

Operációs rendszerek

Állományrendszerek

1.1

Pere László (pipas@linux.pte.hu)

PÉCSI TUDOMÁNYEGYETEM TERMÉSZETTUDOMÁNYI KAR
INFORMATIKA ÉS ÁLTALÁNOS TECHNIKA TANSZÉK



Állományrendszerek



A hosszútávú adattárolásnak a következő feltételeknek kell megfelelnie:

1. Nagy mennyiségű adat tárolását kell lehetővé tennie.
2. Az adatoknak a folyamat befejeződése után is elérhetőnek kell lennie.
3. Az adatoknak egyszerre több folyamat számára is hozzáférhetőnek kell lennie.

A szokásos megoldás az állományok használata háttértárakon.



Állományrendszer

Az állományok kezelése az operációs rendszer feladata. Az, hogy hogyan rendezzük el, milyen módon nevezzük, védjük és használjuk az állományokat, valamint az, hogy miképpen valósítjuk meg a vonatkozó programrészeket, az operációs rendszerek egyik központi kérdésköre.

Az operációs rendszereknek az állományokkal foglalkozó részeit állományrendszernek nevezzük.

A felhasználó szempontja

A felhasználó szempontjából az állomány egy absztrakciós rendszer, amely lehetővé teszi adatok tárolását és visszatöltését.

A felhasználó szempontjából az egyik legfontosabb szempont az állományok elnevezése, hiszen az állományokat nevük alapján különböztetjük meg egymástól.

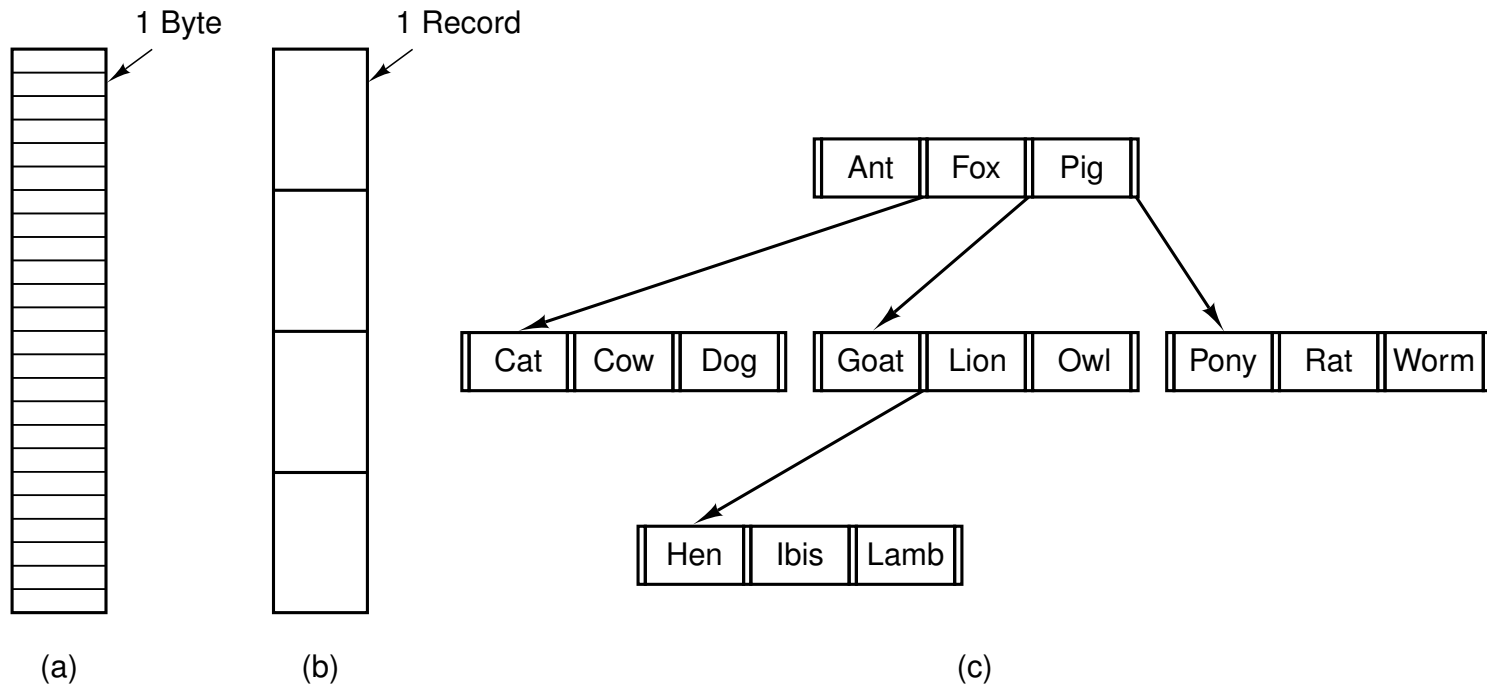
Az állományok elnevezése

- Az állománynevek maximális hossza a különböző rendszereken más-más lehet.
- Néhány rendszer különbséget tesz kis- és nagybetűk között, néhány rendszer pedig nem.
- Néhány operációs rendszer két részre osztja az állományneveket, mások pedig nem.
- Néhány rendszer esetében az állománynév végződés szigorú szabályok szerint, néhány más rendszer esetében pedig csak hagyományok alapján szabályozott.

Állományszerkezet

- Sok rendszer esetében az állományok belső struktúráját nem szabályozzuk. Ez biztosítja a legnagyobb rugalmasságot.
- A szabályozás első fokán rekordszerkezetet követelünk meg az állományokban.
- A harmadik szint a keresőfák használata, ahol könnyen találhatjuk meg az egyes elemeket. Mainframe rendszereken ma is használatos ez a módszer.

Állományszerkezet



1. ábra. Kötetlen, rekord és fa állományszerkezet

Könyvtárbejegyzések

Fejlettebb operációs rendszerekben más könyvtárbejegyzések is léteznek, nem csak egyszerű állományok és könyvtárak.

- regular files (szabályos állományok)
- directories (könyvtárak, mappák)
- character special files (karaktereszköz-meghajtó állományok)
- block special files (blokkeszköz-meghajtó állományok)
- soft links (közvetett hivatkozások)

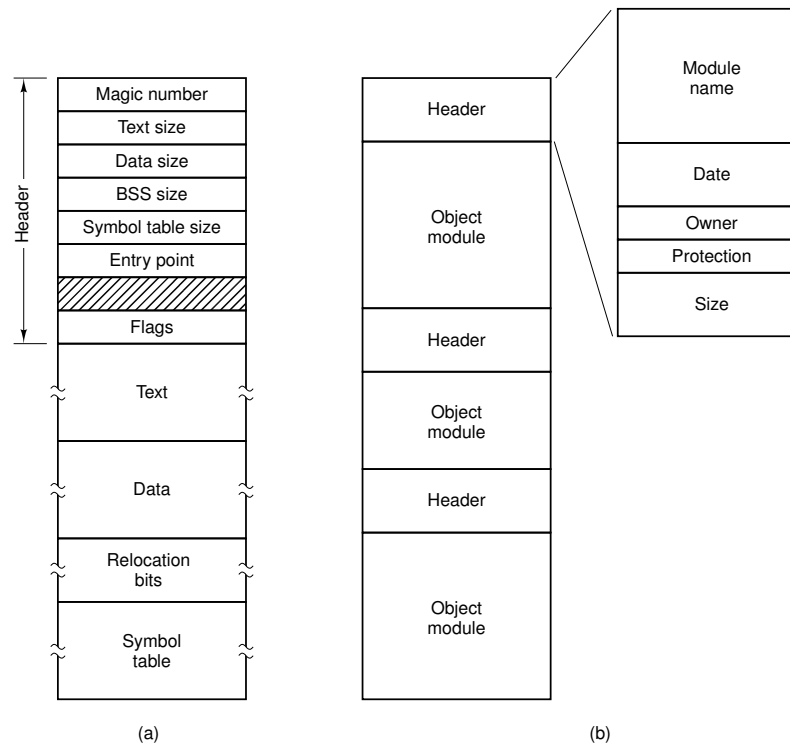
Állományok felosztása

Néhány régebbi operációs rendszer kétféle állományt különböztet meg:

- ASCII (szöveges) állományok.
- Bináris állományok típusukat az ún. *magic number* jelzi.

A modern operációs rendszerek nem tesznek különbséget e két típus között, de az alkalmazások és a programkönyvtárak sokszor igen.

Bináris állomány



2. ábra. Program (a) és arhívum (b) formátum

Állományhozzáférés

Régebbi rendszerek csak szekvenciális állományhozzáférést tettek lehetővé.

Újabb rendszerek címezhető (*random*) elérést is lehetővé tesznek, amely néhány feladat esetében elengedhetetlenül fontos.

Régebbi mainframe operációs rendszerekben már az állomány létrehozásakor meg kell adnunk, hogy szekvenciális vagy random elérést akarunk használni.

Attribútumok

Néhány lehetséges attribútum:

protection Ki, milyen módon használhatja az állományt.

password Az állományhoz tartozó jelszó.

creator A létrehozó személy azonosítója.

owner A jelenlegi tulajdonos.

read only Az állomány csak olvasható.

hidden Ne írassuk ki az állományok lekérdezésekor.

system Az állomány a rendszer része.

archive Szükséges -e az állományt archiválni.

Attribútumok (folyt.)

ascii ASCII vagy bináris állományról van-e szó.

random Lehetséges-e random access elérés az állományon?

temporary Ideiglenes-e az állomány?

lock Zárolt állapot.

record len Milyen hosszú rekordok találhatóak az állományban?

key pos A rekordokban található kulcs helye a rekordon belül.

key length A rekordokban található kulcs hossza.

Attribútumok (folyt.)

creation time A létrehozás időpontja.

time of access A legutolsó használat időpontja.

time of change A legutolsó módosítás időpontja.

current size Az aktuális méret.

maximum size A megengedett legnagyobb méret.

Állományműveletek

- create** Az állomány létrehozása 0 hosszúsággal. A létrehozáskor bizonyos attribútumok beállítódnak (pl. létrehozó, tulajdonos stb.).
- delete** Az állomány törlése, a hozzá tartozó terület felszabadítása. (*nem kuka!*)
- open** Az állományt a használat előtt meg kell nyitni. Sok rendszerben az állomány megnyitásakor történik a jogosultságok kezelése.
Általában lehetőség van arra, hogy a megnyitást és létrehozást egy lépésben végezzük el.

Állományműveletek

- close** Az állomány lezárása, amelyet a használat után azért kell elvégeznünk, hogy a nyitott állományok által foglalt erőforrásokat felszabadítsuk. Sok rendszer a takarékoság kieroszakolása érdekében korlátozza a nyitva tartható állományok számát.
- read** Olvasás, amely általában az aktuális pozíciótól kezdődő adatbájtokat másolja a memóriába adott méretben.
- write** Írás, amely általában az aktuális pozíciótól kezdődően történik. Ha az írás részben vagy egészében az állomány vége utáni területre vonatkozik, az állomány mérete növekszik.

Állományműveletek

append Módosított write, amely mindig az állomány végére történik.

seek Az aktuális pozíció módosítására szolgáló művelet, amelyre random elérésű állományok kezelésekor lehet szükség.

get attr Az attribútumok lekérdezése.

set attr Az attribútumok megváltoztatása.

rename Az állomány nevének megváltoztatása.

Állományműveletek

A következő műveletek valójában könyvtárműveletek, de már itt bemutatjuk őket:

link Új név létrehozása az állomány számára.

unlink Az állományhoz tartozó egyik név törlése. Ha ez a név volt az utolsó, az állomány megsemmisül, az általa foglalt terület felszabadul.

Memóriába ágyazott állományok

A modern operációs rendszerek lehetővé teszik, hogy az alkalmazás kérje az állomány memóriába ágyazását, virtuális memóriához hasonló kezelését.

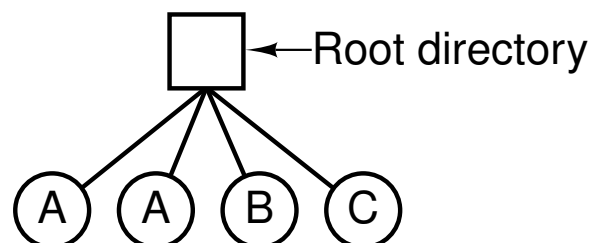
Ezt a módszert használva az operációs rendszer lehetővé teszi, hogy egyszerű memóriaműveletekkel kezeljük az állományokat.

Könyvtárak

Az állományok elrendezéséhez a legtöbb rendszerben könyvtárak (directories, folders) állnak a rendelkezésünkre. Az állományrendszer szerkezete a következők egyike lehet:

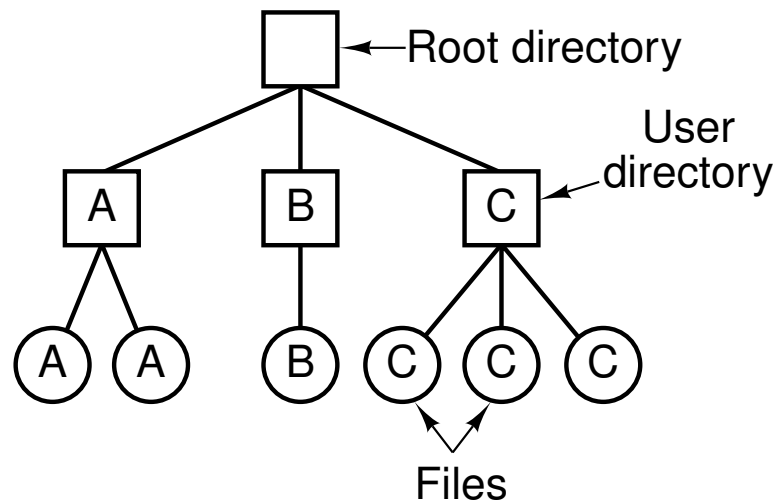
1. Single-level directory systems: Minden egy helyen található, a felhasználók állományai összekeverednek.
2. Two level directory system: Minden felhasználónak van egy könyvtára.
3. Hierarchical directory systems: mindenki ismeri...

Könyvtárszerkezet



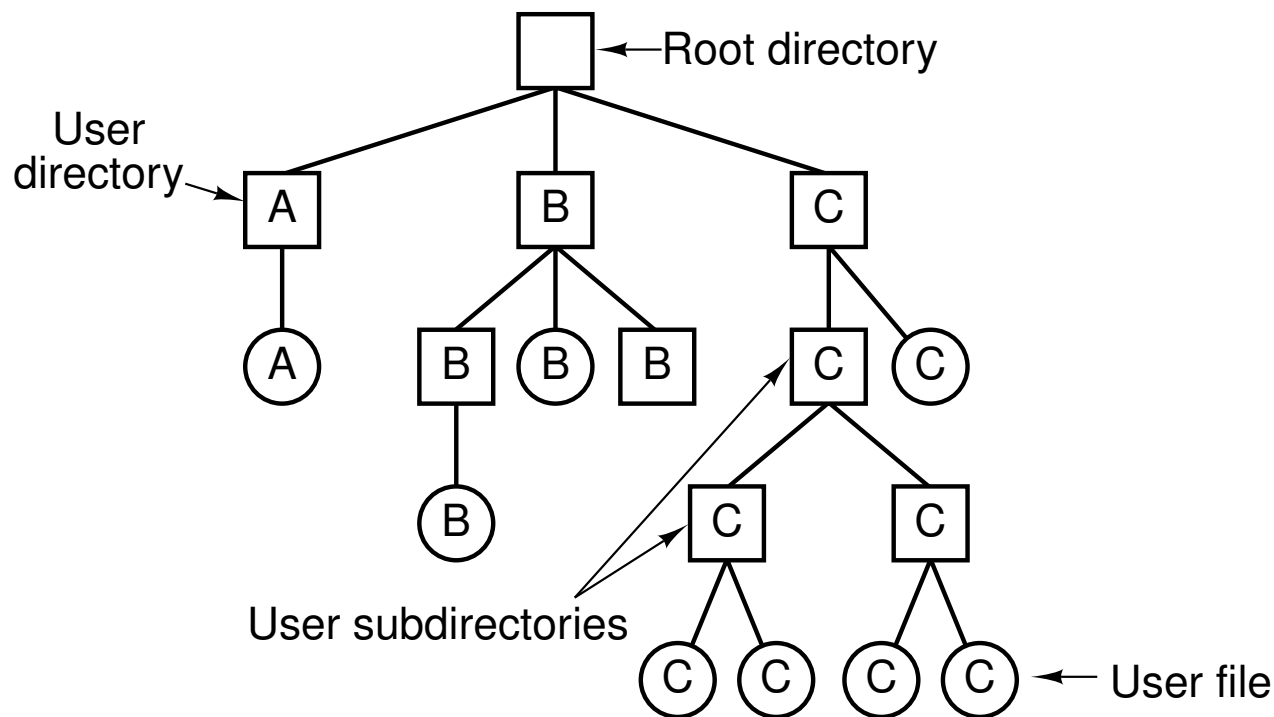
3. ábra. Egyszintű könyvtárszerkezet három felhasználóval

Könyvtárszerkezet



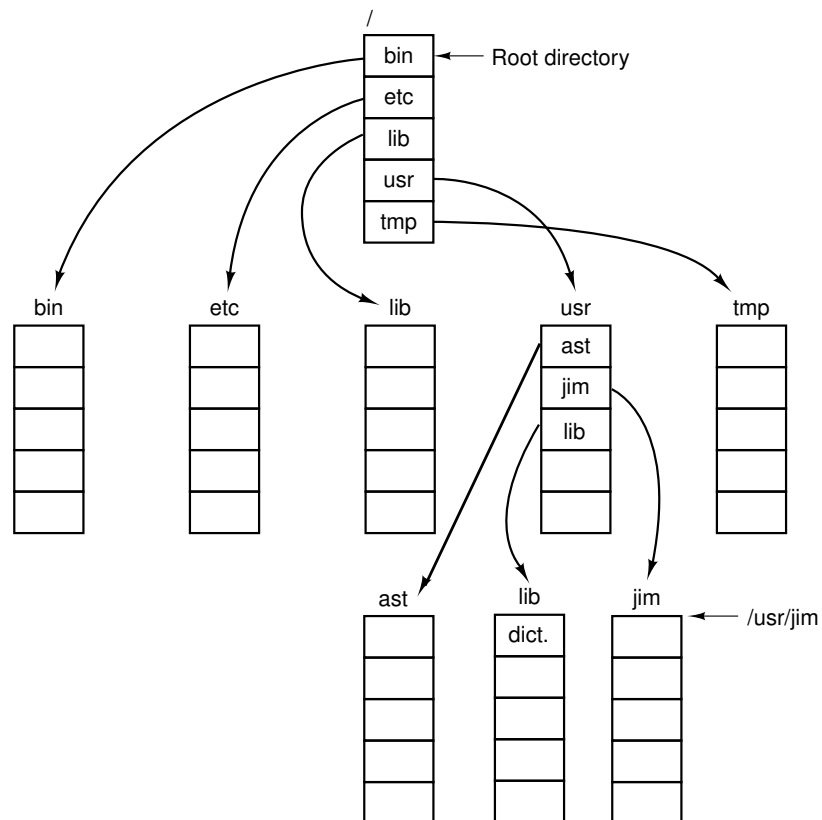
4. ábra. Kétszintű könyvtárszerkezet

Könyvtárszerkezet



5. ábra. Hierarchikus könyvtárszerkezet

Könyvtárak felépítése



6. ábra. A könyvtárfa listákból épül fel

Path names



A következő fogalmakat mindenkinek pontosan kell ismernie annak érdekében, hogy harmadéves lehessen!

- absolute path name
- relative path name
- working directory (current directory)

Az általunk használt nézőpont a konkatenációra alapul, amely jól leírja a könyvtárakra mutató szimbolikus linkek viselkedését!



Műveletek könyvtárakon

create A könyvtár létrehozása. A létrehozott könyvtár általában üres, kivéve a `.` és `..` állományokat, amelyeket az operációs rendszer vagy a segédprogram hoz létre a könyvtár elkészítésekor.

delete A könyvtár törlése. Törölni általában csak üres könyvtárat lehet (a `.` és `..` bejegyzések nem számítanak).

opendir A könyvtárakat olvasás előtt meg kell nyitni.

closedir Amikor a könyvtárat már nem kívánjuk olvasni, le kell zárni, hogy a nyitáskor lefoglalt erőforrásokat felszabadítsuk.

Műveletek könyvtárakon

readdir A könyvtárak megnyitás után olvashatóak. Az olvasás során a könyvtárban található könyvtárbejegyzéseket olvassuk szekvenciális módon.

rename A könyvtár átnevezése.

link Valamiért itt tárgyalja a linkelést a könyv, de tudnunk kell, hogy könyvtárak esetében ez a függvény általában nem használható, csak állományokra.

unlink Az állományok esetében használt technikával megszüntetjük a könyvtár egyik nevét. Mivel a könyvtáraknak általában csak egy neve lehet, ez a művelet a tulajdonképpeni törlés, amely kizárólag üres állományokon végezhető el. Ld. delete

Implementáció

A következőkben az implementáció oldaláról vizsgáljuk meg az állományrendszereket.

Felépítés



1. A merevlemez elején található az MBR (master boot record)
2. Utána következik a partíciós tábla 4 bejegyzésnek fenntartott helyel.
3. Következnek a partíciók, minden partíciónak az elején a boot blokkal.



A rendszerindítás

A rendszerindítás menete (akár így is csinálhatnánk):

1. A BIOS betölti az MBR-ben található kódot és elindítja.
2. A betöltő megvizsgálja melyik partícióbejegyzés az aktív.
3. A betöltő betölti és elindítja az aktív partíció elején található boot blokkot.
4. A betöltött program indítja az operációs rendszert.

Szuperblokk

A partíció elején, a boot blokk után található a szuperblokk, amely a fájlrendszer alapvetően fontos adatait tartalmazza. A szuperblokk része a magic number, amely azonosítja a fájlrendszer típusát. Partíció típusa és szuperblokk magic number összefüggenek, hiszen adott típusú partícióban adott állományrendszert szokás készíteni. A fájlrendszerben szokás elhelyezni a szuperblokk másolatait veszély esetére.

További részek



A szuperblokk után általában az üres és foglalt területek nyilvántartási rendszere következik. Ez után például az inode-ok vagy egyéb adminisztratív elemek következnek és természetesen az adatblokkok.



CD-ROM fájlrendszer

A CD-ROM fájlrendszerek igen egyszerűek, hiszen olyan média számára dolgozták ki őket, amelyeket csak egyszer hozunk létre és soha nem módosítunk. A CD-ROM fájlrendszer ráadásul nem kell, hogy kezelje az üres területeket, ami tovább egyszerűsíti felépítését.

CD-ROM fájlrendszer

A legelterjedtebb CD-ROM fájlrendszer az 1988-ban bevezetett ISO9660 szabvány által leírt fájlrendszer. A szabvány célja olyan fájlrendszer létrehozása volt, amelyet minden operációs rendszer kezelni tud. Sok korlátot építettek a rendszerbe, hogy az olyan primitív operációs rendszerek is kezelni tudják, amilyen például az MS-DOS. Így akadályozta a Microsoft akkoriban a fejlődést...

A CD-ROM felépítése

A CD-ROM nem tartalmaz koncentrikus köröket – mint pl. a merevlemez –, csak egyetlen spirálisan felírt bitsorozat található rajta. Ennek ellenére a CD-ROM drive képes fejmozgató mechanikája segítségével a véletlenszerű adatelérésre.

A felírt bitsorozat 2352 bájtos logikai blokkokra oszlik, amelyből 2048 bájt az adattárolásra használható, hasznos terület. Ilyen módon az adattároló terület lineáris blokkorozatként tárgyalható.

Partíciók

A szabvány lehetővé teszi $2^{16} - 1$ lemezből készített készletek kezelését, továbbá logikai partíciók kialakítását a lemezen, de a gyakorlatban ezeket a lehetőségeket ritkán használjuk ki.

Bootstrap

A CD-ROM első 16 blokkjának szerepét a szabvány nem írja le. Ezt a területet a gyártó betöltőprogram elhelyezésére használhatja fel.

Primary volume descriptor

A következő blokk néhány fontos adatot tartalmaz: system identifier (32bytes), volume identifier (32bytes), publisher identifier (128bytes), preparer identifier (128bytes), létrehozás időpontja, lajázat időpontja (sic!),...

Fontos része a primary volume descriptornak a gyökérvégytár jelző, a gyökérvégytár tartalmazó blokk számát megadó mező. A teljes fájlrendszer elérhető a gyökérvégytárból kiindulva.

További volume descriptor



A primary volume descriptor képes információt hordozni további volume descriptorról, így a lemezen több volume descriptor, több kötet elhelyezhető. Mi ezzel nem foglalkozunk.



A könyvtárak

A könyvtárak változó hosszúságú listák, ahol az utolsó elemet az elemben elhelyezett jelzőbittel jelöljük meg. A könyvtárak elemei maguk is változó hosszúságúak, ezért tartalmazzák a saját hosszukat. Minden könyvtárbejegyzés 10-12 mezőből áll, amelyek közt vannak ASCII és bináris mezők is. A bináris mezők little-endian és big-endian kódolással is szerepelnek a viták elkerülése érdekében.

Könyvtárbejegyzés

- 1 A könyvtárbejegyzés hossza.
- 1 A könyvtárbejegyzéshez tartozhat kiegészítő információ. Ez a bájtt a kiegészítő információk hosszát adja meg.
- 8 Az adatterület elhelyezkedése. Az adatterületek összefüggők, hiszen nem változtatható állományrendszerrel van szó, ezért a blokkszámmal megadható.
- 8 A könyvtárbejegyzéshez tartozó adatterület hossza.

Könyvtárbejegyzés

7 Dátum és idő.

1 Attribútumbitek.

2 Egyszerû CD-ROM-oknál nem használt mezõ.

4 A CD-ROM száma, amelyen az adatterület van. Ilyen módon lehetséges egy master CD-t készíteni, amelyen a könyvtárbejegyzések vannak, a többi lemezen pedig az adatterületek.

1 Az állománynév hossza.

4-15 Az állománynév basename.ext;ver formában.

xxx A további bájtok az esetleges kiegészítésekhez tartoznak.

Korlátok (level 1)

A könyvtárban található könyvtárbejegyzések számára nézve nincs korlátozás.

A könyvtárbejegyzések egymásba ágyazásának maximális mélysége 8. Így nem lehetséges nyolcnál mélyebb könyvtárszerkezetek tárolására.

Az állományok neve 8+3 karakter lehet, a könyvtárak neve 8 karakter. Ezeket a korlátokat elsősorban az MS-DOS miatt vezette be a szabvány és ezek miatt a korlátok miatt több rendszeren nem lehet használni a szabványt (uniform naming!)

Korlátok (level 2)

Az állománynevek hosszára bevezetett korlátot próbálja feloldani a level 2, amely 31 betű hosszú neveket enged meg. Ez persze a legtöbb rendszeren kevés az üdvösséghez.

Korlátok (level 3)

A nevekre ugyanazokat a korlátokat vezeti be, mint a level 2, de itt az állományoknak nem kell folytonos bájtsorozatoknak lenniük. Ha például az állományban van egy ismétlődő rész, nem kell többször felírni.

Rock Ridge extensions

Rock Ridge egy város neve GENE WILDER: *Blazing Saddles* C. filmjében. Innen kapta a nevét a közkedvelt bővítés, amellyel Unix rendszereken is használhatjuk az ISO9660 szabványt.

A R.R. a könyvtárbejegyzések végén található rendszerspecifikus bővítésre fenntartott helyet használja fel, a többi mezőt az eredeti formában használja. A R.R. bővítést nem használó rendszerek a szokásos módon használják a médiát.

R.R. mezők

1. POSIX attribútumok.
2. Major and minor device numbers.
3. Symbolic link.
4. Alternative name.
5. Child location.
6. Parent location.
7. Relocation. (ez jelzi, ha a könyvtárbejegyzés nem ott van, ahol kellene lennie a 8 szint korlát miatt)
8. Time stamps

Joliet



A Microsoft által tervezett bővítés a Joliet, amely hasonló okokból született mint a Rock Ridge. A céljai a következők voltak:

1. Hosszú fájlnevek (64, ha nincs unicode 128).
2. Unicode karakterkészlet.
3. Nyolcnál mélyebb könyvtárstruktúra (egyszerűen eltörölték a szabványban található korlátot).
4. Kiterjesztés a könyvtárnevekben (nem világos, hogy ez miért kellett).

