

Linux fájlrendszerek kezelése Windows alól

Jó hír: nem kell a Microsoft fejlesztéseire várnunk, ha Windows alól is kezelni szeretnénk linuxos fájlrendszereinket...

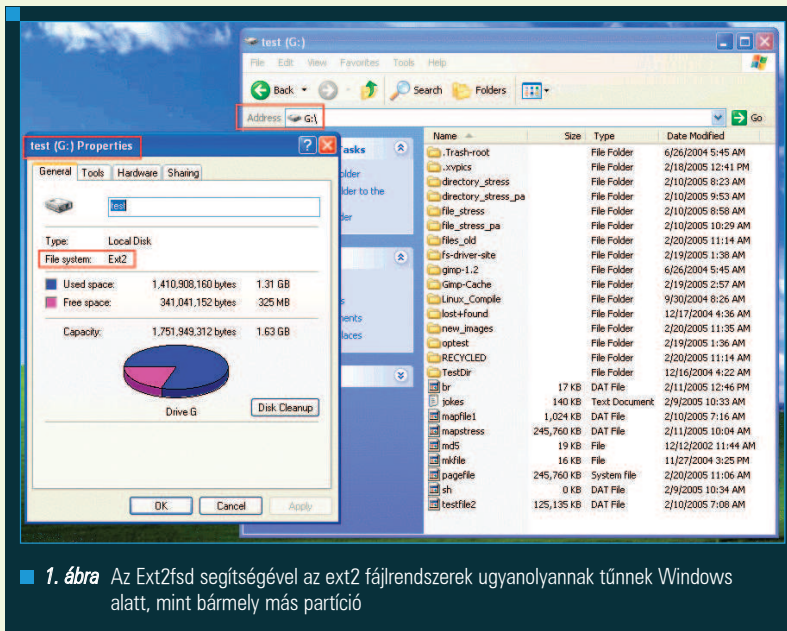
■ Ha egy eddig használt operációs rendszerről egy másikra szeretnénk váltani, annak gyakorta a legnagyobb gátja az a tudat, hogy az új rendszer alól nem fogjuk tudni elérni a régi alatt létrehozott adatokat. És ez valahol érthető is. Senkit nem kárhoztathatunk azért, mert nem fűlik a foga ahhoz, hogy egyik pillanatról a másikra elveszítse az összes adatát. Lehet, hogy az új rendszer szebb és jobb, mint a régi, de az adatok nélkül gyakorlatilag használhatatlan. Ami a *Linuxot* illeti, a fejlesztők mindent megtettek annak érdekében, hogy a *Windows*t is használók hozzáférjenek az ott létrehozott adataikhoz. A *FAT16/32* fájlrendszerek támogatása már régóta gyakorlatilag tökéletes, ami pedig az *NTFS*-t illeti, nos az még mindig nem teljes, de nem áll messze tőle. Itt érdemes megjegyezni, hogy már olyan eszközök is vannak – ilyen például a *Captive-NTFS* – amelyek teljes mértékben támogatják az *NTFS* fájlrendszerek *Linux* alóli kezelését. Az adatokhoz való hozzáférés ráadásul ebben az esetben sem korlátozódik a helyi gépre. *Samba* segítségével a felhasználók a hálózat más *Windows* futtató gépein megosztott könyvtárait is elérhetik. Bátran kijelenthetjük tehát, hogy ha valaki *Windowsról* szeretne *Linuxra* váltani, annak semmiféle problémára nem kell felkészülnie. Minden működni fog a migráció után. Ugyanakkor megfigyelhető egy új trend is. Ahogy a *Linux* egyre elterjedtebbé válik az asztali gépeken, úgy egyre több olyan felhasználóval találkozhatunk, akik bizonyos feladatokra *Linuxot* használnak ugyan, utána viszont újraindítják a gépüket,

Rövid bevezetés a fájlrendszerek elméletébe

Olvasóink közül talán nem pontosan tudják, mi is az a fájlrendszer. A fájlrendszer gyakorlatilag egy tárolási rendszer vagy módszer, amely meghatározza, miként kell a fájlokat a lemezre írni, illetve onnan visszaolvasni. Mármint ebből elég egyenes következik egy logikus kérdés: „Miért van annyiféle fájlrendszer? Miért nem tudják az emberek egyszer és mindenkorra eldönteni, hogy melyik a legjobb tárolási és visszaolvasási rendszer, és miért nem használja mindenki, minden operációs rendszer alatt azt?” Bár a kérdés a maga módján jogos, az igazsághoz az is hozzátartozik, hogy a különböző operációs rendszereknek eltérő a „célközönsége”, és az eltérő igényekhez eltérő fájlrendszerek igazodnak jobban. A *Windows NT*-t például üzleti felhasználóknak szánták, ezért ott nagy jelentősége volt a fájlrendszer biztonsági szolgáltatásainak. Így született meg az *NTFS*. A *Windows 9x* ezzel szemben a „közönséges felhasználóknak” készült, akiknek nem a biztonság, hanem a teljesítmény volt a fontosabb. Éppen ezért ez az operációs rendszer *FAT16/32* fájlrendszerrel működik. A *Windows XP* azért támogatja mindkettőt, mert mindkét célcsoport számára megfelelő szolgáltatásokat kell nyújtania. Hasonlóan *Linux* alatt is többféle fájlrendszert használhatunk. A de facto szabvány itt az *ext2/3* fájlrendszer használata, de léteznek teljesen speciális célokra tervezett adattárolási rendszerek is. Ilyen például a nagy számítási feladatokhoz használt *XFS*, vagy a nagyobb hibátűrést biztosító *ext3* és *ReiserFS*. Ha egy operációs rendszerbe be szeretnénk építeni egy adott fájlrendszer „támogatását”, akkor létre kell hoznunk benne azokat az adatszerkezeteket, amelyek a kérdéses tárolási rendszer kezeléséhez szükségesek. Ilyenkor természetesen sokkal nehezebb dolgunk van egy zárt szabványokra épülő fájlrendszerrel – mint amilyen például az *NTFS* – hiszen ismeretlen, vagy kevéssé ismert a lemezen elhelyezkedő adatok szerkezete, a használt titkosítási algoritmusok, és az egyéb, ezekhez hasonló tényezők. Pont ezzel magyarázható, hogy az *NTFS Linux* alóli támogatása miért jelent olyan nagy kihívást a fejlesztőknek.

és *Windows* alatt folytatják a munkát. Erre általában azért van szükség, mert bizonyos népszerű alkalmazásoknak továbbra sincs *Linux* alatt futó változata, vagy megfelelő képességekkel rendelkező linuxos helyettesítője. Ilyenek például a modernebb játékok, vagy az *Adobe Photoshop*, de más, speciális területekről is említhetnénk olyan példákat, amikor a nyílt forrású

megfelelő egyszerűen nem létezik. Ha tehát a felhasználó kényszerűségből oda-vissza váltogat a két operációs rendszer között, akkor nyilván lesz olyan pillanat is, amikor *Windows* alól szeretné elérni a *Linux* partíciók tartalmát. Ezzel csak egy a gond: a *Windows* hivatalosan semmiféle támogatást nem nyújt a *Linux* fájlrendszerek kezeléséhez. A több operációs



1. ábra Az Ext2fsd segítségével az ext2 fájlrendszerek ugyanolyannak tűnnek Windows alatt, mint bármely más partíció

rendszer telepítőknek tehát mindenképpen speciális eszközöket kell használniuk Windows alatt, ha boldogulni akarnak.

Linux partíciók kezelésére szolgáló eszközök Windows alatt

Amint korábban is említette, linuxos fájlrendszerekhez a Windows natív támogatást egyáltalán nem tartalmaz. Persze azért nincs minden veszve. A nyílt forrású közösség felvette a kesztyűt, és megalkotott néhány egészen kiváló eszközt, amelyekkel a probléma megoldható. Ebben a cikkben elsősorban az **LTOOLS** csomagra fogok koncentrálni. Ez egy egészen kifinomult eszköz, amelyhez többféle interfész is tartozik. Ezekkel a felhasználó több különböző, Linux alatt használatos fájlrendszertípust is kezelhet. Mielőtt azonban belekezdenék ennek a csomagnak a tárgyalásába, fussunk gyorsan végig az egyéb lehetőségeken is, amelyekkel a probléma szintén megoldható.

Ext2fsd

Az **Ext2fsd** az egyik legrégebbi olyan projekt, amely céljával a fenti feladat megoldását tűzte ki. Segítségével **ext2** fájlrendszerek tartalmához férhetünk hozzá Windows alól. A csomagot a <http://sourceforge.net/projects/ext2fsd> címről tölthetjük le. A program fájlrendszer-meghajtóként, és nem közönséges programként települ. Ennek

a módszernek az az előnye, hogy az **Ext2fsd** teljesen transzparens módon teszi elérhetővé az **ext2** fájlrendszereket Windows alól, vagyis első közelítésben semmi különbséget nem látunk valódi windowsos és linuxos partíció között. Az **Ext2fsd** nem csak olvasni, hanem írni is tudja az **ext2** és vele rokon fájlrendszereket. A Linux első de facto szabványos fájlrendszere ugyan az **ext2** volt, mára azonban számos egyéb tárolási rendszer is megjelent. Ilyen például az **ext3** és az újabban felbukkant **ext4**, amelyek azonban mindketten visszafelé kompatibilisek az **ext2**-vel. Ennek megfelelően az **Ext2fsd**-vel ezek is kezelhetők, bár ami azt **ext4**-et illeti, a helyzet egyelőre kisértelmentesen bizonytalan, hiszen egyelőre csak kísérleti jelleggel vették fel a kernelbe. Mindazonáltal az **Ext2fsd**-vel való kompatibilitás azért valószínű, mert amikor Windows alól kezelünk egy **ext3/4** partíciót, akkor a fájlrendszernek csak azokat a funkcióit használjuk, amelyek már az **ext2**-ben is megvoltak, vagyis elvileg nem lehet semmi gond. Ugyanakkor persze az is igaz, hogy az olyan kiterjesztett szolgáltatások, mint a naplózás így nem használhatók.

rfstool

Az utóbbi időben linuxos körökben egyre nagyobb népszerűségnek örvendő a **ReiserFS** fájlrendszer. Ennek elsősorban az általa biztosított nagyfo-

kú hibátűrész az oka. Ehhez is készült egy Windows alatt használható csomag, az **rfstool**, amely azonban csak olvasási műveleteket tud végezni, és a fejlesztők a hivatalos webhely tanúsága szerint nem is tervezik az írási műveletek megvalósítását. A csomagot a <http://fresmeat.net/projects/rfstool> címről szerezhetjük be.

LTOOLS

És ezzel meg is érkezünk a cikk fő témájához, az **LTOOLS** csomaghoz. Az **Ext2fsd**-től és az **rfstool**-tól eltérően ez a fájlrendszereknek nem csupán egyetlen speciális osztályát tudja kezelni, sokkal általánosabb annál. Az **LTOOLS** egyszerre támogatja az **ext2**, **ext3** és **ReiserFS** fájlrendszereket. A csomag tulajdonképpen parancssori eszközök gyűjteménye, de tartozik hozzá grafikus felhasználói felület (**GUI**), sőt webes felület is. Telepíthető gyakorlatilag valamennyi **DOS** és **Windows** változat alá (**XP**, **2000**, **NT**, **ME**, **9x** és **3.x**). Ráadásul nem csak a helyi gépen levő fájlrendszereket tudja kezelni, hanem a hálózati megosztásokat is. Akár **DOS**, akár valamilyen **Windows** alatt dolgozunk is tehát, biztos számíthatunk az **LTOOLS** eszközeire, ha Linux fájlrendszeren levő adatokhoz kell hozzáférni. Az **LTOOLS** amúgy akkor is jól jöhet, ha úgy kell megajánunk egy Linux rendszert, hogy éppen nincs kéznél egy megfelelő **Live CD**. Amint az imént említettem, az **LTOOLS** csomag két különböző grafikus felületet is tartalmaz, amelyekkel a parancssori eszközök teljes funkciókészletét kihasználhatjuk. Van hozzá egy **LTOOLSgui** nevű, Java alapú interfész, illetve létezik egy Microsoft .NET alapú **LTOOLSnet** nevű alkalmazás is. Mindkettővel kezelhetünk helyi és távoli fájlrendszereket egyaránt. Aki még ezzel sem elégedett, például mert kizárólag szabad szoftvereket szeretne használni (sem a Java, sem a Microsoft .NET nem az), az akár egy közönséges webböngészőn keresztül is elérheti az **LTOOLS** szolgáltatásait. Ez utóbbi azért lehetséges, mert a csomag tartalmaz egy **LREADsrv** nevű alkalmazást, ami nem más, mint egy egyszerű webkiszolgáló. Ez a Linux fájlrendszereit bármely webböngésző, akár az Explorer számára is láthatóvá

tudja tenni. Az *LREADsrv* segítségével ráadásul nem csak a *Linux*, hanem a *DOS/Windows* fájlrendszereit is elérhetővé tehetjük a hálózaton keresztül.

Az LTOOLS telepítése

Az *LTOOLS* egy szabványos *Windows* telepítőcsomag formájában érkezik, ami kifejezetten időszerűnek nevezhető. A telepítés folyamatában sincs semmi említésre méltó. A telepítő létrehoz a *Start* menüben egy *LTOOLS* nevű pontot, amely alatt aztán megtaláljuk az összes említett interfészt, amelyeken keresztül máris hozzáférhetünk linuxos fájlrendszereink tartalmához.

Bár – amint említettem – maga az *LTOOLS* támogatja ugyan az összes *Windows* változatot, nem minden szolgáltatása illetve interfésze működőképes valamennyi verzió alatt. A csomag eleve két különböző konzolos változatot tartalmaz a *Win9x/ME* illetve a *Windows NT/XP* rendszerekhez.

Az LTOOLS szolgáltatásainak áttekintése – A parancssori eszközök

A parancssori eszközökkel a leg-
alapvetőbb fájl- és könyvtárkezelési feladatokat végezhetjük el. Mint említettem, a parancsok a linuxos fájlrendszerek írására és olvasására egyaránt használhatók. Minden parancshoz három fájl tartozik. Nézzünk talán egy példát. Az

```
l dir .bat
```

parancs például listázza a könyvtárak tartalmát, ez azonban – mint nevéből is látszik – nem egy program, hanem egy szkript. Ez a szkript – attól függően, hogy milyen operációs rendszer alatt futtatjuk – maga is meghív az *LTOOLS* csomag részét képező programokat. Jelen esetben ténylegesen vagy az *LdirDOS.exe*, vagy az *LdirNT.exe* fog lefutni. Az első a *Win9x/ME* rendszerekre vonatkozik, míg a második a *Windows NT/XP* operációs rendszerekhez való. A legtöbb *LTOOLS* parancs szintaxisa teljesen logikus. A partíciókat például ugyanúgy kell megneveznünk, mint *Linux* alatt. Ha mondjuk a *vars.inc* nevű fájl akarjuk átmásolni a */root* fájlrendszerről (ami fizikailag

1. táblázat *Az LTOOLS parancsai*

<code>lread.bat</code>	Fájlok olvasása és másolása Linux partícióról DOS partícióra. Példa: <code>lread.bat --s/dev/hda2 /root/vars.inc C:\vars.inc</code>
<code>lwrite.bat</code>	Fájlok másolása DOS-ról Linuxra. Példa: <code>lwrite.bat --s/dev/hda2 C:\vars.inc /root/vars.inc</code>
<code>lde1.bat</code>	Fájlok vagy üres könyvtárak törlése Linux fájlrendszerről. Hatása tulajdonképpen azonos az <code>rm -f</code> illetve <code>rmdir</code> parancsokéval. Példa: <code>lde1.bat /root/vars.inc</code>
<code>lchange.bat</code>	Fájlok tulajdonságainak és tulajdonosának megváltoztatása Linux alatt (a <code>chmod</code> parancs megfelelője). Példa: <code>lchange.bat --s/dev/hda2 754 /root/vars.inc</code>
<code>lren.bat</code>	Linux fájlrendszeren található fájlok átnevezése (az <code>mv</code> parancs analógja). Példa: <code>lren.bat --s/dev/hda2 /root/vars.inc /root/var2.inc</code>
<code>lmkdir.bat</code>	Könyvtár létrehozása a Linux partíción (az <code>mkdir</code> megfelelője). Példa: <code>lmkdir.bat --s/dev/hda2 /root/newdir</code>
<code>ln.bat</code>	Szimbolikus lánc létrehozása (az <code>ln</code> megfelelője). Példa: <code>ln.bat --s/dev/hda8 /root/link /root/vars.inc</code>
<code>lcd.bat</code>	Könyvtárváltás (a <code>cd</code> megfelelője). Példa: <code>lcd.bat /home/</code>
<code>ldrive.bat</code>	Az alapértelmezett Linux lemezmeghajtó kiválasztása. Példa: <code>ldrive /dev/hda8</code>

mondjuk a */dev/hda2* partíciót jelenti) a *C:* meghajtóra, akkor a következőt kell tennünk:

```
lread.bat --s/dev/hda2 /root/  
vars.inc C:\vars.inc
```

Teljesen hasonlóan ha ezt a fájlt vissza akarjuk írni a *Linux* fájlrendszerére, a következő parancsot használhatjuk:

```
lwrite.bat --s/dev/hda2  
vars.inc /root/vars.inc
```

Az *LTOOLS* csomag parancsairól az 1. Táblázatban találunk összefoglalót. A fent felsorolt eszközök a *ReiserFS* fájlrendszereket nem támogatják. Éppen ezért az *LTOOLS* csomag tartalmazza az *rfstools* programot is, amit kifejezetten erre a célra írtak. Ahhoz, hogy *Windows NT/2000/XP* vagy *UNIX/Linux* alatt a merevlemez tartalmát olvashassuk, rendszergazdai jogosultságokra van szükségünk. Ha az *LTOOLS* parancsait egy nem adminisztrátori felhasználói fiókból futtatjuk, nem fogunk tudni hozzá-

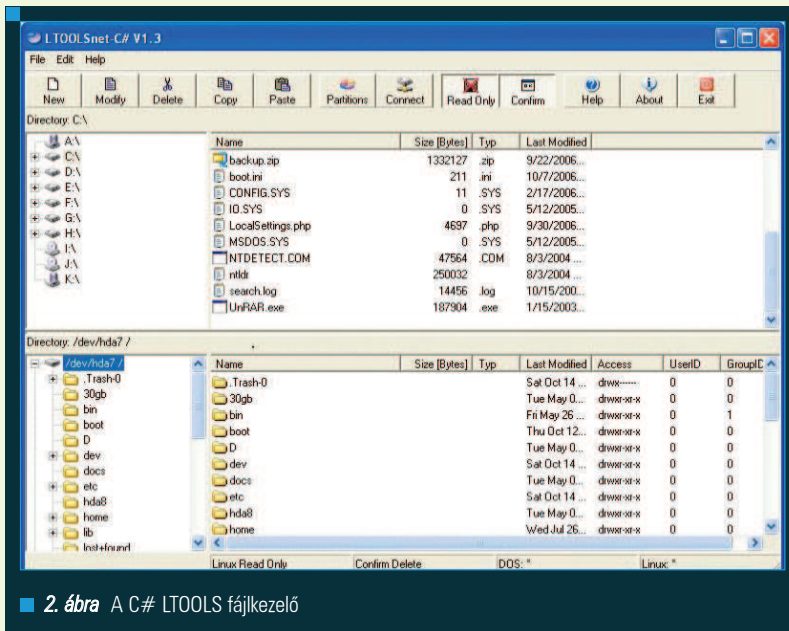
férni a lemez tartalmához.

Az *LTOOLS* emellett nem foglalkozik a *Linux* alatt érvényes „tulajdonviszonyokkal”. Ez egyben azt is jelenti, hogy ha egy felhasználó képes hozzáférni a gyökeri fájlrendszerhez, akkor azon azt alakít át, amit akar – beleértve persze a */etc/passwd* tartalmát is.

Az LTOOLS .NET keretrendszerre támaszkodó ügyfélprogramja

A *.NET* keretrendszerre támaszkodó ügyfélprogram az *LTOOLS* szolgáltatásokban leginkább bővelkedő része. Futtatásához természetesen le kell töltenünk a *Microsoft .NET* keretrendszert a <http://www.microsoft.com/downloads> webhelyről.

Az ügyfél lehetővé teszi valamennyi *Windows* és *Linux* fájlrendszer böngészését. Szabadon átvihetünk köztük fájlokat, törölhetjük, szerkeszthetjük, vagy bármilyen más módon megváltoztathatjuk azokat. Arra is lehetőségünk van, hogy távoli fájlrendszereket becsatoljunk, és azok tartalmát kezeljük. Én magam ezt a lehetőséget



2. ábra A C# LTOOLS fájlkezelő

rendkívül hasznosnak találtam, amikor a webkiszolgálómmal volt némi problémám. *Windows* alól is minden további nélkül be tudtam csatolni a webes dokumentumokat tartalmazó fájlrendszert, és elvégezhettem a szükséges változtatásokat.

Az LTOOLS Java keretrendszerre támaszkodó ügyfélprogramja

Aki *Windows95/98/ME* operációs rendszert használ, az nem tudja telepíteni a *Microsoft .NET* keretrendszerét, mert annak nem készült ezekhez a rendszerekhez megfelelő változata. Ilyenkor jön jól a Java felület. Persze ehhez is le kell töltenünk egy *Java* futtatókörnyezetet (*Java Runtime Environment*), amit a java.sun.com címen találunk meg. A *Java* interfész funkciói mindenben megegyeznek a *.NET* kliens szolgáltatásaival.

Az LTOOLS web-alapú felülete

Tapasztalataim szerint az *LTOOLS* leghasznosabb kezelőfelülete a hozzá mellékelt webes szolgáltatás. Az *LTOOLS* csomag tartalmaz egy *LREADsrv.exe* nevű beépített webkiszolgálót, amit a felhasználó elindíthat, és segítségével egy közönséges webböngészőn keresztül kezelheti a fájlrendszerek tartalmát. Ennek óriási hasznát vehetjük akkor is, ha távoli felhasználókkal szeretnénk fájlokat megosztani. Ugyanakkor biztonsági szempontból nem javasolt az

LREADrv.exe futtatása egy globálisan hozzáférhető kiszolgálón. Ilyenkor ugyanis esetleg olyanok is hozzáférhetnek a fájlrendszer tartalmához, akikkel nem akarunk adatokat cserélni. Az *LTOOLS* saját kiszolgálóját tehát célszerűen csak olyan környezetben futtassuk, ahol egyéb módon biztosított, hogy csak a megfelelő felhasználók férhetnek hozzá a megosztott tartalmakhoz. Ilyen környezet például egy virtuális magánhálózat. Az *LREADsrv.exe*-nek amúgy vannak eleve ismert hibái, amelyeket a fejlesztők a későbbi változatokban bizonyára ki fognak javítani. Ennek megfelelően a program a fejlesztők szerint is alfa állapotú, és vannak bizonyos képességbeli korlátai is. Ezek között egyesek éppen a *HTTP 1.1*-gyes webböngészőkkel való kompatibilitást érintik, amilyen az *Internet Explorer* is. Ilyenkor a kiszolgáló válaszüzeje megnövekszik. Egy másik, adott esetben kifejezetten zavaró korlát abból fakad, hogy az *LREADsrv* program jelen változatában egyszerű, ami azt jelenti, hogy ha egyszerre többet használják, akkor valamennyi változtatás hatása globális. Így versenyhelyzetek alakulhatnak ki, illetve bizonyos szituációkban elveszhetnek módosítások. Ráadásul az *LREADsrv* hibaellenőrzése is meglehetősen gyenge. A legtöbb felhasználói bemenetet (például a fájlneveket és ehhez hasonló információkat) egyáltalán nem

ellenőrzi. Ha tehát a felhasználó elgépel egy fájl nevét, vagy tévesen adja meg egy partíció nevét, a webkiszolgáló instabillá válhat. Az egyetlen jó hír, hogy ez adatvesztést biztosan nem fog okozni.

Összefoglalás

Összességében elmondható, hogy a *Linux* felhasználóknak egyre jobb eszközök állnak rendelkezésére a *Windows* fájlrendszerek kezeléséhez. Szerencsére létezik az ellenkező irányú hozzáférés is, hiszen mint láttuk, az *LTOOLS* számos különféle *Linux* fájlrendszerhez ad hozzáférést *Windows* alól, a legkülönfélébb felhasználói felületeken keresztül. Ugyanakkor az is igaz, hogy maradt még tér a továbbfejlődéshez is, hiszen bizonyos nem is olyan ritkán használt fájlrendszer-típusok *Windows* alatti támogatása ma még meglehetősen hiányos, vagy egyáltalán nem létezik. Ilyen például az *XFS*. Az olyan meghajtók fejlesztése, amelyek a *Linux* fájlrendszerek olyan speciális szolgáltatásait mint például az *ext3* vagy a *ReiserFS* naplózása is képesek kihasználni, egyelőre gyerekcipőben szintén jár. A *Linux* fájlrendszerek és a *Windows* operációs rendszer közötti átjárhatóság megvalósítása ugyanakkor rendkívül fontos feladat, hiszen ennek hiánya sokakat megakadályozhat a nyílt forrású rendszerekre való váltásban. Összességében tehát a nyílt forrású közösségnek tudomásul kell vennie, hogy a *Windows* egyelőre dominánsnak tekinthető az asztali gépeken, és a két rendszer közötti kölcsönös átjárhatóság biztosítása elemi érdekünk.

Linux Journal 2007., 154. szám

Irfan Habib egyetemi hallgató.

Informatikát tanul a Pakisztáni Nemzeti Tudományos és Műszaki egyetemen (National University of Sciences and Technology, Pakistan). Évek óta nagyon komolyan foglalkoztatja a szabad és nyílt forrású szoftverek világa. Gyakran dolgozik heterogén számítógépes rendszerekkel. Éppen innen ered a különböző operációs rendszerek és fájlrendszerek közti átjárhatósággal kapcsolatos érdeklődése.