

# A FreeBSD telepítése és használata más operációs rendszerekkel együtt

Jay Richmond

jayrich@sysc.com

1996. augusztus 6.

**A FreeBSD a FreeBSD Foundation bejegyzett védjegye.**

**Az IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390 és ThinkPad az International Business Machines Corporation védjegyei az Egyesült Államokban, más országokban, vagy mindkettőben.**

**A Linux Linus Torvalds bejegyzett védjegye.**

**A Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media és Windows NT a Microsoft Corporation bejegyzett védjegyei, vagy védjegyei az Egyesült Államokban és/vagy más országokban.**

**A PowerQuest és PartitionMagic a PowerQuest Corporation bejegyzett védjegyei az Egyesült Államokban és/vagy más országokban.**

**A gyártók és terjesztők által használt megnevezések közül sok védjegy jogot követel. Ahol ilyen megnevezés tűnik fel ebben a dokumentumban, és a FreeBSD Projektnek tudomása volt a védjegyről, a megnevezést a "TM" vagy a "®" szimbólum követi.**

Ez a leírás azt tárgyalja, miképpen lehet a FreeBSD-t olyan más népszerű operációs rendszerek, mint mondjuk a Linux® MS-DOS®, OS/2® és Windows® 95 mellé telepíteni és használni. Külön köszönet: Annelise Anderson <andrsn@stanford.edu>, Randall Hopper <rh@ct.picker.com> és Jordan K. Hubbard <jkh@FreeBSD.org>.

*Fordította: Páli Gábor <pgj@FreeBSD.org>*

## 1. Áttekintés

A legtöbben nem tudják az említett operációs rendszereket kényelmesen egymás mellé rakni egy kisebb méretű merevlemezre, ezért a nagyobb EIDE-meghajtókkal kapcsolatos ismeretekről is szó fog esni. Mivel rengeteg kombinációja létezik a különféle operációs rendszereknek és merevlemezeknek, valószínűleg az 5 Szakasz fog a leírás leghasznosabb részének bizonyulni. Itt találhatóak meg ugyanis azok a speciális beállítási sémák, amelyek több operációs rendszer használata esetén alkalmazhatóak.

Ez a cikk feltételezi, hogy a merevlemezünkön már előkészítettünk kellő mennyiségű szabad helyet az újabb operációs rendszer(ek) számára. Minden egyes alkalommal, amikor újra felosztjuk a merevlemezünket, egyúttal kockára tesszük a meglévő partícióinkon levő adataink épségét is. Viszont ha a merevlemezünkön teljes egészében csak a DOS található, akkor a FIPS nevű segédprogramot hasznosnak fogjuk találni (megtalálható a FreeBSD

CDROM-on, a `\TOOLS` könyvtárban, vagy FTP-n (<ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/>). Segítségével anélkül tudjuk particionálni a merevlemezünket, hogy kockára tennénk a rajta levő adatainkat biztonságát. Valamint létezik még egy **PartitionMagic®** nevű kereskedelmi alkalmazás is, amellyel minden komoly következmény nélkül tudunk particiókat átméretezni és törölni.

## 2. A boot managerek áttekintése

Csak röviden bemutatnánk néhány elterjedt boot managert. Közülük, a számítógépünk kiépítésétől függően, egyet vagy többet jó eséllyel tudunk majd használni.

### Boot Easy

Ez a FreeBSD alapértelmezett boot managere. Szinte bármilyen rendszert képes indítani, többek közt a BSD, OS/2 (HPFS), Windows 95 (FAT és FAT32) és Linux típusú rendszereket. Az indítandó particiót a funkcióbillentyűkkel választhatjuk ki.

### OS/2 Boot Manager

Elindítja a FAT, FAT32, HPFS, FFS (FreeBSD) és EXT2 (Linux) particiókat, amelyet a nyilakkal választhatunk ki. Az OS/2 Boot Manager az egyetlen az itt felsoroltak közül, amely a saját particióját használja, miközben az összes többi a Master Boot Record (MBR)-ot. Ennek következtében az 1024. cylinder elé kell telepítenünk, hogy elkerüljük az ezzel kapcsolatos esetleges indítási problémákat. LILO-val telepített Linux-ot csak akkor képes indítani, amikor az a boot szektorban található, nem pedig az MBR-ben. Az Interneten található Linux hogyanok (<http://www.linuxresources.com/LDP/HOWTO/HOWTO-INDEX.html>) között további információkat találhatunk az OS/2 boot manager és a Linux kapcsolatáról.

### OS-BS

Ez egy másik boot manager a Boot Easy mellett. Valamivel több kontrollt ad a rendszerindítási folyamat felett, például beállítható benne az alapértelmezett indított partició és egy várakozási idő. A program béta változatában már a nyilak segítségével lehet kiválasztani az indítandó operációs rendszert. Szintén megtalálható a FreeBSD CD-jén a `\TOOLS` könyvtárban vagy FTP-n (<ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/>).

### LILO, avagy Linux Loader

Ez egy korlátozott képességű boot manager. Képes elindítani a FreeBSD-t, habár ehhez szükség van némi finomhangolásra a hozzátartozó konfigurációs állományban.

**Röviden FAT32-ről:** A FAT32 a FAT állományrendszer kiváltására szolgál, amelyet a Microsoft 1996 végén, a Windows 95 OSR2 béta változatától kezdődően indított útjának, ezzel lecserélve a Windows 95-tel telepített számítógépek alapértelmezett FAT típusú állományrendszerét. Úgy alakítja át a megszokott FAT-ot, hogy lehetővé teszi a kisebb kiosztási egységek használatát nagyobb merevlemezeken is. Továbbá a FAT32-ben megváltoztatták a hagyományos FAT boot szektorát és kiosztási táblázatát is, összeférhetetlenné téve ezáltal néhány boot managerrel.

### 3. Egy átlagos telepítés

Tegyük fel, hogy van két nagyobb EIDE merevlemezünk és szeretnénk rájuk FreeBSD-t, Linux-ot és Windows 95-öt telepíteni.

Íme, hogyan tennénk mindezt az alábbi merevlemezekkel:

- /dev/wd0 (az első fizikai lemez)
- /dev/wd1 (második fizikai lemez)

Mindkét lemeznek 1416 cilindere van.

1. Elsőként indítsunk az MS-DOS vagy Windows 95 rendszerindító lemezével, amelyen az `FDISK.EXE` segédprogram található. Ennek segítségével készítünk egy kis, nagyjából 50 MB méretű elsődleges partíciót (35-40-et a Windows 95-nek, meg hagyunk egy kis helyet levegőzni is) az első lemezen. Ezen kívül még készítünk egy nagyobb partíciót a második merevlemezen, ahol a Windowsos alkalmazásaink és az adataink foglalnak majd helyet.
2. Indítsuk újra a gépet és telepítsük fel a Windows 95-öt a C: partícióra (amit egyébként könnyebb mondani, mint megtenni).
3. Következésként a Linux-ot telepítsük fel. Nem vagyok benne biztos, hogy ez mindegyik Linux-disztribúcióra igaz, de a Slackware (<http://www.slackware.com>) tartalmazza a LILO-t (ld. 2 Szakasz). A Linux-os `fdisk` parancsával tovább partícionálva én a Linux-ot az első lemezre tenném (nagyjából 300 MB elegendő egy kövőbb rendszerpartíciónak és némi lapozóállománynak).
4. Miután feltelepítettük a Linux-ot és éppen a LILO elhelyezése előtt állunk, *mindenképpen* ellenőrizzük, hogy a Linux-os rendszerpartíció boot szektorába telepítjük, nem pedig az MBR-be!
5. A fennmaradó hely mehet mind a FreeBSD-nek. Vigyázzunk, hogy a FreeBSD rendszerslice-a ne kerüljön az 1024. cilinderen túlra. (Az 1024. cilinder az 528. MB-nál található a most feltételezett 720 MB-os lemezükön.) A merevlemez többi részét (nagyjából 270 MB) az /usr és / slice-okra is fel lehet használni. A második lemez fennmaradó részén (aminek a mérete az 1. lépésben kialakított, Windows-os alkalmazásoknak és adatoknak szánt partíció méretétől függ) még elfér a /usr/src slice és a lapozóállomány.
6. Ha most megnézzük a Windows 95 `fdisk` programjával, a merevlemezeket valahogy így láthatjuk:

-----

Partíció információinak megjelenítése

Aktuális merevlemez meghajtó: 1

Partíció	Állapot	Típus	Kötetecímke	Megabájt	Rendszer	Felhasznált
C: 1	A	PRI DOS		50	FAT**	7%
2	A	Non-DOS (Linux)		300		43%

Teljes lemezterület: 696 megabájt (1 megabájt = 1048576 bájt)

A folytatáshoz nyomja meg az Esc billentyűt.

-----

Partíció információinak megjelenítése

Aktuális merevlemez meghajtó: 2

Partíció	Állapot	Típus	Kötet címke	Megabájt	Rendszer	Felhasznált
D: 1	A	PRI DOS		420	FAT**	60%

Teljes lemezterület: 696 megabájt (1 megabájt = 1048576 bájt)

A folytatáshoz nyomja meg az Esc billentyűt.

-----

\*\* Ez FAT16 vagy FAT32 lehet attól függően, hogy OS/2-t használunk-e. Lásd 2 Szakasz.

7. Telepítsük fel a FreeBSD-t. Mindenképpen az első merevlemezről indítsuk el a számítógépet, ezért a BIOS-ban állítsuk "NORMAL"-ra. Ha nem az lenne, adjuk meg a lemez valós geometriáját indításkor (a lekérdezéséhez indítsuk el Windows 95-öt, majd a Microsoft Diagnostics-ot (MSD.EXE, esetleg nézzük meg a BIOS-ban) a `hd0=1416,16,63` paraméterrel, ahol a `1416` megadja a merevlemez cilindreinek számát, a `16` a fejek számát sávonként, valamint a `63` a szektorok számát sávonként.
8. A merevlemez partícionálása során a Boot Easy-t mindenképpen az első lemezre tesszük. A második lemez miatt különösebben ne aggódjunk, semmi bootolni való nincs rajta.
9. Újraindítás után a Boot Easy várhatóan felismeri mind a három indítható partíciót: DOS (Windows 95), Linux és BSD (FreeBSD) néven.

## 4. Különösen megfontolandók

A legtöbb operációs rendszer meglehetősen kényes abban a tekintetben, hogy hova helyezzük őket a merevlemezre. A Windows 95-öt és DOS-t az első merevlemez első elsődleges partíciójára kell telepítenünk. Az OS/2 innen nézve kivételnek számít, mivel egyaránt telepíthető az első vagy a második merevlemezre is, tetszőleges elsődleges vagy kiterjesztett partícióra. Ha nem vagyunk benne biztosak, az indítható partíciókat tesszük mindig az 1024. cylinder elé.

Ha a Windows 95-öt egy már meglévő BSD rendszer mellé telepítjük, “tönkre fogja tenni” az MBR-t, és ezért újra kell telepítenünk a korábbi boot managerünket. A Boot Easy-t a FreeBSD telepítő CDRROM-jának \TOOLS könyvtárában található, vagy az FTP-n (<ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/>) letölthető `BOOTINST.EXE` segítségével tudjuk visszarakni. Másik lehetőség gyanánt elindíthatjuk a telepítőt is, és megkereshetjük benne a partíciószerkesztőt. Itt jelöljük meg FreeBSD-t tartalmazó partíciót indíthatónak (*bootable*), majd válasszuk a Boot Managert és nyomjuk le a W-t (mint (W)rite out) a boot manager tényleges MBR-be írásához. Most már újraindíthatjuk a számítógépet és a Boot Easy pedig felismeri a Windows 95-öt mint DOS.

Nem szabad elfelejtenünk, hogy az OS/2 ugyan képes FAT és HPFS partíciókat olvasni, viszont FFS-t (FreeBSD) és EXT2-t (Linux) nem! Ehhez hasonlóan a Windows 95 csak FAT és FAT32 partíciókat (ld. 2 Szakasz) tud írni és olvasni. A FreeBSD ismeri a legtöbb állományrendszert, de jelenleg nem tud HPFS partíciókat olvasni. A Linux képes HPFS partíciókat olvasni, de nem tudja írni őket. A Linux kernel legújabb (2.x-es) változatai már képesek írni és olvasni a Windows 95 VFAT partícióit (a VFAT az, aminek a segítségével a Windows 95 képes hosszú állományneveket kezelni — egyébként teljesen olyan, mint a FAT). A Linux tehát képes írni és olvasni a legtöbb állományrendszert. Érthető? Remélem!

## 5. Példák

*(ennek a szakasznak szüksége van még némi átdolgozásra, várjuk a hozzászólásokat a témában a <[jayrich@sysc.com](mailto:jayrich@sysc.com)> címre).*

FreeBSD + Windows 95: Ha a FreeBSD-t a Windows 95 után telepítettük, akkor a Windows 95-öt a Boot Easy menüjében DOS-ként kell látnunk. Ha viszont a Windows 95-öt a FreeBSD után telepítettük, olvassuk el a fenti 4 Szakaszt. Amíg nincsenek olyan merevlemezeink, amelyek mérete meghaladná az 1024 cilindert, nem kell különösebben aggódnunk a bootolás miatt. Amikor azonban valamelyik partíciónk az 1024. cylinder fölé merészkedik és DOS (vagy Windows 95) alatt olyan hibüzeneteket kapunk, mint mondjuk a “Rossz rendszerlemez”, valamint a FreeBSD sem képes elindulni, keressünk meg a BIOS-unk beállításai között “> 1024 cylinder support”-ot (1024-nél több cylinder támogatása) vagy a “NORMAL/LBA” nevezetű módot. A DOS-nak ebben az esetben ugyanis szüksége lehet az LBA (Logical Block Addressing) bekapcsolására a bootoláshoz. Ha nem akarjunk minden egyes rendszerindításkor eljátszani ezt, a CD-n található `FBSDBOOT.EXE` segítségével akár a DOS-on keresztül is el tudjuk indítani a FreeBSD-t. (Ez ugyanis megkeresi a FreeBSD-s partíciót és elindítja azt).

FreeBSD + OS/2 + Windows 95: Nincs új a nap alatt. Az OS/2 boot managere képes elindítani mindezen operációs rendszereket, ez a kombináció tehát nem okozhat problémát.

FreeBSD + Linux: A Boot Easy segítségével mind a két rendszer elindítható.

FreeBSD + Linux + Windows 95: (ld. 3 Szakasz)

## 6. Egyéb hasznos helyek

Számtalan Linux hogyan (<http://www.linuxresources.com/LDP/HOWTO/HOWTO-INDEX.html>) foglalkozik az egy merevlemezre telepíthető operációs rendszerek problémájával.

A Linux+DOS+Win95+OS2 mini-hogyan

(<http://www.linuxresources.com/LDP/HOWTO/mini/Linux+DOS+Win95+OS2.html>) az OS/2 boot managerével kapcsolatosan nyújt némi segítséget, valamint a Linux+FreeBSD mini-hogyan

(<http://www.linuxresources.com/LDP/HOWTO/mini/Linux+FreeBSD.html>) is érdekes olvasmány lehet. A Linux-hogyan (<http://www.in.net/~jkatz/win95/Linux-HOWTO.html>) is fontos információkat tartalmazhat.

A Windows NT® Loader Hacking Guide ([http://www.tburke.net/info/ntldr/ntldr\\_hacking\\_guide.htm](http://www.tburke.net/info/ntldr/ntldr_hacking_guide.htm))-ban sok érdekesség megtalálható a Windows NT, Windows 95 és DOS más operációs rendszerekkel együtt történő használatáról.

Hale Landis "Hogyan is működik?" c. leírása is rengeteg hasznos apróságot árul el a különfél lemez geometriákról és a rendszerindítással kapcsolatos egyéb tudnivalókról. Ezt itt

([ftp://fission.dt.wdc.com/pub/otherdocs/pc\\_systems/how\\_it\\_works/allhiw.zip](ftp://fission.dt.wdc.com/pub/otherdocs/pc_systems/how_it_works/allhiw.zip)) találhatjuk meg.

Végezetül, erősen javallott tüzetesen átnézni a FreeBSD rendszermag rendszerindításáról szóló dokumentációját is, amely megtalálható a rendszermag forrásában (alapértelmezés szerint a `/usr/src/sys/i386/boot/biosboot/README.386BSD` helyre kerül).

## 7. Technikai részletek

*(Köszönet érte Randall Hoppernek <[rrh@ct.picker.com](mailto:rrh@ct.picker.com)>)*

Ebben a szakaszban megpróbálunk kellő mennyiségű alapvető ismeretet átadni a használatban levő merevlemezekről, valamint ezen lemezek rendszerindítási folyamatáról, elegendőt ahhoz, hogy le tudjuk küzdeni azokat a leggyakoribb problémákat, amelyek több operációs rendszer indítása során leselkednek ránk. Teljesen a kezdetektől indul, ezért javasolt egészen addig a pontig ugrani az olvasásban, ahol már ismeretlen dolgok is kezdenek feltűnni.

### 7.1. Amit tudni érdemes a lemezekről

Három alapvető jellemző írja le a merevlemezen található adatok pontos helyét: cylinder, fej, szektor. Igazából nem teljesen lényeges tudni, hogy ezek milyen viszonyban is állnak egymással, kivéve annyit, hogy ezek együttesen azonosítják be fizikailag a lemezen található adatokat.

Egy merevlemeznek van adott számú cilindere, feje és szektora az egyes cylinder-fej párosok mentén (amelyet egyébként sávnak is neveznek). Ezek az információk adják meg együttesen a merevlemez "fizikai geometriáját". Általában 512 byte található szektoronként valamint 63 szektor fejenként, azonban a cilinderek és a fejek száma jelentősen változik lemezenként. Ezért a merevlemezen maximálisan tárolható adatok mennyiségét a következőképpen lehet kiszámítani ezek ismeretében:

$(\text{a cilinderek száma}) \times (\text{a fejek száma}) \times (63 \text{ szektor/sáv}) \times (512 \text{ byte/szektor})$

Például, ez egy 1,6 gigabyte-os Western Digial AC31600 EIDA merevlemez esetén:

$(3148 \text{ cylinder}) \times (16 \text{ fej}) \times (63 \text{ szektor/sáv}) \times (512 \text{ byte/szektor})$

amely 1 624 670 208 byte-nak felel meg, ami pedig nagyjából 1,6 gigabyte.

Az egyes merevlemezek fizikai geometriáját (a cilinderek, fejek és a sávonkénti szektorok számát) az ATAID és az Interneten megtalálható egyéb hasonló programokkal lehet lekérdezni. De valószínűleg magán a merevlemezen is megtalálható ez az adat. Azonban nem árt óvatosnak lennünk: ha a BIOS-ban LBA-t állítottunk be (ld. 7.3 Szakasz),

az említett programok egyikét sem tudjuk használni. Ezért sem képes sok más program (pl. az MSD.EXE vagy a FreeBSD fdisk) megállapítani a fizikai lemez geometriát; helyette az *átértelmezett geometriát* (a LBA-ból származó virtuális azámadatokat) adják vissza. Erről még beszélni fogunk.

Még egy apróság ezzel kapcsolatban. A 3 szám — nevezetesen a cilinderek, a fejek és a szektorok sávonkénti száma — ismeretében képesek vagyunk betájolni egy konkrét abszolút szektort (vagyis egy 512 byte-os adatblokkot) a lemezünkön. A cilindereket és fejeket 0-tól, míg a szektorokat 1-től szokták számozni.

Azok számára, akik még jobban el akarnak mélyedni a technikai részletekben, a lemezek geometriájában, a boot szektorok és BIOS-ok stb. titkaiban, mindent megtalálhatnak róluk az Interneten. Keressenek rá bátran a Lycos, Yahoo stb. szolgáltatásokban a `boot sector` vagy `master boot record` szavakra. A sok hasznos ismeret között esetleg találkozni fogunk Hale Landis *Hogyan is működik?* c. leírásgyűjteményével is. Ezzel kapcsolatban ld. a 6 Szakaszt.

Rendben, ennyi elég is lesz a terminológiáról. Beszéljük a bootolásról!

## 7.2. A rendszerindítás folyamata

A merevlemez első szektorában (azaz a 0. cylinder, 0. fej, 1. szektor) lakozik a Master Boot Record (MBR). Ez tartalmazza lényegében a teljes lemez térképét. Legfeljebb 4 *partíciót* képes tárolni, amelyek mindegyike a lemez egy-egy folytonos darabkája. A FreeBSD ezeket a partíciókat egyébként *slice-oknak* hívja annak érdekében, hogy elkerülje a saját partícióival történő összetévesztésüket, habár mi most nem így fogunk tenni. Minden egyes partícióra telepíthető egy-egy operációs rendszer is.

Az MBR-ben található összes partíciós bejegyzésnek van egy ún. *partíció azonosítója*, egy *kezdő cylinder/fej/szektor értéke* és egy *befejező cylinder/fej/szektor értéke*. A partíció azonosítója megadja, hogy az adott partíció milyen típusú (milyen operációs rendszer használja), a kezdő/befejező értéke pedig azt, hol található. A 1 Táblázatban a teljesség igénye nélkül felsoroltunk néhány ismertebb azonosítót.

Táblázat 1. Partíció azonosítók

Az. (hex)	Leírás
01	Elsődleges DOS12 (12 bites FAT)
04	Elsődleges DOS16 (16 bites FAT)
05	Kiterjesztett DOS
06	Elsődleges nagy DOS (> 32MB)
0A	OS/2
83	Linux (EXT2FS)
A5	FreeBSD, NetBSD, 386BSD (UFS)

Megjegyezzük, hogy nem mindegyik partíció indítható (ilyen pl. a kiterjesztett DOS). Egyesek igen — mások pedig nem. Amitől egy partíció bootolhatóvá válik, az a *partíció boot szektora*, amely az egyes partíciók elején található.

Amikor beállítjuk a kedvenc boot managerünket, az tulajdonképpen átnézi az összes merevlemez MBR-jeinek partíciós táblájában található bejegyzéseket és lehetővé teszi számunkra, hogy elnevezzessük őket. Majd amikor elindítjuk a számítógépet, a boot manager az elsőként próbált merevlemez Master Boot Recordjában elhelyezett speciális program segítségével életre kel. Felkeresi a választásunknak megfelelő partíciót az MBR partíciós táblájában, és felhasználva az így megismert kezdő cylinder/fej/szektor adatokat, betölti az adott partíció boot szektorát, majd átadja neki a vezérlést. A partíció boot szektora ezek után már elegendő információt tartalmaz a rajta

levő operációs rendszer indításához.

Egyetlen fontos tudnivalót nem említettünk meg még: minden merevlemezen található MBR. Azonban ezek közül csak az tekinthető fontosnak, amely a BIOS által elsőként próbált lemezen található. Ha csak IDE csatolós merevlemezeink vannak, ez általában az első IDE lemez (pl. az elsődleges lemez az első vezérlőn). Ugyanez a helyzet a csak SCSI-t tartalmazó rendszerekben. Ha viszont van IDE és SCSI merevlemezünk is, a BIOS általában az IDE lemezeket próbálja először elindítani, így az elsőként elindított lemez az első IDE lemez. A boot managert tehát az elsőként elinduló merevlemez MBR-jébe kell elhelyeznünk a fentiekben leírtak szerint.

## 7.3. A rendszerindítás korlátai és veszélyei

Most pedig következzen mindaz, amire nagyon oda kell figyelnünk.

### 7.3.1. A rettegett 1024 cilinderes korlát és hogyan segít ezen az LBA

A bootolás folyamatának első része a BIOS-on keresztül megy végbe (ha még nem ismernénk: a BIOS az az alaplapon található chip, amely a számítógép indításához nélkülözhetetlen rutinokat tárolja). Mint olyan, a folyamat első része tehát a BIOS interfészének korlátozásaitól függ.

Ezen időtartam alatt a BIOS által nyújtott interfészt használjuk a merevlemez olvasására (13H megszakítás, 2-es funkció), amely 10 bitet használ a cilinderek, 8 bitet a fejek és 6 bitet a szektorok számozására. Ezzel lekorlátozza használatát (tehát az MBR-ből induló boot managereket és a boot szektorokban található betöltőket) az alábbiakra:

- legfejlebb 1024 cilinderre
- legfejlebb 256 fejre
- legfejlebb 64 szektorra sávonként (ami ténylegesen 63, mivel a 0. nem használható)

Mostanában azonban a nagyobb merevlemezeknek tengernyi cilinderük van, de nem túl sok fejük, ezért ezek a lemezek szinte kivétel nélkül átlépik az 1024 cilinderes határt. Ha vesszük ezt a tényt és összevetjük a BIOS által kínált interfésszel, rájöhethetünk, hogy nem bootolhatunk akárhol a lemeztől. A rendszerindító kódnak (tehát a boot managernek és az összes indítható partícióban található betöltőnek) az 1024. cilinder alatt kell lennie. Tényekre fordítva a szót, ha van egy átlagos merevlemezünk, aminek 16 feje van, ez nagyjából:

$1024 \text{ cilinder/lemez} \times 16 \text{ fej/lemez} \times 63 \text{ szektor/(cilinder - fej)} \times 512 \text{ byte/szektor},$

ami megfelel a sokszor emlegetett 528 MB-os határnak.

Itt jön a képbe a BIOS LBA (Logical Block Addressing). Ennek segítségével ugyanis a BIOS-hívások használója képes hozzáférni az 1024. feletti fizikai cilinderekhez is a BIOS-on keresztül, még hozzá a cilinderek átdefinálásával. Vagyis újraértelmezi a cilinderek és a fejek számát, és ezzel olyan képzetet ad, mintha a merevlemeznek kevesebb cilindere de több feje lenne, mint a valóságban. Másképp fogalmazva, kihasználja azt a helyzetet, hogy a modern merevlemezekben viszonylag kevés fej és sok cilinder található, ezért eltolja a kettő között nyugvó osztást, aminek köszönhetően mind a két érték az imént említett határok (1024 cilinder, 256 fej) alatt tud maradni.

A BIOS LBA használatával a merevlemezeken ezen korlátozása virtuálisan el is tűnik (nos, valójában csak 8 gigabyte-nyival arrébb kerül). Ha LBA-t támogató BIOS-unk van, akkor a FreeBSD-t és minden más operációs rendszert bárhova pakolhatunk, hiszen így nem fogunk az 1024 cilinderes korlátba ütközni.

Az előbb példaként felhozott 1,6 gigabyte-os Western Digital esetén tehát a fizikai geometria:

(3148 cylinder, 16 fej, 63 szektor/sáv, 512 byte/szektor)

Azonban a BIOS LBA ezt így fordítja át:

(787 cylinder, 64 fej, 63 szektor/sáv, 512 byte/szektor),

ami ugyanazt a tényleges lemezméretet eredményezi, azonban a cylinder- és fejadatok a BIOS-hívások által kezelhető tartományba esnek. (Mellékesen megjegyzem, hogy nekem pont egy Linux és egy FreeBSD partícióm van az egyik merevlemezemen, éppen az 1024. cylinder felett, és mind a kettő remekül bootol, hála az LBA-nak).

### **7.3.2. Boot managerek és a lemez kiosztása**

Egy másik fontos dolog, amire figyelniünk kell a boot managerek telepítése során, az éppen a boot managernek foglalt hely a lemezen. A legjobb erre már előre gondolni, és ezzel elkerüljük egy vagy több operációs rendszerünk újratelepítését.

Ha nyomunkövettük a Master Boot Recordról (avagy hol is található az MBR), a partíciók boot szektoráról és a rendszerindítási folyamatról szóló 7.2 Szakaszt, felmerülhet bennünk a kérdés, hogy a merevlemezükön hova is fog kerülni maga a boot manager. Nos, egyes boot managerek kellően kis méretűek ahhoz, hogy teljes egészében elférjenek a Master Boot Recordban (0. cylinder, 0. fej, 1. szektor), a partíciós tábla mellett. Másoknak ellenben valamivel több helyre van szüksége és tulajdonképpen a 0. cylinder 0. fejének sávjában nyúlnak túl az MBR-en néhány szektornyiival, mivel azok általában szabadon maradnak... általában.

És itt jön a csel! Egyes operációs rendszerek (köztük a FreeBSD is) megengedik, hogy a partíciójuk akár közvetlenül a Master Boot Record után kezdődjön a 0. cylinder 0. fejének 2. szektorában. Tulajdonképpen, ha a FreeBSD telepítőjének egy olyan lemezt adunk meg, amelynek az eleje vagy a teljes egésze éppenséggel üres, a FreeBSD partícióját alapértelmezés szerint közvetlenül ide rakja (legalább is így tette, amikor megpróbáltam telepíteni). Ezután szépen felrakjuk a boot manager, és ha az éppenséggel hajlamos elfoglalni az MBR után következő néhány szektort, akkor ezzel együtt felül is írja az első partíció adatait. A FreeBSD esetében így felülírja a lemezcímjét, amitől a FreeBSD partíció ezáltal bootolhatatlanná válik.

Ha egyszerűen el akarjuk kerülni ezt a problémát (és megadni az esélyt más, kevésbé rugalmas boot managerek számára), akkor hagyjuk szabadon a lemezen található első sávot. Vagyis ne tegyünk semmilyen partíciót a 0. cylinder, 0. fej, 2. szektorától kezdődően egészen a 0. cylinder, 0. fej 63. szektoráig, hanem helyezzük azt a 0. cylinder 1. fejének 1. szektorára. Ugyan nem mernék rá megesküdni, de ha létrehozunk egy DOS partíciót a lemez elején, a DOS alapértelmezés szerint ezt a területet szabadon hagyja (ezért is gondolja úgy néhány boot manager, hogy szabad). Ezt inkább magam szeretem csinálni, ezért létrehozok egy 1 megás DOS partíciót a lemez elején, mivel ezzel ráadásul meg tudom akadályozni, hogy elsőleges DOS meghajtónevek felcserélődjének egy újrapartícionálást követően.

Hivatkozásképpen, a következő boot managerek használják a Master Boot Recordot az adataik és kódjuk tárolására:

- OS-BS 1.35
- Boot Easy

- LILO

Ezek a boot managerek használnak további szektorokat a Master Boot Record után:

- OS-BS 2.0 Beta 8 (2-5. szektorok)
- Az OS/2 boot managere

### **7.3.3. Mit tegyünk, ha nem indul el a rendszer a számítógépünkön?**

Egyes esetekben előfordulhat, hogy a boot managerek telepítése során az MBR-t olyan állapotba sikerül hozni, ahonnan a számítógépünk nem képes elindulni. Ugyan nem valószínű, de megtörténhet, amikor ismételten használjuk az FDISK-et egy már meglevő boot manager alatt.

Ha van a lemezen egy bootolható DOS partíció, akkor indítsuk el azt egy DOS-os rendszerlemezről, és írjuk be:

```
A:\> FDISK /MBR
```

Ennek segítségével vissza tudunk rakni egy egyszerű DOS rendszerbetöltő kódot az MBR-be, ami után be tudjuk tölteni a DOS-t (de csak a DOS-t) a merevlemezről. Másik megoldás lehet, hogy simán újra felrakjuk a boot managerünket egy rendszerindító lemezről.