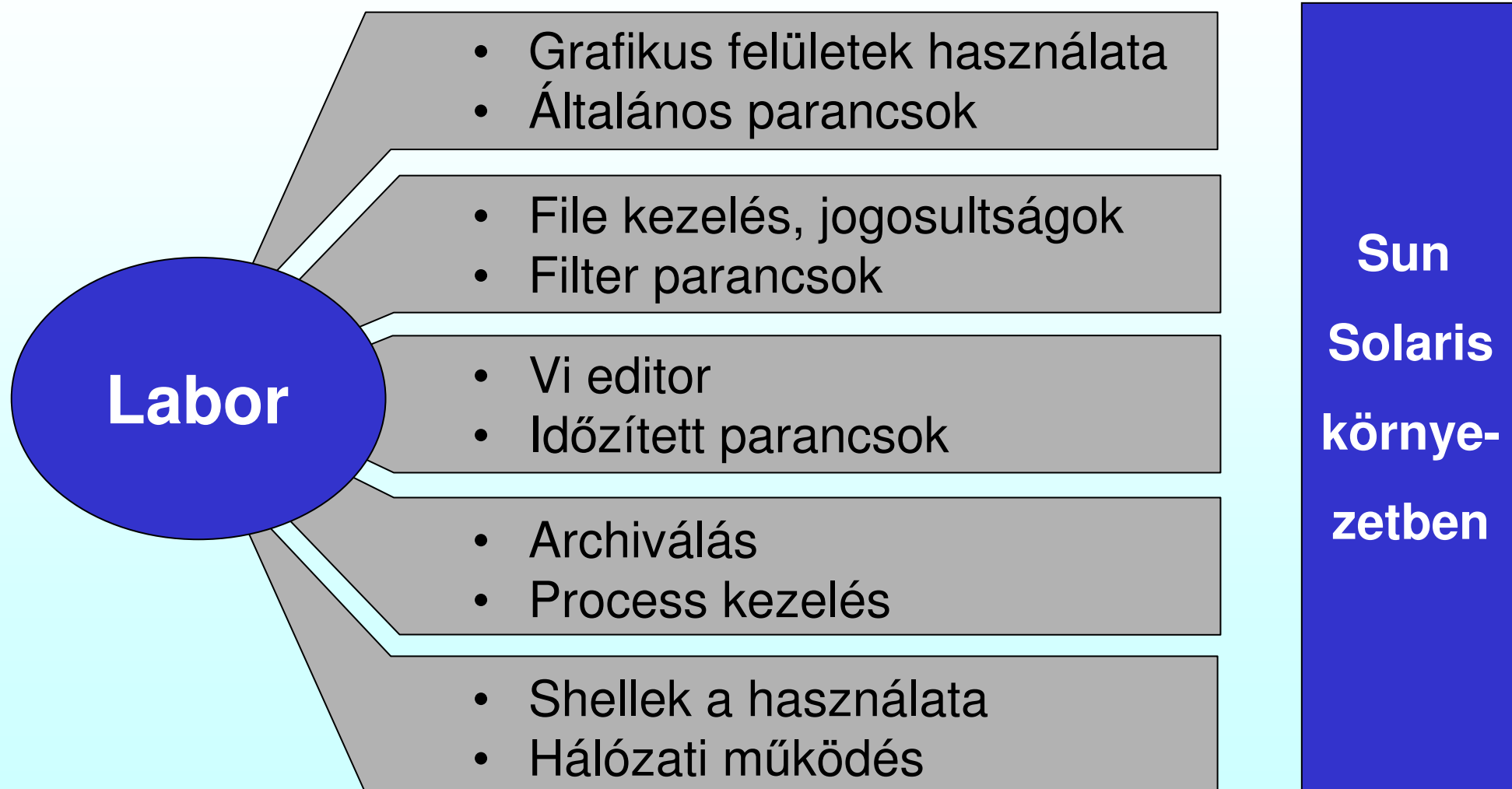


# **Alkalmazott számítástechnika**

# Követelmények, adminisztráció

- Alkalmazott számítástechnika?
  - UNIX + hálózat
- Előadás
  - Előadás:órarend szerint
  - Labor: félév vége
  - Labor: az előadás anyagára épül
- Vizsga
  - Kérdések: Előadás+labor
  - Vizsgajegy: ZH+labor
  - Labor: Kötelező részvétel!!!
- Jegyzet
  - A UNIX operációs rendszer használata
  - Mérési útmutató....
  - Prezentáció

# Labor tematika



# Miért a UNIX? Történeti áttekintés

- 1969 AT&T
  - Kutatási projekt, alapelvek rögzítése
  - Az akkori kor technikai-technológiai színvonala nem determinálta az alapelvek meghatározását!!!
- C programozási nyelv – a kernel is!
  - Előny: portabilitás, hátrány?
- 70-es évek
  - Nem piaci termék, az akadémiai szférában terjedt el
  - Sokféle változat
- Két fő változat
  - AT&T System V UNIX
  - BSD (Berkley Software Distribution) UNIX
- 80-as évek: új szolgáltatások
  - Grafikus felhasználói felület (GUI)
  - Hálózati szolgáltatások (TCP/IP, NFS ...)
  - Piaci termék, de probléma: Az eltérő változatok létezése korlátozta az elterjedést.

# Történeti áttekintés

- Megoldás: szabványosítás
  - Gyártói szervezetek
    - UI (UNIX International): AT&T, SUN ... --- SVR4
    - OSF (Open Software Foundation): IBM, HP, DEC ... --- OSF1
    - COSE (Common Open System Environment): 1993
  - Felhasználói szervezetek
    - POSIX (Portable Operating System Interface X): USA
    - X/Open: Európa
- PC architektúra
  - XENIX
  - SCO
  - LINUX – elterjedt, sokféle disztribúció
- Labor: SOLARIS (SUN UNIX)

# Alapfogalmak

## Multiuseres működés

- Több felhasználó
- A felhasználók azonosítása (név, jelszó ...)
- Különböző jogok/jogosultságok
- Eltérő környezet
- Naplózási lehetőség
- Felhasználói csoportok
- Home könyvtárak

## Multitasking működés

- File, program, process
- A CPU erőforrásainak időosztásos használata
- Kooperatív/Preemptív multitasking
- Interprocess kommunikáció (SIGNAL ...)
- Prioritások, process jogosultságok
- Előtérben/háttérben futó process (Daemon, service, NLM modul ...)

# Alapfogalmak

## Felhasználói interface-ek

- Karakteres, parancsorientált felület
  - Tradicionális
  - Szofisztikáltabb szolgáltatások érhetők el segítségével
- Grafikus felület
  - Open Look
  - MOTIF
  - CDE

## Felhasználó típusok

- Átlagfelhasználó
  - Igénybe veszi a szolgáltatásokat
- Privilegizált felhasználó
  - Üzemelteti a rendszert: rendszergazda, rendszeradminisztrátor, root
- Programozó
  - Az op. rendszert fejleszti

# File-system

**Milyen rendszer szerint tároljuk a fájlokat?**

File system:

- disk
- partíció
- elhelyezkedés

## Felhasználó felőli oldal

- Hierarchikus fa struktúrájú könyvtárrendszer
- File és könyvtár nevek, abszolút és relatív hivatkozások

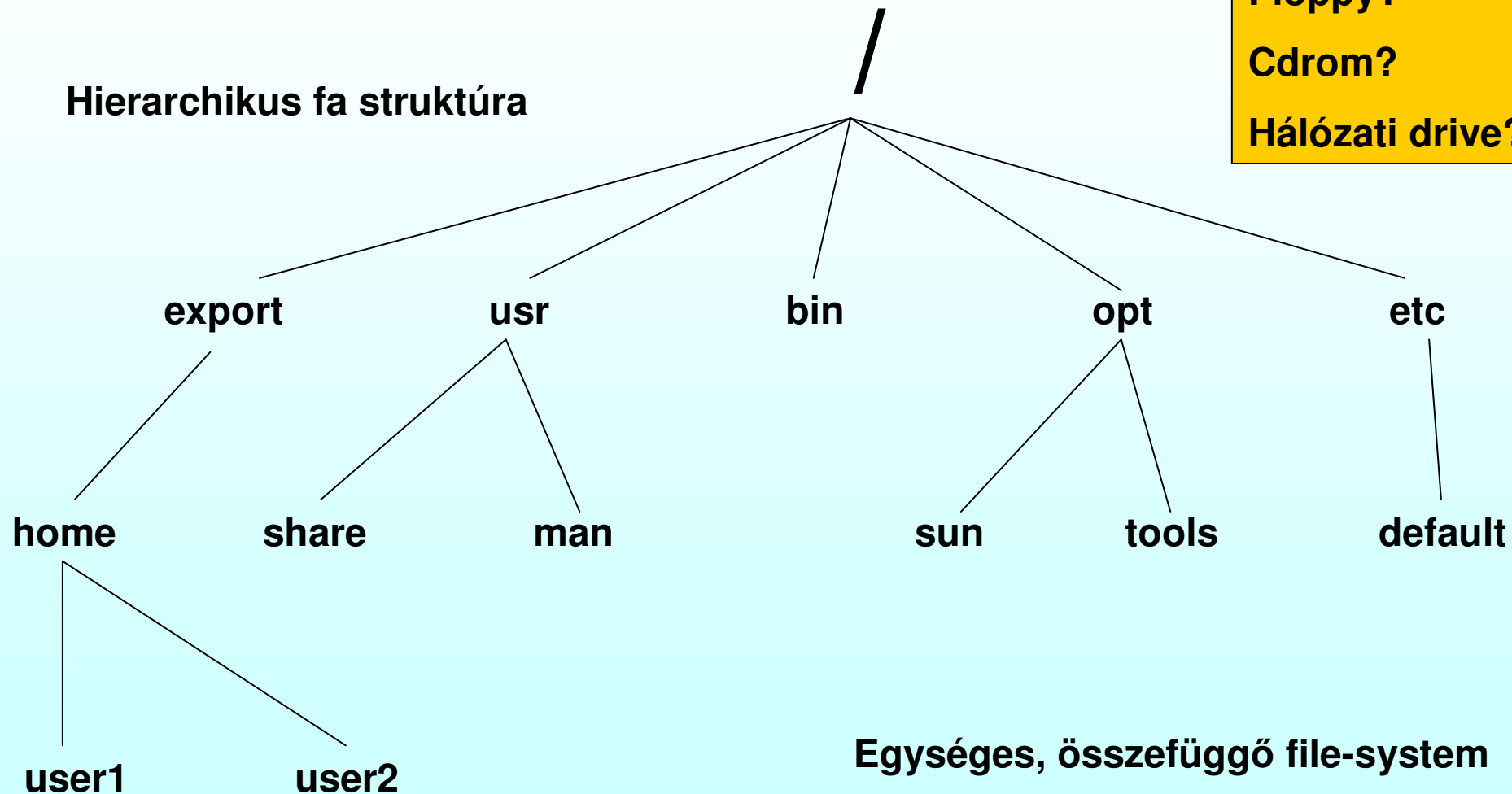
## Op. rendszer felőli oldal

- Táblázatok, szabályok
- UFS, FAT, NTFS, CDFS



# File-system

Hierarchikus fa struktúra



Egységes, összefüggő file-system

# Útvonalak, fájl nevek

## Útvonalak

- Hierarchikus fa strukrúra szerint – a csúcs: /
- Abszolút és relatív hivatkozások
- Szeparátor jel: /
- Spec. karakterek: . .. ~
- Példa: /export/home/users/hallgato111

## Fájl nevek, típusok

- 1-255 karakter hosszúságú string
- Spec. karakterek nem szerepelhetnek benne \* / ? ...
- Rejtett fájl: neve ponttal kezdődik
- Kezdő karakterként nem ajánlott a -
- Közöséges fájl: - , könyvtár: **d**, szimbolikus link: **l**

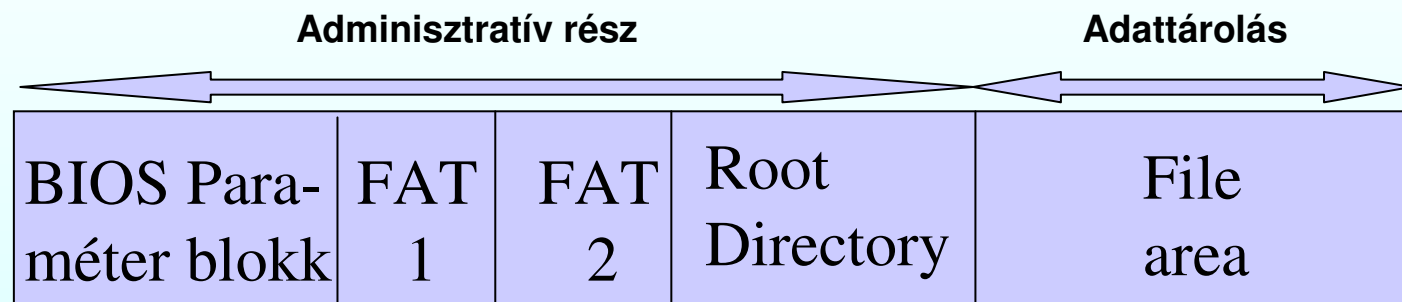
## Karakter használat

- Case sensitive! (A kis és nagybetűk különböznek.)
- Kisbetű orientált
- (A parancsok is kisbetűkkel írandók!)
- Nagybetűk használatának külön oka van (pl. shell változó)

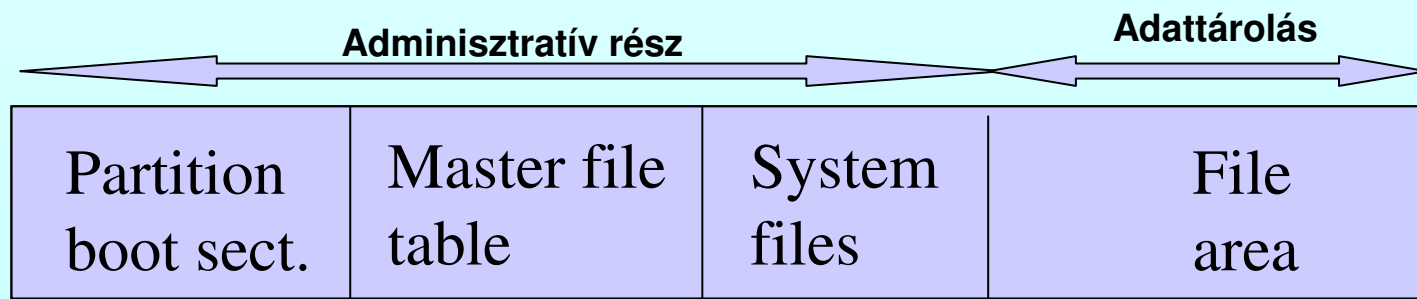
# File-system

## Példák különböző file-systemekre

- FAT



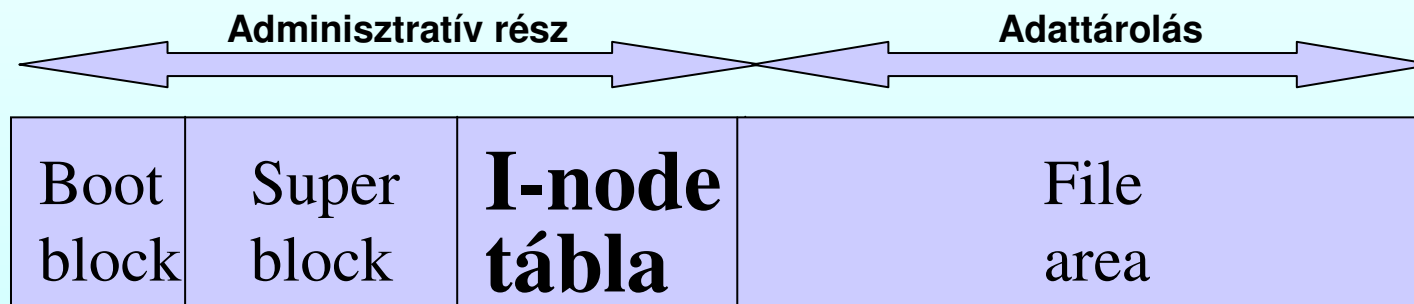
- NTFS



# File-system

## Példák különböző file-systemekre

- UFS
  - A UNIX i-node táblás file rendszere

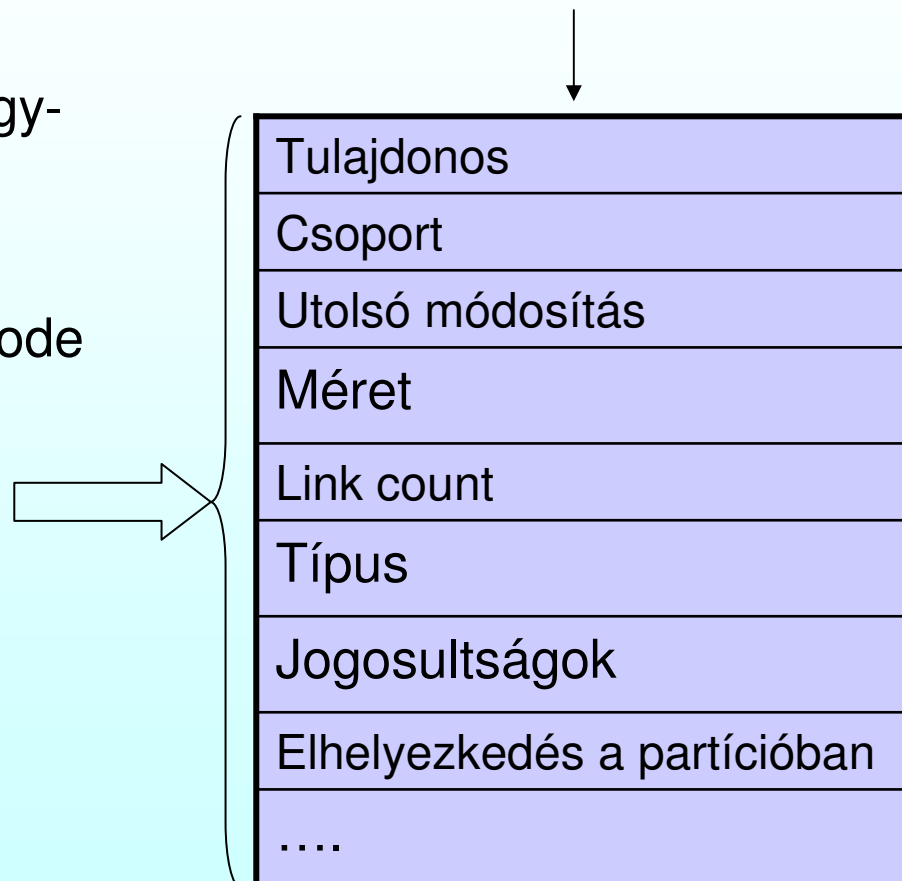


- CDFS
  - ISO9660 (read-only)

# I-node tábla

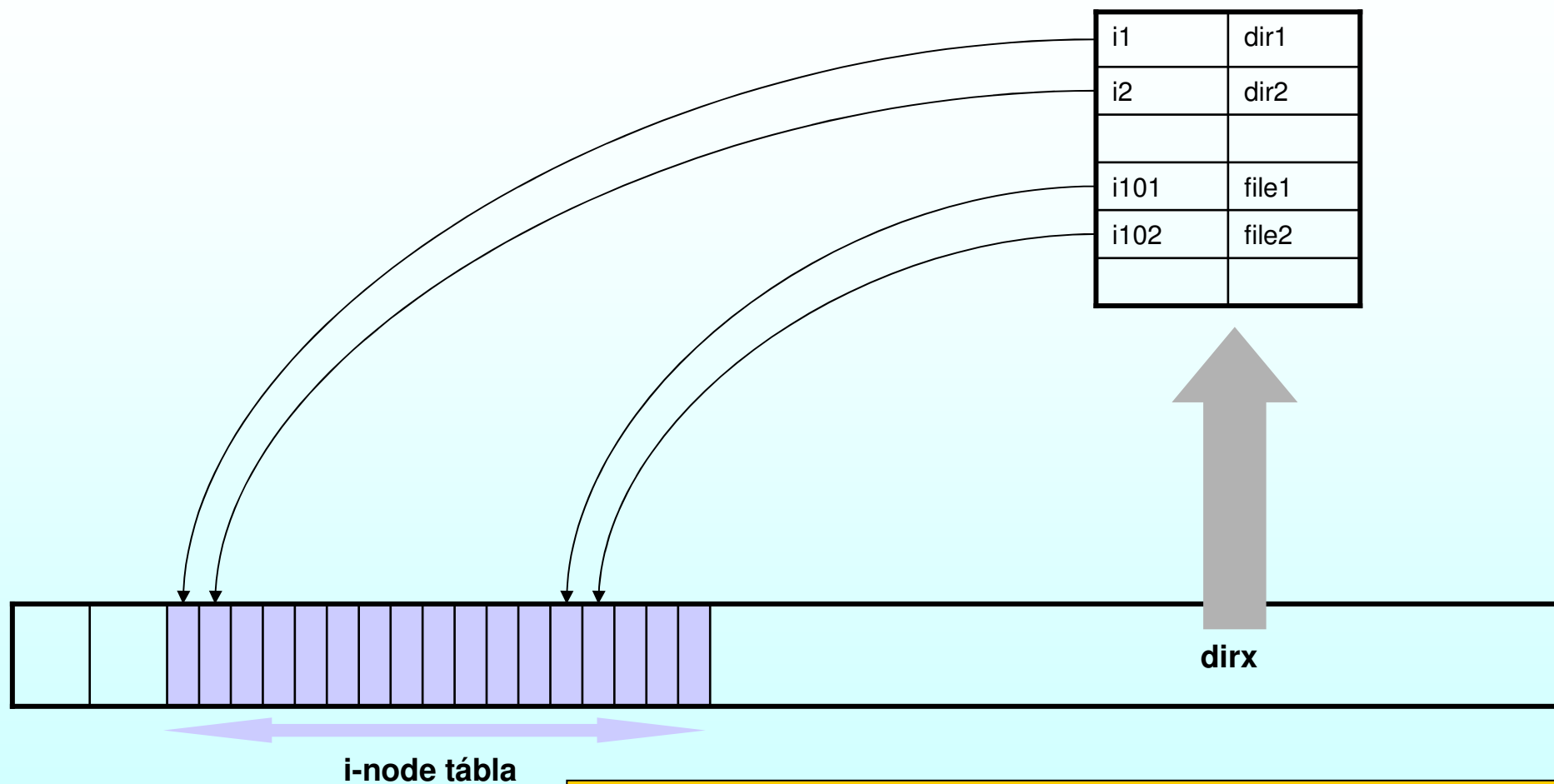
- Sorokból áll, egy-egy sor egy-egy fájlhoz tartozik\*
- A sorok (így a fájlok is) sorszámmal, azonosító számmal rendelkeznek (i-node szám)
- A sorokban egy-egy file attribútumai találhatók

Az i-node tábla egy sora



\* Ha a fájlhoz nem tartozik hard link

# I-node tábla



**A könyvtár speciális formátumú fájl (könyvtári fájl)!**

# Parancsok használata

## Parancs szintaxis

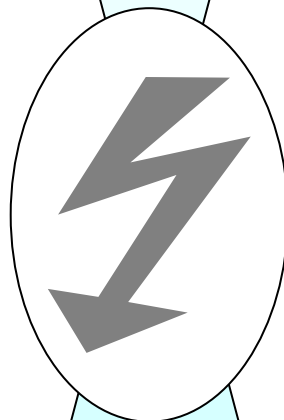
**A legfontosabb  
UNIX parancs: man**

- parancs `–k1 –k2 ... fájl1 fájl2...`
- példa: `ls –l –r –t /etc /opt`
- parancs `–k1k2 ... fájl1 fájl2...`
- példa: `ls –lrt /etc /opt`
- A parancs egyes elemeit „fehér karakterek” választják el egymástól
- Egy parancs több sorban: `;`
- Precedencia gondok: zárójelek
- Példa: `(Sleep 60; echo vege)&`
- (Hosszú parancsok a sor végére tett `\` jellel folytathatók a következő sorban.)

# Speciális karakterek

## Joker karakterek

- A parancsértelmező (shell) értlemezi
- Fájl nevekhez hasonlítjuk
- A parancs végrehajtása előtt kiértékelésre kerül
- **\* ? []**
- Tipikusan a file kezelő parancsok használják
- Példa: echo \*



## Reguláris kifejezések

- A parancs értelmezi
- A fájl tartalmához hasonlítjuk (A fájl mely soraihoz illeszkedik?)
- A parancs végzi az összehasonlítást
- **. \* ^ \$ [] \**
- Tipikusan a filter parancsok használják
- Példa: grep 'day\$' /usr/dict/words



# Process input/output

## A process kommunikációs csatornái

- Minden processhez három kommunikációs csatorna tartozik
  - 0 – stdin default: keyboard
  - 1 – stdout default: display
  - 2 – stderr default: display
- A default értékek megváltoztathatók!

## Átirányítás (redirection)

- Jelek: `<` `>` `<<` `>>`
- Destruktív: az eredeti tartalom felül lesz írva
- Nondestruktív: az eredeti tartalomhoz hozzá lesz fűzve
- Példák
  - `mailx < file`
  - `cat /etc/passwd /etc/shadow 1> újpasswd 2>hibafile`
  - `cat < /dev/term/a >/dev/term/b`

## Csővek használata (piping)

- Jel: `|`
- A process bemenete egy másik process kimenete
- A kommunikáció egy átmeneti memóriaterületen történik - FIFO
- Példák
  - `who|wc -l`
  - `ls -l|more`

# Linkek (láncok)

Egy objektumra több néven történő hivatkozás lehetséges

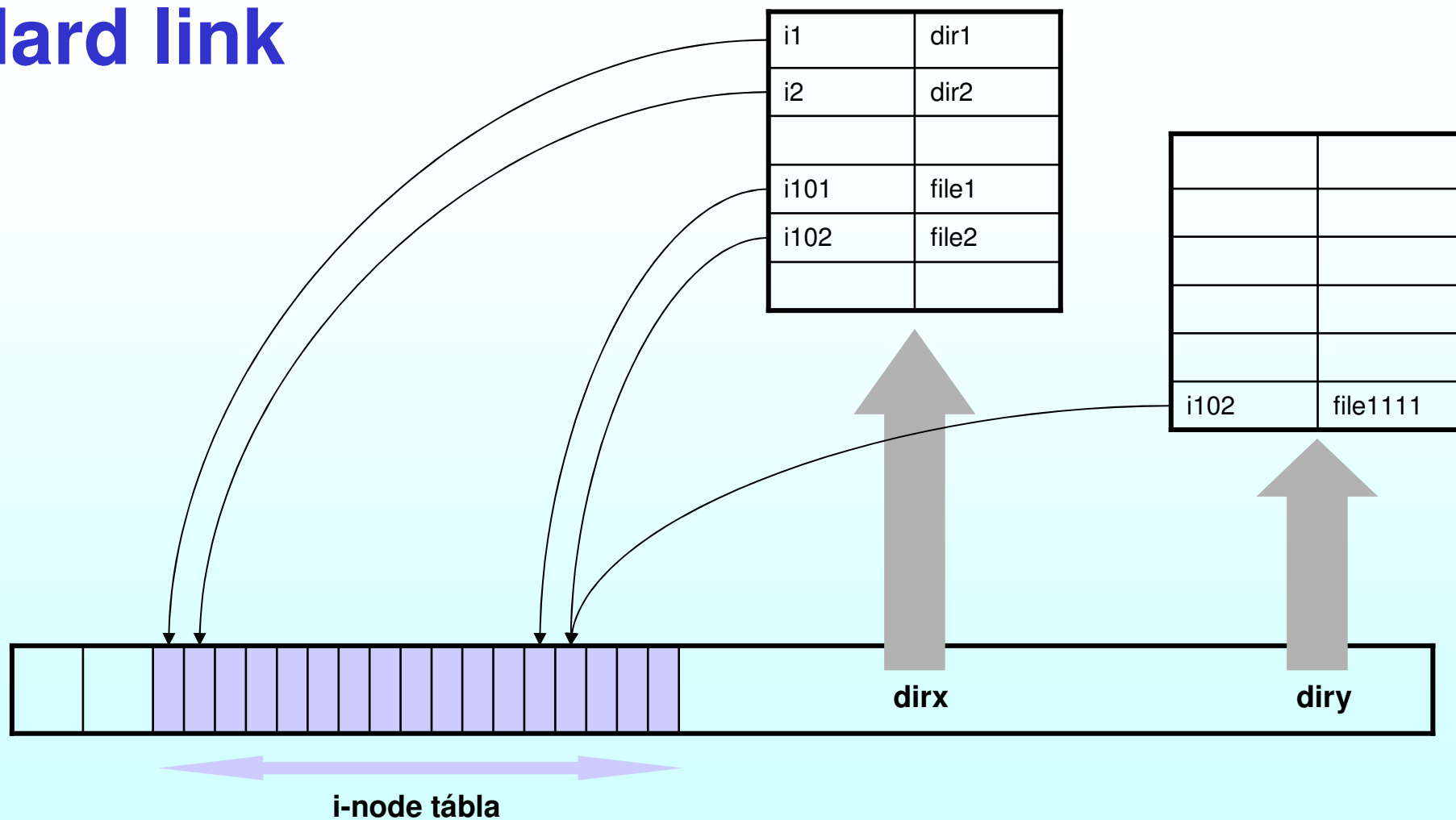
- Hard link

- Az i-node táblabeli sorszám azonosságán alapul a kapcsolat
- A linkelt fájlok teljesen egyenrangúak
- Nem kellőképpen rugalmas
- Tradicionális megoldás

- Szimbolikus (szoft) link

- Speciális file típus (jele: l)
- A fájl tartalma egy útvonal
- A hivatkozás áttevődik a tartalmazott útvonalra
- Rugalmasabb, mint a hard link

# Hard link



A file2 és a file1111 egymás hard linkjei.