UNIX fájlrendszerek alapismeretei

kiegészítő fóliák az előadáshoz

Mészáros Tamás
http://www.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 tárak
  - nem felejtő, nagyságrendekkel nagyobb és lassabb a memóriánál
  - blokkos fizikai tárolás és fájl-alapú logikai szervezés
  - többféle megoldás egyedi jellemzőkkel (HDD, flash, RAID, SAN, iSCSI,...)
- A kernel
  - kezeli a hardver erőforrásokat (köztük a permanens tárakat)
  - 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 Sytems *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
  - ext[234], xfs, ReiserFS, Solaris zfs, (Oracle) Linux btrfs
  - 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
  - helyfoglalás ellenőrzése
  - hely és távoli fájlrendszerek csatolása (részben rendszergazda)
  - fájlrendszerek menedzselése (rendszergazda)
- Programozó (alkalmazás fejlesztő)
  - programozói interfészek (rendszerhívások, rendszerkönyvtárak)
  - fájlleírók, nyitott fájl objektumok és kezelésük
  - zárolási módszerek: kötelező, ajánlott

Felhasználói interfész

- Alapvető felépítés (fa, lásd később) és parancsok

  `ls cp mv rm cd pwd mkdir rmdir`

- Menedzsment (csatlakoztatás, helyfoglalás)

  `mount umount df du /etc/fstab /etc/mtab lsof`

- Fájl- (könyvtár) attribútumok
  - típus (`- d p l b c s`)
  - linkek (hard, soft)
  - eszköz, inode, méret, stb.
  - időbélyegek (ctime, mtime, atime)
  - azonosítási és hozzáférés-szabályozási adatok (lásd következő fólia)
  - listázási parancs: `ls -l`

```
-rw-r--r--  2 root root     189 sze  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
```

Hozzáférési jogosultságok

- POSIX hozzáférési jogosultságok (*alap jogosultságok*)
  - 3 x 3 bit: tulajdonos, csoport, mások x olvasás, írás, futtatás
  - értékek: 4: olvasás, 2: írás, 1: futtatás, 0: nincs jogosultság
    pl.: 740 = tulajdonos: olvasás, írás, futtatás; csoport: olvasás; mások: semmi
  - könyvtárak esetében olvasás és „futtatás” is kell a tartalom listázásához
  - beállítás: chmod <jogosultság> <fájl v. könyvtár>
    pl.: `chmod 750 /home/me` `chmod u+rwx,g+rx,o-rwx /home/me`
- Speciális jogosultságok: SETUID, SETGID, StickyBit
  - SETUID/GID futtatás esetén beállítja az UID/GID értékét a fájlnak megfelelően, chmod u+s setuid_file chmod g+s setgid_file
    Speciális (root) jogokat igénylő műveletekre szokás beállítani: VESZÉLYES!
  - StickyBit könyvtárra beállítva csak a tulajdonosa törölheti a fájlokat
    Jellemzően mindenki által írható könyvtárakra szokás beállítani (/var/tmp)
- POSIX ACL (access control list) (*kiterjesztett jogosultságok*)
  - meghatározott felhasználóknak és csoportoknak külön is jogokat adunk
    pl.: `setfacl -m u:student:r file`
  - az `ls` parancs a jogosultságok utáni `+` jellel figyelmeztet a meglétére

A UNIX fájlrendszer áttekintése

- A fa gyökere (/ avagy ROOT) a kiinduló pont (`ls /`)
  - `/bin` a rendszer működéséhez szükséges alapvető bináris állományok
  - `/sbin` hasonló, de alapvetően a rendszergazda által futtatható programok
  - `/dev` hardver eszközök
  - `/etc` a rendszer konfigurációs beállításait tároló fájlok
  - `/home` a felhasználók saját könyvtárai
  - `/lib` alapvető (megosztott, shared) rendszerkönyvtárak
  - `/mnt` alkalmilag felcsatolt partíciók helye (mount)
  - `/tmp` átmeneti fájlok (programok és felhasználók számára)
  - `/usr` felhasználói programok, programkönyvtárak, dokumentáció, stb.
  - `/var` a rendszerműködés „dinamikus” fájljai, naplófájlok, adatbázisok

  lásd: *diszkhasználat elemző, du, xdu, baobab, kdiskstat, filelight*
- Fájlrendszer „szabványok”, változatok, trendek
  - A fentiek alapvetően igazak, azonban ezeken belül jelentős eltérések
  - A különböző disztribúciók eléggé változatosak
  - FHS: Filesystem Hierarchy Standard (1994, ..., 2004)
  - UsrMove: `/bin`, `/sbin`, ... a `/usr` megfelelő helyére (Solaris11, Fedora)

Gyakorlatok (otthon is kipróbálható!)

- Fájlrendszerek kezelése: `mount umount df mkfs fsck`

  `mount` (Mi az a /proc?) `df` `umount /boot` `mount /boot` (honnan?)

  Hozzunk létre egy új fájlrendszer egy fájlban!

  ```
  dd if=/dev/zero of=filesystem.img bs=1k count=1000
  losetup /dev/loop0 filesystem.img
  mke2fs /dev/loop0
  mount /dev/loop0 /mnt
  ```

  Az egyik tipikus, bosszantó hibajelzés

  ```
  umount: /mnt: device is busy
  ```

  Miért nem sikerül? Valaki foglal (nyitva tart) fájlt, könyvtárat.
  Mit tehetünk? Megnézzük, ki mit tart fogva: `lsof /mnt` (esetleg remount,ro?)

- A UNIX könyvtárstruktúra felépítése: `cd` `pwd` `ls` `mkdir`

  Milyen fájlok és könyvtárak neve kezdődik ponttal?

- Fájlok attribútumai: `ls -la` `ls -laZ` `setfacl`

  Miért van a hozzáférési jogok végén egy pont (v. plusz jel)?

- Fájlműveletek: `cp` `mv`

  Hogyan lehet átnevezni egy fájlt?

Fájlrendszerek hangolása (gyakorlatok otthonra)

- Szabad hely növelése
  - Töltsük tele a korábban létrehozott fájlrendszer-a-fájlban eszközt!
    `cp -r /bin /mnt` (ne root-ként futtassuk!)
  - Miért 0 a szabad hely, miközben nem foglalt minden blokk?
    súgó: `man tune2fs`
  - Fájlrendszerszintű tömörítés bekapcsolása
    pl.: btrfs mount opció: `compress = { zlib | lzo | snappy }`
    (A már létrehozott fájlrendszerre utólag is bekapcsolható.)

- Teljesítménynövelés
  - A `noatime` opció hatása a teljesítményre
    A `/etc/fstab` fájlban módosítsuk az attribútumokat (lásd `man mount`)
    Nézzük meg a `relatime` opciót is!
  - Fájlrendszerszintű tömörítés
    Lényegesen kisebb adatmozgatás, CPU terhelés kismértékű növelése
    Lásd pl.: www.phoronix.com/scan.php?page=article&item=btrfs_lzo_2638
  - Blokkméret beállítása (lásd tipikus fájlméretek vs. blokk címzés később)

Programozói interfész

- Fájlok megnyitása (létrehozása)
  - *open()* rendszerhívás és paraméterei
  - a fájlleíró és a nyitott fájl objektum (lásd később)
  - 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()*
- Könyvtárak kezelése: *opendir(), readdir(), rewinddir(), closedir()*

Fájlrendszerek szervezése (alapismeretek)

- Szervezés a felhasználói felületen
  - fa struktúra
  - csatlakoztatási pontok
  - 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

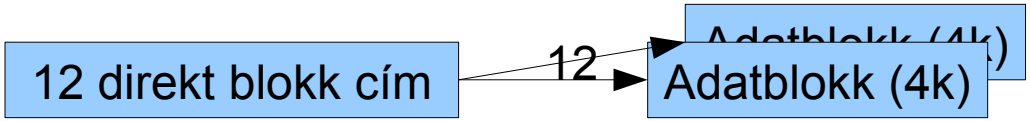
A tárolás megvalósítása

- A diszken elhelyezett fájlrendszer részei
  - szuperblokk (fájlrendszer metaadatok)
  - inode lista (fájl metaadatok)
  - tárolt adatok

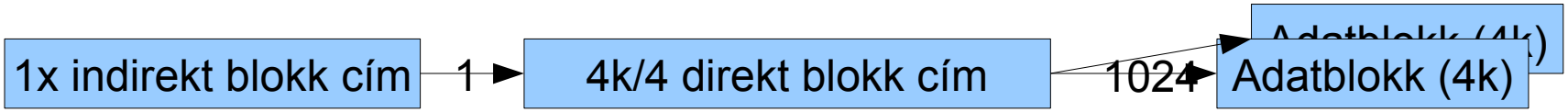
- Szuperblokk
  - a fájlrendszer típusa és mérete
  - szabad blokkok jegyzéke
  - inode lista információk
  - zárolási információk
  - módosítás jelzőbit
  - másolatok elhelyezkedése
  - ...

Az index node (inode)

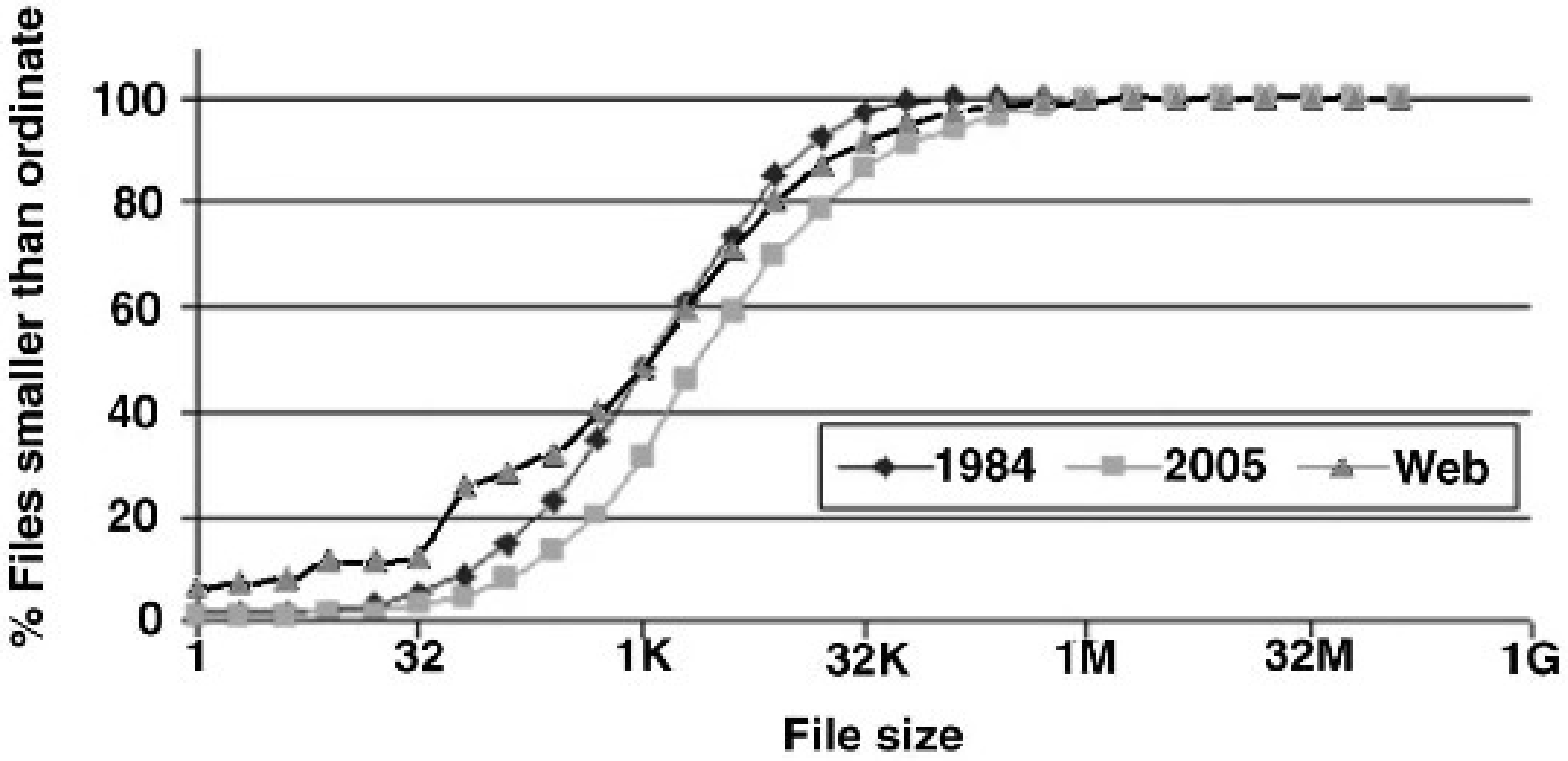
- hitelesítési információk (UID, GID)
- típus
- hozzáférési jogosultságok
- időbélyegek
- méret
- adatblokkok elhelyezkedése (címtábla)
  - 10-15 db direkt blokkcím
  - 1x, 2x és 3x indirekt blokkcímek



Mekkora a maximális fájlméret?



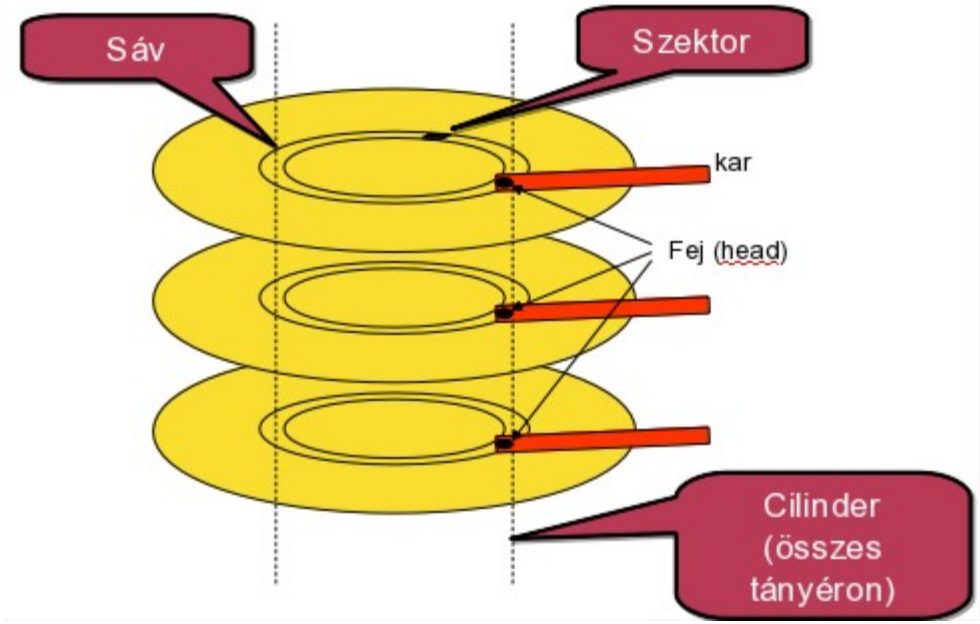
Blokkcímzés és tipikus fájlméretek



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)

Allokáció a diszken

- Szuperblokk, inode lista és adatblokkok elhelyezése a háttértáron
- Szempontok: teljesítmény, megbízhatóság
- Cilinder (blokk) csoport
  - pl.: 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, kevéssé foglalt csoportot keres



Az inode a memóriában

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

(Hasonlóképpen a fájlrendszer leírói is bekerülnek a csatlakoztatáskor.)

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 ⟶ vnode
  - fs ⟶ vfs

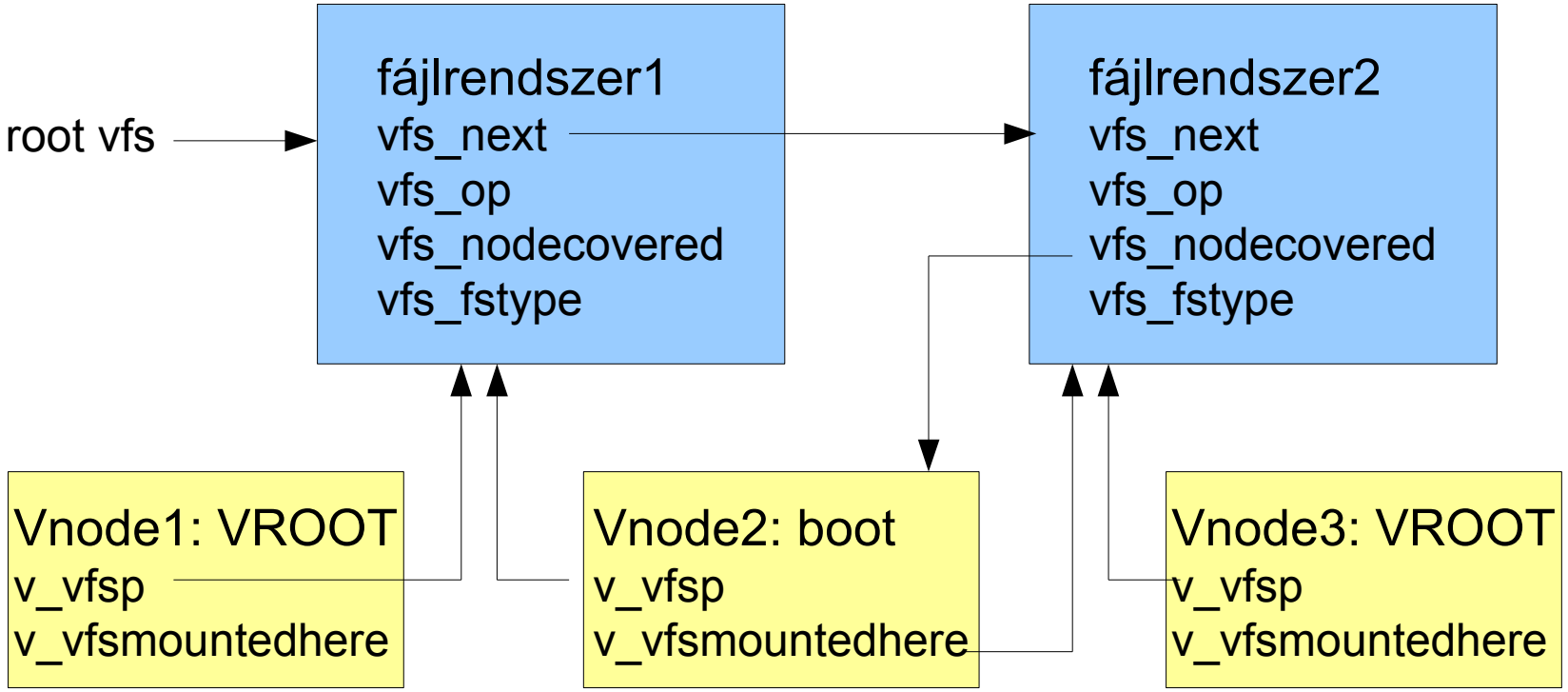
A vnode absztrakció

- 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

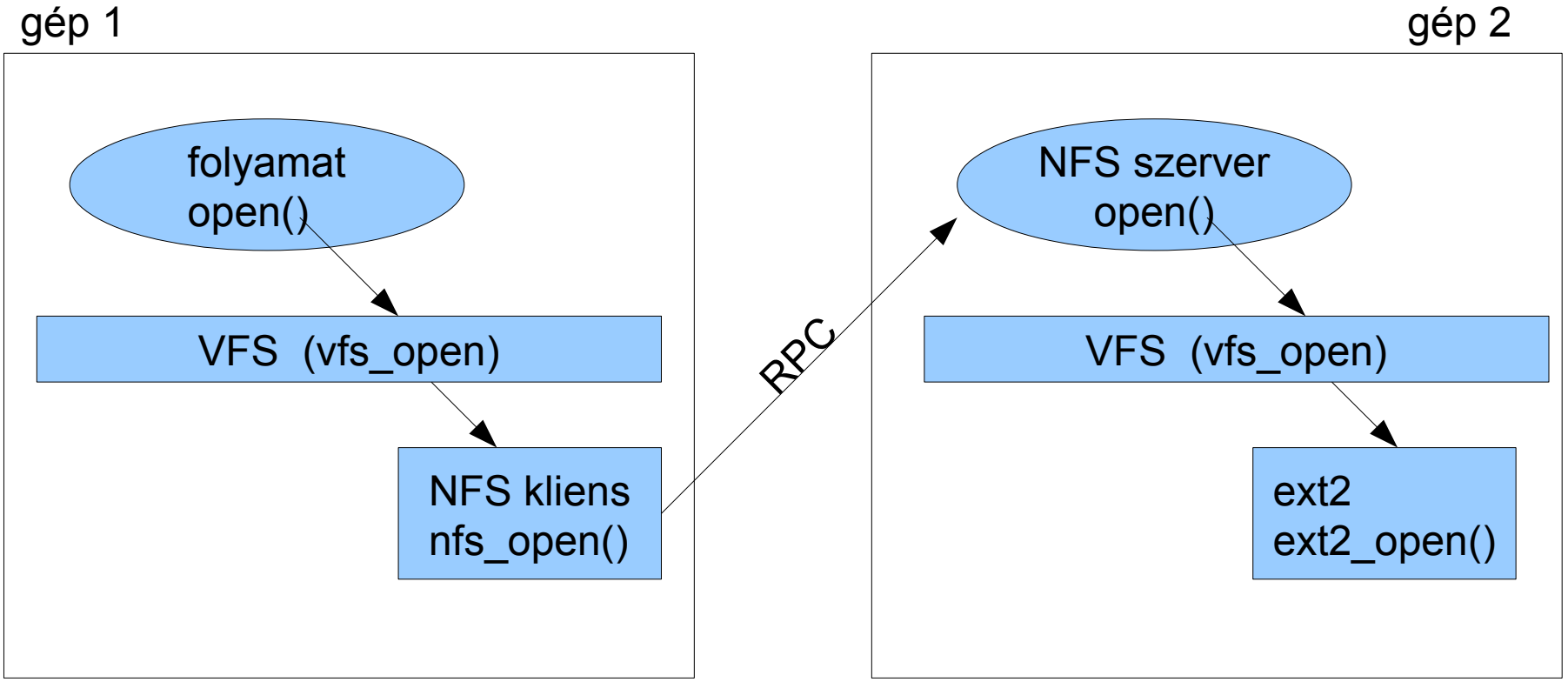
A vfs absztrakció

- 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



Készítsünk saját fájlrendszert VFS alapon!

- Fakultatív házi feladat
  - Aki igazán szeretné megérteni, hogyan működik a VFS.
  - Nem nehéz...

- Dokumentáció

  http://web.archive.org/web/20071027154020/http://www.geocities.com/ravikiran_uvs/articles/rkfs.html
  - Ravi Kiran, a weblapot már levették, ezért a web archívumból érhető el.

  http://us1.samba.org/samba/ftp/cifs-cvs/ols2006-fs-tutorial-smf.odp
  http://ftp.samba.org/pub/samba/cifs-cvs/ols2006-fs-tutorial-smf.pdf
  - Steve French, az IBM mérnöke, aki SAMBA fejlesztéssel foglalkozik

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)