

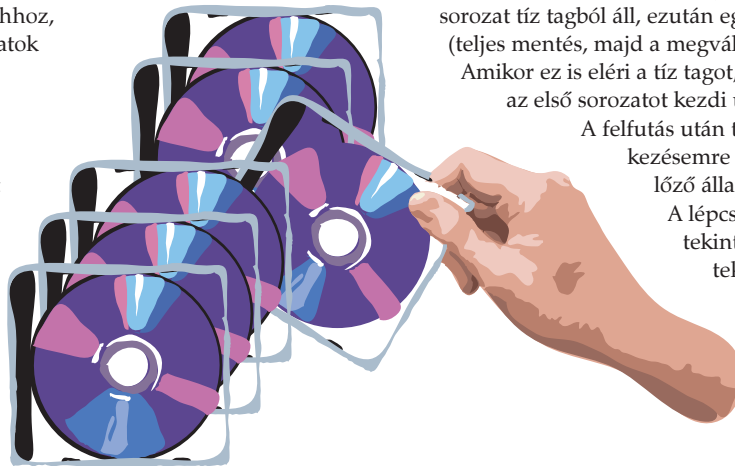
Mentette már az e hetit?

A datainkból egy példány nem példány. Két példány fél példány, három példány „a” példány, és negynél kezdődik a biztonság... Mindig büszke voltam rá, hogy az adataim – aránylag – biztonságban vannak, tanítványaimnak is nemegyszer beszéltem erről.

Mind az otthoni, mind a munkahelyi számítógépeken ugyanarra a munkakörnyezetre van szükségem: ezt egy fiókba (rack) tett kisebb merevlemez biztosítja, míg az operációs rendszer a ki nem vehető merevlemezen van, ami mellett azért marad még elég szabad hely ahhoz, hogy a fiókban lévő munkaadatok másodpéldánya elférjen.

Ablakos operációs rendszerű koromban az adatmentés igen egyszerűen történt: a gép kikapcsolása előtt egy kötegelt feldolgozás kitakarította a szemetet, majd a fiókban az úgynevezett archiv bit felhasználásával a legutóbbi mentés óta megváltozott fájlokat igen könnyedén (egyetlen xcopy utasítással) tükröztem az operációs rendszer melletti szabad helyre.

Ezt a szokásomat a pingvin sem változtatta meg, egészen addig, amíg... amíg elég látványosan rá nem döbentem arra, hogy egyszerű módszerem a hordozott merevlemez tönkremenetele (elvesztése) ellen ugyan megvéd, de a programban rejtőző hibák ellen nem sokat ér. Történt egy emlékezetes napon, hogy fájlrendszerhiba folytán a legutóbb használt két táblázatomból keresztbe láncolódott – természetesen a kulcsfontosságú része veszett el, és az érdektelen megkétszereződött. Az üzemszerű mentés pedig megsemmisítette a kulcsfontosságú fájl előző másolatát... Amikor a jelenség megismétlődött, elhatároztam (mivel mégsem ütött meg a guta),



hogy bonyolítom egy kissé az adatmentési módszeremet. Alapgondolatomban nem új: lépcsőzetes mentésnek (incremental backup) hívják már jó ideje. Először a teljes munkaterület mentése történik meg egyetlen tar *gzip* fájlba, majd minden további alkalommal csak az időközben módosult állományokat mentem egy-egy újabb *tgz*-be. Ilyen módon, ha bármilyen hibára fény derül, jó néhány mentett korábbi állapotom akad, és sokkal nagyobb valószínűséggel találok ép példányt egy fontos, de tönkrement állományból. Természetesen a rendelkezésemre álló hely véges, ezért egy ilyen sorozat tíz tagból áll, ezután egy új sorozat kezdődik (teljes mentés, majd a megváltozott állományok).

Amikor ez is eléri a tíz tagot, a következő mentés az első sorozatot kezdi újra és így tovább.

A felfutás után tehát bármikor rendelkezésemre áll legalább tíz meglévő állapot, legfeljebb húsz.

A lépcsőzetességre való tekintettel a helyigény sem tekinthető túlzottnak.

Természetesen a mellékelt programcska nem általános csodaszer. Első és talán legfontosabb korlátja,

hogy a mentendő fájlok listáját a legutóbbi módosítás dátuma alapján állítja össze. A PC-k belső naptárára viszont semmilyen garancia nincs... A lépcsőzetesség és a valószínűleg felhasználható hely miatt alapvetően irodai, illetve programozói jellegű géphasználat esetén alkalmazható, ahol nem túl sok és nem túl nagy fájl keletkezik vagy módosul (ellentétben mondjuk a videovágással és -szerkesztéssel).

A mentés időigénye teljes mentésnél (a sorozatok első tagja) pár perc, a továbbiakban ennek töredéke. Ennyit szerintem mégér a lényegesen megnövelt adatbiztonság.

Keszthelyi András (KEA@turul.banki.hu)

```
#!/bin/bash
#
# Az alanti 4 változó értéke egyéni ízlés
# szerint állítható
MaxNoOfFiles=10           # Legalacsonyabb
                           # értéke: 2
MunkaKtar="/mnt/hordd"    # Nincs / a
                           # könyvtárnevek végén!
MMP="/mnt/hda4"          # Mentés Mount Pont
MKT="ment"                # Mentés KonyvTar

#####

CKT=$MMP/$MKT             # célkönyvtár
```

```
Ser1="Ment1_"
Ser2="Ment2_"
Teljes="Nem"

# fájlrendszer-befűzések ellenőrzése devfs
# esetén a cutot másképpen kell paraméterezni
if [ `qwq`mount | grep $MunkaKtar |
  ↵cut -d " " -f 3 ` != qwq$MunkaKtar ]; then
  echo "A munkakönyvtár nem érhető el,
  ↵mentés nem történt"
  exit
fi
```

folytatás a következő oldalon

folytatás az előző oldalról

```

# csak a ktar elérhetőséget kell vizsgálnunk,
# vö. lentebb az első futással!
if [ ! -d $CKT ]; then
    echo "A célkönyvtár nem érhető el, mentés
        ↪nem történt"
    exit
fi

# szabad hely ellenőrzése
let Kell=`df | grep $MunkaKtar |
    ↪tr -s " " | cut -d " " -f 3`
let Van=`df | grep $MMP | tr -s " " |
    ↪cut -d " " -f 4`
if [ $Van -le $Kell ]; then
    echo "Nem biztos, hogy elegendő a hely
        ↪a mentéshez, nem csináljuk"
    exit
fi

# kitakarítjuk a szemetet kiterjesztés alapján
# válogatunk, egyéni ízlés szerint bővíthető
echo -n "Takarítás... "
find $MunkaKtar -type f \( -name \*.bak -o
    ↪-name \*.$\* -o -name \*.img -o -name \*.tmp
    ↪-o -name \*.wbk \) -exec rm {} \;
echo "vege"

#Hány sorozat és mennyi darab van:
if [ `ls $CKT | wc -l` -eq 0 ]; then
    echo "Ures kt., elso futas..."
    Teljes="Igen"
    touch -t 198001010000 ${CKT}/mentdatum.txt
    echo "${Ser1}000-000000" >
        ↪${CKT}/lastfile.txt
fi

FC1=`ls ${CKT}/${Ser1}* 2> /dev/null | wc -l`
FC2=`ls ${CKT}/${Ser2}* 2> /dev/null | wc -l`

SerNo=`cat ${CKT}/lastfile.txt | cut -c 5`

if [ $SerNo -eq 1 ]; then
    #Legutóbbi mentés az 1. jelű sorozatba ment
    FNamePrefix=${CKT}/${Ser1}
    if [ $FC1 -ge $MaxNoOfFiles ]; then
        # Megvan a sorozatból a max. darabszám,
        # meg kell kezdeni a 2. jelű sorozatot,
        # (ha van régebbiről, törlendő)
        FNamePrefix=${CKT}/${Ser2}
        rm ${CKT}/${Ser2}*
        find ${CKT} -type f -name ${Ser2}\*
            ↪-maxdepth 1 -exec rm {} \;
        echo "2. sorozat torolve: `date`"
            ↪>> ${CKT}/mentdatum.txt
        Teljes="Igen"
    fi
fi

if [ $SerNo -eq 2 ]; then
    #Legutóbbi mentés a 2. jelű sorozatba ment
    FNamePrefix=${CKT}/${Ser2}
    if [ $FC2 -ge $MaxNoOfFiles ]; then
        # Megvan a sorozatból a max. darabszám,
        # meg kell kezdeni az 1. jelű sorozatot
        # (ha van régebbiről, törlendő)
        FNamePrefix=${CKT}/${Ser1}
        find ${CKT} -type f -name ${Ser1}\*
            ↪-maxdepth 1 -exec rm {} \;
        echo "1. sorozat torolve: `date`"
            ↪>> ${CKT}/mentdatum.txt
        Teljes="Igen"
    fi
fi

# Ezen a ponton: ha az egyik sorozatunk
# elérte a max. darabszámot, letöröltük a
# másikat és beállítottuk a fájlnev elejét,
# meg hogy teljes v. részleges mentést kell
# csinálni. Itt jön maga a mentés.
echo -n "A tenyleges mentes kov.: "

FName=$FNamePrefix$(date +%j)-$
    ↪(date +%H%M%S).tgz

if [ "$Teljes" = "Igen" ]; then
    tar -czf $FName $MunkaKtar 2 > /dev/null
    echo -n "teljes mentes... "
else
    # csak a frissebbeket kell menteni
    echo -n "frissebbek mentese... "
    find $MunkaKtar -type f -anewer
        ↪${CKT}/mentdatum.txt -exec ls {} \;
        ↪> /tmp/mentlista.$$
    tar -czf $FName -T /tmp/mentlista.$$
        ↪2 > /dev/null
    if [ -f /tmp/mentlista.$$ ];
        ↪then rm /tmp/mentlista.$$; fi
fi
basename `echo $FName` > ${CKT}/lastfile.txt
basename `echo $FName` >> ${CKT}/mentdatum.txt
echo "rendben(?)."

###EOF

```

