

Zacks Kernel-News

Global File System ist wieder frei

Nach Jahren proprietärer Entwicklung ist das Global File System (GFS) wieder unter der GPL verfügbar. Es ist aber ungewiss, ob das 2001 abgespaltene Open-GFS-Projekt mit dem nun wieder freien Code fusionieren wird.

GFS hat bereits eine bewegte Geschichte hinter sich. Ursprünglich wurde das verteilte Dateisystem von der Firma Sistina unter der GPL entwickelt. Zunächst sah es auch so aus, als würde die Aufnahme in den Linux-Kernel kurz bevorstehen. Doch 2001 beschloss Sistina, den Code unter die neu geschaffene Sistina Public License zu stellen. Diese Lizenz gewähr-

te zwar Zugang zum Quellcode, beim Code-Vertrieb fielen aber Gebühren an. Das widersprach den Prinzipien von Open-Source-Software und führte zu heftigen Reaktionen in der Entwicklergemeinschaft. Alan Cox beispielsweise behauptete, mit der Lizenzänderung hätte Sistina gegen die GPL verstoßen, unter der die Firma selbst den Code ursprünglich veröffentlicht hatte.

In der Folge gründeten Entwickler, die an GFS mitgearbeitet hatten, das Open-GFS-Projekt, das auf der letzten GPL-Version des Sistina-Codes aufbaute. Kein ungewöhnlicher Vorgang: Auch

als sich Mosix von der GPL abwandte und beschloss proprietär zu werden, wurde fast unmittelbar darauf das Open-Mosix-Projekt ins Leben gerufen, ebenfalls auf Basis der letzten freien Quellen.

Während Open-GFS die Entwicklung fortführte, versuchte Sistina, die vorherigen Beiträge der Entwicklergemeinschaft zu Geld zu machen. Das führte offensichtlich nicht zum Erfolg: Anfang 2004 kaufte Red Hat Sistina und stellte den GFS-Code wieder unter der GPL zur Verfügung. Die Community heißt das Global File System wieder in der Welt der freien Software willkommen. ■

»ikconfig«

Wer von der Konfigurationsinformation begeistert war, die mit vorkompilierten 2.6-Kerneln ausgeliefert wurde, wird sich über Randy Dunlaps Vorschlag freuen, auch die Kernelversion und das Datum der Übersetzung hinzuzufügen.

Die Aufnahme von Konfigurationsdaten in den Kernel war lange umstritten: Altgediente Entwickler warnten vor dem unnötigen Aufblähen des Kernels, während Anwender und Distributoren den größeren Komfort begrüßten. Schließlich konnte Randy eine Implementierung von »ikconfig« vorschlagen, die Alan Cox überzeugte. ■

User Mode Linux und die Kunst des Patchens

Das Projekt User Mode Linux (UML, [<http://user-mode-linux.sf.net>]), hat in letzter Zeit große Vorstöße in Richtung des offiziellen Linux-Kernels gemacht, doch nun befindet es sich wieder im Abseits. Jeff Dike leistet zwar gute Arbeit, doch das Patch ist mittlerweile auf eine solche Größe angewachsen, dass seine Aufnahme in den Kernel enormen zusätzlichen Aufwand verursachen würde.

Es hat sich eingebürgert, Patches für den Linux-Kernel in kleinen Stücken einzureichen, die jeweils einen einzi-

gen Zweck erfüllen, etwa einen Bug beheben oder ein neues Feature umsetzen. Eingeführt wurde diese Praxis von Linus Torvalds und Leute wie Alan Cox, Andrew Morton und Marcelo Tosatti führen sie fort.

Ein Meister dieser Vorgehensweise ist Alexander Viro, Maintainer des Virtual File System (VFS). Laut Linus ist jedes einzelne von Alexanders Patches durch ein relevantes Detail gerechtfertigt und rundet gleichzeitig das gesamte Design des VFS-Subsystems ab. Leider fallen

Meister der Patch-Kunst nicht vom Himmel und auch die zurzeit verfügbaren Werkzeuge schaffen nicht gerade ideale Voraussetzungen dafür, dies zu ändern.

Jeff Dike zieht daraus den Schluss, dass die Werkzeuge für das Patch-Management gründlich zu überarbeiten sind. Andrew Morton und andere empfehlen Quilt. Dieses Tool scheint für Leute wie Andrew maßgeschneidert zu sein, denn mit ihm lässt sich eine Sammlung einzelner Patches für den Quellcode anderer Entwickler verwalten.

Jeff wird gegen Widerstände kämpfen müssen, um seine Arbeit in den Kernel zu bekommen. Doch Andrew Morton hat schon Bereitschaft signalisiert, Jeffs Arbeit in seine »-mm«-Patches aufzunehmen, die bei Entwicklern breiten Einsatz finden und den offiziellen Kernelquellen sehr nahe stehen. Dieser Schritt würde UML zu mehr Aufmerksamkeit verhelfen, die benötigten Bug-Reports generieren und das Patch aktuell halten, bis Jeff die Umsetzung in viele Einzelpatches vorgenommen hat. ■

Linus Torvalds bezieht Stellung zu Compiler-Built-ins

Die GNU Compiler Collection (GCC) ist und bleibt ein zentrales Tool der Kernelentwicklung. Es wird viel diskutiert, welche GCC-Version sich am besten für eine bestimmte Kernelversion eignet. Unlängst haben einige Entwickler darüber nachgedacht, ob man nicht für den Zweig 2.7 ein Upgrade auf GCC 3.3 und höher voraussetzen sollte, und welche Built-ins, also in den Compiler eingebaute Funktionen, sich für die Verwendung in Kernelcode eignen. Zur Frage der Built-ins hat Linus Torvalds deutlich Stellung bezogen. Er meinte, die Verwendung dieser Funktio-

nen könne zu Problemen führen, wenn ein Anwender eine ältere GCC-Version einsetzt. Auch gebe es Prozessor-Architekturen, auf denen ein Built-in zu schlechterer Performance führt als der Code, den es ersetzt. Die Vorteile hält Linus selbst im besten Fall für minimal. Daher befürwortet er ihre Verwendung lediglich in zwei Fällen: Bei Built-ins, die sich bewährt haben und möglichst für alle Architekturen vorhanden sind, oder wenn die Funktion messbare Vorteile gegenüber einer Alternative bringt. In allen anderen Fällen sei ihr Einsatz nicht gerechtfertigt.

Möglicherweise ist seine reservierte Haltung auch auf frühere Diskussionen zwischen den Kernelentwicklern und dem GCC-Team zurückzuführen, die von scharfer Kritik und Unnachgiebigkeit auf beiden Seiten geprägt waren. Mehr als einmal haben Linus und seine Kollegen sich schon geweigert, ein Upgrade ihrer GCC-Installationen vorzunehmen, weil sie der Meinung waren, der Compiler entwickle sich in die falsche Richtung. Diese Taktik ging nicht immer auf: Die GCC-Entwickler haben sehr eigene Vorstellungen und fühlen sich allen Projekten verpflichtet, die auf

GCC setzen, nicht nur dem Kernel. Außerdem sind sie nicht von Linux abhängig, während der Linux-Kernel beim Kompilieren lange Zeit ganz allein auf GCC angewiesen war. Noch bis vor kurzem war GCC der einzige Compiler, der den Linux-Kernel übersetzen konnte. Doch das hat sich mit Intels C-Compiler (ICC) geändert. Ständig findet neuer Code Aufnahme in den Kernel, der die Portabilität zwischen verschiedenen Compilern verbessern soll. ICC hat diese Tendenz verstärkt, auch wenn GCC bei den Entwicklern immer noch beliebter ist. ■

SMP für Software-Suspend

Das Software-Suspend-Projekt (Swsusp) versucht den Status eines laufenden Systems beim Shutdown so zu speichern, dass er beim Neustart korrekt wiederhergestellt werden kann. Das Projekt bemüht sich seit einiger Zeit um SMP-Unterstützung. Nun glauben Pavel Machek und seine Mitstreiter, sie hätten eine Lösung gefunden. Wegen der großen Schwierigkeiten, die mit Software-Suspend verbunden sind, wurde das Vorhaben fast zum „Heiligen Gral“ der Linux-Entwicklung. Eines der größten Hindernisse besteht darin, dass sich der Hardwarestatus eines Systems weder vor dem Shutdown noch vor dem Wiederherstellen zuverlässig ermitteln lässt. Versuche, dieses Problem zu lösen oder we-

nigstens vernünftig zu handhaben, gaben häufig Anlass zu hitzigen Auseinandersetzungen in der Entwicklergemeinde. Patrick Mochel beispielsweise nahm einen Fork weg von Pavels Code vor. Eine Zeit lang herrschte ein verbissener Wettbewerb zwischen den zwei Versionen. Doch anscheinend haben sich die beiden – teilweise auf Drängen von Linus Torvalds – wieder versöhnt und arbeiten seither zusammen. Dabei hat Pavel wie zuvor die Rolle des Projektleiters. Die Windows-Systeme beherrschen Software-Suspend schon seit einiger Zeit. Das ist ein Stachel im Fleisch vieler Linux-Entwickler und auch eine große Motivation dafür, Swsusp weiter zu verbessern. (mhu) ■

