

# Zacks Kernel-News

## SysFS in ständiger Entwicklung

Der Code für das Sys-Filesystem im Kernel 2.6 durchläuft derzeit eine sehr aktive Entwicklungsphase. Es ist zwar nicht ungewöhnlich, dass Teile des Kernels während der stabilen Serie fertig gestellt werden, aber bei SysFS treffen Patches vieler unterschiedlicher Quellen ein. Jeder möchte sicherstellen, dass sein Treiber mit SysFS Schritt hält, dabei ändert sich SysFS selbst ständig.

Zum Beispiel hat Benjamin Herreschmidt Anfang Februar eine neue Eigenschaft namens »devspec« in SysFS vorgeschlagen. Sie soll den kompletten Open-Firmware-Pfad für Devices am PowerPC und PowerPC64 zur Verfügung stellen, die mit Open-Firmware-Support ausgestattet sind. Tatsächlich war der

ursprüngliche Gedanke, dass die PCI-Einträge selbst einen Open-Firmware-Pfad enthalten; letztendlich führte das zu einer stets wachsenden Komplexität des User-Codes und des SysFS-Verzeichnisses selbst.

Wie Linus Torvalds in einer Diskussion auf der Kernel-Mailingliste verdeutlicht hat, sollten die PCI-Layer keine „magischen“ Kenntnisse spezifischer Plattformen besitzen. Stattdessen sollten die Plattform-Schichten diese Informationen bei Bedarf von der PCI-Schicht beziehen. Entwicklungen wie diese führen zu Widerspruch: Einige Entwickler, allen voran Alexander Viro, regen jetzt schon an, die Entwicklungsarbeiten an SysFS etwas zu bremsen. ■

## Input Driver FAQ

Die Input-Driver-FAQ ist ein ausführliches Dokument von Vojtech Pavlik und beispielsweise unter <http://kerneltrap.org/node/view/2199> zu finden. Es zeigt Lösungen für verschiedene Treiberprobleme, die in Kernel 2.6 auftreten können.

Bei der Entwicklung wurden die Input Layer gründlich überarbeitet. Viele Treiberentwickler haben daher einige Mühe, auf den neuesten

Stand zu kommen, zumal Input Devices eine recht kritische Komponente des Gesamtsystems sind. Mit der FAQ soll es Programmierern gelingen, wieder Anschluss zu finden.

Das Dokument ist aber auch für normale User ganz nützlich, die auf Fehlermeldungen und seltsames Verhalten stoßen und keine eigene Erklärung oder Lösung des Problems finden können. ■

## Was ist ein Maintainer?

Die Diskussionen auf der Kernel-Mailingliste stellen gelegentlich auch scheinbar Selbstverständliches in Frage, so zum Beispiel, was genau unter einem Maintainer zu verstehen sei. Laut Linus Torvalds „besitzt“ dieser keinesfalls den Code in dem Sinn, dass er allein entscheidet, was in seinen Tree hineinkommt und was nicht.

Bei genügend Druck von anderen Entwicklern müsse ein Maintainer auch Code akzeptieren, der ihm persönlich

nicht gefällt. Schließlich ist es bei Open-Source-Software möglich, dass ein anderer die Pflege des Projekts übernimmt, wenn es einen Konsens gibt, dass der Maintainer schlecht arbeitet. Maintainer, die sich für die Wünsche der Entwickler sperren, so Linus, werden diese Position wohl kaum lange bekleiden. Es gab in der Tat einige Fälle, in denen verärgerte Entwickler autoritären Maintainer die Kontrolle über den Code entzogen haben. ■

## FAT-Unterstützung

Der wichtigste Grund, heutzutage noch an der Unterstützung für FAT-Dateisysteme zu arbeiten, ist es, alte Medien weiterhin auslesen zu können. Kaum jemand setzt das Dateisystem ein, weil ihn das Design überzeugt. Frodo Loosjgaard bemüht sich derzeit, den FAT-Support auf das Äußerste zu erweitern.

Über die Jahre hat es viele Implementierungen des FAT-Dateisystems gegeben, die alle auf irgendeine Weise in Konflikt mit der offiziellen Spezifikation standen. Sinnvoll ist es aber, die Microsoft-Implementierung von FAT-FS höher zu bewerten als die Spezifikation selbst, denn diese muss unterstützt werden und nicht die abstrakte

Spezifikation. Frodo hat das eventuell ignoriert.

Sein Code bemüht sich, den Datei-Index für ein Verzeichnis auch dann zu lesen, wenn er auf übliche Art und Weise auf der Festplatte gespeichert ist. In diesem speziellen Fall ist das Verzeichnis-Layout vielleicht in der Spezifikation so festgelegt worden, aber die Spezifikation weicht etwas von der Microsoft-Implementierung ab.

Wie H. Peter Anvin schon gesagt hat, ist der richtige Ansatz, sich an die WWDD-Regeln (das ist die Abkürzung für „What would DOS do?“, also „Was würde DOS tun?“) und damit besser an die Vorgehensweise von Microsoft zu halten. ■

## Wunschliste für verschlüsseltes Dateisystem

Michael A. Halcrow hat sich dazu entschlossen, sein eigenes verschlüsseltes Dateisystem für Linux neu zu konzipieren und möglicherweise zu implementieren – zusätzlich zu den zahlreichen bereits bestehenden und meist gescheiterten Ansätzen. Michaels erster Schritt bestand darin, so viele Menschen wie möglich nach den Features zu fragen, die sie in einem verschlüsselten Dateisystem erwarten. Hier unterscheidet er sich wohltuend von vielen Programmierern. Auf dieser Wunschliste steht die nahtlose Verschlüsselung weit oben. Nicht verschlüsselte Dateien sollten mit minimalem Overhead an die Festplatte übergeben werden und die Verschlüsselungsschicht sollte als Modul realisiert sein, das über das bevorzugte Dateisystem (Ext 3, ReiserFS oder was auch immer) des Users gestülpt wird. Bei der Organisation der verschlüsselten Daten bevorzug-

ten die meisten Befragten die Verknüpfung eines bestimmten Schlüssels mit einer bestimmten Datei statt mit einem Festplattenblock oder einem anderen Mechanismus. Außerdem sollen möglichst Metadaten mit verschlüsselt werden, zum Beispiel auch, ob das Executable-Bit gesetzt ist oder nicht. Zu den weiteren beliebten Features zählt die Möglichkeit, grafische Dateimanager wie Nautilus und Konqueror nahtlos um die Funktionalität der Verschlüsselung zu erweitern. Außerdem wünschen sich die Benutzer auch ein Feature zum Schreddern von Dateien bei der Löschung, um eine Wiederherstellung weitestgehend unmöglich zu machen. Ob Michael diese Wunschliste in Angriff nehmen wird, steht momentan in den Sternen, aber immerhin wissen jetzt auch alle anderen Entwickler, welche Features besonders gefragt sind. ■

## Patch für falsche Bios-Meldungen

Ein Patch von Tony Lindgren führt Plausibilitätskontrollen durch, um die Gültigkeit der Bios-Meldungen für das laufende System zu überprüfen. Er hat aus eigener Erfahrung feststellen müssen, dass einige Bios-Varianten falsche Angaben zu Prozessortakt und Spannung liefern. Der konkrete Anlass war das Bios seines Athlon-64-Notebooks M6805 des Herstellers EMachines, das statt der tatsächlichen CPU-Frequenz

von 1800 MHz nur 1600 lieferte. Fehler wie dieser sind ein immer wiederkehrendes Problem bei der Entwicklung von Betriebssystemen und auch Patches dieser Art gab es schon viele. Die meisten Linux-Entwickler wünschen sich deshalb, dass die Chiphersteller den Quellcode für ihre Bios-Implementierungen einfach veröffentlichen oder wenigstens schnell genug auf Bug-Meldungen reagieren. ■

## Treiber für Infrarot-Fernbedienungen

Gerd Knorr entwickelt ein Modul, das die Interaktion von Infrarot-Fernbedienungen mit dem Kernel über den neuen Input-Layer ermöglicht. Der Treiber selbst stellt nur Helper-Funktionen zur Verfügung und unterstützt selbst keine Hardware, sondern wird von anderen Treibern, etwa dem BTTV-Treiber, dazu benutzt, um Fernbedienungen indirekt zu unterstützen. Ein weiteres Feature des Treibers ist seine Beziehung zu den verschiedenen Kernel-Trees. Die Modulschicht in Kernel 2.6 unterscheidet sich stark von der des Vorgängers. Gerts Modul vermeidet ganz

bewusst einige der neuen Konstrukte, um deren Support sich die meisten anderen Module bemühen. Er möchte, dass der Treiber unverändert unter 2.4- und 2.6-Kernels kompilierbar ist, ohne dafür viele »#ifdef«-Präprozessorbefehle vorauszusetzen. Rusty Russell, der am stärksten an der Überarbeitung des Treiber-Handling-Codes beteiligt war, hat Gerts Anforderungen unterstützt. Er hat Backports für einige Kompatibilitätsfunktionen von 2.6 in 2.4 erstellt, sodass spezifische 2.6-Konstrukte unter 2.4 kompilieren, ohne dass der Treiber Funktionalität einbüßt. (Zack Brown/uwo) ■

