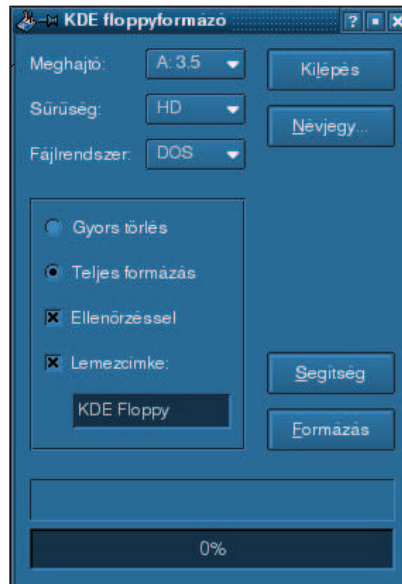


Fűzzünk be!

Sorozatunk e részében a Unixok egyik érdekes sajátosságával, a lemezegegyein lévő fájlrendszerek be-, illetve kifűzésével foglalkozunk részletesebben.

Aszóban forgó művelet mount-nak hívják. Akik ez idáig kizárólag DOS-, illetve Windows-rendszereket használtak, megszokhatták, hogy az összes különálló lemezegegyen helyet foglaló fájlrendszer egymástól teljesen el van választva. Ez a gyakorlatban úgy néz ki, hogy a rendszer azonosítóként minden meghajtóhoz egy betűt rendel – a későbbiekben ennek segítségével hivatkozhatunk rá. Nem nehéz belátunk, hogy rendkívül merev rendszerről van szó, amelyben elég nehéz jól szervezett könyvtárszerkezetet létrehozni. Mivel a Unix tervezésénél kiemelt figyelmet fordítottak a rugalmasságra, a háttértárolókon található fájlok ábrázolásánál nem ezt a sémát alkalmazták. Az ötlet a következő: legyen egy úgynevezett fő (root) fájlrendszer (/), amelynek könyvtárai alá további fájlrendszereket be lehet fűzni. Így tulajdonképpen egy fájlrendszerben látjuk az összes lemezegegyünkön található összes állományt és könyvtárat. Ezzel a megoldással azonban némi gondunk akad – meg kell mondanunk a rendszernek, hogy a gépünkben található sok lemezerületből melyik látja el a fő fájlrendszer szerepét, illetve hogy mely könyvtárak alá fűzze be a többi lemezerületen lévő fájlrendszert. A Linuxszal most ismerkedő olvasóknak szánt sorozatunk mostani részében ennek módját próbáljuk bemutatni. Rendszerünk futása során fájlrendszerünkhöz bármikor új lemezegegyeket fűzhetünk, illetve a már feleslegessé váltakat kifűzhetjük. A Unix világában azt a műveletet, amelynek során új lemezegegyet fűzzünk be, „mountolásnak” nevezzük. A rendszer indításakor a fő lemezerület befűzéséről maga a rendszermag gondoskodik. Azt, hogy melyik lemezerületünk legyen a fő lemezerület, az általunk használt Linux indításkézelőjében (például LILO) kell beállítanunk. Természetesen terjesztésünk telepítőprogramja erről már gondoskodott, nekünk csak akkor szükséges megbolygatnunk, ha új fő lemezerületet szeretnénk létrehozni (a Linux indításkézelővel sorozatunk egy későbbi részében foglalkozunk). Előfordulhat, hogy nemcsak a fő lemezerületet szeretnénk befűzni, hanem például a Linux betöltésekor a DOS/Windows lemezerületeket is önműködően be akarjuk fűzni, vagy rendszerünket több lemezerületre telepítettük, ekkor használhatjuk a `/etc/fstab` fájlt. Az összes terjesztés lehetővé teszi, hogy pingvinünk ne egyetlen lemezerületen terpeszkedjen, hanem bizonyos részeit, mint például az alkalmazások binárisait tartalmazó `/usr`, vagy a felhasználói könyvtárakat tartalmazó `/home` könyvtárakat külön-külön lemezerületen helyezhessük el. Ez első látásra értelmetlennek tűnhet, valójában azonban nagyon sok gyakorlati haszna van, mint azt a későbbiekben látni is fogjuk.



Ahhoz tehát, hogy a rendszer indulásakor további lemezegegyek is önműködően befűzésre kerüljenek, értékeiket először a `/etc/fstab` fájlban meg kell adnunk. Nyissuk meg ezt az állományt! Figyeljünk arra, hogy ezt is, mint a rendszer összes beállítóállományát, csak a rendszergazda szerkesztheti. Az `fstab` felépítését vizsgálva láthatjuk, hogy minden bejegyzés külön sorban foglal helyet. Az összes terjesztés telepítője a legfontosabb dolgokat beállítja helyettünk, az intelligensebbek még a cserélhető lemezegegyeket (például hajlékonylemez, CD-ROM), illetve az idegen (nem linuxos) lemezerületeket is felveszik. Ha mégsem történt volna meg, ez a feladat ránk vár. Ám előbb ismerkedjünk meg az `fstab` felépítésével! Minden bejegyzés hat oszlopból áll. Az első oszlopba az adott lemezegegy nevét kell írunk. Sorozatunk korábbi részében már említettük, hogy a Unixban minden fizikai

eszközre az úgynevezett eszközfájlok segítségével hivatkozhatunk. Ezek olyan különleges állományok, amelyek a `/dev` könyvtár alatt találhatók. Minden eszközhöz egyedi eszközfájl tartozik. Minket jelenleg csak a lemezegegyeket azonosító eszközállományok érdekelnek, úgyhogy most csak azokat vesszük sorra. Az első IDE vezérlőnk master csatlakozójára rákötött eszköz neve `hda`, a slavere csatlakozó pedig `hdb`. A második IDE-meghajtó esetében `hdc`, illetve `hdd`. Merevlemez esetében a rajta található lemezerületeket is azonosítanunk kell, ezt az eszköz neve után írt számmal tehetjük meg. A szabály ez esetben egyáltalán nem bonyolult: 1–4-ig az elsődleges lemezerületeket azonosítjuk (egy merevlemez legfeljebb négy darab elsődleges lemezerületet tartalmazhat), 5-től felfelé pedig a logikai tartományokat vagy lemezerületeket (partition). Az SCSI merevlemezekenél `hd` helyett `sd`-t kell írunk. Az első hajlékonylemez meghajtó az `fd0`, a második az `fd1`. A második oszlopban az úgynevezett `mount point`-ot kell megadnunk. Ez nem más, mint az a könyvtár, amely alá az adott lemezegegy fájlrendszerét be szeretnénk fűzni. Figyelem! `Mount point`-nak csak létező könyvtárat adhatunk meg, tehát a befűzés előtt létre kell hoznunk egyet (például a `mkdir` paranccsal). A harmadik oszlop a fájlrendszer típusa. Linux-lemezerület esetében `ext2`, illetve az újabb rendszerek esetében már `ext3` vagy más típusú lemezerületet is használhatunk. Az `ext3` viszonylag új (ám már a 90-es évek elejétől létező módszerre épül), a „hagyományos” fájlrendszerek működésétől gyökeresen eltérő, úgynevezett naplózó fájlrendszer. Előnye az `ext2`-vel szemben, hogy több kisebb blokk írása esetében gyorsabb, és kevésbé sérülékeny az áramszünet okozta leállításokra. Windowsos lemezerület esetében a `vfat`-ot használjuk (DOS esetében az

Példa a /etc/fstab állományra

/dev/hda8	/	ext3	defaults	1	1
none	/proc	proc	defaults	0	0
/dev/hda9	swap	swap	defaults	0	0
/dev/cdrom	/mnt/cdrom	iso9660	noauto,ro,user	0	0
/dev/fd0	/mnt/floppy	auto	noauto,rw,user	0	0
/dev/hda1	/mnt/c	vfat	defaults	0	0
/dev/hda5	/mnt/d	vfat	defaults	0	0
/dev/hda6	/mnt/e	vfat	defaults	0	0
/dev/hda7	/mnt/f	vfat	defaults	0	0

msdos-t), CD-ROM-nál pedig az iso9660 kulcsszót. A következő oszlopban további szolgáltatásokat határozhatunk meg az egység csatolására vonatkozóan. Nézzük meg a legfontosabbakat:

- ro: az egységet kizárólag olvasásra fűzzük be. Az ily módon befűzött fájlrendszerbe még a rendszergazda sem írhat! Az esetleges vírusátadások megelőzéseként nem rossz ötlet, ha ilyen módon fűzzük be a binárisokat tartalmazó *usr* könyvtárat, ám új alkalmazás telepítése előtt írásra is be kell fűzni.
- rw: az egység írásra és olvasásra is be lesz fűzve. user/nouser: a rendszergazdán kívül más felhasználónak is megengedjük/megtiltjuk az egység ki- és befűzését.
- exec/noexec: az adott lemezegegről bináris futtatásának engedélyezése/tiltása. Jó szolgáltatásokat tehet a rendszergazdának, ha meg szeretné akadályozni, hogy a felhasználók saját – a /home-ban található – könyvtárukból maguk hozta programokat futtassanak.
- suid/nosuid: a sorozatunk előző részében bemutatott Set UID-es programok futtatásának engedélyezése/tiltása az adott egységről. Leginkább biztonsági célokat szolgál.
- auto/noauto: az adott egység a rendszer indulásánál önműködően be legyen-e fűzve vagy sem. Cserélhető lemezegegeknél a noauto kapcsolót használjuk!
- defaults: a következő kapcsolókat foglalja magában: rw, auto, nouser, exec és suid.

A maradék oszlopok értéke csak egész szám lehet. Az utolsó előtti a dump parancs használja annak meghatározására, hogy szükséges-e a fájlrendszerrel biztonsági másolatot készíteni. Véleményünk szerint ide 0-t érdemes írunk.

A hatodik és egyben utolsó oszlopban adhatjuk meg, hogy a fájlrendszer-ellenőrző fsck program milyen sorrendben ellenőrizze a lemezerületeket. Az 1-es az indító (boot) lemezerület. Amennyiben egy lemezerülethez 0-t írunk, nem fog ellenőrzésre kerülni.

Listánkon szerzőnk */etc/fstab* állományát láthatjuk. E sorozat írója ugyan tisztában van a Linux-rendszer minden előnyével, ám mégsem volt szíve megválni windowsos rendszerétől, sőt, mindennek tetejébe még merevlemezének első (primary) lemezerületére telepítette (*/dev/hda1*). Teljes Linux-rendszerének csupán egyetlen logikai tartományt szentelt (*/dev/hda8*). Láthatjuk, hogy az *fstab* állomány a fő lemezerület feltüntetésével kezdődik, a második sorban pedig a csereterület (swap) jellemzőinek megadása következik (*/dev/hda9*). A csereterületet is a telepítés során kell létrehozni, amelyet a rendszer virtuális memóriaként kezel. Szerzőnk gépében még egy hajlékonylemez-meghajtó (*/dev/fd0*), illetve egy, a második IDE-vezérlő master csatlakozójára kötött CD-ROM (*/dev/hdc*) is található. Mivel néhány dolgot a windowsos lemezerületeken

tárol, időnként szükséges, hogy ezekhez Linux alól is hozzáférjen, ám ezeket a lemezerületeket felesleges minden alkalommal önműködően befűzni. Szükség esetén a befűzést kézzel végezzük (lásd a későbbiek folyamán).

Szemfülesebb olvasóink biztos felgyeltek már a zavarbaejtő második sorra, ahol mountpoint-ként a */proc* van megadva. Ez nem valódi, hanem virtuális fájlrendszer, amelyet a */proc* könyvtár alá fűztünk be. Az itt található állományok az éppen futó

folyamatokról, illetve a rendszer mag különböző környezeti változóinak állapotáról tartalmaznak értékes adatokat. Ezeket az állományokat egyik felhasználó sem írhatja, csupán olvashatja. A */proc*-ra nagy szükség van, ugyanis ha nincs befűzve, rengeteg szolgáltatás elérhetetlenné válik, sőt, olyan alapvető parancsokat sem használhatunk, mint a ps.

Ezek után nézzük meg, miként tudunk lemezegegeket kézzel be- és kifűzni! A befűzésre a mount parancs használható. Ha az adott lemezegege már szerepel a */etc/fstab* állományban, akkor beállításaként elégséges csak a mountpoint-ot vagy az eszköz nevét megadnunk, például: mount /mnt/cdrom. Amennyiben mégsem írtuk bele, a mountpoint-on és az eszköznévén kívül a fájlrendszer típusát is meg kell mondanunk. Ezt a következő formában tehetjük meg: mount -t *fszjrendszer-t pus eszk zn0v* mountpoint.

A lemezegegegek kifűzését az umount utasítással végezhetjük. Itt kapcsolóként elég csak a mountpoint-ot megadnunk, ugyanis a */etc/mstab* nevű fájlban az összes idáig befűzött lemezegege adatai megtalálhatók.

Meg kell említenünk, hogy manapság már rengeteg segédprogram létezik a lemezegegegek ki- és befűzésének könnyebbé tételére, illetve az *fstab* állomány szerkesztésére. Legtöbbjük grafikus, mindenki számára könnyedén kezelhető, és a „felhasználóbarátabb” terjesztések önműködően fel is telepítik őket. Tehát aki nagyon elveszettnek érzi magát a befűzés világában, annak számára létezik segítség. Például a KDE-hez is számos apró alkalmazás készült.

Most már csak néhány lemezekezeléssel kapcsolatos segédprogramról kell szót ejtenünk. Ilyen például az fsck, amely linuxos fájlrendszerünk ellenőrzésére, illetve javítására szolgál. Az fsck általában minden 20. rendszerindításkor az összes linuxos lemezerületen ellenőrzést végez.

A linuxos lemezerületek formázására is lehetőségünk nyílik. Szerencsére ez a művelet FAT-es társaihoz képest sokkal gyorsabb. Ext2 lemezerületet az mkfs *eszk zn0v* utasítással formázhatunk.

Röviden ennyit a Linux alatti lemezekezelés rejtelméről. Sorozatunk következő részében ismét visszatérünk az általunk annyira kedvelt parancssorhoz, és kicsit magasabb szinten tekintjük át a héj- vagy parancs- (shell) műveleteket, továbbá alapszinten parancsfájlt írni is megtanulunk.

Garzó András

(garzoand@interware.hu) Körülbelül három éve foglalkozik Linux- és más Unix-rendszerekkel. Legjobban az operációs rendszerek lelkivilága érdekli, de nyitott egyéniség. Kedvence étele a palacsinta, és van egy Richard nevű macskája. Minden észrevétel, megjegyzés, levelet szívesen fogad.