

# Linuxon alapuló automatizált mentési rendszer

Hogyan gondoskodjunk a biztonsági mentésekről egyszerűen és ötletesen?

**H**ajlamosak vagyunk azt hinni, hogy a számítógépen tárolt adatok biztonságban vannak. Ez a hiedelem lehet az alapja annak, hogy bár fontos adatokat dolgozunk fel vagy tárolunk a számítógépen, az adatokat mégsem mentjük rendszeresen. A rendszergazdák természetesen tisztában vannak a megbízható mentések fontosságával, de elképzelhető, hogy azok az emberek, akikkel együtt kell dolgozniuk, nem fordítanak erre elegendő figyelmet, és az ő adataik elvesztésekor is a rendszergazdákat veszik elő.

Egy tudományos kutatócsoportban dolgozom. Munkahelyünk korszerű, gyakorlatilag minden adatfeldolgozást Windows 95-öt futtató számítógépekkel végzünk. Tömören összefoglalva az egész tevékenységünk nem más, mint a számítógépekben tárolt adatok feldolgozása. Az adatvesztés több százezer forintba kerül, különösen, ha az adatokat előállító munkatársak fizetését is figyelembe vesszük. A pótolhatatlan adatok elvesztésének megakadályozására kidolgoztam egy automatizált hálózati mentési rendszert. A költsége elenyésző volt, ugyanis egy parlagon heverő 486/66 számítógépet és egy 3 GB tárhelyű merevlemez használtam fel hozzá, így összesen alig több mint harmincezer forintba került az egész. Több különféle rendszert kipróbáltam az elmúlt két évben, Windows 95-tel kezdtem, végül is egy gyors és hatékony Linux-rendszerrel kötöttem ki. Ezt a rendszert könnyű kiépíteni, olcsó, hatékony és megbízható. Feltéve, hogy rendelkezünk egy hálózatra kapcsolt Linuxot futtató számítógéppel, az itt leírtak segítségével rövid idő alatt kiépíthető az automatizált mentési rendszer.

## Szükséges eszközök

Az automatizált mentési rendszerhez szükséges összes eszköz megtalálható a legtöbb Linux-változatban. Az egyik a Samba, egy kiváló nyílt forráskódú csomag, amely lehetővé teszi, hogy Unix-alapú rendszerek Windows-alapú rendszerekkel a TCP/IP protokollon keresztül fájlmegosztást végezzenek. A linuxos változat tartalmazza az smbmount segédprogramot, amely kihasználva a rendszer mag egyedülálló smb fájlrendszer támogatását, lehetővé teszi, hogy bármely windowsos gépen elhelyezkedő könyvtárat a Linux fájlrendszerébe fűzzünk, és úgy használjuk az ott lévő fájlokat, mintha a linuxos gép merevlemezén lennének. Így a mentést végző program (frissítési módomból még azelőtt meg tudja állapítani a windowsos gépen lévő fájlról, hogy kell-e menteni, mielőtt átmozgatná a hálózaton keresztül. Ez jelentősen csökkenti a hálózati forgalmat, a processzor és a merevlemez terhelését.

Rendeteg biztonsági másolatot készítő program található Linux alatt, például a tar, a bzip2, vagy az egyszerű cp parancs. Választásom mégsem ezekre, hanem a nyílt forráskódú Info-ZIP eszközeire esett. Ezek kicsik, gyorsak, a legtöbb Linux-változat tartalmazza, más felületeken is elérhetők, és kiválóan együttműködnek a Windowszal is. Rádásul az Info-ZIP programok tömörítési képességeinek köszönhetően a tároló-fájl mérete jelentősen csökken.

## Előzetes lépések

A windowsos gépeken létre kell hozni azokat a hálózati megosztásokat (merevlemez vagy egy alkönyvtár), amelyeket menteni szeretnénk. Ha a fájlmegosztás nincs engedélyezve, be kell állítanunk a Vezérlőpult Hálózatok részében. Ezután a Windows Intézőben kattintsunk jobb egérgombbal arra a meghajtóra vagy könyvtárra, ame-

### 1. lista A DATA könyvtár mentése

```
#!/bin/sh
# Mentés a megadott SHARE megosztásról
SHARE="//higgins/d_drive";
DATADIR="DATA";
USERNAME="notneeded"
PASSWD="mjdomo";
BACKUPFILE="higgins_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
export PASSWD;
echo 'go';
smbmount $SMBMP;
smbmount $$SHARE $$SMBMP -N -n $USERNAME;
if (ls $$SMBMP$DATADIR) then
  echo "$SHARE mentése folyamatban";
  mount -t vfat $BACKUPDRIVE $BACKUPMP;
  cd $SMBMP;
  zip -r -u -v $BACKUPMP$BACKUPFILE
  $DATADIR;
  cd /;
  umount $BACKUPDRIVE;
fi;
smbmount $SMBMP;
```

lyet meg szeretnénk osztani, és válasszuk a Megosztás pontot. Javasolom, hogy „csak olvasható” megosztásokat hozzunk létre, hogy a betörők ne tehessenek kárt a fájlokban akkor sem, ha valahogy megszerzték a jelszavakat. Jegyezzük meg jól a megosztások neveit. Jó ötlet beírni a windowsos gépek netbios nevét, DNS-nevét és IP-címét a Linux /etc/hosts fájljába (ahogy az a /etc/hosts megjegyzéseiben le van írva), esetleg a névkiszolgálóba, különösen akkor, ha a számítógépek több különböző alhálózatban vannak.

Ha ez megvan, elő kell készíteni a linuxos gépet az adatok elérésére és tárolására. Hozunk létre egy könyvtárat, ahová a windowsos megosztásokat fogjuk befűzni: `mkdir /mnt/smb`. Ezután el kell döntenünk, hogy hová tesszük a lementett adatokat.

Én egy különálló 1 gigabájtos vfat (Windows) lemezterületen tartom a lementett fájlokat, ez nincs a rendszerbe illesztve, csak a mentés idején. Így a fájlok a lehető legjobban védettek az áramkimaradásból eredő károkkal szemben, a meghajtó kivehető a linuxos számítógépből és áttehető egy windowsos gépbe az adatok helyreállításának céljából. Ennek megfelelően egy /mnt/backups nevű befűzési pontot is létrehoztam.

## Parancsállományok

A parancsállomány egy szövegfájl, amely olyan parancsokat tartalmaz, amelyeket az ember általában a parancssorban ad ki. Bonyolult feladatokat ismétlődő jelleggel elvégeztethetünk velük. A parancsállomány létrehozható bármelyik szerkesztőben. A fájl elkészülte után `chmod u+x` segítségével tehetjük futtathatóvá.

## 2. lista Példák windowsos gépekhez

```
# ----- VÁLTOZÓK a Rick nevű géphez -----
SHARE="//rick/rick_c";
DATADIR="Data";
USERNAME="notneeded"
PASSWD="icepick";
BACKUPFILE="rick_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
# -----
# ----- VÁLTOZÓK a TC nevű géphez -----
SHARE="//tc/data";
DATADIR="*";
USERNAME="obrien"
PASSWD="flyme";
BACKUPFILE="tc_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
# -----
```

Az 1. lista egy parancsállományt mutat be, amely a Higgins nevű gép d\_drive megosztásáról a DATA könyvtárat menti le. A parancsállomány neve root/backup/higgins.

Az első, megjegyzésnek tűnő sor tulajdonképpen arra utasítja a számítógépet, hogy a basht használja a parancsállomány futtatásához. Ezután következnek a mentéskor használt változók. Ha a parancsállomány elején található változókban tároljuk az adott gépre érvényes adatokat, akkor a fájl elejét átírva könnyen hozhatunk létre újabb parancsállományokat a különböző gépekhez. A 2. lista egy másik változókészletet mutat be egy Windows 98-at futtató gépre (a gép neve Rick és a C: meghajtó van megosztva) és egy Windows NT-t futtató gépre (a gép neve TC és a „data” könyvtár van megosztva). Ne feledjük, hogy Windows NT esetén meg kell adni a felhasználói nevet és a hozzá tartozó jelszót! A többi sor végzi el a tulajdonképeni munkát. Az <<export paswd>> parancs a jelszót egy környezeti változóba rakja, amelyet az smbmount program automatikusan beolvas. Az smbmount programot azért hívjuk meg, hogy leválasszuk az esetlegesen befűzött SMB megosztásokat. (Ha semmi nincs ott, akkor az smbmount egy értelmetlen hibáüzenetet ad, és a parancsállomány tovább fut.) Az smbmount megkísérel befűzni a távoli megosztást. A -N kapcsoló arra utasítja, hogy ne kérje el a jelszót, hanem a <<paswd>> környezeti változóból vegye. A -n kapcsolóval adható át a felhasználói név.

Egy if szerkezet megvizsgálja, hogy a megadott mentendő fájlok léteznek-e, mielőtt a mentés megkezdődne, hiszen elképzelhető, hogy a hálózat nem működik, vagy a távoli számítógép ki van kapcsolva. Ilyenkor a parancsállomány a befűzési pont újrafelszabadítása után leáll.

Ha a linuxos gép el tudja érni a távoli fájlokat, a mentést a zip parancs végzi. A -r kapcsoló a rekurzívítást biztosítja, azaz a zip az adatokat tartalmazó könyvtár minden alkönyvtárát is végignézi. A -u kapcsoló állítja be a frissítési üzemmódot, amely csak akkor adja hozzá a fájlt a tárolófájlhoz, ha az még nincs benne, vagy újabb, mint ami a tárolófájlban található. A -v kapcsoló bőbeszédű üzemmódra készíti a zip programot, minden feldolgozott fájl nevét kiírja – ez hasznos segítség a hibakeresés során.

Miután az összes számítógéphez elkészítettük a mentést végző parancsállományt, készíthetünk egy főparancsállományt (nevezzük masternek), amely egymás után végrehajtja az összes mentést végző parancsállományt. Az én főparancsállományom a 3. listán olvashatóhoz hasonló.

## 3. lista Főparancsállomány

```
#!/bin/sh
# A mentést végző parancsállományok meghívása
/root/backup/higgins
/root/backup/rick
/root/backup/tc
```

## A rendszer beindítása

A parancsállományok megírása után tegyünk egy, a főparancsállományra mutató közvetett hivatkozást (symbolic link) a /etc/cron.d alkönyvtárainak egyikébe, hogy a számítógép automatikusan elvégezze a mentést. Én az alábbi parancsral értem el, hogy hetente lefusson:

```
ln -s /root/backup/master /etc/cron.d/weekly/master
```

Szükség esetén napi mentést is végezhetünk, hiszen a tárolóprogramok frissítési üzemmódja nagymértékben csökkenti az erőforrásigényt. A mentést végző parancsállomány első futtatása azonban nagy sávszélességet és sok processzoridőt igényel. A leghelyesebb, ha ezt a feladatot először kézzel indítjuk, vagy az at parancs segítségével éjjel.

## Figyelmeztetések

1. Minden jelszót tartalmazó parancsállományt rejtünk el mindenki elől. Ehhez használjuk a `chmod go-r` parancsot.
2. A nagyon fontos adatok védelmére külön biztonsági intézkedéseket kell tennünk, elkerülendő, hogy ipari kémek betörjenek gépünkre, és ellopják az egy helyre összegyűjtött adatokat.
3. Az smbmount program kissé különbözik az egyes Linux-változatokban. Ezért, ha az itt megadott parancsállományok nem működnek, nézzük végig az smbmount parancs leírását (man smbmount).
4. A Windows-felhasználókat meg kell tanítani, hogy adataikat egy központi helyen tárolják, például az „adatok” könyvtárban, ahelyett hogy mindenfelé szétszórják a fájljaikat. Sok ember lusta áthelyezni az adatait egy területre, annak ellenére, hogy ez csak öt másodpercet venne igénybe. Elképzelhető, hogy nekünk kell átraknunk a fájlokat, amíg ők is megtanulják, hogy egy központi könyvtárban tartsák az adataikat. Jegyezzük meg, hogy ezek a lusta emberek jelentik a legnagyobb veszélyt egy szervezetben az adatbiztonság tekintetében, hiszen saját adataik mentésével sem törődnek.
5. A merevlemez nagyon célszerű tárolóeszköz a pótolhatatlan adatok számára. Az én rendszeremen a tárolófájlok kevesebb mint 400 megabájtot foglalnak, ami 1,5 GB adatnak felel meg. Megfontolandó esetleg egy nagy tárhelyű eltávolítható eszközre író meghajtó beszerzése. Így időszakonként kimenthetjük a tárolófájlokat a merevlemezről az eltávolítható eszközre, és hazavihetjük, ezzel védekezve a gép fizikai megsemmisülése vagy ellopása ellen.

## Összefoglalás

Ha több, hálózatba kapcsolt számítógépről kell lementeni a pótolhatatlan adatokat, a Linux-alapú hálózati mentési rendszer olcsó, megbízható, könnyen telepíthető és könnyen bővíthető megoldást jelent a feladatra. Ha egy lemez meghibásodik, a befektetett egyórás munka sok pénz és időt takaríthat meg csoportunknak vagy vállalatunknak azáltal, hogy az adatok nem vesznek el. Nálam például a Pentium 150-es gép több év alatt felhalmozott fontos adatot tárol három épületben és két külön alhálózaton lévő nyolc különböző számítógépről. Alig egy perc alatt tudok egy új gépet hozzáadni a rendszerhez, hála az állományok elején lévő változóknak.

Michael O'Brien (mobrien@unm.edu) az Új-mexikói egyetem végzős hallgatója, optikát tanul. A számítógép egyszerűre a kedvtelése és segítőtársa a munkában. Szabadidejében egy kis számítógépszobát felügyel.