

Linux alapú , több terabájt kapacitású háttértár építése otthon



© Kiskapu Kft. Minden jog fenntartva

Építsünk alacsony költségű, Linux alapú, terabájt méretű háttérkiszolgálót digitális képeink, audio és videóanyagaink tárolására.

A terabájt méretű, vagy ennél is nagyobb otthoni háttértárak építése a *Linux* felhasználók számára immár elérhető alternatíva az adatmentésben. Ebben a cikkben egy ilyen rendszer építésével, beállításával és használatával kapcsolatos lehetőségeket fogjuk áttekinteni.

A kiszolgáló megtervezése

Manapság a kifejezetten nagy kapacitású merevlemezek szinte elképesztően alacsony áron szerezhetőek be, legalábbis ha a mindössze néhány évvel ezelőtti árakhoz hasonlítjuk a mai költségeket. Ezzel párhuzamosan, köszönhetően elsősorban a *Linux* multimédiás képességek területén tapasztalható fejlődésének, egyre több *Linux* felhasználó foglalkozik zenei CD-k tömörítésével, digitális fényképezéssel és filmezéssel, filmvágással és feldolgozással (például a *MythTV* segítségével). Így érthető, hogy egyre nagyobb az igény a nagy kapacitású háttértárakra is, hiszen az elkészült adatok

biztonságos mentése és tárolása immár kritikus elvárás egy ilyen célú rendszer esetében. Ha egy merevlemez meghibásodása miatt elveszítjük a gyermekeinkről készült képeket vagy filmeket – vagy egyszerűen csak a nagy gondal összeválogatott zenegyűjteményünket – az egyenesen katasztrófa. Szerencsére ma már egész egyszerűen megépíthetünk egy a biztonsági mentéseket tároló nagy kapacitású háttérkiszolgálót linuxos szoftverekkel, és egyszerű, bármelyik számítástechnikai szaküzletben megvásárolható olcsó merevlemezektől. Gyakorlatilag bármilyen otthoni felhasználásra szánt *PC* alkalmas lehet arra, hogy egy ilyen kiegészítő háttérkiszolgáló szerepét betöltse. Ha viszonylag ritkán kell adatainkat menteni, és egy-egy frissítés nem igényli különösebben sok anyag átvitelét, akkor a számítógép sebessége feltehetőleg nem lesz akadály. Ugyanakkor győződjünk meg róla, hogy a hálózatunk képes a rendelkezésre álló idő alatt átvinni a szükséges adatmennyiséget. Az öregebb gépek esetében a szűk keresztmetszetet leginkább maguknak a lemezeknek az adatátviteli sebessége (30-1510 Mbps a technológiától függően) jelentheti.

Ügyeljünk rá, hogy számos, általános felhasználásra tervezett számítógép hűtési kapacitása csupán két merev-

lemezhez elegendő. Persze a legtöbb alaplap eleve legfeljebb négy merevlemez csatlakoztatását teszi lehetővé. (Ez elsősorban a csak *ATA/IDE* lemezeket támogató rendszerekre érvényes, manapság azonban egyre gyakoribb a két *ATA/IDE* plusz két *SATA* csatlakozó megléte.) Használhatunk külső, *USB* portra csatlakoztatható lemezeket is. Öregebb géphez, amiben csak *USB1*-es csatlakozó van érdemes venni egy *PCI USB2* kártyát, ami tízszer gyorsabb.

A *SCSI* rendszereknél sokkal kevesebb korlátozással kell számolnunk, viszont ezek kifejezetten drágák, illetve maga a technológia kicsit összevissza fejlődik, rengeteg változattal és kiterővel. A lemezek számát legegyszerűbben egy úgynevezett *HBA* kártya (*Host Bus Adaptor*) segítségével növelhetjük. Ilyeneket gyárt például a *Promise Technology*. A *HBA* egy olyan *PCI* sínre csatlakozó lemezvezérlő, amelynek rendszerint semmiféle kiegészítő szoftverre nincs szüksége a működéséhez, és mivel saját *BIOS*-a van, így a *PC BIOS*-ában esetleg meglévő korlátok (például bizonyos méretnél nagyobb lemezek kezelésének képessége) sem gátolják. Egy *HBA* kártya segítségével tehát olyan gépben is használhatunk 120 GB-nál nagyobb lemezeket, amelyek elvileg nem lennének képesek ezek kezelésére,



■ **1. ábra** Egy öreg kiszolgálóból épített lemeztömb (kilenc IDE lemezt tartalmaz, amelyek közül öt egy SCSI RAID tagja) A kiegészítő IDE csatolók egy Promise HBA kártyán találhatók.

1. táblázat *Néhány lehetséges összeállítás a kisegítő fájlserverre, az egy gigabájtra vetített költséggel együtt*

Típus	Konfiguráció	Kapacitás (TB)	Ár/Gigabájt (\$)
ATA/SATA lemezek	Belső lemez	0.4	0.56
Linux asztali gép*	Három belső lemez	1.2	0.84
Linux asztali gép*	Három belső plusz két külső USB lemez	2.0	0.73
LaCie 2TB Storage	Speciális célú hálózati kiszolgáló	2	1.15
Linux kiszolgáló**	Hat belső lemez	2.4	1.21
Linux kiszolgáló**	Hat belső és két USB-s külső lemez	3.2	1.08

* Intel Celeron D 478 325 2.53 GHz, 256 MB RAM.

** Intel SC5275 ház, Intel ATX alaplap, két darab 3 GHz-es Xeon CPU, 2 GB RAM.

ATA-33-ról ATA-150-re fejleszthetjük a rendszer képességeit, illetve egyszerűen használhatunk ATA és SATA lemezeket. Megfontolhatjuk egy dedikált fájlkiszolgáló beszerzését is. Egy hat lemez fogadására képes úgynevezett barebone kiszolgálót (összeszerelve, de lemezek és operációs rendszer nélkül) körülbelül 1500 dollárért megvehetünk. Ezzel a viszonylag jelentősebb kezdeti beruházással gigabájtontként körülbelül 0,8 dollárért bővíthetjük a rendelkezésre álló lemezterületet, illetve ha elfogyott a hat hely, akkor használhatunk külső USB-s lemezeket is. Ha eldöntöttük, hogy hány lemezre

lesz szükségünk, a következő lépés a helyigén, a zaj és a szükséges hűtés felmérése. Az 1. ábrán egy olyan otthoni fájlkiszolgálót láthatunk, aminek az alapját egy kiöregedett otthoni gép szolgáltatta. A kapacitása jóval 1 gigabájt fölött van. Még ha úgy döntünk, hogy saját kezűleg építjük meg a fájlkiszolgálónkat, akkor is egészen alacsonyan tartható az egy gigabájtra vetített költség. Ennek persze leginkább az az oka, hogy a merevlemezek ára az elmúlt időben drasztikusan esett. Az 1. táblázatban felsoroltam néhány lehetséges konfigurációt, a hozzájuk tartozó egy gigabájtra vetített költséggel együtt.

(Az árak természetesen csak becült értékek, és nem tartalmazzák a szállítási költséget illetve az adókat.) Amint látható, egy 2 terbájtot is meghaladó fájlkiszolgáló egy gigabájt tárterületre vetített költsége nem haladja meg az 1,50 dollárt, miközben egy ekkora lemeztömbön már igazán jókora mennyiségű filmet, digitális képet és zenesámot tárolhatunk.

A kiszolgáló megépítése és a lemezek behelyezése

Jómagam az elmúlt néhány évben több Red Hat Linux 9 alapú háttérkiszolgálót építettem, de természetesen használhatunk bármilyen más terjesztést is. Az én választásom azért esett éppen a Red Hat-re, mert stabil, szabadon hozzáférhető, és – a Fedora Legacy Projectnek köszönhetően – folyamatos a frissítése. Telepítése és beállítása is egészen egyszerű. Aki új számítógépet vásárol, annak persze érdekesebb valamilyen ennél frissebb terjesztéssel próbálkoznia. Az alacsony költségvetéssel készülő fájlkiszolgálókon én nem szoktam RAID-et használni, mivel itt az ár – értelemszerűen – elsődleges szempont. Ugyanakkor érdemes ezen a lehetőségen is elgondolkozni.

Ami a szoftvert illeti, egy linuxos háttérkiszolgáló igényei nyugodtan nevezhetők minimálisnak. Amire – az alapokon kívül – biztosan szükségünk lesz, az az alapvető hálózati adminisztráció, beleértve az SSH-t is, valamint az rsync. Utóbbi egy gyors, inkrementális másoló/szinkronizáló segédprogram, amit a legtöbb Linux terjesztés eleve tartalmaz. Az SSH és az rsync segítségével gyakorlatilag bármilyen alapvető adatmentési feladat elvégezhető. Hasznos, ha a kisegítő kiszolgáló egyben fájlkiszolgálóként is működik, így én mindig telepítem rá a Samba-t is. Azért ezt a megoldást használom, mert az MS Windows ügyfelek egyrészt alapértelmezésként SMB-n keresztül kommunikálnak, másrészt a megfelelő Samba ügyfél segítségével az így létrehozott hálózati megosztások bármilyen UNIX alapú rendszerről (még a Mac OS X-et is beleértve) elérhetőek. Akinek homogén UNIX-os hálózata van, az használhat NFS-t is, de ezt a lehetőséget itt részletesen nem tárgyalom.

Új partíciók létrehozása

Ha a hdb eszközhöz tartozó lemezen akarunk új partíciókat kialakítani, adjuk ki a következő parancsot:

```
fdisk /dev/hdb
```

Az fdisk promptjánál gépeljük be az m parancsot, amely a sűgót jeleníti meg. Itt is látható, hogy új partíciót az n segítségével hozhatunk létre, előbb azonban válaszolnunk kell néhány kérdésre:

```
Command action
  e   extended
  p   primary partition (1-4)
  p
```

Ha az egész lemezen egyetlen partíciót akarunk kialakítani, használjuk a p parancsot:

```
Partition number (1-4):1
```

A program ekkor megkérdezi az elsődleges partíció sorszámát. (Ha egyetlen partíciót hozunk létre, akkor a válasz 1 kell legyen.) Határozzuk meg a partíció méretét az első és utolsó cylinder megadásával. Tekintve hogy mi most a teljes lemezt akarjuk fölhasználni, nincs más dolgunk, mint elfogadni az alapértelmezett értékeket.

```
First cylinder (1-48641, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-48641, default
  48641):
Using default value 48641
```

A partíciós tábla frissítéséhez adjuk ki a w parancsot. Ezzel létrehoztuk a /dev/hdb1 eszközt, amely egy a teljes lemezt elfoglaló partíciónak felel meg.

Ezután válasszuk ki a céljainknak megfelelő fájlrendszer típust (én az ext3-mal szoktam használni), majd formázzuk meg a lemezt az mkfs parancs segítségével:

```
mkfs -t ext3 /dev/hdb1
```

Aztán hozzuk létre a csatolási pontot (esetben /data1)

```
mkdir /data1
```

majd csatoljuk be az új lemezrészét:

```
mount -t ext3 /dev/hdb1 /data1
```

Ellenőrizzük az írhatóságát és olvashatóságát, végül pedig szűrjük be az automatikus befűzéséhez szükséges sort a /etc/fstab fájlba:

```
# Device mountpoint fstype options freq pass_no
/dev/hdb1 /data1 ext3 defaults 1 2
```

A lemezek csatlakoztatása előtt először is győződjünk meg róla, hogy van elegendő tápcsatlakozónk és (IDE/SATA/SCSI) adatkábelünk. Esetleg érdemes azt is ellenőrizni, hogy a beépítendő lemez kompatibilis-e a Linuxszal (a legtöbb igen). Kapcsoljuk ki a gépet, és húzzuk ki a tápkábelt is. Csatlakoztassuk a lemezeket, majd kapcsoljuk be újra a rendszert. A Linuxnak automatikusan fel kell ismernie az új elemeket. Ha ez valamiért mégsem történik meg, akkor vagy speciális meghajtóra lesz szükségünk, vagy a kérdéses hardverelem nem használható Linux alatt. A bootolás közben megjelenő üzeneteket a dmesg paranccsal hívhatjuk elő újra. Egy sikeresen felismert IDE lemezzel kapcsolatos üzenet valahogy így néz ki:

```
hdb: ST3400832A, ATA DISK drive
```

Minden IDE/ATA (sőt néhány SATA) merevlemezhez egy hdx névszerkezetű eszközön keresztül férhetünk hozzá, ahol x az ábécé valamelyik betűje (esetünkben b). Hasonlóan az USB, SCSI és egyes SATA merevlemezek csatlakoztatása után a bootolás közben megjelenő üzenetek között egy sdx eszközre vonatkozó jelenik meg, ahol x ismét valamilyen betű.

A legtöbb Linux terjesztés általában tartalmaz valamilyen grafikus felületen át használható lemezkezelő programot. Ezekkel az alkalmazásokkal feloszthatjuk (particionálhatjuk) a merevlemez, illetve meg is formázhatjuk az így kialakított lemezrészeket (jómagam minden háttérkiszolgálóba helyezett lemezen egyetlen partíciót használok), sőt rendelhetünk hozzájuk befűzési pontot is (például /data1, /data2). Mindezt természetesen parancssorból is megoldhatjuk az fdisk és egyéb parancsok segítségével.

A szoftverkomponensek beállítása – rsync és SSH

Az rsync program a legtöbb Linux terjesztésnek része. Erre és az SSH rendszerre az ügyfélen és a kiszolgálón egyaránt szükségünk lesz. Azt, hogy az rsync telepítve van-e, a legegyszerűbben úgy ellenőrizhetjük, ha a parancssorban kiadjuk az rsync parancsot, vagy átvizsgáljuk a telepített csomagok listáját. Ha esetleg nem

SSH kulcspár létrehozása

Jelentkezzünk be a megfelelő felhasználó nevében arra a gépre, amelyen a kulcspárral majd hitelesíteni szeretnénk magunkat (jelen esetben tehát Bob-nak kell bejelentkeznie a bar szerverre), majd adjuk ki a

```
ssh-keygen -d
```

parancsot. Ha a kulcsot más felhasználók is használják, vagy olvashatják, akkor adjunk meg jelszót is, ellenkező esetben az ezzel kapcsolatos kérdésre válaszoljunk az *Enter* megnyomásával. Lépünk be a *.ssh* könyvtárba, és másoljuk be az engedélyezett kulcsok listájába a most létrehozott nyilvános kulcsot:

```
cd ~/.ssh
cp id_dsa.pub authorized_keys2
```

A titkos kulcsot másoljuk át annak a gépnek a *.ssh* könyvtárába, amelyről be fogunk jelentkezni (jelen esetben tehát a foo gépen a Bob könyvtára alatt található *.ssh* könyvtárba). A kiszolgálóról (bar) egyidejűleg töröljük a titkos kulcsot:

```
scp id_dsa root@foo:~/.ssh/id_dsa
rm id_dsa
```

A bejelentkezéshez használt gépen indítsuk el az *SSH* ügynököt, és vegyük fel a most átmásolt titkos kulcsot a listájába (ha az imént megadtunk jelszót is, akkor az ügynök meg fogja azt kérdezni):

```
eval `ssh-agent`
ssh-add
```

És ezzel készen is vagyunk. Ettől kezdve *Bob* a *foo* gépről a bar kiszolgálóra jelszó nélkül jelentkezhet be:

```
ssh bob@foo
```

találjuk az *rsync* csomagot sem a rendszeren, sem a telepítőlemezen, akkor letölthetjük a forrását is a program hivatalos weblapján található linkeket követve (lásd a kapcsolódó címekeket). Az *rsync* biztonságos futtatásának leg-egyszerűbb módja az, ha önálló alkalmazásként indítjuk, a titkosításhoz pedig az *SSH*-t használjuk. A program ugyan futtatható démonként is, sőt így a szolgáltatásainak a köre is valamivel bővebb, de erre esetünkben nem lesz szükség. A következőkben a parancssori program használatát fogom bemutatni. A példákban az ügyfelet *foo*-nak, a kiszolgálót pedig *bar*-nak hívják.

Ha a *foo* nevű, *Linuxot* futtató gép */home* könyvtárát akarjuk átmásolni a bar kiszolgáló */data1/foo* könyvtára alá az *rsync* és *SSH* kombinálásával, gépeljük a következőt:

```
rsync -az /home -e ssh
bob@bar:/data1/foo
```

A rendszer kérni fogja bob jelszavát, majd a foo gépen található */home* könyvtárakat átmásolja a bar-on található */data1/foo/home* könyvtárba. (Ehhez Bobnak természetesen rendelkeznie kell egy felhasználói fiókkal a kiszolgálón, illetve írási joga kell legyen a */data1/foo* könyvtárra.)

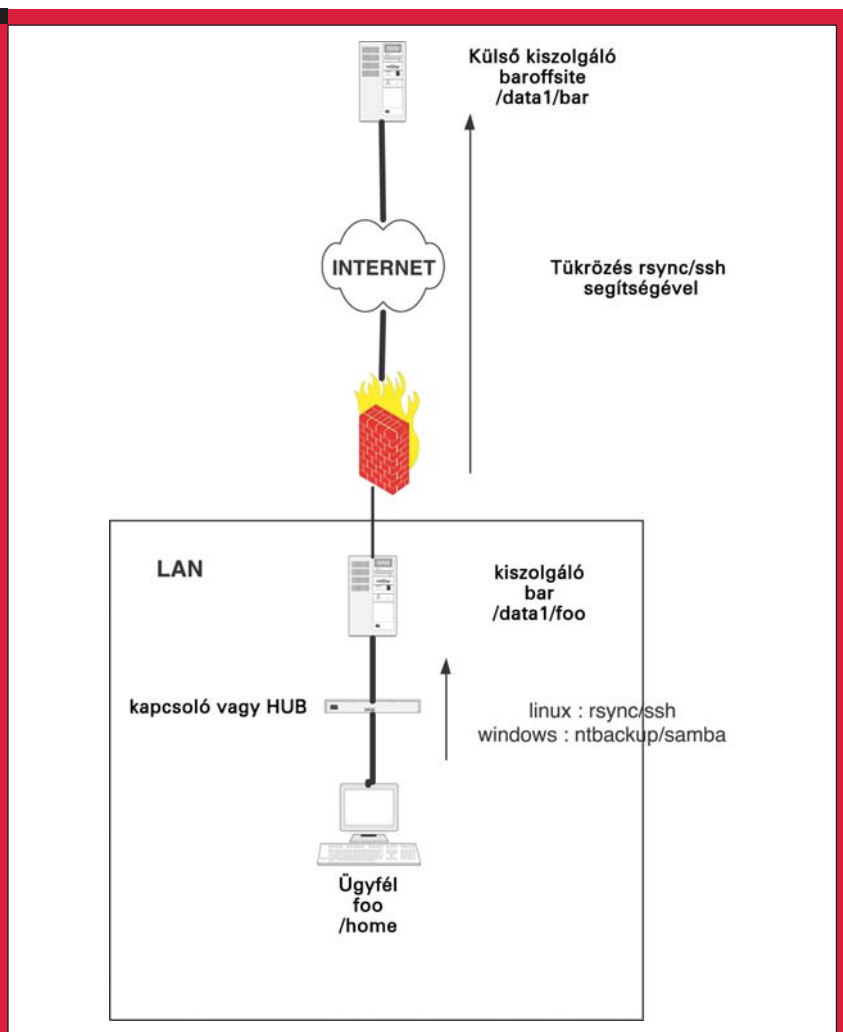
Ha el akarjuk kerülni, hogy minden egyes alkalommal be kelljen gépelnünk a kérdéses jelszót, hosszunk létre egy jelszó nélküli nyilvános/titkos kulcspárt, és azt használjuk az *SSH* hitelesítéshez.

A mentést természetesen nem csak kézzel végezhetjük el, hanem írhatunk rá egy szkriptet is, ami a *foo* gépen lefuttatva elvégzi a mentett anyag frissítését. Ehhez azonban előbb

1. Lista Teljes és inkrementális mentés az *rsync* segítségével

```
#!/bin/sh
# Ez a szkript menti a foo gép kijelölt anyagait a bar kiszolgálóra.
# A mentés módja: 7 napos, inkrementális.
# A program Andrew Tridgeld eredeti kódja alapján készült.
# A menteni kívánt könyvtár BDIR=/home
# Célkönyvtár a kiszolgálón BACKUP_HOME=/data1/foo
# A mentés során használandó felhasználói fiók BACKUP_LOGIN=bob
# A kiszolgáló neve BSERVER=bar
BACKUPDIR=`date +%A`
OPTS="-force -ignore-errors
↪ -delete -backup
↪ -backup-dir=$BACKUP_HOME/
↪ $BACKUPDIR -av"
export PATH=$PATH:/bin:/usr/
↪ bin:/usr/local/bin
# A kimenetet egy fájlba irányítjuk
date > /var/log/
↪ backup.$BACKUPDIR.log
# Ez a sor törli az előző héten végrehajtott
# inkrementális mentésekhez tartozó könyvtárat
[ -d /tmp/emptydir ] || mkdir
↪ /tmp/emptydir
rsync -delete -a /tmp/emptydir/
BACKUP_LOGIN@$BSERVER:
↪ $BACKUP_HOME/$BACKUPDIR/
rmdir /tmp/emptydir
# Itt kezdődik a tényleges átvitel
rsync $OPTS $BDIR BACKUP_
↪ LOGIN@$BSERVER:$BACKUP_HOME/
↪ current >>
/var/log/backup.$BACKUPDIR.log
```

célszerű elolvasni az *rsync* dokumentációját, hiszen a program több mint 70 parancssori kapcsolója van. Különösen a *-delete* opcióhoz tartozó részt olvassuk el figyelmesen, ennek nem megfelelő használata ugyanis katasztrofális következményekkel járhat. Az 1. Lista egy hét napos inkrementális mentést megvalósító



■ 2. ábra Példa külső telephelyen elhelyezett adatmentő kiszolgálóval. A helyi ügyfél foo, a helyi adatmentési kiszolgáló bar, a külső telephelyen elhelyezett tükrőszerver pedig baroffsite.

szkriptet mutat. Azok a fájlok, amelyeket módosítottunk vagy töröltünk az adott napnak megfelelő könyvtárba kerülnek (utóbbit a `-backup-di r` kapcsolóval lehet beállítani), a legfrissebb mentést pedig mindig a `current` nevű könyvtárban találjuk.

Aki tömörített formában szeretne mentést készíteni, az természetesen a `tar`-t is használhatja a hálózati mentéshez a következőképpen:

```
tar cvfz - /home | ssh bob@bar
↳ dd of=/data1/foo/
↳ current.tar.gz
```

Inkrementális mentéshez ilyenkor a `-newer` kapcsolót kell használnunk. Mindazonáltal az `rsync` jóval hatékonyabb, mint a `tar`, mert előbbi csak a jelenlegi és az előző állapot közti különbségeket másolja át.

Összességében bár az `rsync` és az `SSH` segítségével kiválóan elboldogulhatunk még `MS Windows` környezetben is, azért valós körülmények között sokkal célszerűbbnek tűnik egy mentési célokra szolgáló fájlkiszolgáló üzembe helyezése. Ez különösen akkor célszerű, ha a hálózatban sok windowsos ügyfél található. Ilyen esetben nyilván olyan szkriptet kell írunk, amely a `Samba` kiszolgáló egy megadott megosztására írja a menteni kívánt adatokat.

Samba fájlkiszolgáló üzembe helyezése

Ha az általunk használt `Linux` terjesztés támogatja az `SMB`-n keresztüli fájl-megosztást, akkor a `Samba` rendszer minden valószínűség szerint alapértelmezésként települt. Ha mégsem így lenne, vagy valamiért a `Samba` legfris-

sebb kiadását szeretnénk használni, töltsük le a forráskódját, és fordítsuk le. Mind megannyi más programnál, itt is rendelkezésre állnak különböző bináris csomagok is (lásd a kapcsolódó címeket). A fordítás és a kezdeti beállítás menetéről a dokumentációban olvashatunk.

Amint feltelepítettük a csomagot, a `Samba` kiszolgáló valamennyi beállítását az `smb.conf` megfelelő átszerkesztésével végezzük el. Ez a fájl rendszerint a `/etc/samba/smb.conf`, vagy a `/usr/local/samba/lib/smb.conf` helyen található. Léteznek grafikus felülettel rendelkező beállítóprogramok is. Ilyen például a `SWAT`, amit a legtöbb `Linux` terjesztés szintén tartalmaz. A `Samba` indításával és leállításával kapcsolatos műveletekről a program dokumentációjában olvashatunk. A legegyszerűbb nyilván az, ha úgy állítjuk be rendszerünket, hogy a gép indulásakor a `Samba` kiszolgáló is automatikusan elinduljon.

Visszatérve korábban említett mentési példánkhoz, a `bar` kiszolgálón – amelyen immár a `Samba` is telepítve van – hozzunk létre egy egyszerű `smb.conf` fájlt, vagy másoljuk be egy már létező végére a következő szakaszt:

```
[bob]
comment = foo backup account
path = /data1/foo
valid users = bob
public = no
writable = yes
```

A következő lépésben hozzunk létre egy `bob` nevű `Samba` felhasználót egy kellően biztonságos jelszóval ellátva. (Ne felejtjük el azt sem, hogy `bob`nak egy közönséges felhasználói fiókkal is rendelkeznie kell, illetve írási és olvasási joga kell legyen a `/data1/foo` könyvtárra:

```
smbpasswd -a bob
New SMB password: somepassword
Retype new SMB password:
somepassword
Added user bob
```

A `Windows` futtató ügyfeleken hozunk létre egy a `\\bar\bob` megosztásra mutató hálózati meghajtót (network drive) természetesen a `bob` bejelentkezési névvel és `bob` `Samba` fiókjának jelszával „felszerelve”. Én általában

2. Lista Adatmentés
külső telep helyre az rsync
segítségével

```
#!/bin/sh
# Mirror /data1 on bar to
/data1/bar on baroffsite.
#Backup directory on bar
BACKUP=/data1
#Backup directory on baroffsite
BACKUP_OFF=/data1/bar
# Give the day of week as name
of backup
BACKUPNAME=`date +%A`
# Offsite server
BSERVER=baroffsite
# Backup account on backup
server
BAC_ACC=backup
date > /var/log/
↳ backup.$BACKUPNAME.log
/usr/bin/rsync -avz -delete -e
↳ ssh $BACKUP
↳ $BAC_ACC@$BSERVER:$BACKUPOFF
>> /var/log/
↳ backup.$BACKUPNAME.log
# Email the log to
administrator
cat /var/log/
↳ backup.$BACKUPNAME.log | mail
↳ -s 'Mirror Check'
backup_guy@mycompany.com
```

a szabadon használható ntbakup program segítségével szoktam a kiszolgálóra felmásolni a *blkf* fájlokat. Az ntbakup egyébként része a *Windows 2000* és *Windows XP* rendszernek is, futtatható automatikusan, időzített módon, így minden probléma nélkül megoldható vele a *Windows 2000* és *XP* ügyletek időszakos mentése. A dolognak ráadásul az az előnye is megvan, hogy ezek a mentések a rendszer teljes állapotát is tartalmazzák, beleértve a regisztrációs adatbázis (registry) is. A *UNIX* különböző változatait, illetve *Mac OS X*-et futtató ügyletek mentése szintén zökkenőmentesen megoldható *Samba* megosztások segítségével. A *Mac OS X*-en alapértelmezésként telepítve van az *smb* ügyfél, a különböző *Linux* terjesztéseknél pedig vagy ugyanez a helyzet, vagy csomagból kell telepítenünk ezt a szoftvert. Ha ez

3. Lista Egy egyszer időbélyeg- és
lemezterület-megfigyelő program

```
#!/bin/sh
# Szabad hely lekérdezése
minden partíción
# Időbélyegek megjelenítése
időrendi sorrendben
BACKUPS=/data1 # Az
ellenőrizendő könyvtárak
azonosítása
# A mentés neve a megfelelő nap
lesz
BACKUPNAME=`date +%A`
#Timestamp
date > /var/log/
↳ backup.$BACKUPNAME.log
# Lemezterület a partíciókon
df -k > /var/log/
↳ backup.$BACKUPNAME.log
echo ' ' >> /var/log/
↳ backup.$BACKUPNAME.log
# Az adatmentő kiszolgáló
időbélyegeinek listája
# Az ls -lRt parancs sokkal
beszédesebb
ls -lRt $BACKUPS/* >> /var
↳ /log/backup.$BACKUPNAME.log
# Levélben elküldjük a naplót
a rendszergazdának
cat /var/log/
↳ backup.$BACKUPNAME.log | mail
↳ -s 'Backup Check'
backup_guy@mycompany.com
```

megtörtént, akkor az *smb* megosztást a *foo* gép */backup* könyvtára alá kell becsatolnunk:

```
mount -t smbfs -o username=bob,
↳ password=somepassword
↳ //bar/foo /backup
```

Ha ezt a műveletet automatikusan szeretnénk elvégeztetni a rendszerrel már a bootolás során, akkor a következőképpen kell módosítanunk a */etc/fstab* fájl tartalmát:

```
//bar/data1/foo /backup smbfs
↳ rw,username=bob,password=some
↳ password 0 0
```

Biztonságos adatmentés külső telephelyre

Ha még nagyobb biztonságban szeretnénk tudni adatainkat, készíthetünk

róluk másolatot egy olyan külső telephelyen található kiszolgálóra is, amely kellően messze van mind a munka közben használt gépünktől, mind pedig a helyi kisegítő kiszolgálótól. Megtehetjük például azt, hogy a belső, adatmentésre használt kiszolgáló tartalmát hetente egyszer áttükörözzük erre a külső telephelyre. Ha a belső gépekkel valamiféle katasztrófa történik, például leég az iroda, még mindig lesz egy teljes másolatunk az adattartalomról. Egy ilyen hálózat elvi vázlatát mutatja a 2. ábra.

A 2. Lista egy olyan egyszerű szkriptet mutat, ami a belső kiszolgáló tartalmát áttükörözi a külsőre az rsync segítségével. Az ilyen mentéseket nyilván adott időközönként automatikusan érdemes végrehajtani. Szintén fontos, hogy a mentés folyamatáról napló készüljön, illetve hogy ennek tartalmát rendszeresen ellenőrizzük is.

Az adatmentő kiszolgáló megfigyelése

Egy a cikkben vázolt adatmentési rendszerhez mindenképpen tartoznia kell egy megfigyelő programnak is, ami egyrészt ellenőrzi, hogy az ütemezett mentések valóban a tervnek megfelelően történnek-e, felügyeli magát a folyamatot, és megfigyeli a mentés céljaként használt gép állapotát (például hogy nem telt-e be a merevlemez). A 3. Lista egy ilyen megfigyelést végző kódot mutat, ami a cron segítségével szintén szabályos időközönként fut le, és jelentést küld a végrehajtott mentési műveletekről, illetve a pillanatnyilag rendelkezésre álló lemezterületről.

Linux Journal 2006. 141. szám



Duncan Napier

Brit Columbiában, Vancouver területén dolgozik számítógépes és műszerezési tanácsadóként.

KAPCSOLÓDÓ CÍMEK

A cikkhez tartozó források a következő címen találhatóak:

↳ www.linuxjournal.com/article/8635

© Kiskapu Kft. Minden jog fenntartva