

## 4. Tétel

Fájlrendszerekkel kapcsolatos elvárások; a FAT fájlrendszer felépítése, szabad helyek kezelése, blokkok fejléce, adatterületek tartalma könyvtárak és fájlok esetében, példa egy fájl beolvasására; az UFS fájlrendszer felépítése, szabad helyek kezelése, blokkok fejléce, példa egy fájl beolvasására; fájlrendszer felcsatolása („mount”-olás) Unix rendszerben; az /etc/fstab és /etc/mtab fájlok; a /dev könyvtár; loopback device, soft és hard linkek; FUSE, VFS, NFS;

### Fájlrendszerekkel kapcsolatos elvárások:

A számítástechnika egy fájlrendszer alatt a számítógépes fájlok tárolásának és rendszerezésének a módszerét érti, ide értve a tárolt adatokhoz való hozzáférést és az adatok egyszerű megtalálását is.

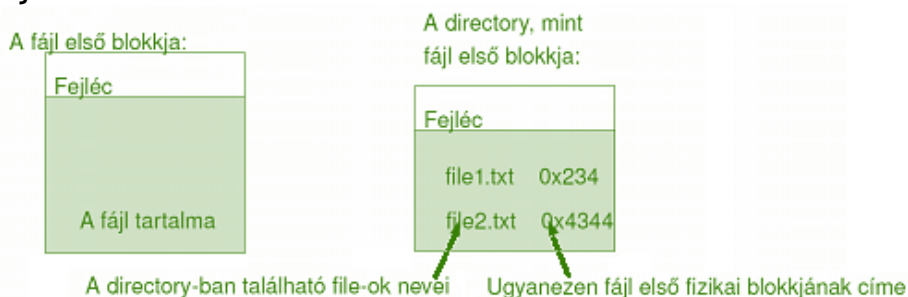
Elsődleges elvárások:	Másodlagos elvárások:
<ul style="list-style-type: none"> <li><b>namespace:</b> nevekkel lehessen hivatkozni az adatokra (fájl nevek)</li> <li><b>szabad terület és foglalt terület managementje,</b> azaz tudjuk, hogy mere vannak az adatok a tárolón</li> <li><b>transzparencia:</b> minden hardware-t ugyanúgy lehessen kezelni azaz gyártótól mentes legyen egy hardware kezelése.</li> </ul>	<ul style="list-style-type: none"> <li><b>tools:</b> eszközök legyenek hozzá formázásra</li> <li><b>robosztusság:</b> ha elmegy a táp ne korruptálódjon azaz ne boruljon a sorrendiség a nevek ne romolhassanak el</li> <li><b>Hozzáférés és jogosultság szabályozása</b></li> <li><b>fragmentation policy:</b> az adatok fregmentációjának csökkentése</li> <li><b>cache:</b> legalább olvasásra</li> </ul>

a FAT fájlrendszer felépítése, szabad helyek kezelése, blokkok fejléce, adatterületek tartalma könyvtárak és fájlok esetében, példa egy fájl beolvasására:

A FAT fájlrendszer a Windows NT operációs rendszer által támogatott fájlrendszerek legegyszerűbbike, legfőbb sajátossága a fájlkiosztási tábla FAT, ami a kötet legfelső szintjén elhelyezkedő adattáblázat. A kötet védelme érdekében a fájlkiosztási táblának két példánya van, ha az egyik megsérül a másik segítségül hívható. FAT-ra formázott kötetek alkotóegységei a klaszterek (clusters). A klaszter méretét a kötet mérete határozza meg 512 bájt és 64 KB közötti lehet. Megkülönböztethetünk FAT12, FAT16 és FAT32-t. A FAT-ek után szereplő számok a bitekre utalnak amelyekkel azonosítja a klasztereket, így minél nagyobb bit annál több klasztert tud azonosítani és így felhasználni. Manapság már csak a 32-est használják. A FAT tábla tartalmazza azokat az adatokat, hogy egy fájl tartalmának végigolvasásához melyik klaszter után melyik klasztert kell olvasni.

**Szabad helyek kezelése:** A fájl egy folytonos fizikai blokk sorozatot foglal el. Hozzáférés egyszerű és gyors HDD esetén (a sáv folytonosan olvasható a forgó lemezzel). Növekvő méretű fájloknak a helyfoglalás problémás. Milyen méretű szabad helyet allokálunk? Új fájlok számára megfelelő szabad hely megtalálása nehéz, külső tördelődés lép fel. Fájl törlése után a méretének megfelelő számú blokk felszabadul. Erre a helyre kisebb vagy egyenlő méretű fájl írható. Ugyan azokat az algoritmusokat használhatjuk, mint a memória foglalás során (First fit, Next fit, Best fit, Worst fit). Növekvő fájlok esetén a Best fit allokációs stratégia különösen veszélyes. A fájl nagyobb szabad helyre másolása (erőforrás igényes).

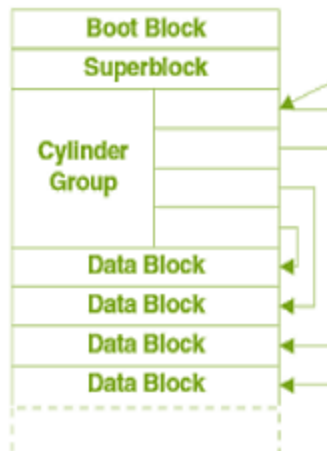
### Blokkok fejléce:



az UFS fájlrendszer felépítése, szabad helyek kezelése, blokkok fejléce, példa egy fájl beolvasására:

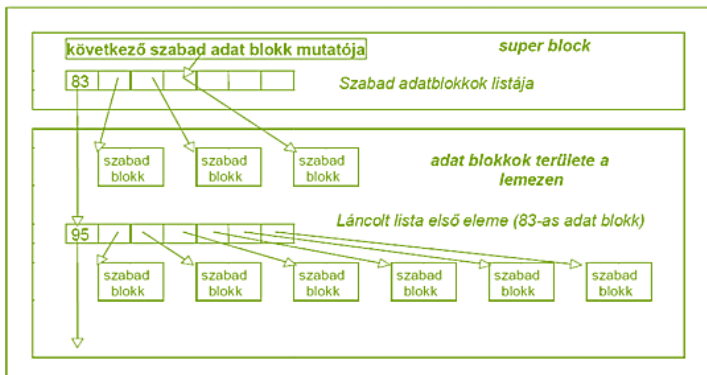
Az UFS fájlrendszert 1982-ben vezették be. Sok Unix és Unix-alapú operációs rendszer használta. A maximális fájl méret 273 bájt (8ZiB) lehet. A maximális fájl név hosszúság 255 bájt.

az UFS fájlrendszer felépítése:



1. **Boot szektor:** a partíció elején található pár blokk, amit a fájlrendszerből külön kell inicializálni.
2. **superblokk:** ez tartalmazza a „magic number” és más alapvető számokat, amik leírják a fájlrendszer geometriai, statisztikai és viselkedésbeli finombeállításait.
3. **Cilinder csoportok:** Minden cilinder csoport tartalmazza a superblokk biztonsági másolatát a cilinder csoport fejrészét i-node-ok számát (mindegyik tartalmazza a fájl attribútumait) adatblokkok számát.
4. **i-node-ok:** fájl típusa (speciális, adat, könyvtár bejegyzés, PIPE/FIFO) a linkek számát, amik az adott inode-ra mutatnak az eszköz és az i-node ID-t fájl elérési jogosultságokat módosító flageket a fájlhoz tartozó adatblokkok elérési információját.

Szabad helyek kezelése: Két tárolási korlát figyelhető meg:



1. **fájlrendszer méretére vonatkozó** (fizikai méretnél nem lehet nagyobb mint a logikai)
2. **A szabad adatblokkok tárolása** szabad adatblokkokra mutat tömbök láncolt listájával történik.

Példa egy fájl beolvasására:

1. fopen (nyitás módja , fájl neve)
2. nyitáskor az eltolási érték 0
3. jogok ellenőrzése megtörténik
4. új bejegyzés a process FDT-ben
5. read(olvasandó byte szám)
6. chown uid megváltoztatása
7. utimes időpontok módosítása

## fájlrendszer felcsatolása („mount”-olás) Unix rendszerben:

Használatba vétel előtt a fájlrendszert csatolni kell a rendszerünkhöz. Ehhez be kell olvasni a szuperblokkot és a fájlrendszer táblában egy új bejegyzést kell létrehozni. Csatolás után a fájlrendszer fájllai elérhetővé válnak.

az `/etc/fstab` és `/etc/mtab` fájlok:

**fstab**= file system table

Az `/etc/fstab` szerepe UNIX rendszerben, hogy az itt lefektetett szabályok megkönnyítik az állományrendszerek használatát. Megléte rendkívül fontos, mivel az `fsck` és a `mount` (tehát a boot során az automata `mount` is) ebből tájékozódik az elérhető állományrendszerekről. Az `fstab` kilistázza az elérhető lemez partíciókat és egyéb típusú file rendszereket az olyan adat forrásokat, amik nem lemez elhelyezkedésűek.

fs_specs	fs_file	fs_vfstype	fs_mntops
fizikailag hol van a kezelendő fájlrendszer	hová kerüljön a montolásra a fájlrendszer	állományrendszer típusa	mountolási beállításokat tárolja

**mtab**= mounted file system table. Az `mtab` file egy rendszer információs file. Kilistázza az összes jelenlegi mountolt file rendszert a beállítási opciókkal együtt.

a `/dev` könyvtár:

**dev**= device file, két részből áll az egyik azonosítja a kontrollert, a másik a tényleges eszközt és drivert. A `dev` könyvtár a speciális eszközfájlokat tartalmazza az összes eszköz számára. Az eszközfájlok a telepítéskor jönnek létre, illetve később a `/dev/MAKEDEV` szkript segítségével. A `/dev/MAKEDEV.local` egy szkript, melyet a rendszeradminisztrátor ír. példák:

<code>/dev/dsp</code>	<code>/dev/fd0</code>	<code>/dev/loop0</code>	<code>/dev/null</code>	<code>/dev/md0</code>	<code>/dev/pt0</code>
hangkártya és a hangot létrehozó program	első hajlékonylemez meghajtó	első loopback eszköz	ami ide kerül az megsemmisül	RAID eszköz számára	párhuzamos potra csatlakoztatott szalagos eszköz.

## loopback device, soft és hard linkek:

**Loopback device:** Egy olyan mechanizmus mely a file-okat úgy értelmezi, mintha igazi eszközök lennének. Az előnye ennek, hogy minden igazi lemezentárolt eszközt tudunk használni a loopback segítségével. KB mint egy image file.

**Soft link:** A soft link (symbolic link (symlink)) egy speciális fájltypust jelöl, amely valójában egy hivatkozás egy másik fájlra vagy könyvtárra. A soft linkek nem közvetlenül mutatnak adatra, hanem egy elérési útvonalat tartalmaznak, amelyből a rendszer képes egy hard linket (vagy egy másik soft linket) beazonosítani.

**A hard link:** rögzített hivatkozás egy hivatkozás vagy mutató egy adattároló eszközön elérhető adatra. A legtöbb fájlrendszerben az elnevezett fájlok hard linkek. Hard linkkel csak azonos fájlrendszerben létező adatra lehet hivatkozni.

## FUSE, VFS, NFS:

**FUSE:** A FUSE (Filesystem in Userspace) segítségével képesek vagyunk felhasználói térben filerendszert megvalósítani. A FUSE kommunikációsfelülete egyszerű, hatékony, biztonságos és emellett támogatja a szokásos filerendszer szemantikákat.

**VFS:** Virtual File System. Egy szoftverben megvalósított absztrakció, aminek lényege hogy filerendszerként tekintünk valamit, ami valójában klasszikus értelemben nem az (pl. FTP szerverek listája vagy egy .ZIP file tartalma), vagy pedig több fizikailag különböző filerendszer eléréséhez biztosítunk egy API-t, amit szintén szokás VFS-nek nevezni.

**NFS:** Network File System. A SUN által kitalált hálózati filerendszer, mely erősen támaszkodik az távoli hozzáférésre. Lehetővé teszi hálózaton keresztül adatok megosztását. (UNIX, más oprendszereknél nem elterjedt).