Partíció, formázás, fájlrendszerek

Partíciók

A merevlemezek több *partíció*ra oszthatók, melyek mindegyike külön merevlemezként viselkedik. Az alapötlet az, hogy ha egy merevlemezed van, de mondjuk két operációs rendszered, a lemezt két partícióra oszthatod az egyes operációs rendszerek számára. Az operációs rendszerek csak a saját partíciójukat használják, a másikról nem vesznek tudomást, így képesek békésen egymás mellett élni azonos merevlemezen. (Ugyanakkor például a Linux képes más rendszerek [FAT32, NTFS stb.] partícióit használni, valamint komolyabb Linux rendszer maga is több partíción helyezkedik el. - a lektor) A partíciók nélkül külön merevlemez kellene minden operációs rendszer számára.

A hajlékonylemezeket nem particionáljuk. Ugyan nincs technikai ok, mely ez ellen szólna, de olyan kicsik, hogy nem lenne értelme tovább osztani őket. A CD-ROM-ok rendszerint szintén nem particionáltak, mivel egyszerűbb egyetlen nagy lemezként használni őket, s csak igen ritkán kell több operációs rendszert egyetlen CD-ROM-ra feltenni. (A több menetben írt CD lemez nem tévesztendő össze a több partícióval. - a lektor)

Formázás

A *formázás* az a folyamat, melynek során a mágneses adathordozóra jeleket írunk, melyek a sávokat és szektorokat jelölik meg. Egy lemez formázása előtt a mágneses felszín csak összevissza jeleket tartalmaz. A formázás bizonyos rendet visz ebbe a káoszba oly módon, hogy lényegében vonalakat húz a sáv- és szektorhatárokra. A pontos részletek ettől egy kicsit eltérnek, de most ez lényegtelen. Az viszont fontos, hogy formázás nélkül nem használhatók a lemezek.

A szóhasználat kicsit zavaros itt: MS-DOS és MS Windows alatt a "formázás" a fentieken kívül még a fájlrendszer létrehozását (lásd lentebb) is jelenti. Ott a két folyamat gyakran keveredik, különösen a hajlékonylemezek esetében. Ha a kettőt fontos megkülönböztetni, akkor a valódi formázást *alacsony szintű formázás*nak, a fájlrendszer létrehozását *magas szintű formázás*nak nevezik. Unixos körökben ezeket a folyamatokat formázásnak illetve fájlrendszer létrehozásnak nevezik. Ezt a szóhasználatot követjük a továbbiakban.

IDE és SCSI lemezeknél a formázást többnyire már a gyárban elvégzik, és sosem kell megismételni, így általában nem is kell vele törődni. A merevlemezek "házi" formázása rosszabb eredményekhez vezethet, mert például a lemezt speciálisan kell formázni, hogy a hibás szektorok automatikus áthelyezése működhessen.

A merevlemezek, melyeket meg kell vagy meg lehet formázni, gyakran igényelnek speciális programot, mivel a formázás meghajtón belüli logikája meghajtóról meghajtóra más és más. A formázó program gyakran a vezérlőben lévő BIOS-ban van, vagy egy MS-DOS program; egyik sem használható könnyen Linux alól. (Igaz, többnyire nincs is rájuk szükség.)

A formázás során rossz "foltokat" észlelhetünk a lemezen, amit *hibás blokk*nak vagy *hibás szektor*nak nevezünk. Ezeket néha maga a meghajtó is tudja kezelni, de még ebben az esetben is tennünk kell valamit, hogy biztosan elkerüljük a hibás részek használatát, ha a hibás

szektorok száma nagyon megnövekszik. Ennek logikája bele van építve a fájlrendszerbe; a hibás szektorokról szóló információ megadásáról lásd a lentieket. Egy másik lehetőség, hogy létrehozunk egy olyan partíciót, amely csak a lemez rossz részeit fedi le; ez jó ötlet lehet, ha a hibás rész nagyon nagy, mivel ez megzavarhatja a fájlrendszerek működését.

Fájlrendszerek

*Fájlrendszer*en (filesystem) azokat a módszereket és adatstruktúrákat értjük, melyeket egy operációs rendszer használ egy lemez vagy partíció fájljainak kezelésére; azaz ahogyan a fájlok elrendeződnek a lemezen. Ezt a szót még a lemez adott típusú fájlrendszert tartalmazó partíciójára is használják, vagy csak a típus megjelölésekor. Így például mondhatjuk, hogy "Két fájlrendszerem van.", ami azt jelenti, hogy két partíciónk van, melyeken a fájlokat tároljuk. Azt is mondhatjuk, hogy valaki "kiterjesztett fájlrendszert" használ, ami viszont az általa használt fájlrendszer típusát jelenti.

A lemez (vagy partíció) és a rajta lévő fájlrendszer közti különbség nagyon fontos. Néhány program, mint például a fájlrendszereket létrehozó programok, közvetlenül a lemez szektoraival dolgoznak. Ilyen program egy már esetleg meglévő fájlrendszert súlyosan megrongálhat, vagy meg is semmisíthet. A legtöbb program viszont csak a fájlrendszert használva ír a lemezre, ezért ezek csak megfelelő fájlrendszert már tartalmazó partíción működnek.

Mielőtt egy lemezt vagy partíciót fájlrendszerként kezdünk használni, inicializálni kell, és a nyilvántartó adatstruktúrákat a lemezre kell írni. Ezt a folyamatot *fájlrendszer készítés*nek nevezzük.

A legtöbb UNIX fájlrendszernek hasonló az általános felépítése, bár a részletek egy kicsit változhatnak. A központi fogalmak: *szuperblokk, inode, adatblokk, könyvtárblokk* és az *indirekció blokk*. A szuperblokk a fájlrendszer egészéről tartalmaz információkat, mint például a teljes méret. A szuperblokk pontos tartalma fájlrendszerfüggő. Az inode-ok egy-egy fájl minden adatát tartalmazzák a nevén kívül. A név ugyanis a könyvtárban tárolódik az inode sorszámával együtt. Egy könyvtárbejegyzés a fájlnévből és a fájlt jelképező inode számából áll. Az inode több adatblokk sorszámát tartalmazza, melyek a fájl adatait tárolják. Az inode-ban azonban csak néhány adatblokk sorszámának van hely, és ha ennél több szükséges, automatikusan több terület kerül lefoglalásra. Ezek a dinamikusan lefoglalt blokkok az indirekt blokkok; a név azt jelzi, hogy az adatblokk megtalálásához előbb az indirekt blokkban kell megkeresni a megfelelő blokk sorszámát.

Az UNIX fájlrendszerek rendszerint megengedik a fájlbeli *lyukak* (hole) létrehozását (lásd az `lseek()` rendszerhívás kézikönyv oldalát), ami azt jelenti, hogy a fájlrendszer megjegyzi, hogy a fájl egy adott helyén egymás után adott számú 0 byte van, de nem foglal ezen nullák számára blokkot. Ez kisebb lemezhasználatot eredményez. Egymást követő nullák gyakran előfordulnak kis bináris programokban, a Linux megosztott programkönyvtáraiban, adatbázisokban és néhány más esetben. (A lyukak megvalósítása úgy történik, hogy egy speciális érték kerül beírásra az adatblokk sorszámának helyére az inode-ban vagy az indirekt blokkban. Ez a speciális érték azt jelenti, hogy a fájl azon részéhez nem tartozik adatblokk, azaz lyuk van a fájlban.)