

Operációs rendszerek

2. gyakorlat: UNIX rendszergazdai ismeretek 2

Az operációs rendszer feladata, hogy gondoskodjon a fájlrendszer konzisztenciájáról. Ennek biztosításához elengedhetetlenül fontos, hogy a gép kikapcsolása előtt *umount*-oljuk. Ha mégis valami okból (pl. áramszünet, hardver hiba) ez elmarad, a fájlrendszer sérülhet, mert a diszkre írt adatok egy része még nem került fizikailag a diszkre csak a buffer-ben van.

Most próbáljunk meg tönkre tenni az UmL egy virtuális diszkjén lévő fájlrendszert. Ehhez állítsuk le a virtuális gépet, majd egy szövegszerkesztővel szerkesszünk bele az egyik diszket reprezentáló fájlba (pl. a banyek nevű fájl). Lehetőség szerint ne a gyökér fájlrendszert, mert az bonyolultabb hibákhoz vezethet.

Sérült fájlrendszer helyreállítása

Ha sérült fájlrendszert mountolunk, akkor bizonyos hibák már ilyenkor kiderülhetnek. Bizonyos események (hibák, áramszünet, stb.) után érdemes és szükséges lehet megvizsgálni a fájlrendszert, hogy konzisztens-e.

A Linux rendszerek 2nd extended fájlrendszere nyilvántart egy számlálót, ami a mountolások számát tartalmazza. Ha ez a számláló elér egy (a fájlrendszer létrehozásakor megadott) értéket, akkor jelez, hogy a fájlrendszer ellenőrzése ajánlott.

A fájlrendszer ellenőrzését UNIX rendszereken az *fsck* program végzi. Az utasítást a rendszergazda adhatja ki, pl. az alábbi formában:

```
% fsck -afV /dev/ubd/1
```

Az utasítás kiadásának feltétele, hogy a megadott eszköz fájlrendszere ne legyen mountolva. Ennek hatására a megadott eszközön lévő fájlrendszer ellenőrzi és automatikusan helyreállítja. Az program ellenőrzi a fájlrendszert és ha hibát talál, akkor megkérdezi, hogy javítsa-e. Az „a” kapcsoló hatására nem tesz fel kérdést, automatikus igennel válaszol. Az „f” kapcsoló hatására az ellenőrzés mindenképpen végbemegy, akkor is, ha még nem érte el a számláló a kellő értéket. A „V” kapcsoló hatására beszédes kiírást ad a program.

A fájlrendszer többféle sérülése lehetséges, és ezek nem mindegyike javítható maradandó veszteségek nélkül. Bizonyos sérülés esetén viszont az *fsck* találhat a

fájlrendszeren olyan, blokkokat, melyek a sérülés miatt nem részei egyetlen fájlnek sem. Ezeket a blokkokat fájlként a program bemásolja a fájlrendszer gyökerében lévő *lost+ found* jegyzékbe. Ezután, ha nagyon szükséges, ezekből a blokkokból össze lehet szedni az elveszett fájlkat.

Az operációs rendszer gyökerét képző fájlrendszer ellenőrzése azért bonyolultabb, mert az már a kernel betöltése után mountolódik, még az init processz indulása előtt. Ha ez a fájlrendszer sérül, akkor a rendszergazda 1-es üzemmódba állítja a rendszert (egy felhasználós üzemmód) és a gyöker fájlrendszer mountolását csak olvashatóra változtatja. Ilyenkor van lehetőség az *fsck* futtatására, ami helyreállítja a fájlrendszert.

```
# init 0
# mount /dev/ubd/0 / -o remount,ro
# sync
# fsck -aV /dev/ubd/0
# mount /dev/ubd/0 / -o remount,rw
# init 2
```

Ennek persze megvan az a veszélye, hogy a fájlrendszer valamilyen, az operációs rendszer szempontjából létfontosságú fájlja helyrehozhatatlanul megsérül. Az ilyen katasztrófák után a helyreállítás csak úgy lehetséges, ha van biztonsági mentésünk a sérült fájlrendszerről.

Biztonsági mentések készítése és visszatöltése

Mivel az *fsck* csak a fájlrendszer struktúráját képes helyreállítani, nem garantált, hogy az adatok nem vesznek el. Adatveszteség ellen az egyetlen hatékony eszköz, ha rendszeres biztonsági mentéseket készítünk a rendszer fájljairól. A biztonsági mentések készítéséről és kezeléséről a számítógép üzemeltetője köteles biztonsági politikát készíteni, amit a rendszergazda, vagy a biztonsági mentésekért felelős megbízott hajt végre.

Biztonsági mentések készítésére a UNIX rendszerek a *tar* (Tape ARchiver) programot biztosítják. A program jellegzetessége, hogy szalagos egységre történő mentések készítéséhez készült és csak külön opcióként lehetséges, hogy fájlba készüljön a mentés. Mivel azonban egy egyszerű és jól használható programról van szó, így szabványosnak tekinthető és minden UNIX rendszeren megtalálható. Nemcsak a rendszergazda használhatja, hanem minden felhasználó a saját jogosultsági körében.

Megjegyzés: szabványossága és elterjedtsége miatt gyakori, hogy a UNIX rendszerekre készült alkalmazásokat ezzel a programmal csomagolva kerülnek (köz-) forgalomba.

A programot az alábbi formában hívhatjuk a parancssorból.

```
# tar <commands> <args>
```

Az argumentumok attól függenek, hogy milyen utasítást akarunk végrehajtani a programmal.

Lehetséges műveletek:

Create (c) – archívum létrehozása

Pl.

```
# tar c /dev/tape01 /home
```

Ennek hatására a */home* jegyzék tartalmát lementi a *tape01* jelű szalagos egységre.

File (f) – átirányítás fájlba. Ez az utasítás csak valamelyik másik utasítással együtt használható.

Pl.

```
# tar cf /mentes.tar /home
```

Most az előbbi műveletet végzi, de a mentés a */mentes.tar* fájlba kerül.

Speciális funkció, ha fájlnevként egy „-” adunk meg, akkor a program az stdin/stdout adatfolyamokat használja.

Pl.

```
# tar cf - /home | ssh másik_gép.neve.hu -l felhasználó 'cat > mentes.tar'
```

Enne hatására a program mentést készít a */home* jegyzékről, amit a standard kimenetén küld ki. Ezt átirányítottuk az *ssh* bemenetére, amivel egy másik gépre, más felhasználóként bejelentkezve egy *mentes.tar* fájlba irányítjuk az adatfolyamot.

Extract (x) – archívum kicsomagolása

Pl.

```
# tar xf /mentes.tar
```

Az előbbi archívum fájl kicsomagolja az aktuális jegyzékbe.

Test (t) – az archívum tartalmának listázása.

Pl.

```
# tar tf /mentes.tar
```

Kilistázza, hogy az archívumban milyen fájlok és jegyzékek vannak.

Update (u) – archívum frissítése. Az ún. inkrementális mentés azt jelenti, hogy egy már meglévő mentés vizsgál a program és csak azokat a fájlokat menti le újra, amelyek megváltoztak a mentés készítése óta.

Pl.

```
# tar uf /mentes.tar /home
```

Figyelem! Egy archívum mérete a frissítés során csak növekedni képes akkor is, ha a mentett fájlok mérete csökkent.

Append (r) – archívumhoz való hozzáfűzés.

Pl.

```
# tar rf /mentes.tar /etc
```

A már létező archívumhoz hozzáfűzzük az /etc jegyzékről készített mentést.

Verbose (v) – beszédes kimenetet produkál, láthatjuk, hogy éppen melyik fájlt vagy jegyzéket dolgozza fel a program.

Dereference (h) – szimbolikus linkek helyett az általa mutatott fájlt menti.

Local filesystem (l) – csak annak a fájlrendszernek a tartalmát menti, amin a kiindulási pont is van.

Multi-volume (M) – kötet tördelés. Akkor használjuk, ha előreláthatólag az archívum nem fog elférni egyetlen adathordozón és több részre kell tördelni.

Zip (z) – az archívum tömörítése. Alapértelmezés szerint a *tar* program csak csomagol és nem tömörít. Ennek az utasításnak a megadásával az archívumot egy *gzip* (újabbán *bzip2*) tömörítő programmal tömöríti. A tömörített archívumok szokásos kiterjesztései *.tar.gz* vagy *.tgz*.

A program használatával kapcsolatosan Bartók-Laufer: UNIX felhasználói ismeretek c. könyv 383. oldalán találunk példákat. A kellő gyakorlat megszerzése érdekében ezeknek a feladatoknak az elvégzése javasolt.