

Filesystem in Userspace

Jens Spiekermann

Aufbau

- Was ist FUSE?
- Grundlagen
- Wie funktioniert FUSE?
- Eigenschaften
 - Vorteile
 - Nachteile
- Wofür kann man FUSE nutzen?
- Wie wird FUSE benutzt?
- Abschluss
- Quellen

Was ist FUSE?

- Kernel-Modul für Unix-Systeme
- Unabhängig vom Betriebssystem
- Verlagerung von Dateisystemtreibern
- Ermöglicht Einbindung verschiedener Dateisysteme

Grundlagen: Dateisysteme

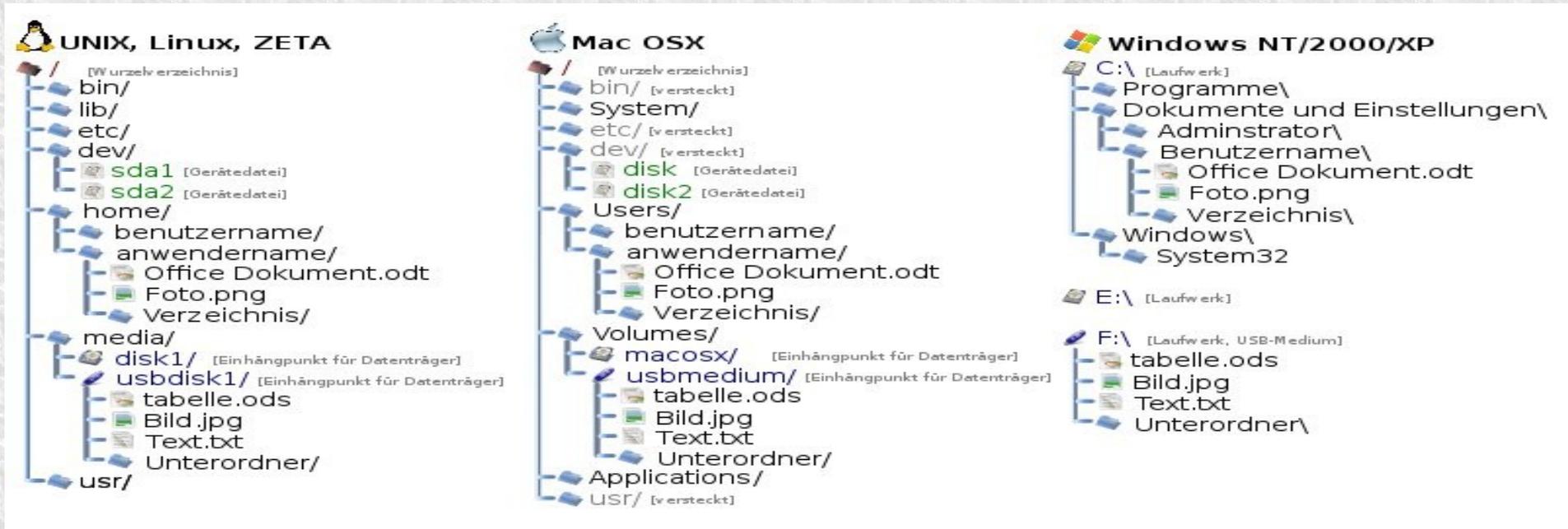


Abb. 1

- Organisiert die Ablage von Dateien auf einem Computer
- Wesentliche Aufgaben: Speichern und Wiederfinden
- Nur ein Verzeichnisbaum in Unix-Systemen

Grundlagen: Betriebssystemkern



Abb. 2

- Legt Prozess- und Datenorganisation fest
- Schnittstelle zwischen Anwendungsprogrammen und der Hardware
- Anwendungsprogramme greifen auf den Kernel zu

Grundlagen: Benutzerrechte

- Unter Windows:
 - Benutzerkontensteuerung als Sicherheitsinfrastruktur
 - Bis Windows XP: Auslegung auf Einbenutzerbetrieb
 - Ab Windows Vista: Aktivitätsbestätigung durch einen Administrator bei bestimmten Aktivitäten
- Unter Unix:
 - Immer mit Benutzernamen angemeldet
 - Systemadministrator setzt Zugriffsrechte fest
 - Änderungen im Heimatverzeichnis betreffen andere Benutzer nicht

Filesystem in Userspace

- Allgemeines
 - FUSE ist frei erhältlich
 - Ursprünglich Teil von AVFS
 - Eigenes Projekt auf sourceforge.net (15.10.2004)
 - Aufnahme in den Linux-Kernel am 13.9.2005
 - In C implementiert

Filesystem in Userspace

- Betriebssysteme, die FUSE unterstützen
 - Linux
 - MacOS
 - OpenSolaris
 - FreeBSD
 - NetBSD

Filesystem in Userspace

- Besteht aus Kernel-Modul und Userspace-Bibliothek
- Kernel-Modul wird in den Betriebssystemkern geladen
- Programmlogik läuft im Usermode ab
- Funktionalität: Aufrufumleitung

Filesystem in Userspace

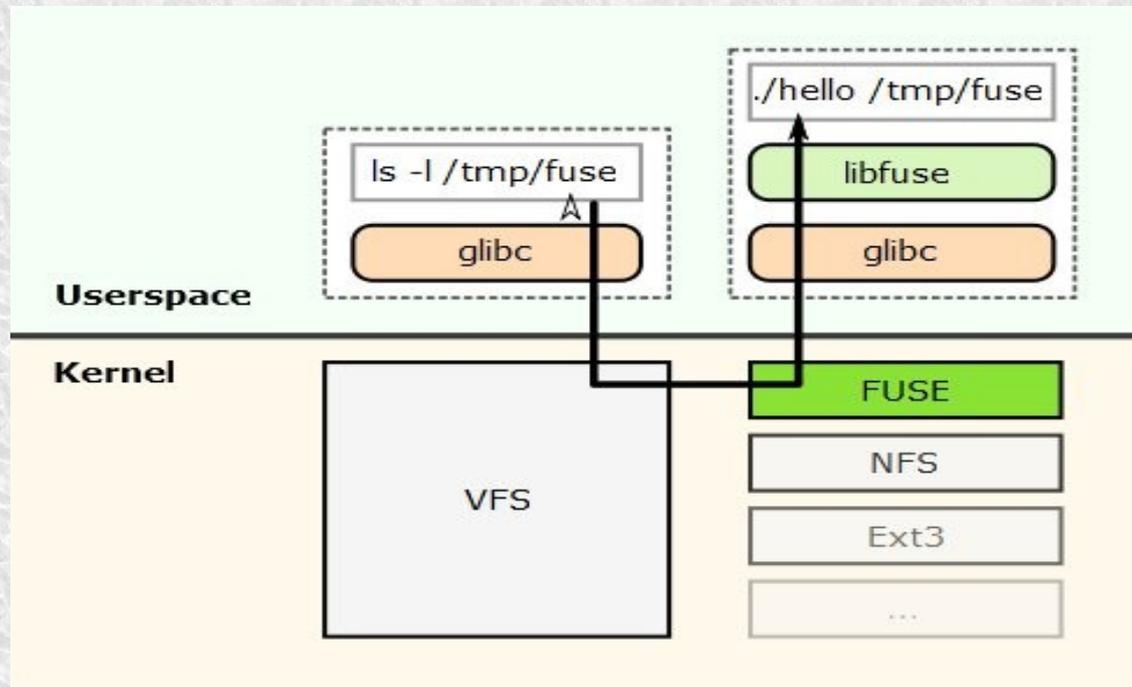


Abb. 3

- Systemaufruf wird im Kernel vom FUSE-Treiber umgeleitet
- Kommunikation zwischen Bibliothek und Kernel-Modul via "file descriptor"

FUSE: Eigenschaften

- Vorteile
 - Einbindung vieler verschiedener Dateisysteme
 - Verwendbar auf allen Unix-Systemen
 - Verwendbar für Benutzer mit eingeschränkten Rechten
 - Open Source
 - Einfache Programmierbarkeit

FUSE: Eigenschaften

- Nachteile
 - Leistungsprobleme
 - Abhängig vom eingebundenen Dateisystem
 - Durch Wechsel zwischen Betriebssystemkern und Usermodus

Einbindbare Dateisysteme

- Unterteilbar in verschiedene Kategorien
 - Archiv-Dateisysteme
 - Ermöglicht Zugriff auf Dateien, die sich in Archiven befinden (z.B. tar, zip)
 - Datenbank-Dateisysteme
 - Ablegen von Dateien in relationalen Datenbanken
 - Medien-Dateisysteme
 - Speichern von Dateien auf Mediengeräten

Einbindbare Dateisysteme

- Nicht ursprüngliche Dateisysteme
 - Zugriff auf Dateisysteme, die keine Linux-Standards sind (z.B. NTFS, ZFS)
- Verschlüsselte Dateisysteme
 - Sichere Dateilagerung durch Verschlüsselung
- Netzwerk-Dateisysteme
 - Dateiablage auf entfernten Rechnern

Einbindbare Dateisysteme

- NTFS
 - Nicht ursprüngliches Dateisystem
 - Einbindung virtueller Festplatten
 - NTFS-3G Open-Source-Implementation von NTFS
 - Alternativ mit Captive NTFS
 - Lese- und Schreibfähigkeit

Einbindbare Dateisysteme

- SSHFS
 - Netzwerk-Dateisystem
 - Basiert auf dem SSH File Transfer Protocol
 - Secure Shell erlaubt gesicherten Datenaustausch zwischen vernetzten Geräten
 - Laufwerke werden über SSH vom Nutzer eingehängt
 - Ermöglicht den Zugriff auf Dateisysteme über Netzwerk/Internet

Einbindbare Dateisysteme

- GmailFS
 - Netzwerk-Dateisystem
 - Dateisystem benutzt Gmail-Account als Speichermedium
 - FUSE bietet das Dateisystem an und kommuniziert mit Gmail
 - Begrenzte Geschwindigkeit
 - Verstößt gegen die Benutzerbedingungen von Google

Einbindbare Dateisysteme

- Weitere Möglichkeiten
 - YoutubeFS
 - Durchblättern des Youtube-Accounts vom PC aus
 - gphoto2-fuse-fs
 - Ermöglicht den Zugang auf gphoto2 basierte Digitalkameras durch Standardprogramme
 - SiefS
 - Erlaubt die Nutzung von Siemens-Handys als entfernbaren Datenträger

Einbindbare Dateisysteme

- EncFS
 - Bietet ein verschlüsseltes Dateisystem an
- Relfs
 - Lagert Informationen über Daten in einer relationalen Datenbank
- LoggedFS
 - Protokolliert jede Operation, die in einem Dateisystem geschieht

Anwendungsbeispiel

- Benutzung von SSHFS
 - Verwendung von Konsolenbefehlen
 - "\$ usermod -a -G fuse klaus"
 - Erlaubt dem Benutzer FUSE zu benutzen
 - "\$ mkdir sshfstest"
 - Erstellt ein neues, leeres Verzeichnis
 - "\$ sshfs host:/path ~/localpath"
 - Bezeichnet Host und Zielpfad und hängt das Dateisystem ein
 - "\$ fusermount -u ~/localpath"
 - Hängt das Dateisystem wieder aus

Abschluss

- Grundlagen
- FUSE
 - Was
 - Wie
- Eigenschaften
 - Vor- und Nachteile
- Einbindbare Dateisysteme
- Einbindung von Dateisystemen

Quellen

- Abb. 1: <http://upload.wikimedia.org/wikipedia/de/1/1f/Filesystem.svg>
- Abb. 2: http://upload.wikimedia.org/wikipedia/commons/b/b3/Linux_Schichten.svg
- Abb. 3: http://upload.wikimedia.org/wikipedia/commons/0/08/FUSE_structure.svg
- Abb. 4: http://upload.wikimedia.org/wikipedia/commons/2/25/CPU_ring_scheme.svg
- <http://en.wikipedia.org/>
- <http://de.wikipedia.org/>
- <http://www.selflinux.org>
- <http://fuse.sourceforge.net/>
- http://www.linfo.org/context_switch.html
- <http://blog.tsunanet.net/2010/11/how-long-does-it-take-to-make-context.html>
- <http://www.techrepublic.com/blog/opensource/use-fuse-To-mount-remote-ssh-directories/225>
- <http://www.cs.rochester.edu/u/cli/research/switch.pdf>
- <http://www.linuxjournal.com/article/8904>