLOTS überprüft, die in einem vorangegangenen Kapitel besprochen wurden.

Benutzer, die andere Distributionen und ihre Paket-Management-Systeme neben Portage kennen, sind vielleicht etwas darüber verwirrt, dass Portage nicht nur ein blindes Aktualisieren der Pakete vornimmt, einfach nur anhand der Versionsnummern (Wie es bis Gentoo 1.0 gehandhabt wurde).

Viele der Pakete die in Gentoo's Portage-Tree sind, stehen in verschiedenen Versionen zur Verfügung. Eine ältere oder neuere Version eines Paketes kann mit der Software, die auf sie aufbaut inkompatibel sein. Blindes Aktualisieren von Bibliotheken und Programmen, ohne Rücksicht darauf, dass sie von anderen Paketen gebraucht werden, kann schnell zu vielen schwerwiegenden Problemen führen. Um dies zu verhindern lässt Portage beim Aktualisieren Vorsicht walten und bezieht die Abhängigkeiten aller Pakete, basierend auf den Angaben in den einzelnen ebuilds, mit ein.

Das Herz von Portage's-World-Update ist das world-Profil. Anders als das System-Profil, welches nur von den Entwicklern definiert wird und nie vom Benutzer verändert werden sollte, wird das World-Update-Profil indirekt mit der Zeit durch Aktionen des Benutzers erstellen. Das world-Profil funktioniert in etwa wie eine "Favoriten-Liste". Pakete die vom Benutzer manuell mit Hilfe von emerge installiert werden, werden in der Datei world aufgezeichnet. Diese Datei findet sich unter /var/cache/edb/world . Portage macht dies, da sie ihm mitgeteilt haben es zu installieren (per emerge) und es annimmt, dass sie ein Interesse daran haben, das Paket immer auf dem aktuellsten Stand zu halten.

Die Datei world besteht aus einem Paketnamen mit Kategorie pro Zeile und sollte in etwa wie folgt aussehen:

```
Beispiel 7

net-im/gaim
net-www/skipstone
net-www/galeon
app-editors/vim
app-text/ispell
net-mail/evolution
dev-util/ltrace
sys-apps/xfsprogs
=net-www/mozilla-0.9.8-r3
sys-apps/attr
sys-apps/dmapi
sys-kernel/linux-sources
sys-apps/acl
app-office/gnucash
app-cdr/xcdroast
```

Nahezu alle Einträge in diesem Beispiel wurden von Portage automatisch hinzugefügt, als der Benutzer eines der Pakete manuell "ein-merge-te". Diese Pakete werden aktualisiert, wenn eine neuere Version verfügbar ist.

Anmerkung: Um Zeit zu sparen und sicher zu stellen, dass alle Ihre bevorzugten Pakete aktuell gehalten werden, können Sie die Datei world selbst bearbeiten und so Einträge für diese Pakete hinzufügen. Wenn Sie eine älteren Version von Portage aktualisieren, müssen sie das world-Profil erstellen und dem System bekannt machen. Bei aktuellen Installationen von Gentoo und Portage sollte das world-Profil wärend der Installation erzeugt werden.

Ein interessanter Eintrag ist der für das Mozilla-Paket (=net-www/mozilla-0.9.8-r3). Dieser Eintrag wurde von Hand hinzugefügt, um eine exakte Version festzulegen. Paketeinschränkungen, wie sie im Abschnitt "Unmergen (Deinstallieren) von Paketen" in diesem Handbuch besprochen wurden, können dazu verwendet werden Portage zu zwingen nur spezielle Versionen beim Aktualisieren von Paketen zu verwenden. Der obige Eintrag hat z.B. den Effekt, dass Portage auf das Paket mozilla-0.9.8-r3 als einzig verfügbare Version festgelegt ist. Somit wird es dieses Paket im Verlaufe eines World-Update nie aktualisieren.

World-Updates werden durch den folgenden Befehl gestartet:

Beispiel 8 emerge –update world

Portage wird nun versuchen alle Pakete die in der Datei world stehen und (wenn nötig) deren Abhöngigkeiten aktualisieren. Abhöngigkeiten werden auf die neuste verfügbare Version, die vom zu aktualisierenden Paket gebraucht wird, gebracht. Pakete die nicht in der Datei world aufgeführt sind und keine Abhöngigkeiten von denvorherr genannten Paketen sind, werden nicht aktualisiert.

Warnung: Portage wird keine Dateien in Verzeichnissen überschreiben, die durch die "Configuration File Protection" (Schutz von Konfigurationsdateien) geschützt sind. Es ist notwendig, dass Sie selbst die Unterschiede zwischen Ihren bestehenden und den neuen Dateien, die von Portage generiert wurden, ausgleichen. Wenn Sie Ihre Konfigurationsdateien nicht aktualisieren, werden verschiedene Programme nicht mehr funktionieren. Bitte schauen sie für weitere Informationen unter "Schutz der Konfigurationsdateien" im Kapitel "Portage konfigurieren" nach oder benutzen sie den Befehl emerge –help config .

Um eine Liste mit den Paketen zu bekommen, die von einem World-Update betroffen wären, können sie das Argument –pretend verwenden, so wie es bereits in einem vorangegangen Abschnitt in diesem Kapitel besprochen wurde.

Anmerkung: Durch ein World-Update wird gleichzeitig auch ein System-Update durchgeführt. Außerdem können Kernpakete nicht auf Versionen in world festgelegt werden, da sie vom aktuellen Portage-Profil immer überschrieben werden!

Ein praktischer Nebeneffekt der Art wie World-Update arbeitet, ist für Benutzer interessant, die ein komplettes neu-Übersetzen aller installierten Pakete auf einem System wünschen. Da World-Update alle Pakete und deren Abhängigkeiten, die in der Datei world stehen aktualisiert, gibt einem die Option –emptytree die Möglichkeit ein Neuübersetzung sämtlicher Pakete und aller Abhängigkeiten - mit Ausnahme der glibc - zu erzwingen. Das ist z.B. für Leute nützlich, die ihre Compiler-Optionen oder ihre USE-Variablen geändert haben und wollen, dass diese Veränderungen von der gesamten Software die sie benutzen verwendet wird - ohne das sie nun jedes Paket selbst erneut "ein-mergen" müssen. Dazu müssen sie einfach die Datei world mit allen Paketen, die sie verwenden auffüllen und den folgenden Befehl verwenden:

Beispiel 9
emerge –update world –emptytree

Sie können die Option –pretend mit diesem Befehl verwenden, um eine Liste mit den Paketen, welche neu-Übersetzt würden, zu bekommen.

2.3.4 System aufräumen

Portage hat die Fähigkeit verschiedene Versionen eines Paketes parallel zu installieren. Es gibt einige Pakete in Gentoo's Portage-Tree die diese Funktion nutzen (z.B. zur Kompatibilitätssicherung, wenn ältere Programme mit neueren Versionen inkompatibel sind).

Denken Sie daran, dass wenn eine neuere Version eines Paketes installiert wird, in den meisten Fällen ein Großteil des älteren Paketes überschrieben wird und alles was zurückbleibt sind einige Dokumentationsdateien und andere, für das System unwichtige. Mit der Zeit können diese "Dateileichen" sehr viel Festplattenplatz verschwenden.

Um dies zu verhindern bietet Portage einen einfachen Weg an, Rückstände veralteter Dateiversionen vom Benutzersystem zu entfernen. Diese Funktionalität ergibt sich aus der emerge-Option clean und kann folgendermaßen benutzt werden:

Beispiel 10 emerge clean

emerge wird nun eine Liste ausgeben mit Paketrevisionen und -versionen die entfernt werden und die Versionen die erhalten bleiben. Außerdem gibt es dem Benutzer Zeit, die Aktion mit STRG+C abzubrechen. Auf einem normalen System werden nun eine Vielzahl von Aktionen durchgeführt, die lange Listen mit Dateien die entweder gelöscht oder erhalten wurden, ausgibt.

Naheliegenderweise wird Portage die Aufräumaktion auf die Datei world (alle installierten Pakete) anwenden. Sie können den Umfang der Säuberung durch Optionen wie world , system , eine Liste von Paketnamen oder eine Einschränkung auf Paketversionen, wie es im Abschnitt "Unmerge" in diesem Kapitel besprochen wurde, beeinflussen.

Beim Herausfinden, welche Paketversionen entfernt werden sollten, überprüft Portage die verschiedenen Profile, die Beziehungen zu anderen Paketen und den SLOT eines Paketes. Vorausgesetzt dass alle Paketabhängigkeiten für alle Pakete richtig definiert sind, wird emerge clean nur veraltet Pakete vom System entfernen und nicht solche, deren Entfernung die Funktionalität des Systems einschränken würde.

2.3.5 Pakete säubern

Portage bietet außerdem die Funktion ein Paket zu säubern (engl. prune). prune ist eine unsichere Variante von clean . Es entfernt alle Versionen aller Pakete, ausgenommen der zuletzt installierten Version. Es führt nur wenige Überprüfung aus, die clean durchführt und kann grundlegende Abhängigkeiten von Ihrem System entfernen! Wenn Sie diese Option nutzen, können sie sehr schnell ihr System unbrauchbar machen. Somit wird diese Variante nicht empfohlen und sollte nur in wenigen Ausnahmenfällen verwendet werden.

Die Aktion prune akzeptiert die selben Optionen wie die Aktion clean und kann wie folgt angewendet werden:

Beispiel 11 emerge prune

2.3.6 Den Portage-Tree durchsuchen

Portage-Trees, wie jener der das Herzstück von Gentoo Linux bildet können sehr groß sein. Der Befehl emerge bietet eine Suchfunktion an, welche Suchanfragen in Form eines regulären Ausdrucks, dieser muss von Anführungszeichen eingeschlossen sein, akzeptiert. Reguläre Ausdrücke sind sehr komplizierte Biester und Anwendern die sich dafür interessieren, sei ein gutes Buch zum Thema empfohlen.

Die meisten einfachen Suchen könne ohne Wissen, wie ein regulärer Ausdruck zu bilden ist durchgeführt werden. Das folgende ist ein Beispiel für eine einfache Suche nach einem Paket, das "gcc" heißt oder "gcc" im Namen hat:

Beispiel 12 emerge search gcc

Für jeden Treffer gibt der Befehl den Paketnamen, die neuste Version, die neuste installierte Version, seine Homepage und eine Beschreibung über die Software im Paket aus.

2.3.7 Hilfe erhalten

Mehr Informationen über die zahlreichen Optionen und Aktionen die emerge unterstützt, erhalten Sie durch die Eingabe von:

Beispiel 13 emerge –help

2.3.8 Nützliche Werkzeuge

Verschiedenste Werkzeuge wurden von Gentoo-Nutzern erstellt, um das Leben mit emerge zu erleichtern. Diese Werkzeuge sind im Paket app-admin/gentoolkit im Gentoo-Portage-Tree zu finden.

etc-update : Shell-Skripte für vim , die dabei helfen die Dateien in /etc abzugleichen (es ist gefährlich diese falsch zu verwenden!)

qpkg : Werkzeug für Datenbankanfragen

epm : ein weiteres Anfragewerkzeug für die Datenbank mit RPM-ühnlicher Syntax

2.3.9 Grafisches Frontend zu emerge

Es gibt auch grafische Frontend's für emerge z. B. kemerge oder kportagemaster für KDE.

3 Runlevel

Im Gegensatz zu anderen Init-Systemen, bestehen Gentoos Runlevel nicht aus festen Namen oder Nummern, sonder vielmehr aus eigenen Namen, die auf die standard Runlevel von init abgebildet werden. Anmerkung: Standardmäßig gibt es drei Runlevel, namentlich: "boot", "default" und "nonetwork".

Das "boot" Runlevel sollte der standard Typ für die meisten Setups sein, und ist, wie der Name sagt, das erste Runlevel das nach dem booten ausgeführt wird. Als nächstes kommt "default", welches, wie der Name schon andeutet, das standard Runlevel welches nach dem booten gestartet wird. Zuletzt folgt "nonetwork", welches ausschließlich als Beispiel dient. Die Runlevels liegen in /etc/runlevels, in einem Unterverzeichnis das nach dem Namen des Runlevels benannt wurde; dieses Unterverzeichnis enthält symbolische Links zu den Diensten, die in diesem Runlevel geladen werden sollen.

Anmerkung: Der bevorzugte Weg um einen Service hinzuzufügen oder zu löschen wird später in dem Abschnitt "Einrichten der Runlevels" behandelt.

Wie bereits erwähnt, kann der Name frei gewählt werden, solange die Datei /etc/inittab entsprechend dem neuen Namen des Runlevels angepasst wird.

Wichtig: Eine Ausnahme dieser Regel, die jedoch erwähnt werden sollte, stellt das Runlevel "boot" dar.

Warnung: Bitte ändern Sie den Namen des "boot" Runlevels NIEMALS, da es einige Dinge kaputt machen würde.

Die ganze Arbeit wird vom /sbin/rc Script erledigt, mit diesem können Sie auch im laufenden Betrieb zwischen den virtuellen Runleveln wechseln.

3.0.10 Virtuelle Runlevel

Da Runlevel nicht statisch auf die von Init gebunden sind, kann man wesentlich mehr haben als init unterstützt. Das ermöglicht es dem Benutzer nach seinen Bedürfnissen Profile und Virtuelle Runlevel zu erstellen Zum Beispiel könnte ein Laptop Benutzer zwei standard Runlevel haben, "online" und "offline". Das würde erlauben ein aktives Runlevel zu benutzen, wenn die PCMCIA Netzwerkkarte eingesteckt ist, und ein weiteres Runlevel zu haben, wenn sie es nicht ist. Die PCMCIA Scripts könnten dann so konfiguriert werden, dass sie "/sbin/rc online" oder "/sbin/rc offline" aufrufen, um die richtigen Dienste zu starten oder zu stoppen, jeweils abhängig vom Status der PCMCIA Netzwerkkarte.

3.0.11 Einrichten der Runlevels

Um einen Service einem Runlevel zuzuordnen benutzt man den Befehl rc-update.

```
rc-update add xdm default
```

add : hinzufügen

xdm : Grafischer Login

default : Runlevel default

Zum Löschen benutzt man den Befehl wie folgt.

```
rc-update del xdm default
```

4 Quellen

Hauptquelle http://www.gentoo.de und die MAN-Pages