

FUSE, SSHFS és társaik - bármit láthatunk lokális fájlként!

Szeredi Miklós

miklos@szeredi.hu

SUSE Labs/Novell

- Fjúz
- Fu-se
- ~~• Fjúzi~~
- ~~• Fusze~~



Tartalom

- FUSE alapú fájlrendszerek használata
 - Általános tudnivalók
 - Egyes fájlrendszerek bemutatása
- Elméleti háttér
 - Fájlrendszerek
 - FUSE működése
 - Továbbfejlesztési lehetőségek
- FUSE alapú fájlrendszerek fejlesztése
 - Egyszerű példák

I. rész

FUSE alapú fájlrendszerek használata

Általános tudnivalók

- Nem kell „root” jog
- De a csatolási ponthoz írás jog szükséges
- `fuse` csoport tagság kell egyes disztribúciókon:

```
$ sudo adduser $USER fuse
```
- Normál program (`ps`, `kill`, `stb.`)
- Háttérben fut (démon)

Indítás

- `hello`: példa fájlrendszer

<http://fuse.sourceforge.net/ipszilon/>

```
$ hello /tmp/valami
```

Fájlrendszer neve



Csatolási pont



Leállítás

- **Segédprogram:**

```
$ fusermount -u /tmp/valami
```

- **Root-ként:**

```
$ sudo umount /tmp/valami
```

- **Erőszakos módszerek:**

```
kill -9
```

```
umount -f
```

```
/sys/fs/fuse/connections/N/abort
```

Általános opciók

- h segítség
- d nyomkövetés
- f előtérben futás
- s egy szálú futás
- o *opció*

mount(8) opció, pl. ro, noatime, suid, ...

fuse opció

fájlrendszer saját opciói

Néhány FUSE opció

-o allow_other

Más felhasználó is láthatja a fájlrendszer tartalmát

-o kernel_cache

Fájlok tartalmának tárolása a kernelben

-o nonempty

Nem üres könyvtárra is csatolható

-o big_writes

Nagyobb mint 4kb méretű írás műveletek

Modulok

- „Szűrő”

- o `modules=M1[:M2...]`

- Például karakterkészlet konvertálás:

- o `modules=iconv,from_code=cp852,`
`to_code=utf8`

NTFS-3G (Szakacsits Szabolcs)

- Első GPL-es írható NTFS implementáció
- Legjobban elterjedt FUSE alapú fájlrendszer
 - Sok felhasználó → sok tesztelő → kevés hiba
 - Mozgatóerő FUSE más operációs rendszerekre való hordozása mögött

```
$ mount /dev/sda1 /mnt/ntfs
```

SSHFS

- Ötletadó: Florin Malita (LUFS szerzője)
- SFTP az SSH része:
 - nincs szerver oldali konfigurálás
- Más mint a többi hálózati fájlrendszer
 - NFS: egy példány → több felhasználó
 - SSHFS: egy példány → egy felhasználó

SSHFS (folyt.)

```
$ sshfs [név@]hoszt:[könyvtár] ...
```

- **Hasznos opciók:**

-C

tömörítés

-o *SSH-OPCIÓ*

(man ssh_config)

-o reconnect

újracsatlakozás

-o cache_timeout=*N*

kesselési idő

-o workaround=rename

-o transform_symlinks

mount program használata

- Például csatolás */etc/fstab*-ból indításkor
- Régi:

```
sshfs#név@hoszt:/ /mnt/hoszt fuse allow_other 0 0
```

- Új:

```
név@hoszt:/ /mnt/hoszt fuse.sshfs allow_other 0 0
```

CurlFtpFS (Robson Braga Araujo)

- FTP fájlrendszer, libcurl-ra épül
- FTP protokoll nem fájlrendszerekhez lett kitalálva

```
$ curlftpfs ftp://ftp.fsn.hu/ ...
```

- Érdeemes hosszú ideig kesselni:

```
-ocache_timeout=600
```

afuse (Jacob Bower)

- Automatikus csatolás (automount-hoz hasonló)
- Opciók:

```
mount_template="sshfs %r: / %m"
```

```
unmount_template="fusermount -u -z %m"
```

```
timeout=600
```




EncFS (Valient Gough)

- Titkosított fájlrendszer

```
$ encfs /tmp/titkos /tmp/valami
```

- Hasznos opciók:

```
-i PERC      Automatikus kilépés
```

FuseSMB (Vincent Wagelaar)

- Windows megosztott könyvtárak automatikus csatolása

```
$ fuse smb /tmp/halozat
```

```
/tmp/halozat/WORKGROUP/HOST/SHARE/
```

- Konfiguráció: `~/ .smb/fusesmb.conf`
- CIFS:
 - Egy könyvtár csatolása egyszerre
 - Nincs automatikus feltérlépezés

gphotofs (Philip Langdale)

- Fényképezőgépen található képekhez hozzáférés
- gphoto2 program fájlrendszer változata
- Több mint 1300 géptípust támogat

```
gphotofs /tmp/fenykepek
```

AVFS

- Virtuális fájlrendszerek gyűjteménye
 - Archívumok: tar, zip, rar, ar, rpm, deb, ...
 - Tömörítők: gzip, bzip2, lzma, ...
 - Hálózati: FTP, http, ssh, rsh, ...
 - Egyéb: lslR, patch, „avfsstat”, ...

```
$ avfsd /tmp/avfs
```

- Példa:

```
/tmp/avfs/tmp/xyz.tgz#/...
```



ObexFS

(Christian W. Zuckschwerdt)

- OBEX protokoll (Infra, USB, Bluetooth)
- Például egyes mobiltelefonokkal kapcsolat:

```
$ obexautofs /tmp/valami
```

```
/tmp/valami/00:11:22:33:44:55/C:/...
```

- Néha elszáll:

```
Transport endpoint is not connected
```

FuseISO (Dmitry Morozhnikov)

- iso9660 implementáció (CD/DVD image)

```
$ fuseiso xyz.img /tmp/valami
```

- Hasonló mint a

```
$ mount -o loop xyz.img /tmp/valami
```

- ...de nem kell hozzá root jog

mountlo

- Szintén a „`mount -o loop`” helyettesítés
- Összes Linux által támogatott fájlrendszer
- UML (User Mode Linux) segítségével:
 - Egy teljes virtuális Linux példány fut minden egyes csatolásnál

vmware-mount

- VMWare diszk image-ek csatolása

```
$ vmware-mount OpenSUSE.vmdk 2 ...
```

```
$ vmware-mount -d ...
```


Egy kis kombinálás

```
curlftpfs ftp://ftp.fsn.hu/.../11.3/iso/ /tmp/ftp
fuseiso /tmp/ftp/...-DVD-i586.iso /tmp/dvd
avfsd ~/avfs
cd /tmp/avfs/tmp/dvd/suse/i586/
cd sshfs-2.2-5.1.i586.rpm#/
cd CONTENTS.cpio#/usr/bin/
./sshfs
```

II. rész

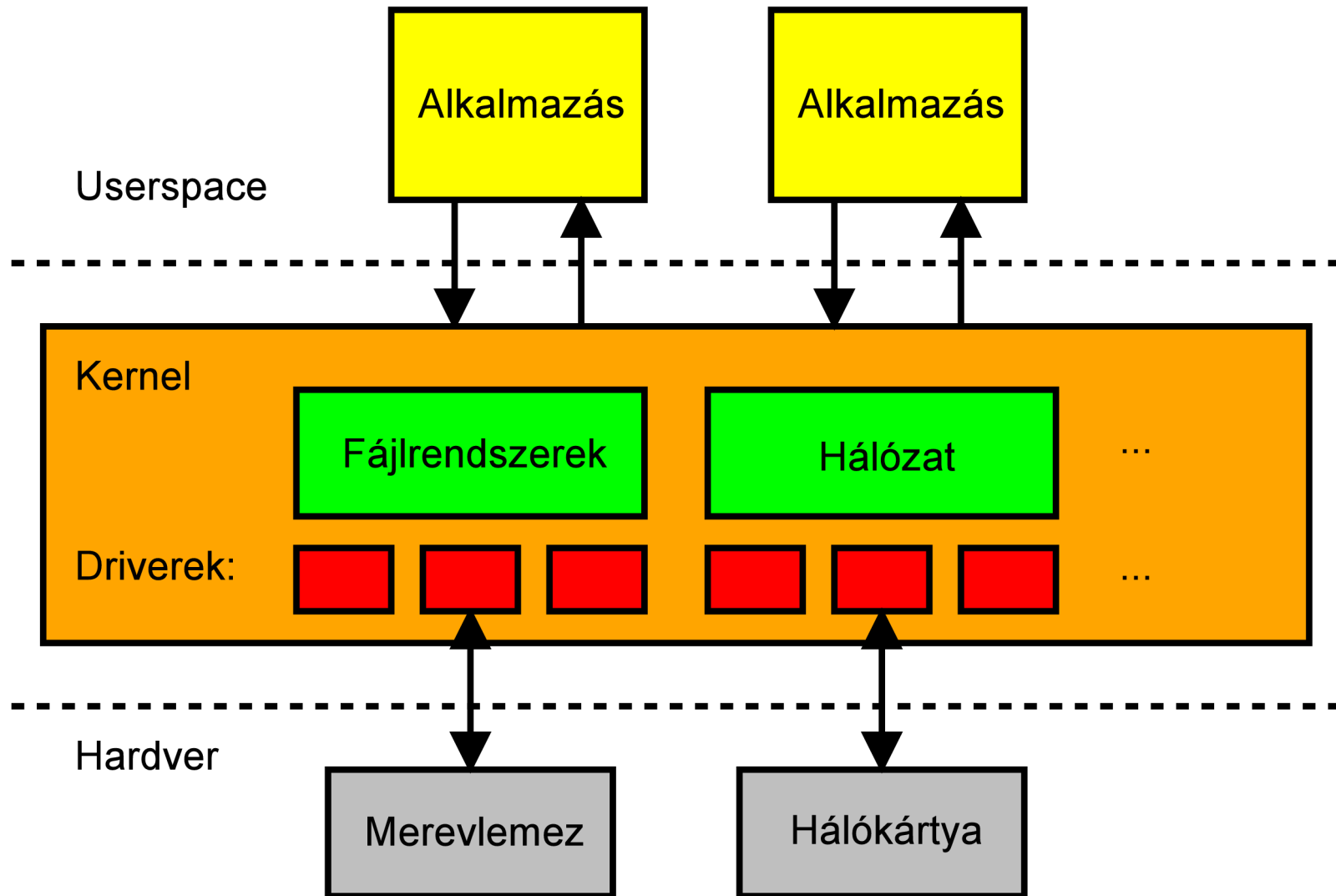
Elméleti háttér

Fájlrendszerek

Modern fájlrendszerek

- UNIX (Dennis Ritchie, Ken Thompson)
- Hierarchikus
- Egységes névtér → csatolás
- „Bármit láthatunk fájlként” koncepció
 - Pipe, eszközök (terminál, diszk), socket, ...

Operációs rendszer felépítése



Rendszerhívások nyomkövetése

```
$ strace cat valami
```

```
...
```

```
open("valami", O_RDONLY) = 3
```

```
read(3, "semmi\n", 4096) = 6
```

```
write(1, "semmi\n", 6) = 6
```

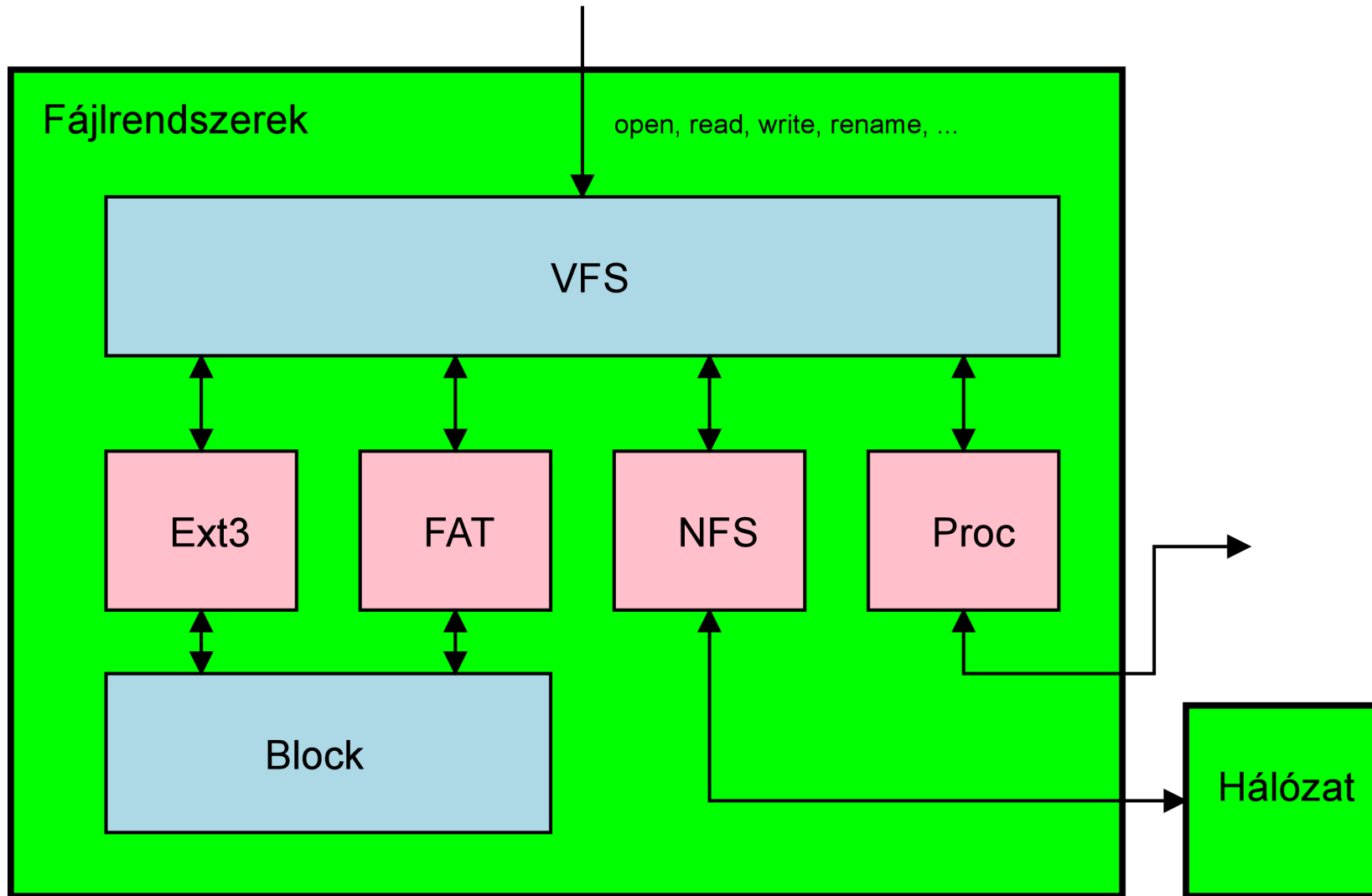
```
read(3, "", 4096) = 0
```

```
close(3) = 0
```

Fájlokat közvetlenül érintő rendszerhívások

- stat, access
- chmod, chown, utimes, truncate
- readlink, symlink
- open, read, write, fsync, lseek, getdents, close
- mknod, creat, unlink,
- mkdir, rmdir, link, rename
- statfs, sync
- (getxattr, setxattr, listxattr, removexattr)

Fájlsziszterek



VFS

- Elérési útvonal feldolgozása
 - Csatolások követése
 - A . és .. kezelése, szimbolikus linkek követése
 - Megfelelő fájlrendszer hívása
- Kesselés
 - fájlok adatai és metaadatai
 - útvonal kikeresés (lookup)

VFS adatstruktúrái

- Dentry:
 - Egy könyvtárbejegyzés
 - Fájlnev
 - Hivatkozás az inode-ra
- Inode:
 - Köznyelven: fájl
 - Fájl adatok és attribútumok
 - Hard link esetén *több dentry* → *egy inode*

VFS adatstruktúrái (folyt.)

- File:
 - **Megnyitott fájl**
- File descriptor:
 - Adott processzen belül egyedi azonosító
 - *több file descriptor → egy file*

FUSE: Fájlrendszer Userspace-ben

Miért jó?

- „Bármit láthatunk fájlként” koncepció továbbvitele
 - Archívumok, tömörített fájlok
 - Titkosítás
 - Különleges hardverhez hozzáférés
 - Különböző kütyük adatai
 - Távoli gépeken lévő fájlok
 - Adatbázisok tartalma
 - Régi verziók CVS/SVN/...-ben
 - stb...

Probléma: kernel programozás

- Nem használhatóak könyvtárak
- Csak „C” nyelven
- Különleges szabályok
- Egy apró hiba is fagyáshoz vezethet
- Nehéz hibát keresni

Előzmények

- Fájl menedzserek
 - Norton Commander, ...
 - Gnome VFS, KDE KIO
 - Korlátozott körben működnek
- Összes alkalmazás használhatja:
 - userfs (Jeremy Fitzhardinge)
 - PODFUK (Pavel Machek)
 - AVFS
 - LUFS (Florin Malita)

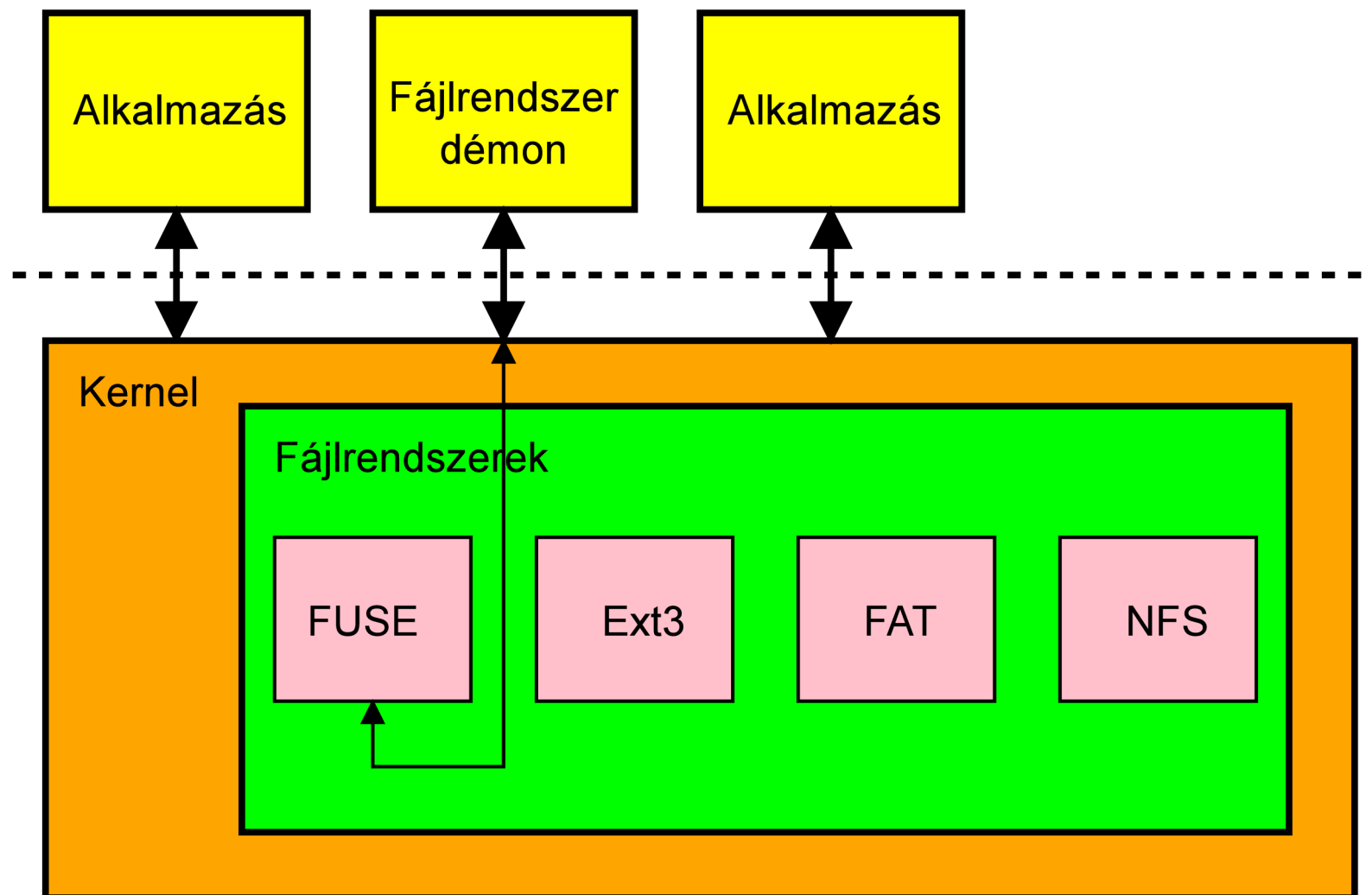
Történet

- 2003.02 – FUSE-1.0
- 2004.12 – FUSE-2.1
- 2005.09 – Támogatás a hivatalos kernelben
- 2005.11 – FreeBSD port (Henk Csaba)
- 2007.01 – MacFUSE (Amit Singh)
- 2010.09 – FUSE-2.8.5
- ?????.?? – FUSE-3.0

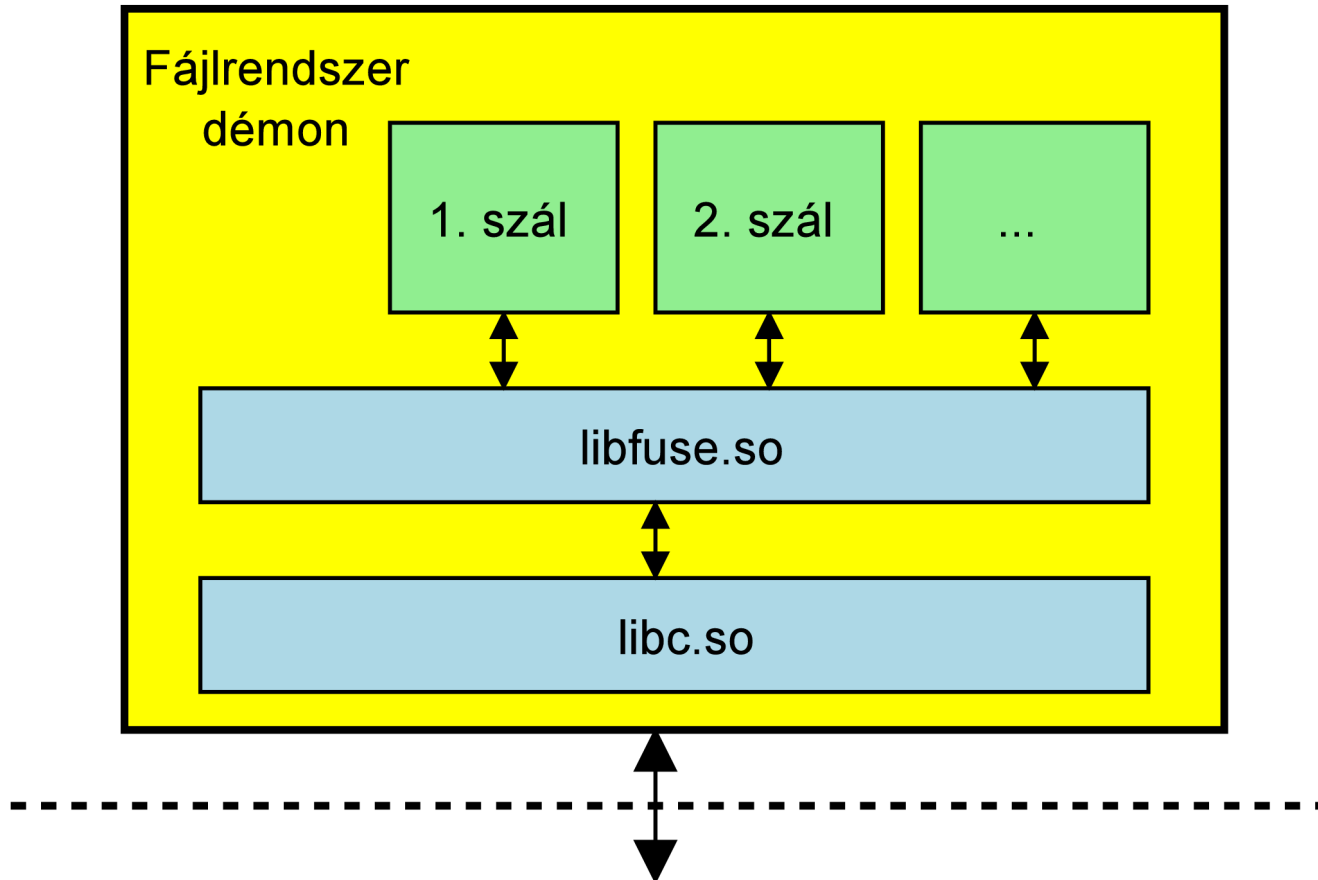
FUSE részei

- Kernel modul
 - ~8 500 sor
- libfuse könyvtár (nem kötelező, de ajánlott)
 - ~10 000 sor
- Nyelvi illesztő („C” esetén nem kell)
- Fájrendszer démon

FUSE kernel modul



Libfuse



Statisztika

- Web: <http://fuse.sourceforge.net/>
- Wiki: <http://sourceforge.net/apps/mediawiki/fuse>
- > 500 000 letöltés
- 198 FUSE alapú fájlrendszert regisztráltak
- 22 programozási nyelvi illesztő
- 7 operációs rendszer

Továbbfejlesztési lehetőségek

TODO lista

- Jelenleg kb. 50 elemű
- libfuse 3.0-ás verzió: rendcsinálás
 - Nem kell kompatibilisnek maradni
 - Rossz/kevésbé használt API-k ki lesznek dobálva
 - Jó API-k megmaradnak nagyjából változatlanul
- Memória fogyasztás optimalizálása
- Dokumentáció készítése

Kernel

- Felfüggesztés/hibernálás problémák
- Privilegium mentes csatolás
- Jobb támogatás valódi fájlrendszereknek
- Hibrid objektumok:
 - Fájl mint könyvtár → AVFS ahogyan ki lett találva
- Gyors fájl másolás

III. rész

FUSE alapú fájlrendszerek fejlesztése

Interfészek

- Kernel ↔ userspace kommunikáció:
 - /dev/fuse
 - Különböző méretű üzenetek
- libfuse API-k:
 - alacsony szintű (<fuse_lowlevel.h>
 - hasonlít a VFS API-hoz
 - magas szintű (<fuse.h>)
 - hasonlít a POSIX API-hoz

hello.c

- Egy 100 soros fájlrendszer
- Magas szintű API
- Kiindulási pontul szolgálhat új fájlrendszerekhez
- Program eleje:

```
#define FUSE_USE_VERSION 26  
#include <fuse.h>
```

hello.c: program vége

```
static struct fuse_operations hello_oper = {  
    .getattr      = hello_getattr,  
    .readdir     = hello_readdir,  
    .open        = hello_open,  
    .read        = hello_read,  
};
```

```
int main(int argc, char *argv[])  
{  
    return fuse_main(argc, argv, &hello_oper, NULL);  
}
```

Első példa

- `hello.c` módosításával
- `'hello'` helyett hálózati interfészek
- Tartalom: összes eddigi forgalom byte-ban
- Módosítások, egyenként kipróbálva:
 1. `hello_readdir()`
 2. `hello_getattr()`
 3. `hello_read()`

```
static int hello_readdir(...)
{
    FILE *f;
    char ifname[256];
    ...
    filler(buf, "..", NULL, 0);
    f = popen("ip link show|grep '^[0-9]|cut -f2 -d:", "r");
    if (f == NULL)
        return -errno;

    while (fscanf(f, "%255s", ifname) == 1)
        filler(buf, ifname, NULL, 0);

    pclose(f);
    return 0;
}
```

```

static int hello_getattr(...)
{
    ...
    stbuf->st_nlink = 2;
} else {
    char cmd[320];

    snprintf(cmd, sizeof(cmd), "ip link show %s", path+1);
    res = system(cmd);
    if (res == 0) {
        stbuf->st_mode = S_IFREG | 0444;
        stbuf->st_nlink = 1;
        stbuf->st_size = 32;
    } else {
        ...
    }
}

```

```
static int hello_read(...)
{
    ...
    char num_str[32];
    unsigned long long total_bytes;

    if (strlen(path) > 256)
        return -EIO;

    total_bytes = get_interface_bytes(path + 1);
    len = snprintf(num_str, sizeof(num_str), "%llu\n", total_bytes);
    if (offset < len) {
        if (offset + size > len)
            size = len - offset;
        memcpy(buf, num_str + offset, size);
    }
    ...
}
```

Második példa

- `xmp.py`:
 - 188 sor
 - Tükrözi a gyöker könyvtár tartalmát (loopback)
 - hasonló: `fusexmp.c`, `fusexmp_fh.c`
 - szintén jó kiindulópont más fájlrendszerekhez
- Módosítsuk `xmp.py`-t, hogy valóban „tükrözze” a fájlokat

Transzformáció

```
def transform(path):  
    q = "."  
    for name in path.split("/"):   
        q += "/" + name[::-1]  
    return q
```

\$ nib/tac corp/fles/spam

Vége