

Adataink biztonságos tárolása és mentése

Akivel már megtörtént, hogy fontos adatot veszített bármilyen okból kifolyólag, az egészen biztosan kínosan ügyel arra, hogy még egyszer ilyen elő ne fordulhasson vele. Akivel ez még nem történt meg, az egyfelől nagyon szerencsés, másfelől jobb ha okos ember módjára más hibájából tanul, nem a sajátjából.

© Kiskapu Kft. Minden jog fenntartva

Statisztikailag ugyanis minél többet foglalkozunk a számítógépünkkel, annál nagyobb az esélye annak, hogy egyszer velünk is megtörténik a katasztrófa és erre jobb felkészülni, mint akkor kapkodni fűhöz, fához.

Adatbiztonság

Az adatbiztonság egy meglehetősen összetett kérdés, több pusztán adataink tárolásánál. Amikor adatbiztonságról beszélünk, akkor az adatok tárolásáról, azok hozzáférhetőségéről beszélünk annak a kockázatnak a vizsgálata mellett, hogy illetéktelenek miként tudnak az adatainkhoz hozzáférni, miként tudják azokat ellopni, módosítani, mások felé minket megsemmélyesíteni. Adataink biztonságos tárolása ugyan nem az egyik legfontosabb pontja a klasszikus értelemben vett adatbiztonságnak, ám a hétköznapi számítógép használat alkalmával legalább olyan fontos kérdéssé válik, mint az illetéktelen hozzáférés kiszűrése.

Biztonságos adattárolás – módszerek és technológiák

Adataink biztonságos tárolásáról ma már hétköznapi eszközök igénybevételel is meglehetősen egyszerűen gondoskodhatunk. Gondoljunk csak a legegyszerűbb megoldásra: ma már minden háztartásban fellelhető egy CD-író eszköz. Ha ezt rendszeresen használjuk adataink archiválására, már sokat tettünk állományaink biztonságáért érdekében.

Érdeemes tehát legalább a legegyszerűbb eszközöket arra használni, hogy

adatainkat biztonságba helyezzük.

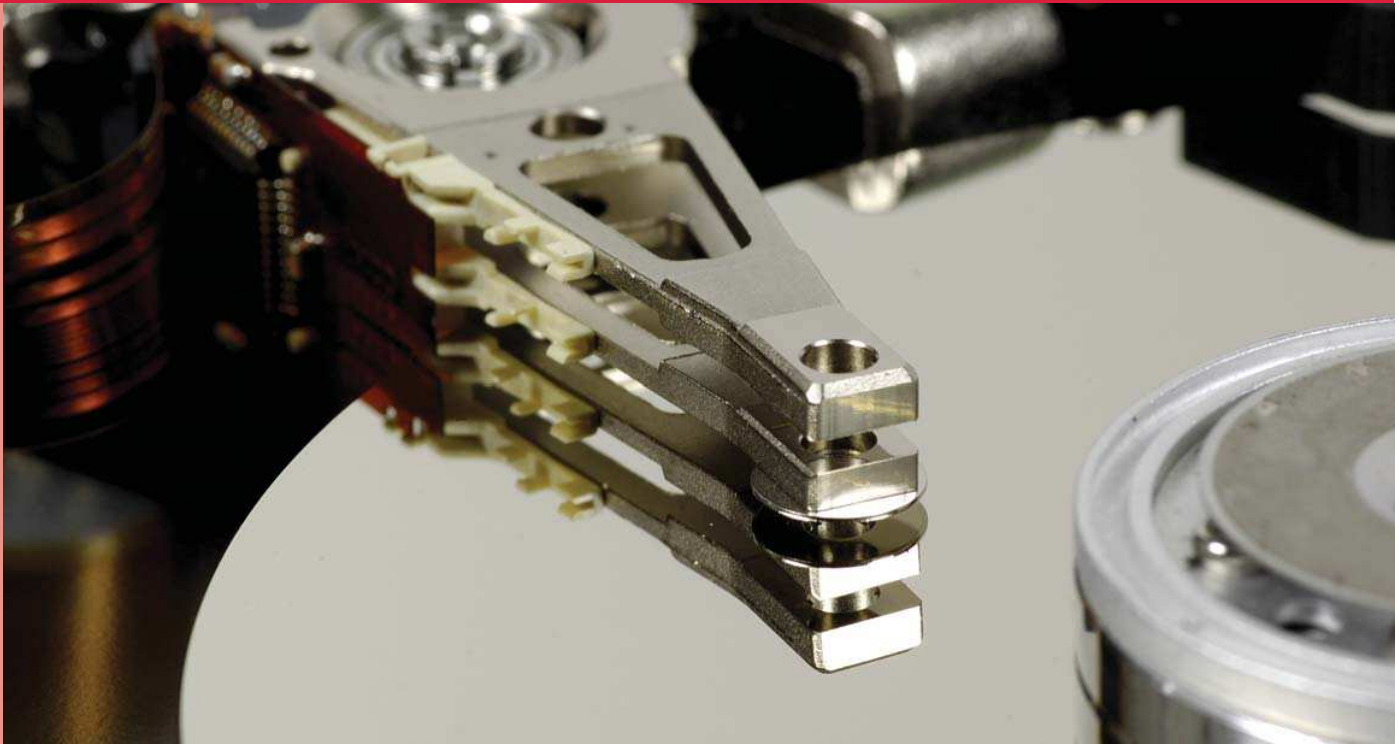
Azt azonban nem árt átgondolni, hogy melyik médiára milyen típusú adatokat írunk ki. Ha a fényképalbumunkat szeretnénk archiválni, akkor arra nem érdemes többször írási médiumot használni, ezeket ugyanis egyszer kiírjuk, utána maximum visszanézzük a képeket. Itt jegyezném meg, hogy az írási CD-n és DVD-n elhelyezett adatainkat is érdemes néha újra és újra megnézni, mert sajnos az idő múlásával ezek a lemezek veszítenek minőségükből, így elképzelhető, hogy egy-egy fényképgyűjtemény a kukába kerül, mert négy-öt év múlva egyszerűen nem fogjuk tudni visszaolvasni az állományokat. Persze okos ember erre is felkészül, néha újra kiírja az ilyen fontos anyagokat, továbbá az is jó megoldás lehet, ha több másolatot is készítünk egy-egy lemezből.

A fényképekkel, családi videókkal ellentétben az általunk készített dokumentumokat érdemes olyan médiára írni, amelyik többször írási. Ez azért jó megoldás, mert általában a dokumentumokat tartalmazó könyvtárunk nem olyan struktúrában készül el, hogy magán a könyvtárstruktúrán követni lehessen az idő változását, így két adott időpont között készült állományok mentése problémás

lehet. De ha ügyesek voltunk és lementettünk, akkor egy ilyen formában készült mentésen bármilyen megalátni az egy újabb hívás lesz.

Érdeemes tehát ilyenkor az adott könyvtárstruktúrát, vagy annak részleteit egyben időnként egy többször írási lemezre menteni. Ez a többször írási lemez lehet egy CD-RW, egy DVD-RW lemez, de talán praktikusabb egy DVD-RAM-ot, vagy külső merevlemez, esetleg flash meghajtót használni. Hogy miért? Rengeteg időt megspórolhatunk több gigabájt mentésekor, ha úgy készítjük a mentést mondjuk egy második merevlemezre, hogy a mentett dokumentumok mappáinkat oly módon írjuk felül az eredeti dokumentumok mappával, hogy ott csak a módosult





állományok kerüljenek mozgatásra. Ezzel rengeteg időt megspórolhatunk például egy **USB**-n csatlakoztatott külső merevlemez esetén.

A RAID

Ugyan nem kapcsolódik szorosan mostani témánkhoz, de mindenképpen meg szeretném említeni, hogy a biztonságos adattárolás bizonyos esetekben semmit nem ér anélkül, hogy magán a gazda számítógépen is biztonságban legyenek az állományok. Erre nyújt megoldást a redundáns lemezkezelés, ismertebb nevén a **RAID (Redundant Array of Independent Disks)**. A **RAID** egy nagyon jó megoldás abban az esetben, ha a gazda számítógépünkön – például egy kiszolgálón – elviselhetővé szeretnénk tenni azt a problémát, amit egy merevlemez meghibásodása, kiesése okozhat. A **RAID** éppen arra nyújt különböző konfigurációkban megoldást, hogy egy, vagy több lemez kiesése esetén is működőképes maradjon a rendszer.

Gyorsan nézzük át, hogy melyek a legnépszerűbb **RAID** konfigurációk:

RAID 0

Napjainkban már sok középkategóriás alaplap is támogatja a **RAID 0** konfigurációt, ám attól eltekintve, hogy a neve **RAID**, nem sok használat van a biztonságos adattárolás területén.

A **RAID 0** konfiguráció arra használható, hogy több független lemezt úgy fűzzünk össze, hogy azok a végén az operációs rendszer számára egy logikai lemezként látszódjanak. Így a sok kis merevlemezünk-ből tudunk egy nagy logikai lemezt készíteni. Ám bármelyik lemez kiesése esetén az egész tömb használhatatlanná válik.

RAID 1

Talán a legegyszerűbb **RAID** konfiguráció. Szintén sok középkategóriás alaplap támogatja, valamint nagyon jó támogatással rendelkezik a Linux rendszermagba építve is. A módszer lényege, hogy veszünk legalább két ugyanolyan lemezt, vagy szoftveres megoldás esetén ugyanakkora partíciót és ezeket úgy használjuk, hogy tükrözzük egymásra az adatokat. Így tehát egy elmentett állomány nem csak az egyik lemezen kerül elmentésre, hanem a tömb minden lemezén. Az eredmény, hogy attól függően, hogy hány lemezt tettünk a tömbbe, $n-1$ lemez kiesés esetén a rendszer még használható. Ellenben a tömb teljes tárterületeként annyit látunk csak, mintha egy lemezt használnánk csak. Ez a biztonság ára.

A **RAID 1** egy nagyon egyszerű és könnyen használható megoldás, akár hardveres, akár szoftveres megoldás esetén.

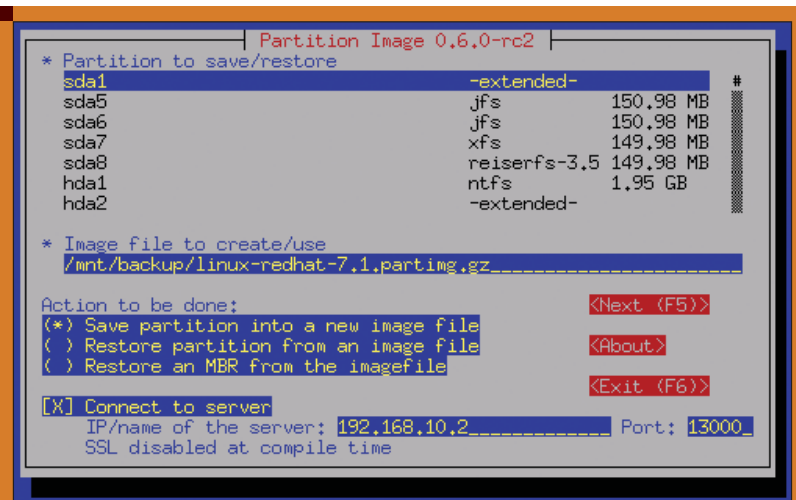
RAID 10

A **RAID 10** a **RAID 0** és **1** ötvözése, oly módon, hogy a **RAID 0** által létrehozott logikai meghajtókat tükrözéssel teszi redundáns állományrendszeré. A **RAID 10** egy jó megoldás volna, de van ennél egy sokkal frappánsabb és olcsóbb megoldás.

RAID 5

A **RAID 1** mellett a legelterjedtebb redundáns megoldás. Óriási előnye a **RAID 1**-hez képest, hogy amellett, hogy redundáns, tetszőlegesen nagy logikai lemezek hozhatók létre a segítségével. A **RAID 5** konfigurációhoz legalább három darab egyforma lemezre van szükségünk és ezt a három lemezt fűzzük össze egy logikai meghajtóvá oly módon, hogy a teljes tárterület kétharmada fog rendelkezésre állni, mint szabad terület. A működés egy zseniálisan egyszerű ötleten alapul, a logikai **XOR** (kizáró vagy) műveletet használja. Vesz két bitet, amit el akar tárolni, ezeket kiírja az első és a második lemezre, majd a két bit **XOR** művelettel vett eredményét a harmadikra. Ennek a megoldásnak óriási előnye, hogy a **XOR** művelet miatt bármelyik lemez kiesik, a másik két lemezről egyértelműen előállítható az eredeti tartalom. A **RAID 5** rendszerbe természetesen nem csak három, hanem annál több lemez is összefogható egy logikai egységgé, valamint a redundancia

© Kiskapu Kft. Minden jog fenntartva



■ 1. ábra Partimage – Mentési folyamat indítása

mértéke növelhető úgynevezett spare, tartalék lemezek beállításával. A *Linux* kernel ezt a *RAID* megoldást is támogatja rendszermag szinten, így szoftveres *RAID 5* lemezeket is használhatunk. Ilyenkor mindössze arra kell figyelni, hogy a *Linux* rendszerek jelenleg nem tudnak szoftveres *RAID 5*-ről indulni, ezért létre kell hoznunk egy kisebb partíciót a rendszertöltő és a rendszermag számára. Számolni kell továbbá azzal a plusz terheléssel, amit a bitenkénti *XOR* művelet jelent a processzor számára.

Biztonsági mentés Linux alatt

Térjünk rá a cikk fő témájára, a linuxos adatmentő szoftverekre. Két fő szoftvercsoportot fogunk megvizsgálni, mindkettőnek egy-egy képviselőjét. Az első csoport azokat a szoftvereket foglalja magában, amely szoftverek segítségével egy fizikai lemezről olyan képállományt tudunk készíteni, amelyet utána bármely másik számítógépre vissza tudunk tölteni. A második csoportba pedig azok a programok tartoznak, amelyek segítségével élő rendszereket tudunk lementeni úgy, hogy ahhoz semmilyen szolgáltatást nem kell leállítani.

Partimage

Bizonyára sokan ismerik és talán többen használták is már korábban a *Symantec*-féle *Ghost*-ot arra, hogy számítógépükről biztonsági mentést készítsenek. Ezzel a programmal könnyedén meg lehetett csinálni, hogy időnként az ember elmentette

a számítógép teljes tartalmát és amikor valami probléma előjött, akkor gyorsan, percek alatt visszavarázslottunk egy korábban készített mentést és ezzel megúsztuk egy órákig, vagy éppen napokig tartó újratelepítést. Nos a *Ghost* annak idején egy nagyon jópofa szoftver volt, amelyből aztán egy meglehetősen komoly üzleti rendszert fejlesztett a *Symantec*, csak éppen a magánszemélyek és kisebb vállalkozások számára nem biztos, hogy megfizethető áron. Persze elkeseredésre most sincs semmi okunk, mert itt van nekünk a *Partimage*! Ez a program tökéletesen helyettesíti az előbb említett termékcsaládot és mint az egy jó linuxos rendszerhez illik nyílt forráskódú és ingyenesen használható. Működését tekintve a *Partimage* legfontosabb szolgáltatása, hogy számos

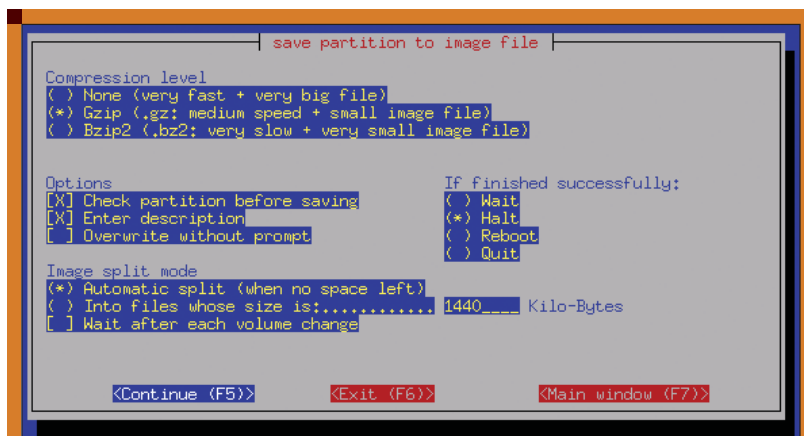
különböző állományrendszerről képes úgynevezett lenyomatot (image) készíteni, ezt tetszőlegesen lokális, vagy hálózati erőforráson elhelyezni, illetve ilyen image-ből a lementett rendszert percek alatt visszatölteni.

A programhoz tartozik egy nagyon jól használható konzolos felhasználói felület, így lépésről lépésre tudjuk beállítani a mentés folyamatát, de természetesen a program a megfelelő paraméterekkel parancssorból is futtatható, ennek megfelelően nagyon jól szkriptelhető.

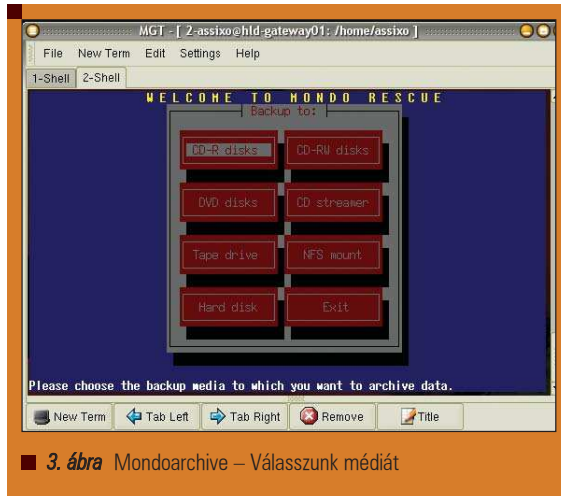
A program indítása után kiválasztjuk, hogy melyik partíciót szeretnénk menteni és megadjuk, hogy hol szeretnénk ezt az állományt elhelyezni.

Erre rögtön két mód is kínálkozik, az első, hogy egy a lokális állományrendszeren megadjunk egy könyvtárat és egy állományt, vagy pedig egy távoli kiszolgálón helyezzük el a mentést. Utóbbi esetben több dologra is szükségünk van. Először is telepíteni kell a *partimaged* csomagot a kiszolgálóra, ahová a mentést tenni szeretnénk, valamint rendelkezniük kell az adott gépre olyan hozzáféréssel, hogy ezt a mentési műveletet végrehajtsuk.

Távoli kiszolgáló használata esetén lehetőség van arra, hogy *SSH* segítségével titkosítsuk a kommunikációs csatornát. Tulajdonképpen ez a megoldás pofon egyszerűen működik, bárki percek alatt elsajátítja a használatát, ám mégsem biztos, hogy ez a legjobb megoldás, ugyanis meglehetősen lassú ez a mentési mód. Amennyiben lehetőségünk van rá, kiszolgálóra való mentéshez lokális hálózaton inkább *NFS* megosztást használjunk, csatoljuk be a megosztást a megfelelő helyre



■ 2. ábra Partimage – Mentési paraméterek beállítása



■ 3. ábra Mondoarchive – Válasszuk médiát

és után mint egy lokális mappába, úgy mentjük le a gépet. A sebességben nagyságrendi eltérést érhetünk így el. Ha kiválasztottuk a mentés helyét, akkor a következő képernyőn választanunk kell a tömörítés módozatai közül, a használható legnagyobb mentési állományméretről, valamint egyéb a mentés folyamatát érintő opciók közül. Tömörítésnek érdemes a *gzip*-et használni, mert bár a *bzip2* tömörítés kisebb állományt eredményez, nagyon lelassítja a mentést. Persze aki nek kevés a helye és sok az ideje... Ha mindezzel megvagyunk, akkor még adhatunk egy leírást az adott állománynak és indulhat a mentés. Ha a program végzett, akkor erről értesít minket és visszalép a terminálba. Miután megvan a mentésünk, nincs is más hátra, mint a visszaállítás. Ehhez ugyanúgy futtassuk a *partimage* parancsot és a megjelenő ablakban adjuk meg a visszaállítandó partíció helyét, az állományt, amelyből visszaállunk, majd válasszuk ki visszaállítás funkciót. A következő képernyőn még választhatunk, hogy csak szimulációt végzünk, vagy ténylegesen visszaállítjuk a rendszert és miután indítottuk a visszaállítást egy utolsó figyelmeztetés után a rendszer felülírja a kiválasztott partíciót. A *Partimage* program használható az összes linuxos állományrendszer, valamint *FAT16*, *FAT32* és *NTFS* partíciók mentésére és visszaállítására. Tapasztalatunk szerint a program tökéletesen működik, hibát eddig nem tapasztaltunk nála. A program nagy előnye, hogy megfelelő szkripteléssel akár arra is használható, hogy egy nagyobb rendszerben egy parancs futtatására

program és egy megfelelő *NFS* kiosztással már indulhat is a telepítés.

Éles rendszer mentése

A *Partimage*-el az előbb megtanultunk könnyen és gyorsan biztonsági mentést készíteni a munkaállomásunkról, most viszont nézzük meg, hogy hogyan tudjuk lementeni a szervereinket úgy, hogy rendszerösszeomlás esetén is percekben, de legalábbis órákon belül újra működőképesek legyünk. Linuxról biztonsági mentést készíteni nem egy nagy ördögösség, egy egyszerű másolással is meglehetősen jó mentést lehet készíteni, a *tar* program használatáról nem is beszélve. Azonban vannak már olyan programok, amelyek az esetlegesen zárolt állományok mentését, vagy a mentés alatti változásokat is jól kezelik, arról nem is beszélve, hogy akár egy tucatnyi mentési médiát is támogatnak. Az én kedven ilyen programom a *Mondo*. Amellett, hogy egy nagyon barátságos és könnyen használható felhasználói felület tartozik hozzá, amely mellett természetesen nagyon jól kezelhető parancssorból is, rengeteg mentési módot ismer. Tud menteni szalagos meghajtóra, *CD*-re, *DVD*-re, *ISO* állományba, *NFS*-re és természetesen lokális állományba is. Emellett elég jól állítható a menteni kívánt és a mentésből kihagyni kívánt állományok és könyvtárak listája. A program két részből áll, a *mondoarchive* és a *mondorestore* parancsokkal

indítható. A *mondoarchive* rész való a rendszerünk mentésére, míg a *mondorestore* a visszaállítási folyamatokhoz. A *mondoarchive* indítása után azonnal kiválaszthatjuk, hogy hová szeretnénk a mentést készíteni. Én legtöbb esetben a kiszolgálókat szalagos meghajtóra mentem, szerintem ez a legjobb módja a biztonsági mentés készítésének, de természetesen bármely más módozat is választható. Miután kiválasztottuk a médiát a mentéshez, a program rákérdez, hogy milyen tömörítési szintet szeretnénk használni. Itt is elmondható, amit a *partimage*-nél már említettem, ha van idő a mentésre, akkor érdemes a nagyobb tömörítést választani. Ha ezzel megvagyunk, akkor meg kell adni a menteni kívánt könyvtárstruktúra kezdőpontját. Egy teljes mentés esetén ez a gyökérkönyvtár, tehát a */*. Következő lépésben megadhatjuk azokat a könyvtárakat, amelyeket az előbb megadott struktúrában belül nem szeretnénk menteni. Ilyenek lehetnek az ideiglenes állományok tárolására szolgáló könyvtárak, vagy a töménytelen mennyiségű zenét, filmet tartalmazó könyvtárak. Ha ezzel is végeztünk, a program rákérdez, hogy szeretnénk-e a mentés végeztével ellenőrzést tartani a biztonsági mentés állapotáról. Ez ugyan tovább növeli a mentés idejét, de mindeképpen megéri elvégezni ezt a folyamatot, ugyanis rossz dolog, ha egy mentés használhatatlanságával akkor szembesülünk, amikor használni kéne. Következő lépésben arról kell nyilatkoznunk, hogy szabványos



■ 4. ábra Így indul az Unison – Válasszuk profilt

© Kiskapu Kft. Minden jog fenntartva

rendszermagot használunk-e. Mivel én a legtöbb gépen saját általam fordított kernelt szoktam használni, én itt általában nemmel felelek, de aki gyári kernelt használ, nyugodtan nyomjon igent. Erre azért lesz a későbbiekben szükség, mert szalagos mentés készítése esetén a mondo készít egy indítólemez, amellyel egy rendszerösszeomlás esetén a gépet indítva szinte egy gombnyomásra vissza lehet állítani a rendszert. Így tehát fontos, hogy az indítólemezre olyan rendszermag kerüljön, amivel a gép el tud indulni. Ezután már nincs is más hátra, mint elindítani a mentést. Miután lefutott a mentésünk, a rendszer végrehajtja az ellenőrzést a mentett állományon és az esetleges eltéréseket, amelyek a mentés közben keletkeztek kilistázza nekünk.

CD-re, vagy DVD-re való mentéskor lehetőségünk van arra, hogy a mondo olyan lemezeket készít, amelyekről a gépünk el tud indulni, így a lemezt a gépbe helyezve szintén egy gombnyomásra indítható a teljes rendszer-visszaállítás.

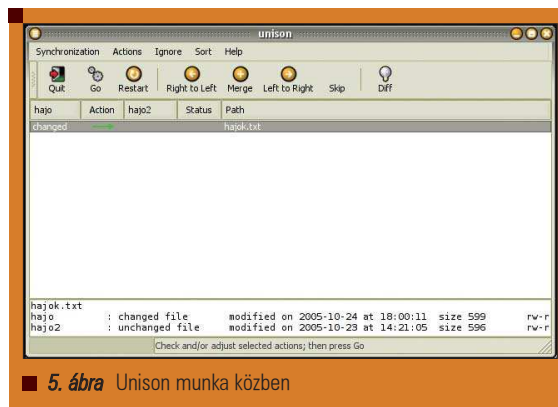
A rendszer visszaállítása

Az előbb már említettem, hogy kazetás és CD/DVD-s mentés esetén pofon egyszerűen tudunk teljes rendszer-visszaállítást indítani. Ám mi van akkor, ha csak néhány állományt szeretnénk visszatölteni a kazettáról, vagy a CD-ről. Nos, ehhez indítsuk el a *mondorestore* programot. Itt megint, ahogy a mentésnél is, meg kell adnunk, hogy milyen médiát szeretnénk használni. Ezek után meg kell adni, hogy melyik könyvtárba szeretnénk a visszaállított állományokat tárolni és egy fástruktúrából ki kell választani, hogy melyik könyvtárakat szeretnénk visszaállítani. A könyvtárak kiválasztása után lehetőségünk van egyesével megadni, hogy az adott könyvtárakon belül melyik állományokat szeretnénk visszanyerni a mentésből. Ha ezzel is megvagyunk, akkor már csak idő kérdése és az állományok ott lesznek a megadott könyvtárban. Egyszerű és tényleg nagyszerű. Szerencsére eddig összesen egyszer kellett a mentésekhez nyúlnom rendszer-visszaállítás céljából, de akkor nagyon jól jött, hogy ilyen egyszerűen használható rendszert választottam.

Egyéb megoldások adatok mentésére

Most, hogy megnéztük, hogy milyen megoldások vannak munkaállomások és szerverek mentésére, be szeretnék mutatni még egy megoldást, amelyet a hordozható számítógépek tulajdonosai használnak előszeretettel. A program neve

Unison és arra használható, hogy egy hálózati erőforrás és egy lokális mappa állományait szinkronban tartsuk vele. Tegyük fel, hogy van egy céges hálózati dokumentumok mappánk, amit az egész cég használ arra, hogy abban közös használatú állományokat helyezzen el. Azonban mi szeretnénk, ha dokumentumaink nem csak akkor állnának rendelkezésünkre, ha az irodában vagyunk, hanem kapcsolat nélkül is szeretnénk azokat használni. Mit tudunk ilyenkor tenni? Mondjuk minden alkalommal lemásoljuk a mappa teljes tartalmát és amikor visszaérkezünk az asztalunkhoz, visszamásoljuk a módosított állományokat. Végül is ez is egy megoldás, ám elég körülményes lekezelni azt, ha közben a hálózati mappa tartalma is megváltozott. Az *Unison* éppen erre nyújt megoldást, végignézni a két könyvtár tartalmát és ahol eltérést talál valamelyik oldalon azt jelzi a felhasználónak. Ahol pedig olyan eltérést talál, hogy mindkét oldali állomány megváltozott, ott lehetőségünk van arra, hogy megnézzük a két állomány közötti különbséget. Természetesen ennek legnagyobb haszna a szöveges állományoknál van, mint például egy *HTML* kód, vagy egy *XML* fájl. Egy *OpenOffice* dokumentum, vagy egy *PDF* állomány kódjának összehasonlításából kevés ember tud mélyreható következtetéseket levonni. A működés elve nagyjából hasonló a *CVS*-hez, mintha minden állomány bináris formában töltenék be egy *CVS*-be. A program ismeri a lokális könyvtárak szinkronizációját, így egy becsatolt *NFS* mappával való szinkronizáció pofon egyszerű. A program ezen felül azonban tud több különböző protokoll



5. ábra Unison munka közben

felett is szinkronizálni, így ismeri például az *SSH*-t. Ez abból a szempontból nagyon jó dolog, hogy így például otthonról is biztonságos csatornán le tudjuk szinkronizálni az állományainkat. A program meglehetősen stabilan működik, beállítása és használata nagyon egyszerű. *Gnome*-ot használó felhasználóknak pedig azzal kedveskedik, hogy egy nagyon kulturált *GTK2*-es felhasználói felülettel együtt is telepíthető. Szerintem érdemes kipróbálni ez a segédprogramot is. Zárásként megjegyezném, hogy természetesen a fenti három programnál jóval több megoldás létezik mentések készítésére, én ezt a hármat azért emeltem ki, mert rendszeresen használom őket és működésükkel nagyon meg vagyok elégedve. Természetesen most is érvényes a mondás, érdemes minél több dolgot kipróbálni, hogy ki tudjuk választani azokat az eszközöket, amelyek számunkra és az adott feladathoz a legjobban használhatóak. Kezdeként talán érdemes az *LVM* állományrendszerrel és annak a segédprogramjaival is megismerkedni, mert sok érdekes megoldást rejtenek, akár csak a *CODA* fájlrendszer az *NFS* és az előbb említett *Unison* kiváltására. Bár megjegyezném, hogy a *CODA*-val még voltak rossz tapasztalataink, ami a rendszert stabilitását illeti. Mindenesetre ezek is olyan projektek, amelyekre érdemes odafigyelni és fejlődésüket követni.



Illés Viktor
 (illes.viktor@assixo.com)
 Mérnök informatikus,
 az Assixo Kft. munkatársa.