



LINUX





UNIX történet

Elozmény: 1965 Multics

- **1969 Unix (Kernighen, Thompson)**
- **1973 Unix C nyelven (Ritchie)**
- **1980 UNIX (lényegében a mai forma)**
 - AT&T - System V
 - Microsoft - Xenix
 - Berkeley egyetem - BSD Unix
- **1991 LINUX (Linus Torvalds)**



LINUX történet (kernel !!!)

- **Elozmény: 1985 intel 386 DX**
 - 32 bit, multiprogramozás, virtuális
- **1991 Linux 0.01**
 - *folyamat és memóriakezelés*
- **1994 Linux 1.0**
 - *hálózatkezelés, hálózati protokollok*
 - *új perifériák (CD, egér, hangkártya)*
- **1995 Linux 1.1** ✍
- **1996 Linux 2.0**
 - *fájlkezelés, javított memóriakezelés*
 - *multiplatform (Alpha, PowerPC, MIPS stb.)*



GNU projekt

- **freeware, de NEM public domain (tehát a copyright megmarad!)**
- **a GPL alatt készülő programokból senki nem húzhat anyagi hasznot, a használatban, fejlesztésben senki sem korlátozható (forráskódban terjeszthető)**

www.gnu.org

www.linux.org



Linux változatok

- **kernel** (rendszermag)
 - monolitikus, egységes, GNU
- **system** (rendszer)
 - C fordító, hálózat menedzselés, egyéb shell-ek, X-Window, stb.
- **distribution** (szoftver csomag)
 - telepítő segédprogramok
 - hasznos alkalmazások: StarOffice, Web
 - pl. Slackware, Red Hat, Debian



Felhasználói felület

- **Hagyományosan karakteres**
 - Bourne (Again) Shell
 - C shell
 - Korn shell
- **Indítás elotérben, illetve háttérben (&)**
- **Standard *input, output, error***
- **Átirányítási lehetőség - Pipe**
- **Shell script (kb. DOS batch fájl)**
- **X-Window**



Állománykezelés

- Hierarchikus katalógus szerkezet
- Minden fájl (eszközök is ! - *dev*)
- Kvázi szabványos elnevezések

/	bin	(végrehajtó rendszerállományok)
	etc	(jelszavak, csoportok)
	dev	(konzol, nyomtató)
	lib	(rendszerkönyvtárak)
	tmp	(ideiglenes állományok)
	usr	(alkalmazások)
	user	(felhasználók katalógusai)

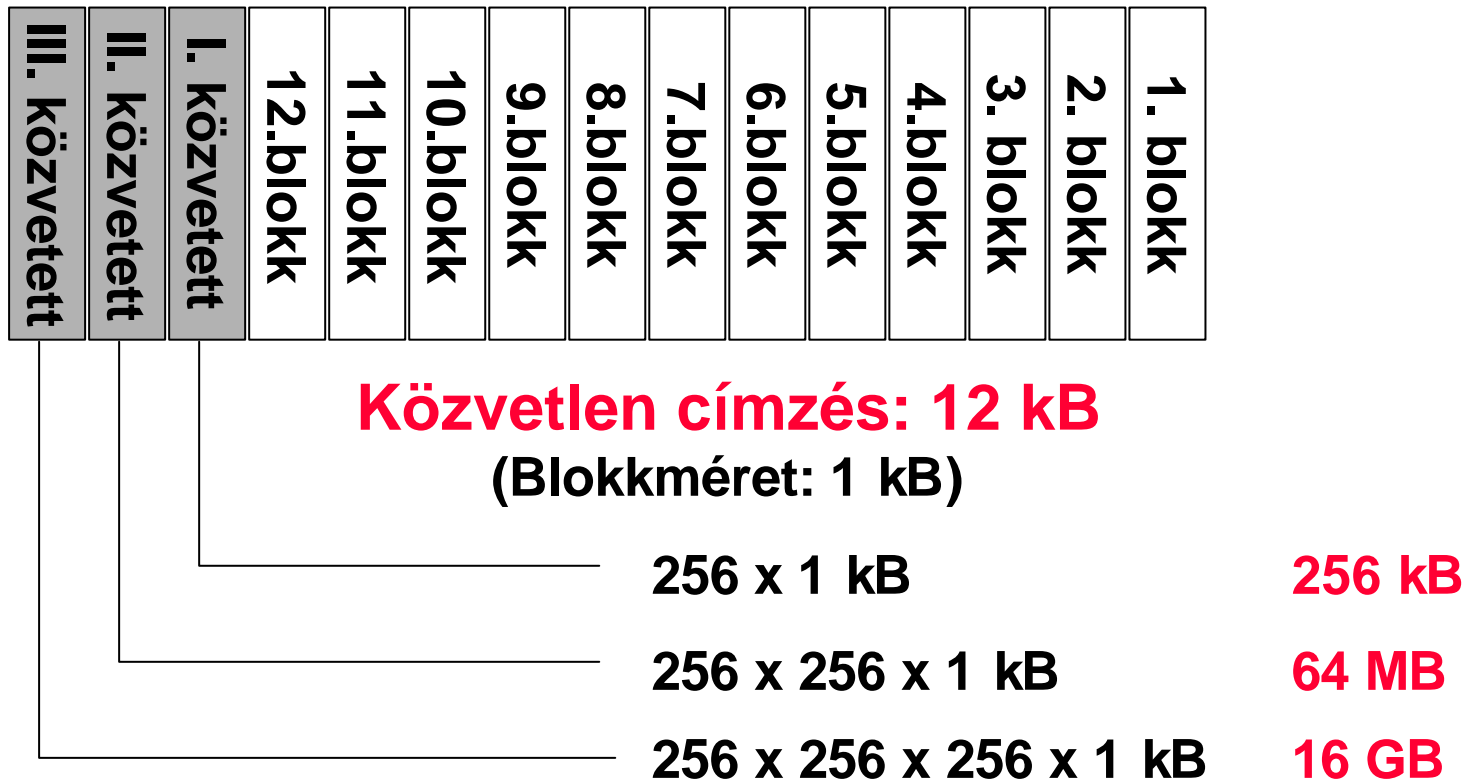


Katalógus - INODE kapcsolat

- **Katalógus tartalma**
 - Fájlnev hossza
 - Fájlnev
 - Inode mutató (4 byte)
- **INODE tartalma**
 - Típus (fájl, katalógus, pipe, fájlrendszer)
 - Tulajdonos (owner)
 - Jogosultságok (ld. később)
 - Méret
 - Létrehozás, utolsó módosítás időpontja
 - Hány katalógus bejegyzés hivatkozik rá ?
 - Index tábla



Indexelt fájl elhelyezés





Fájrendszer

- **Probléma** - Sok eltéő rendszer létezik
- **Megoldás** - Virtual File System (VFS)
 - Objektum orientált szemlélet
 - Objektumok: File, Inode, File system
 - Az objektumok mutatót tartalmaznak
 - az adatokra
 - a függvényekre (szabványos felület, a konkrét megvalósítástól függetlenül)
- **EXT2FS** - lemezes fájlrendszer
- **PROC** - rendszerstatisztikák fájl-jelmezben



Az objektumok függvényei

- **File system objektum**

Visszaadja a hivatkozott állományt leíró INODE címet (az FS címmel együtt)

- **Inode objektum**

Jogok, jellemzők, fizikai elhelyezkedés, a fájllal végezhető műveletek listája

- **Fájl objektum**

Folyamathoz tartozik

File mutató, Megnyitási mód, Használati gyakoriság



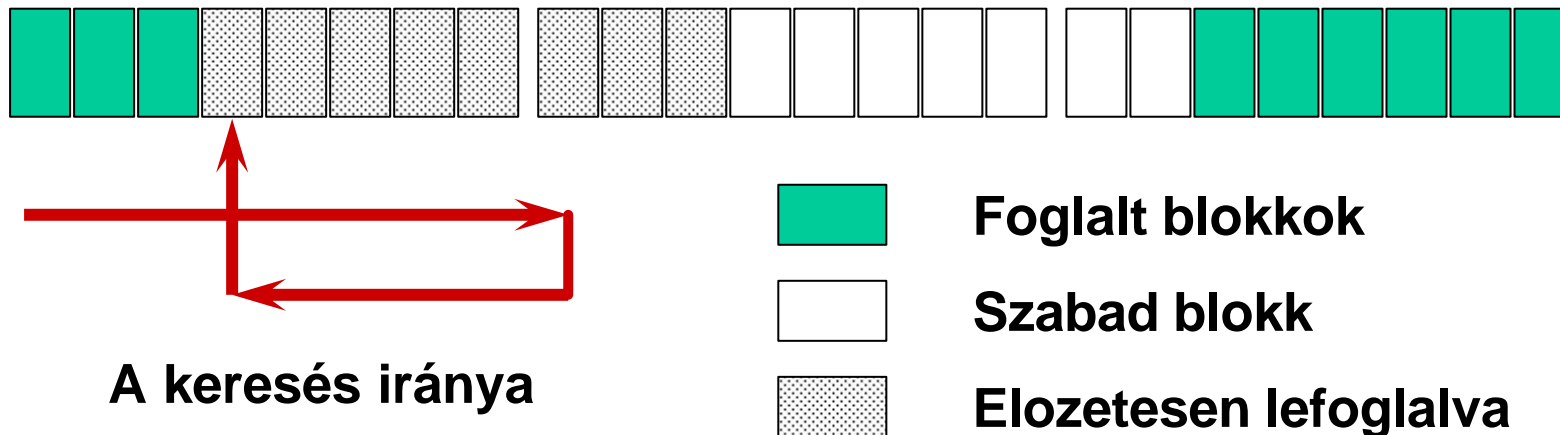
EXT2FS fájlrendszer

- **Ellentmondás:**
 - Jó helykihasználás -> Kis blokkméret
 - Gyors elérés -> Nagy blokkméret
- **Megoldás:**
 - Kis blokkméret alkalmazása (1-2-4-8 kB)
 - Blokkcsoportok, lehetőleg folyonos elhelyezés



Helyfoglalás EXT2FS alatt

- A keresés az első üres blokkcsoportnál indul (ha van ilyen)
- Visszafelé folytatódik a jó helykihasználás miatt





Folyamatkezelés, szinkronizálás

- **Probléma**

A kernel folyamatok gyakran kritikus szekciót futtatnak, ennek kezelése adminisztrációt igényel (ld. szemafor)

- **Megoldás**

- rendszerhívás, hardver megszakítás

- A kernel folyamat non-preemptívek

- kivétel (pl. a kernel által okozott laphiba)

- Nem szabad virtuális memóriát használni



CPU ütemezés

- **Interaktív folyamatok - időosztás**
 - cél: igazságozan bánni mind a CPU, mind a periféria igényes folyamatokkal
- **Valós idejű folyamatok - prioritás**
 - feltétlen elsőbbség az interaktív folyamatokkal szemben
- **Több processzor (SMP)**
 - a 2.0-nál nem optimális, kernel folyamat mindig csak egy processzoron fut



Idoosztásos ütemezés

- a folyamatok meghatározott kredittel rendelkeznek
- az ütemező az éppen futó folyamat kreditjét futásakor eggyel csökkenti
- Mindig az a folyamat futhat, amelynek a legtöbb pontja van
- Ha egy folyamat pontszáma eléri a nullát, a rendszer felfüggeszti
- Ha nincs futóképes folyamat, a kredit pont újra számolódik:

$$\text{Új kredit} = \frac{\text{Régi kredit}}{2} + \text{Prioritás}$$



Valós idejű ütemezés

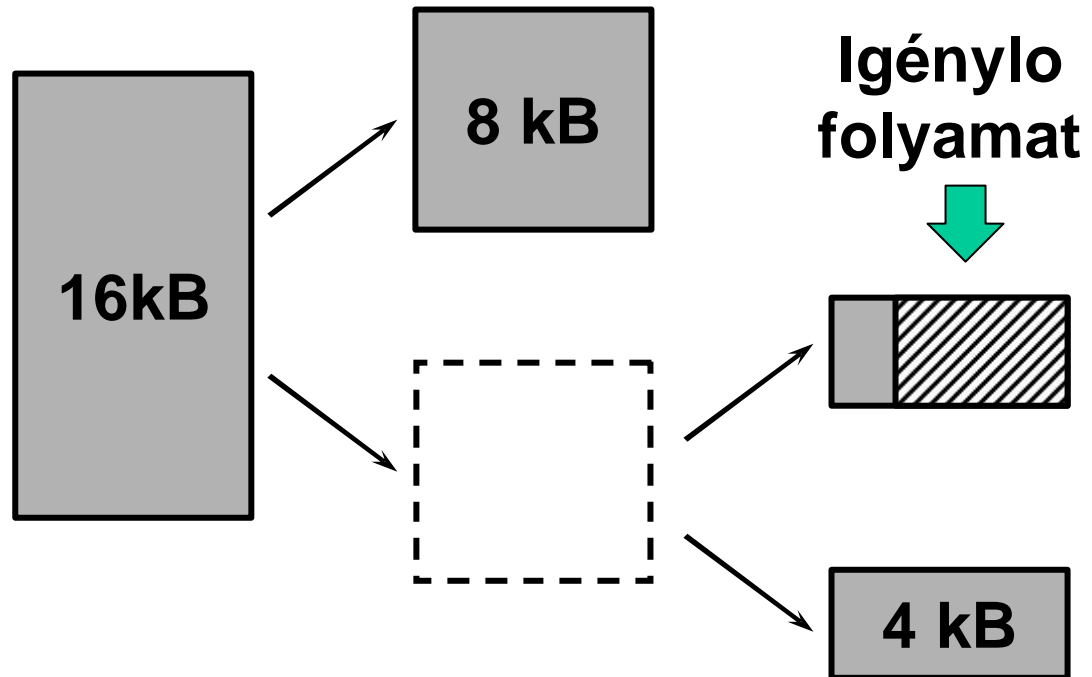
- **Két várakozási sor**
 - Mindkettőnek prioritása van az időosztásos sorral szemben
 - A folyamatok érkezési sorrendben futnak
 - Az egyik sor preemptív, RR jellegű
 - A másik sor non-preemptív, FCFS jellegű



Fizikai memóriakezelés

- **Buffer cache**
- **Page cache**
- **Virtuális memória**

- **Buddy heap**



- A különböző blokkméretek külön listán
- A szomszédos, üres blokkok összevonva



Virtuális memória kezelés

- **Igény szerinti, szegmentált lapozás**
 - egyéni régiók: a változások csak kilapozáskor mentodnek
 - megosztott régiók: a változás azonnal érvényesül
- **Lapcsere**
 - módosított „második esély”
 - a lapoknak „életkora van”
 - az ütemező periodikusan csökkenti
 - a laphoz fordulásakor no az életkor
 - azt a lapot cserélik, amelyik a „legfiatalabb”



Biztonság

- **Jelszavas védelem**
- **Hozzáférési jogok**
 - **jogok: read (r), write (w), execute (e)**
 - **szereplők: tulajdonos (u), group(g), others(o)**
 - **speciális esetek:**
 - **root - mindenhez van joga**
 - **setuid - a folyamat elindítója idolegesen átveheti a tulajdonos jogait**

uid	-rwx (user)	gid	-rwx (group)	-rwx (others)	setuid
------------	------------------------	------------	-------------------------	--------------------------	---------------



Következik ...

- **Dr. Orczán Zsolt**
 - **A Linux, ahogy a rendszergazda látja**
- **Brucker Péter**
 - **A Linux az Internet szolgáltató szemével**