

FUSE

Filesystem in Userspace

Michael Burian Bernhard Guillon Stefan Huber

Institut für Computerwissenschaften
Universität Salzburg

Wissenschaftliche Arbeitstechniken und Präsentation

Gliederung

- 1 Was ist FUSE?
 - Allgemein
 - Wh. Betriebssysteme und FUSE-Ansatz
 - Motivation
- 2 Implementierung
 - concatfs
 - Source Code
- 3 Anwendungen
 - Überblick
 - Beispiele
 - ntfs-3g, sshfs, gmailfs, clamfs
 - Kurioses

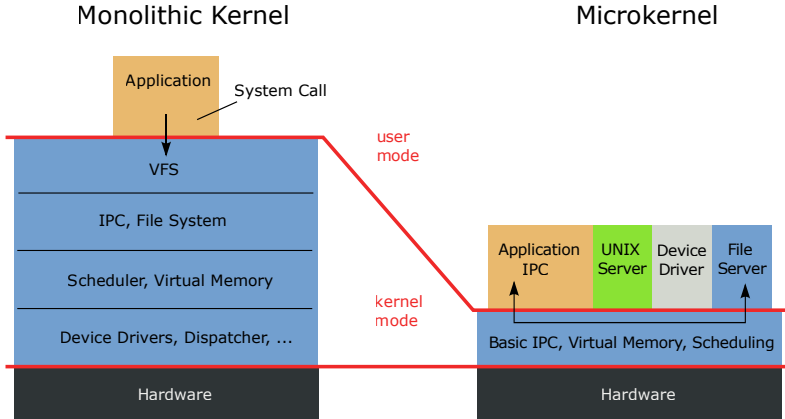
Gliederung

- 1 Was ist FUSE?
 - Allgemein
 - Wh. Betriebssysteme und FUSE-Ansatz
 - Motivation
- 2 Implementierung
 - concatfs
 - Source Code
- 3 Anwendungen
 - Überblick
 - Beispiele
 - ntfs-3g, sshfs, gmailfs, clamfs
 - Kurioses

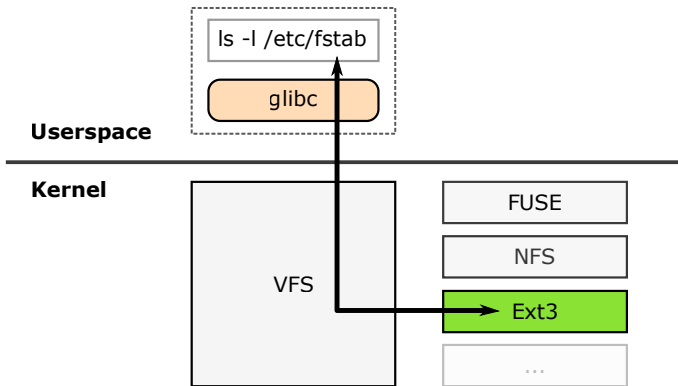
FUSE stellt sich vor

- FUSE ist ein Userspace Dateisystem Framework
- FUSE besteht aus
 - Kernel Modul (fuse.ko)
 - Userspace Library (libfuse.*)
 - Userspace mount (fusermount)
- FUSE öffnet neue Möglichkeiten für Dateisysteme

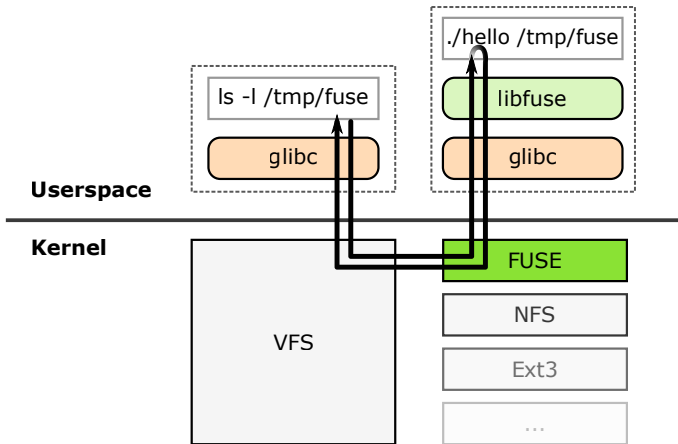
Monolithischer- vs. Micro-Kernel



Kernelmode Dateisystem Treiber



FUSE Dateisystem Treiber



Vorteile

- portabel (Linux, FreeBSD, OpenSolaris, ...)
- viele Sprachbindungen (z.B. Java, C#, Haskell, ...)
- closed source Treiber möglich
- einfache Fehlersuche
- Kernelstabilität wird nicht beeinträchtigt
- experimentelle Treiber

Nachteile

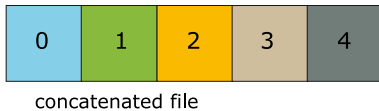
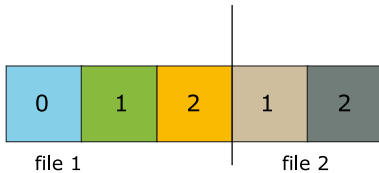
- langsamer durch viele Kontextwechsel

Gliederung

- 1 Was ist FUSE?
 - Allgemein
 - Wh. Betriebssysteme und FUSE-Ansatz
 - Motivation
- 2 Implementierung
 - concatfs
 - Source Code
- 3 Anwendungen
 - Überblick
 - Beispiele
 - ntfs-3g, sshfs, gmailfs, clamfs
 - Kurioses

Grundlagen

• Aufgabenstellung



Entwicklungszyklus

Sample Session

```
# $EDITOR concat.c
# make
cc -Wall -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse      -pthread
-lfuse -lrt -ldl      concat.c      -o concat
# mkdir mountpoint
# concat clean.sh clean.sh mountpoint/
# ls -l mountpoint/
total 0
-r--r--r-- 1 root root 96 Jan  1  1970 concat
# cat mountpoint/concat
latexmk -C
rm -f *.sty *.tex *.ps *.pdf *.svg

latexmk -C
rm -f *.sty *.tex *.ps *.pdf *.svg

# umount mountpoint/
# ls -l mountpoint/
total 0
```

Makefile

Makefile

```
CFLAGS := -Wall $(shell pkg-config fuse --cflags)
```

```
LDFLAGS := $(shell pkg-config fuse --libs)
```

```
targets = concat
```

```
all: $(targets)
```

```
clean:
```

```
  rm -f *.o
```

```
  rm -f $(targets)
```

Datenstrukturen

```
struct fuse_operations concat_oper = {
    .getattr = concat_getattr,
    .readdir = concat_readdir,
    .open = concat_open,
    .read = concat_read,
};

struct entry {
    char *name;
    off_t size; // the physical size in bytes
    off_t start; // absolute start address of this file within the concatenated file
    int fd; // file descriptor
};

// global argc,argv style list of files to concatenate
struct entry **files = 0;
int filesc = 0;
off_t totalsize = 0;
```

Ablauf

- 1 Kommandozeile parsen: `fuse_opt_parse`
- 2 Files verwalten: `add_file`
- 3 Callbacks registrieren, FUSE starten: `fuse_main`

Read Callback (1)

```
static int concat_read(const char *path, char *buf, size_t size,
off_t offset, struct fuse_file_info *fi)
{
    size_t total = 0;
    size_t res = 0;
    int i = 0;

    // we are only interested in read access to our single file
    if (strcmp(path, concat_path) != 0) {
        res = -ENOENT;
        goto out;
    }
    // attempt to read beyond end of file
    if (offset >= totalsize) {
        goto ok;
    }
    // if we cannot satisfy the (usually last) read due to EOF
    // calculate the remaining size
    if (offset + size > totalsize) {
        if (totalsize < offset)
            goto ok;
        size = totalsize - offset;
    }

    // to be continued ...
}
```


Read Callback (2)

```
static int concat_read(const char *path, char *buf, size_t size,
off_t offset, struct fuse_file_info *fi)
{
    // ... continued, with sanity checks omitted ...
    for (i = 0; i < filesc; i++) {
        while (offset > files[i]->size) {
            // skip files before our offset
            // TODO: linear search, O(n)
            offset -= files[i++]->size;
        }

        size_t tmp = pread(files[i]->fd, buf + total, size,
offset);
        size -= tmp;
        total += tmp;
        offset = 0;
    }

    ok:
    res = total;

    out:
    return res;
}
```

Gliederung

- 1 Was ist FUSE?
 - Allgemein
 - Wh. Betriebssysteme und FUSE-Ansatz
 - Motivation
- 2 Implementierung
 - concatfs
 - Source Code
- 3 Anwendungen
 - Überblick
 - Beispiele
 - ntfs-3g, sshfs, gmailfs, clamfs
 - Kurioses

Überblick

- ArchiveFileSystems (tar, cpio, zip, etc.)
- DatabaseFileSystems (MySQL, BerkeleyDB, etc.)
- MediaFileSystems
- NetworkFileSystems
- NonNativeFileSystems
- etc.

Beispiele - ntfs-3g

- NTFS-Treiber
- Bearbeiten, Löschen und Anlegen von Dateien
- Folgendes ist nicht möglich:
 - auf verschlüsselte Dateien zugreifen
 - komprimierte Dateien schreiben (lesen funktioniert)
 - den Dateibesitzer und die Zugriffsrechte ändern

Beispiele - ntfs-3g (cont.)

- **Verwendung:**

```
mount -t ntfs-3g /dev/hdX /mnt/windows  
fusermount -u /media/ntfs
```

- **Mount während des Bootvorgangs über /etc/fstab:**

```
/dev/sda1 /mnt/windows ntfs-3g defaults 0 0
```

Beispiele - sshfs

- SSH File Transfer Protocol (SFTP)
- Verwendung:

```
sshfs user@host:/path/ ~/mnt
```
- Kombination mit SSH-Authentifizierung über Public-Keys
- Automount über /etc/fstab

```
sshfs#USERNAME@REMOTE_HOST:REMOTE_PATH  
MOUNT_POINT fuse SSHFS_OPTIONS 0 0
```

Beispiele - gmailfs

- gmail-account als externer Speicherplatz (ca. 7 GB)
- libgmail um mit Gmail zu kommunizieren
- meisten Dateioperationen möglich
- Dateien als Attachment

Beispiele - gmailfs (cont.)

- Metainformation im Betreff:
 - Informationen zur Datei
 - fsname-Field (filesystem name)
- Probleme:
 - Alpha-Phase, d.h. buggy
 - sehr langsam
 - nicht von Google unterstützt

Beispiele - clamfs

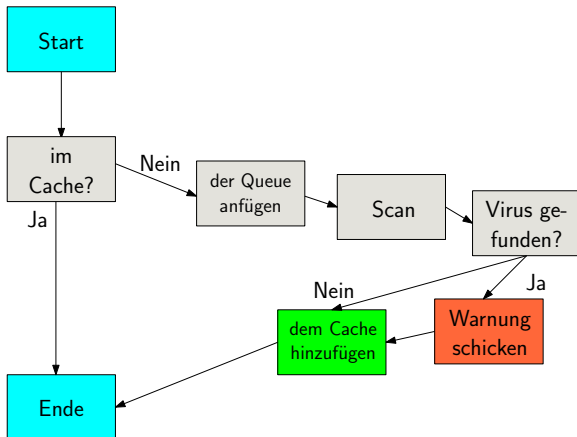
- transparente Anti-Virus-Überprüfung durch den clamd Daemon
- clamd: File-Scanning-Service entwickelt vom ClamAV-Projekt

Beispiele - clamfs (cont.)

- Konfiguration über XML-File
 - Angabe des root-Verzeichnis und mountpoint
 - weitere Optionen: readonly, public, etc.
 - Whitelist und Blacklist
- Aufruf des System-Daemon:

```
clamfs /etc/clamfs/<some_name>.xml
```

Beispiele - clamfs (cont.)



Kurioses - Hyppocampus

- Zitat: “All files (...) are jails. In those jails, informations are imprisoned between formatting rules and useless data” (Roberto -MadBob- Guido)
- Experimentelles relationales Filesystem
- Abbildung der hierarchischen Organisation in eine relationale Struktur
- Implementiert mit FUSE und libdbi

Kurioses - Hyppocampus

- Metainformation
- jede Datei besitzt eine ID
- Abfrage der Metainformation SQL-Queries

Kurioses - Hippocampus (cont.)

Beispiel von der Webpage:

```
madtablet:/# hattr -l "SELECT META_NAME WHERE META_ID
= 2"
pippo
  1 = pippo
  3 = 33152
  ...
  9 = 0
10 = 6
11 = text/plain
madtablet:/# ls "/mnt/hippo/SELECT 10 WHERE 1
= 'pippo' "
6
```

Quellen I

- [1] Mikrokern <http://en.wikipedia.org/wiki/Mikrokernel>
- [2] FUSE Sourceforge <http://fuse.sourceforge.net/>
- [3] Linux Kernel Source <http://kernel.org/>
- [4] Wikipedia FUSE
http://de.wikipedia.org/wiki/Filesystem_in_Userspace
- [5] NTFS-3g <http://www.ntfs-3g.org/>
- [6] SSH Filesystem <http://fuse.sourceforge.net/sshfs.html>
- [7] Jones R.: Gmail Filesystem
<http://richard.jones.name/google-hacks/gmail-filesystem/gmail-filesystem.html>
- [8] Burghardt K.: ClamFS <http://clamfs.sourceforge.net/>

Quellen II

- [9] Guido R.: Hyppocampus
<http://lobotomy.sourceforge.net/Hyppocampus.php>
- [10] Grafik Fuse Structure adaptiert von Sven
http://upload.wikimedia.org/wikipedia/commons/thumb/0/08/FUSE_structure.svg/320px-FUSE_structure.svg.png
(GNU Free Documentation License)
- [11] Grafik OS Structure adaptiert von
<http://upload.wikimedia.org/wikipedia/commons/6/67/OS-structure.svg>
(Public Domain)

Quellen III



Andrew S. Tanenbaum
Modern Operating Systems, 2/E .
Addison-Wesley, 2001.