

>> Gentoo Linux Kernel Guide

[Bitte Kapitel auswählen] 

1. Einleitung

So wie bei Allem in Gentoo-Linux ist es auch das Ziel des Gentoo-Kernel-Teams Ihnen, dem Anwender, so viele Wahlmöglichkeiten wie möglich zu geben. Wenn man sich die Ausgabe von *emerge -s sources* ansieht, wird deutlich, wie viele Kernel bei Gentoo zur Auswahl stehen. In diesem Dokument möchten wir einen kurzen Überblick über die Ziele der einzelnen Patchesätze geben die wir entwerfen und außerdem die anderen Kernelquellen erklären, die wir anbieten.

2. Die Auswahlmöglichkeiten, Teil I

2.1 gentoo-sources

Für die meisten Anwender sind die empfehlenswertesten Kernelquellen die *gentoo-sources*. Das *gentoo-sources* Paket enthält speziell auf Leistung getrimmte Kernelpatches die bestimmte Aufgaben wie z.B. Kompilieren, gleichzeitiges Musikhören und im Web surfen optimieren sollen. Viele von Ihnen haben möglicherweise noch nie ein System benutzt, bei dem man gleichzeitig kompilieren und die tägliche Arbeit erledigen kann.

Sie werden vielleicht feststellen, dass die Benutzung der *vanilla-sources* (den offiziellen Kernelquellen von <http://www.kernel.org>) dazu führt, dass normale Aufgaben wie Musikhören, Mausbewegungen und ähnliches "Sprünge" aufweisen können.

Die *gentoo-sources* enthalten ein erweitertes ACPI-Subsystem und basieren auf Con Kolivas High-Performance Kernelpatches (den sog. *ck-sources*). Wir unterstützen außerdem grSecurity (einen Satz von sicherheitsbezogenen Patches mit Unterstützung für ACLs), EVMS(2) (ein hochflexibles Speicherverwaltungs-System mit einfacher Partitionsgrößenveränderung), JFS (ein Hochleistungs-Dateisystem von IBM), die neuesten NTFS-Treiber und vieles, vieles mehr.

Da die *gentoo-sources* auf Höchstleistung ausgelegt sind, stellen sie außerdem eine gute Grundlage für Spiele dar.

Die folgenden USE-Flags können zur Auswahl von optionalen Patches gesetzt werden:

Flag Beschreibung

aavm Andrea Arcangeli's Speicherverwaltung nutzen

evms2EVMS 2.0.1 anstelle von EVMS 1.2.1 nutzen

crypt Kryptographie-Patches benutzen

usagi USAGI beibehalten, superfreeswan, patch-int und loop-jari nicht beibehalten

2.2 vanilla-sources

Die nächsten Kernelquellen mit denen Sie als Linux-Anwender vielleicht vertraut sind, sind die *vanilla-sources*. Wie ich vorher schon kurz erwähnt habe, sind dies die offiziellen Kernelquellen von <http://www.kernel.org>. Diese Quellen werden (entgegen dem weit verbreitetem Glauben) nicht von Linus Torvalds selbst, sondern von Marcelo Tosatti verwaltet. Linus ist der "Chef" der aktiven Kernelentwicklung und da er eben nur eine Person ist, übergibt er die Verwaltung des stabilen Kernel-Zweigs an Personen, denen er dies zutraut. So ist zum Beispiel Alan Cox Verwalter der Kernel-Serie 2.2 und Marcelo Tosatti wurde eben Verwalter der Kernel-Serie 2.4. Dies ist exakt das, worauf alle Patchesätze der Serie 2.4 basieren. Marcelo hat bei der Verwaltung hervorragende Arbeit geleistet und man kann auf diesen Kernel in Sachen Stabilität, Aktualität (betrifft nicht noch in der Entwicklung befindliche Teile) und Hardwareunterstützung wirklich zählen.

Die *vanilla-sources* sind möglicherweise die stabilsten aller verfügbaren Kernelquellen, da sie auch die am meisten getesteten sind und alle anderen Kernelquellen auf ihnen basieren. Wenn Sie keines der von den anderen Kernels mitgebrachten Extras brauchen, sollten Sie sich für die *vanilla-sources* entscheiden.

2.3 gs-sources

Für Anwender, die Arbeitsplatzleistung gegenüber Verlässlichkeit und Hardwareunterstützung bevorzugen, haben wir die *gs-sources* im Angebot. GS steht für Gentoo Stable (wie kreativ, gel?).

Dieser Patch-Satz ist im Hinblick auf Unterstützung von neuester Hardware erweitert und getestet und stellt sicher, dass kritische Server dann verfügbar sind, wenn sie benötigt werden. Dieser Kernel lässt einige der überaggressiven in den *gentoo-sources* verfügbaren Leistungs-Tuning-Patches außen vor, aber die hervorragende Leistung die man von den Vanilla Kernels kennt und liebt sind immer noch enthalten. Dort wo es ohne Stabilitätseinbußen möglich ist, fügen wir serverbezogene Patches dem Kernel hinzu.

Dieser Kernel bietet Unterstützung für das neueste ACPI Subsystem, EVMS, ECC (wird für Hochverfügbarkeitskernel benötigt), verschlüsselte Loopbackgeräte, NTFS, Win4Lin und XFS. Er enthält außerdem Aktualisierungen für IDE, ext3, diverse Netzwerkkarten und noch einige andere Patches.

Mit anderen Worten sind diese Quellen perfekt für Server und Hochverfügbarkeitssysteme.

Die folgenden USE-Flags können zur Auswahl von optionalen Patches gesetzt werden:

FlagsBeschreibung

crypt Kryptographie-Patches benutzen

2.4 gentoo-test-sources

gentoo-test-sources werden nach viel Testarbeit und Qualitätssicherung der Nachfolger der *gentoo-sources* werden. Patches für die *gentoo-sources* werden zu Testzwecken zuerst an den *gentoo-test-sources* angewendet. Wenn Sie also die Leistung der *gentoo-sources* zusammen mit den neuesten Patches haben wollen, dann benutzen sie die *gentoo-test-sources*.

Die folgenden USE-Flags können zur Auswahl von optionalen Patches gesetzt werden:

Flag Beschreibung

aavm Andrea Arcangeli's Speicherverwaltung nutzen

evms2EVMS 2.0.1 anstelle von EVMS 1.2.1 nutzen

crypt Kryptographie-Patches benutzen

usagi USAGI beibehalten, superfreeswan, patch-int und loop-jari nicht beibehalten

2.5 hardened-sources

Die *hardened-sources* enthalten neben Patches für die verschiedenen Unterprojekte von Gentoo Hardened auch einige andere Stabilitäts- und Sicherheitserweiterungen. Besuchen Sie <http://www.gentoo.org/proj/en/hardened/> für weitere Informationen.

Die folgenden USE-Flags können zur Auswahl von optionalen Patches gesetzt werden:

Flags Beschreibung

selinuxErsetze grSecurity mit Unterstützung für SELinux

2.6 xfs-sources

Die *xfs-sources* enthalten Unterstützung für EVMS, ACPI, grSecurity und - was Sie sich vielleicht schon gedacht haben - die neuesten Patches für die Unterstützung von XFS. Falls Sie es wissen möchten: die Gentoo LiveCD benutzt die *xfs-sources* :-).

Mehr Informationen über XFS gibt es bei <http://oss.sgi.com/projects/xfs/>.

Die folgenden USE-Flags können zur Auswahl von optionalen Patches gesetzt werden:

FlagsBeschreibung

crypt Kryptographie-Patches benutzen

2.7 Architektur-spezifische Kernel

alpha-sources, *arm-sources*, *hppa-sources*, *mips-sources*, *ppc-sources* und *sparc-sources* sind - wie der Name vermuten lässt - für die bestmögliche Lauffähigkeit auf den entsprechenden Plattformen angepasst. Sie enthalten ausserdem einige der oben und unten genannten Hardware- und Feature-Patchesätze.

2.8 ppc-sources-benh

Die *ppc-sources-benh* Ebuilds bieten zusätzliche Hardwareunterstützung für die *ppc-sources*-Kernel. Sie sind ein wenig experimenteller als die *ppc-sources*.

2.9 ppc-sources-crypto

Die *ppc-sources-crypto* Ebuilds bieten Crypto-API Unterstützung für den Gentoo Linux PPC-Kernel. Mehr Informationen über CryptoAPI erhalten Sie unter <http://www.kernel.org/about/>.

2.10 ppc-sources-dev

Die *ppc-sources-dev*-Pakete bieten die Entwicklerquellen für die *ppc-sources*. Jeder Patch der in die *ppc-sources* einfließen soll muß seine Funktionsfähigkeit erst in den *ppc-sources-dev* beweisen.

2.11 compaq-sources

Die *compaq-sources* bieten die Kernel Quellen von RedHat für Alpha, die von Compaq verwaltet werden.

3. Die Auswahlmöglichkeiten, Teil II

Nun werde ich versuchen, kurz einige der anderen in *sys-kernel/*-sources* befindlichen Kernelquellen, die Sie bei *emerge -s sources* vorbeihuschen gesehen haben, zu erklären. Lassen Sie sie uns in alphabetischer Reihenfolge durchgehen.

3.2 aa-sources

Zu allererst haben wir die *aa-sources*. Hierbei handelt es sich um Andrea Arcangelis Patch-Satz. Sein Kernelpatch hat einige der am aggressivsten getunten VM (Virtual Memory) Patches die bekannt sind. Als ich das letzte Mal nachgesehen habe, enthielten Sie auch SGIs XFS-Dateisystem und den O(1)-Scheduler von Ingo Molar (welcher der Default-Scheduler in Linux 2.6 werden wird).

Sie bieten außerdem Unterstützung für User-Mode-Linux (siehe auch unseren [UML Guide \(englisch\)](#) für weitere Informationen) und den neuesten TUX-Webserver (ein im Kernel befindlicher Webserver).

Wenn Sie bei anderen Kernels Probleme mit der Speicherverwaltung haben, dann könnten die *aa-sources* die Lösung des Problems darstellen. Wenn Sie die Linux Speicherverwaltung für Ihr System optimieren wollen, dann sollten die *aa-sources* **auf jeden Fall** das sein, was Sie brauchen.

Besuchen Sie <http://www.kernel.org/pub/linux/kernel/people/andrea/kernels/v2.4> um mehr Informationen über alle Patches in diesen Kernelquellen zu erhalten.

3.3 ac-sources

Als Nächstes haben wie die *ac-sources*. Es handelt sich hierbei um Alan Cox Patchsatz für die Kernel Serie 2.4. In diesem Patchsatz finden Sie den O(1) Scheduler, die neuesten Aktualisierungen für das IDE-System von 2.4 und viele andere Patches die auf eine mögliche Integration in die Kernel Serie 2.4 warten.

Dieser Kernel ist dafür bekannt, anständige Unterstützung für einige zusätzliche Hardware-Komponenten zu haben und kann ein Kandidat für Sie sein, wenn Sie einen stabilen aber weniger konservativen Kernel als die *vanilla-sources* möchten.

Besuchen Sie <http://www.kernel.org/pub/linux/kernel/people/alan/linux-2.4/> um einen Überblick über Alan Cox Arbeit zu bekommen.

3.4 ck-sources

Die *ck-sources* sind Con Kolivas Kernelpatch-Satz. Dieser Kernel ist **höchst** optimiert für Desktopleistung mit dem Nachteil von Durchsatz und ein paar Fähigkeiten des Schedulers bei der Priorisierung von Anwendungen. Con Kolivas "benchmark" Kernel, um die bestmögliche Kombination der einzelnen Features für die Benutzung auf Desktops zu finden. Besuchen Sie <http://kernel.kolivas.org>, um mehr Informationen über Con und seine Patches zu bekommen.

3.5 development-sources

Die *development-sources* sind die Kernel des aktuellen Entwicklerzweiges. Dies ist der Zweig des Linux-Kernels, den Linus Torvalds selbst verwaltet. Man kann sich diesen Kernel als einen Spielplatz vorstellen, wo alle Features für den nächsten stabilen Kernelzweig angepasst, weiterentwickelt und getestet werden bevor sie an die übrigen Linux-Anwender weitergegeben werden.

Wenn Sie die neuesten Treiber, Erweiterungen und das neueste experimentelle Kernsystem möchten, dann ist dieser Kernel das was Sie haben wollen. Bedenken Sie aber bitte, das diese Kernelquellen **äußerst** experimentell sind und es **nicht empfehlenswert** ist, sie auf einem unbedingt benötigten System einzusetzen.

Warnung

Wir weisen Sie hiermit darauf hin, dass Gentoo-Linux keine Fähigkeiten der *development-sources* oder der darauf basierenden sources unterstützt, da sich diese zu oft ändern und evtl. Schäden anrichten können.

3.6 gaming-sources

Die *gaming-sources* basieren auf den *ck-sources* und sind folglich auf Höchstleistung getrimmt. Sie enthalten außerdem die neuesten Patches für Multimedia-Hardware (Grafikkarten, Soundkarten, usw.).

Wenn Sie ein Hardcore-Gamer sind, dann ist dies der Kernel Ihrer Wahl.

3.7 mm-sources

Die *mm-sources* basieren auf den *development-sources* und enthalten Andrew Mortons Patchsatz. Dieser enthält einige andere Patches, wie z.B. erweiterte Attribute für ext2/3 und ACLs (Access Control Lists), Page Table Sharing, den Orlov Allocator, nicht-lineares Mapping-Verhalten, usw.

Wenn Sie sich wirklich auf Messers Schneide bewegen wollen und der Meinung sind die *development-sources* wären nur etwas für Memmen, dann versuchen Sie die *mm-sources*.

3.8 mosix-sources

Die *mosix-sources* sind speziell für MOSIX-Operationen bei Clustern angepasst. Ein Cluster ist ein Satz von Knoten (=PCs) mit spezieller Software, die es Ihnen erlaubt Aufgaben verteilt zu erledigen. Mit Clustern entfällt die Notwendigkeit für Supercomputer, um grössere Aufgaben zu erledigen. Wenn Sie weitere Informationen möchten, dann besuchen Sie bitte <http://www.mosix.org>.

3.9 openmosix-sources

Die *openmosix-sources* sind für die Unterstützung von openMosix Systemen optimiert (dasselbe wie MOSIX, aber Open Source). Für weitere Informationen besuchen Sie bitte <http://www.openmosix.org>.

3.10 redhat-sources

Die *redhat-sources* sind - wie der Name schon vermuten lässt - die Quellen des RedHat Kernels. Dank der Vorteile von Open Source kann jeder an den Früchten der Arbeit, die die RedHat-Ingenieure in ihren Kernel investieren, teilhaben. Wir von Gentoo bieten ein entsprechendes Ebuild an, damit Sie auf möglichst einfachem Wege in den Genuß kommen.

3.11 rsbac-sources

Die *rsbac-sources* enthalten die Patches von <http://www.rsbac.org>. RSBAC steht für **Rule Set Based Access Control**. Diese Kernelpatches erlauben Ihnen, Benutzer anhand von Regeln anstatt Kombinationen von UID-/GID-Berechtigungen zu authentifizieren.

3.12 selinux-sources

Die *selinux-sources* von <http://www.nsa.gov/selinux> sind Patches zur Unterstützung der LSM (Linux Security Modules) und Flask Security Architecture.

3.13 usermode-sources

Die *usermode-sources* sind die Quellen, die Sie für einen Usermode-Kernel brauchen. Dies ist ein spezieller Kernel, der dafür entwickelt wurde, um Linux innerhalb von Linux innerhalb von Linux innerhalb von Linux... auszuführen. User-Mode Linux ist zu Testzwecke und zur Unterstützung von

virtuellen Servern vorgesehen. Für weitere Informationen über diese phantastische Hommage an die Skalierbarkeit von Linux, besuchen Sie bitte <http://user-mode-linux.sourceforge.net>.

Für weitere Informationen über UML und Gentoo, lesen Sie bitte [den Gentoo UML Guide](#).

3.14 win4lin-sources

Die *win4lin-sources* sind zur Unterstützung der Win4Lin Werkzeuge speziell angepasst. Diese erlauben Linux-Anwendern viele Anwendungen, die eigentlich für Microsoft Windows entwickelt wurden mit fast normaler Geschwindigkeit auszuführen. Weitere Informationen gibt es unter <http://www.netraverse.com/>.

3.15 wolk-sources

Die *wolk-sources* enthalten den **Working Overloaded Linux Kernel** von <http://sourceforge.net/projects/wolk>. Dieser Kernel enthält viele verschiedene Patches, die alle mit größter Vorsicht in den Kernel eingepasst wurden. Dies erlaubt Ihnen als Anwender fast jeden davon entweder bei der Kompilierung des Kernels zu berücksichtigen oder dies eben zu lassen - der Kernel selber wird bei fast jeder Kombination von einkompilierten Patches funktionieren.

Wenn Sie eine bestimmte Kombination von Patches brauchen, die Sie nicht in anderen Kernelquellen finden können, dann ist WOLK sicherlich einen Versuch wert.

