

UNIX fájlrendszerek alapismeretei

kiegészítő fóliák az előadásokhoz

Mészáros Tamás

<http://home.mit.bme.hu/~meszaros/>

*Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék*

Az előző részekben történt...

- A folyamatok
 - a felhasználói programok futás alatt álló példányai
 - a programokat permanens tárból töltjük be
- A permanens táarak
 - nem felejtő, nagyságrendekkel nagyobb és lassabb a memóriánál
 - blokkos fizikai tárolás és fájl-alapú logikai szervezés
 - többféle megoldás egyedi jellemzőkkel (HDD, flash, usb, RAID, SAN, ...)
- A kernel
 - kezeli a hardver erőforrásokat (köztük a permanens táarakat)
 - a hardverkezelő réteg felett többszintű fájlrendszer réteg található
 - háttértár kezelés, fájlrendszer szervezés, logikai felépítés (fájl, könyvtár)
 - adminisztrálja a fájlok blokkjait és az üres helyeket a permanens tárban
 - elvégzi a fizikai és logikai szervezés közötti leképezést
 - programozói interfészt nyújt az alkalmazásfejlesztők számára

Alapfogalmak

- Fájl (más néven állomány, de biztosan nem „file” magyarul)
 - adattárolási hely
- Fájlrendszer (állományrendszer)
 - fájlok tárolásának szervezése, hozzáférés biztosítása
- Fájlrendszerek felhasználói felülete
 - programozói (API, rendszerhívások)
 - parancssori (illetve grafikus)
- Fájlrendszerek szervezési felülete
 - diszk szervezés

UNIX fájlrendszerek történeti áttekintése

- System V első fájlrendszer *s5fs*
 - 80-as évek, alap implementáció, egyszerű szervezés
- 4.2 BSD Fast File Systems *FFS (a Linux ext2 alapja is)*
 - megnövelt teljesítmény
 - új szolgáltatások
 - akkori diszk hardver felépítéshez optimalizált rendszer
- Virtuális fájlrendszerek *vnode/vfs*
 - moduláris, objektum-orientált
 - cserélhető szervezési modulok, akár hálózati is
- Elosztott fájlrendszerek
 - NFS: transzparens hálózati fájlrendszer RPC megvalósítással
- Modern fájlrendszerek
 - ext3, ext4, xfs, ReiserFS, Solaris ZFS
 - felhasználói fájlrendszerek gnome-vfs, fuse: ftp, smb, dav, stb. célra
 - Klaszter fájlrendszerek, pl. Red Hat GFS

A fájlrendszer felhasználói szemmel

- Operációs rendszer felhasználó
 - parancssori és grafikus felület
 - könyvtárszervezés, speciális könyvtárak
 - fájlok és könyvtárak kezelése, attribútumaik
 - fájlrendszerek menedzselése (rendszergazda)
- Programozó (alkalmazás fejlesztő)
 - programozói interfészek (rendszerhívások, rendszerkönyvtárak)
 - fájlleírók, nyitott fájl objektumok és kezelésük
 - zárolási módszerek: kötelező, ajánlott

Felhasználói interfész

- Diszkek, partíciók, fájlrendszerek
- Tipikus UNIX könyvtárszerkezet
- Fájlrendszerek csatlakoztatása a könyvtárszerkezethez
- Fájl attribútumok
 - típus (- d p l b c s)
 - linkek (hard, szoft)
 - eszköz, inode, méret, stb.
 - időbélyegek (ctime, mtime, atime)
 - azonosítási és hozzáférés-szabályozási adatok
 - listázási parancs: `ls -la`

```

-rw-r--r--  2 root root      189 szept  8  2006 /etc/hosts
-rwxr-xr-x  1 root root 616920 nov 17 01:29 /bin/bash
srwxr-xr-x  1 clamilt clamilt 0 ápr 23 10:16 clamav.sock
crw-rw----  1 root tty      4, 0 ápr 20  2007 /dev/tty0
-r-s--x---  1 root apache 10760 jan 14 14:22 suexec
          
```

Demók (otthon is kipróbálható!)

- Csatolt fájlrendszerek: mount, umount, df
Mi az a /dev/shm?
- A UNIX könyvtárstruktúra felépítése: cd, pwd, ls
Milyen fájlok és könyvtárak neve kezdődik ponttal?
- Fájlok attribútumai: ls -la, ls -laZ
Miért van a hozzáférési jogok végén egy pont (v. plusz jel) Fedora alatt?
- Fájlműveletek: cp, mv
Hogyan lehet átnevezni egy fájlt?
- A noatime opció hatása a teljesítményre (**házi feladat**)
A /etc/fstab fájlban módosítsuk az attribútumokat (lásd man mount)!
Nézzük meg a relatime opciót is!

Programozói interfész

- Fájlok megnyitása (létrehozása)
 - *open()* rendszerhívás és paraméterei
 - a fájlleíró és a nyitott fájl objektum
 - több folyamat által megnyitott fájl és a *fork()*
- Írás és olvasás: *read()*, *write()*
- Fájlok zárolása
 - kötelező (mandatory): *fcntl()*, *lockf()*
 - ajánlott (advisory): *flock()*
- Fájlok lezárása: *close()*

Fájlrendszerek szervezése

- Csatlakoztatás
 - csatlakoztatási pont
 - elfedés
- Szervezés a háttértáron
 - blokkos tárolás
 - fájlok leírói (diszk inode)
 - szabad helyek kezelése
- Szervezés a memóriában
 - csatlakoztatás nyilvántartása
 - fájlok leírói (memória inode)
 - kapcsolat a nyitott fájl objektumokhoz

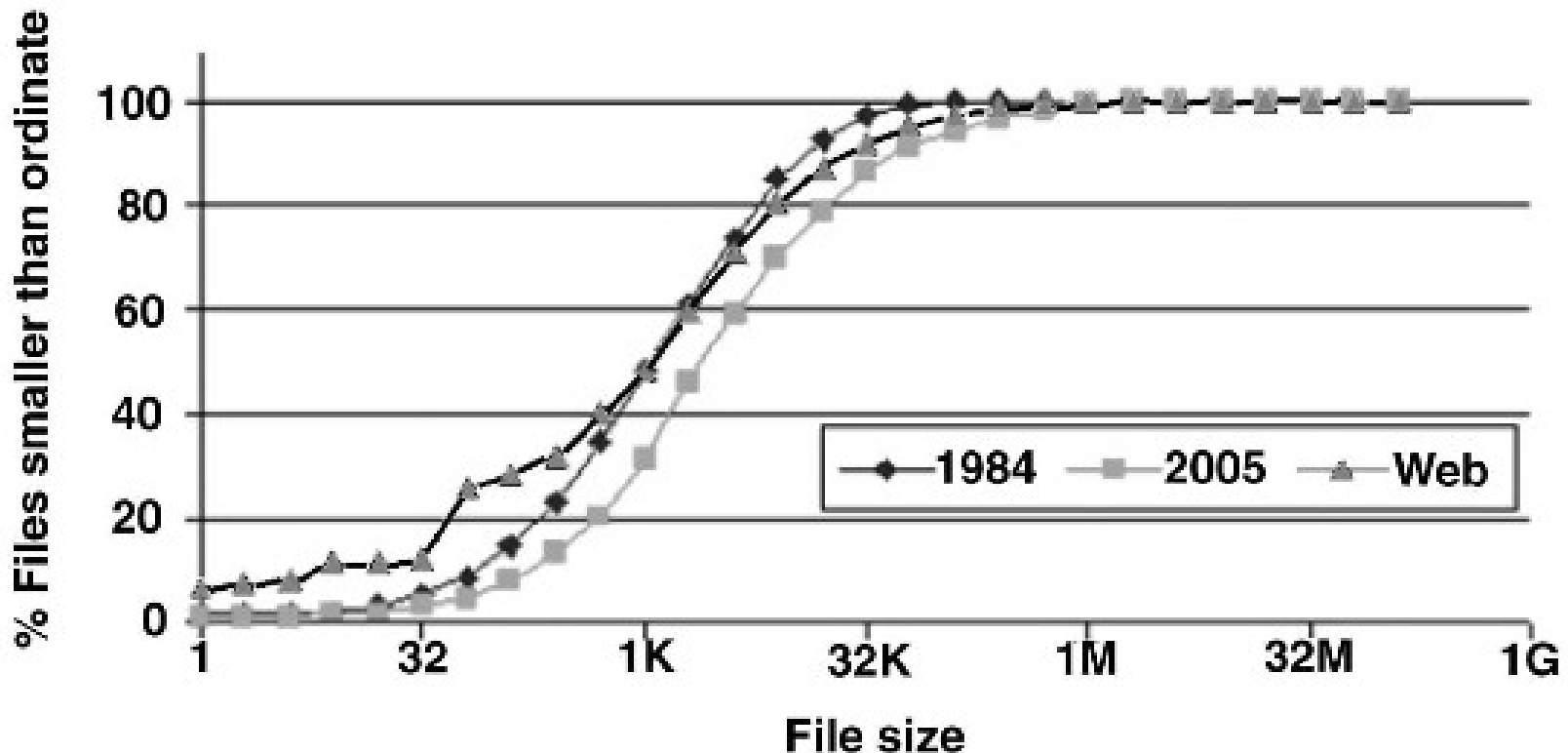
A tárolás megvalósítása

- A diszken elhelyezett fájlrendszer részei
 - szuperblokk (fájlrendszer metaadatok)
 - inode lista (fájl metaadatok)
 - tárolt adatok
- Szuperblokk
 - a fájlrendszer mérete
 - szabad blokkok jegyzéke
 - zárolási információk
 - módosítás jelzőbit

Az index node (inode)

- hitelesítési információk (UID, GID)
- típus
- hozzáférési jogosultságok
- időbélyegek
- méret
- adatblokkok elhelyezkedése (címtábla)
 - 10-15 db direkt blokkcím
 - 1x, 2x és 3x indirekt blokkcímek(számítási példa)

Címzés és tipikus fájlméretek



Andrew S. Tanenbaum, Jorrit N. Herder, Herbert Bos: File size distribution on UNIX systems: then and now. Operating Systems Review 40(1): 100-104 (2006)

Az inode a memóriában

- a nyitott fájl objektumhoz kapcsolódik
- diszk inode tartalma bekerül a memóriába
- az aktív használat információival bővül
 - státusz (zárolt, módosított, stb.)
 - háttértár eszköz (fájlrendszer) azonosítója
 - hivatkozás számláló (fájlleírók)
 - csatlakoztatási pont adminisztrációja
 - ...

Allokáció a diszken

- Szempontok: teljesítmény, megbízhatóság
- Cylinder (blokk) csoport (FFS, ext2, ...)
- Allokációs elvek
 - szuperblokk másolása minden csoportba
 - inode lista és szabad blokkok csoportonként kezelve
 - egy könyvtár – egy csoport
 - kis fájlok egy csoportba
 - nagy fájlok „szétkenve” több csoportba
 - új könyvtárnak egy új, kevésbé foglalt csoportot keres

A virtuális fájlrendszer

- Implementáció-független fájlrendszer absztrakció
 - a modern unix fájlrendszerek alapja

- Célok:
 - többféle fájlrendszer egységes egyidejű támogatása
 - egységes kezelés a csatlakoztatás után (programozó IF)
 - speciális fájlrendszerek (hálózati, processz, stb.)
 - modulárisan bővíthető rendszer

- Absztrakció
 - inode \longrightarrow vnode
 - fs \longrightarrow vfs

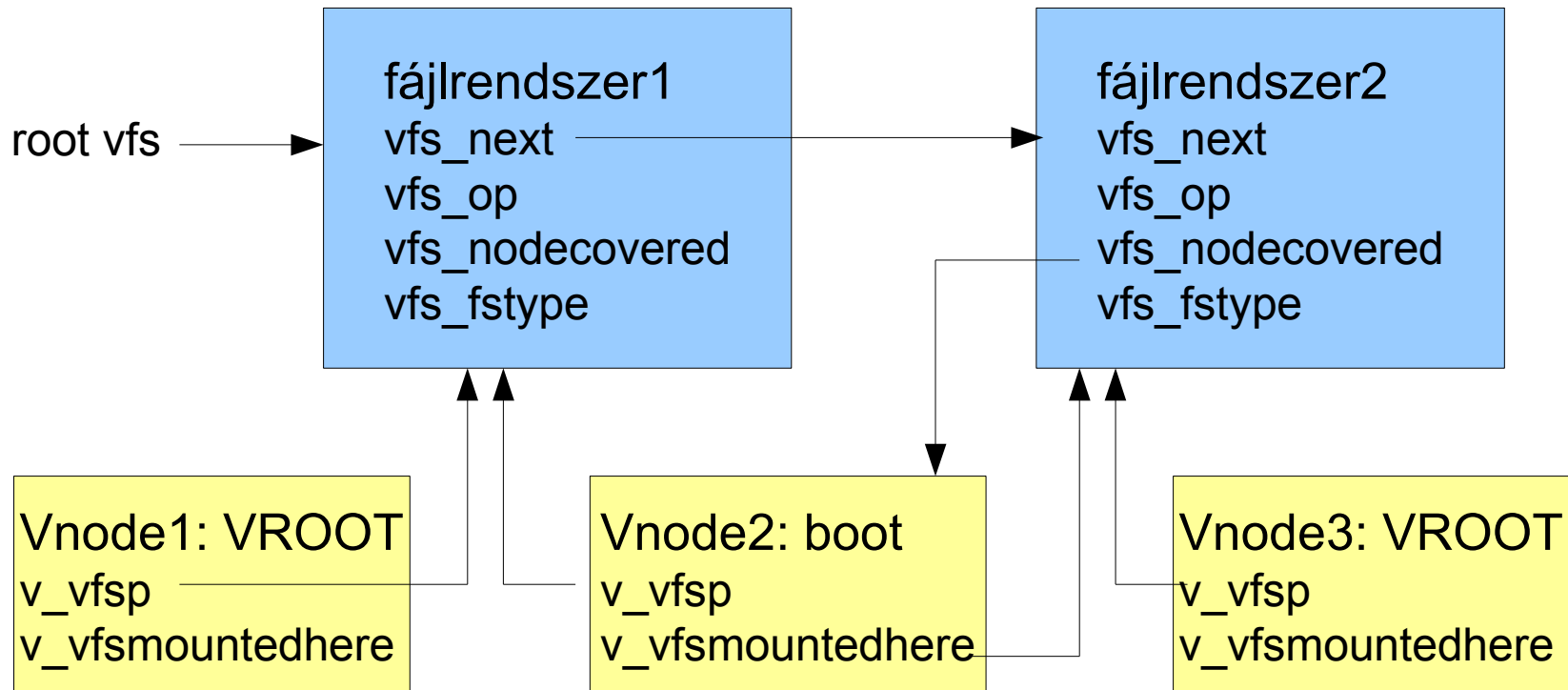
A vnode absztrakció

- adatmezők
 - közös adatok (típus, csatlakoztatás, hivatkozás száml.)
 - `v_data`: állományrendszertől függő adatok (inode)
 - `v_op`: az állományrendszer metódusainak táblája
- virtuális függvények
 - állományrendszertől független: `vop_open`, `vop_read`,...
 - a tényleges metódusokra helyettesítődnek be
- segédrutinok, makrók

A vfs absztrakció

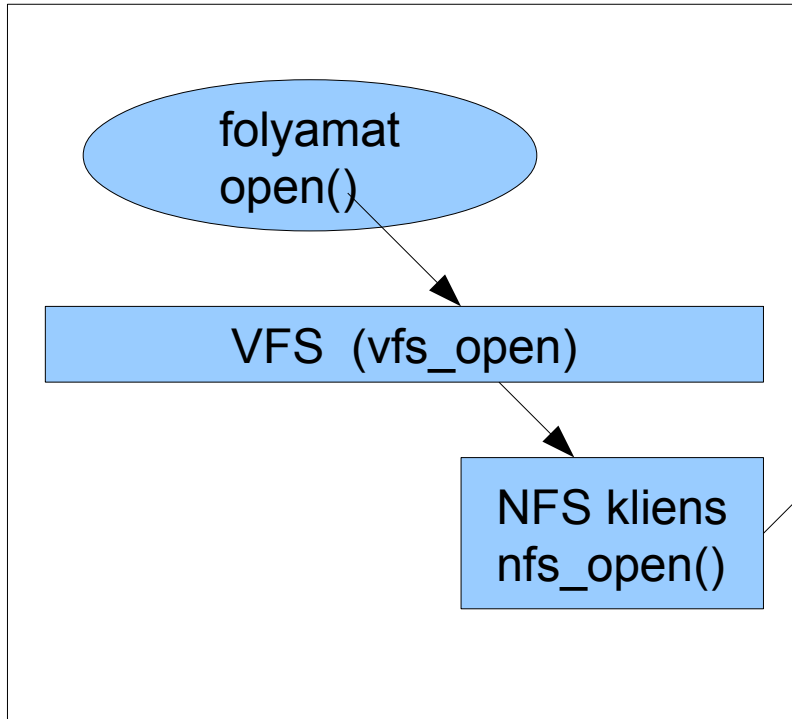
- adatmezők
 - közös adatok (fájlrendszer típus, csatlakoztatás, hivatkozás, `vfs_next`)
 - `vfs_data`: állományrendszertől függő adatok
 - `vfs_op`: az állományrendszer metódusainak táblája
- virtuális függvények
 - állományrendszertől független: `vfs_mount`, `vfs_umount`, `vfs_sync`,...
 - a tényleges metódusokra helyettesítődnek be
- segédrutinok, makrók

A vfs és a vnode kapcsolata

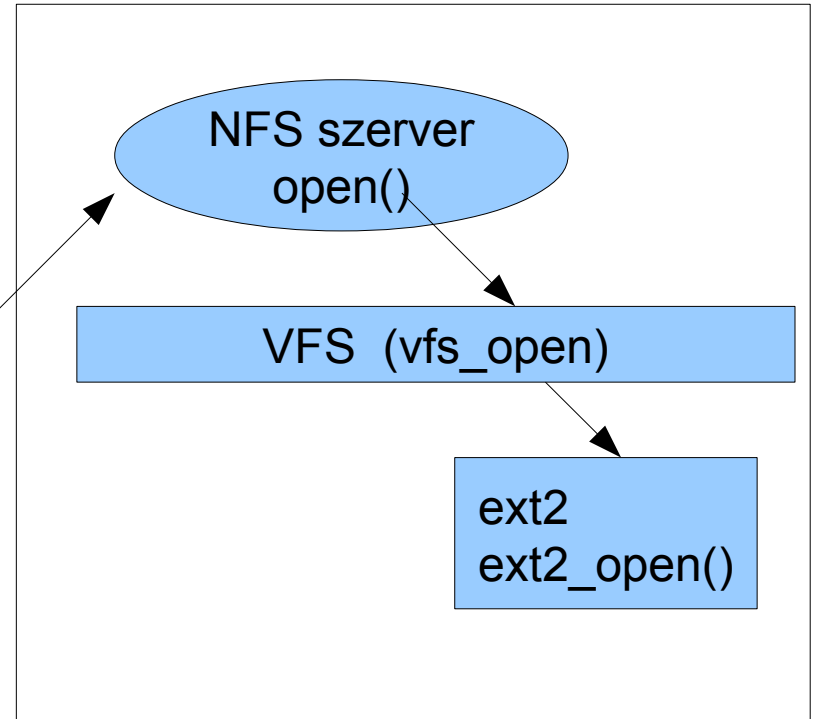


Alkalmazási példa: NFS egyszerűsített felépítés

szgép 1



szgép 2



RPC

Összefoglalás: UNIX fájlrendszerek alapismeretei

- Alapok
 - Fájl (esetleg állomány, nem file!), fájlrendszer, API, diszk szervezés
 - Az s5fs-től a ZFS-en át az elosztott és felhasználói fájlrendszerekig
- Felhasználói alapismeretek
 - Könyvtárszervezés, speciális könyvtárak
 - Fájlok és könyvtárak attribútumai
 - Felhasználói eszközök (cd, pwd, ls, cp, mv,
 - Adminisztrátori eszközök (mount, umount, df, ...)
- Fájlrendszerek megvalósítása
 - adatstruktúrák (**inode**), interfészek
 - diszk szervezés, index címtábla, allokáció
 - virtuális fájlrendszer felépítés, alkalmazási példa (NFS)