Dióhéjban az ext3 fájlrendszerről

Ha biztonságban szeretnénk tudni féltett adatainkat, használjunk valamilyen naplózó fájlrendszert, például az ext3-at.

A Linux „hagyományos” fájlrendszere az `ext2`. Ez egy jól kipróbált, megbízható rendszer, azonban akad egy gyenge pontja: érzékeny a hirtelen leállásokra. Az `ext2` a hatékonyság növelése érdekében a merevlemezt úgynevezett aszinkron módban használja. Ennek az a hátránya, hogy ha az adatátvitel akár csak egyetlen pillanatra is megszakad, fennáll az adatvesztés kockázata.

A metaadatok szerepe

Az, hogy történik-e adatvesztés, csak azon múlik, hogy mikor következett be a hirtelen leállás. Ha az adatok lemezre mentése után történik, akkor nincs is semmi baj, megúsztuk. Ha a kiírás előtt, akkor sajnos az új (még ki nem írt) adatok örökre elvesznek, de legalább a régiek megmaradnak. A legrosszabb eset az, amikor pont az írás folyamata közben következik be a sajnálatos esemény. Ekkor ugyanis könnyen előfordulhat az, hogy az állomány első fele az új, a másik fele még a régi adatokat tartalmazza. Ez rendkívül kellemetlen, de korántsem annyira, mint amikor az úgynevezett metaadatok írásakor áll le a rendszer. A metaadatok olyan adatok, amelyeknek a nyilvántartását a lemezen lévő állományokról maga a fájlrendszer végzi. Ilyen például a fájl neve, a mérete, létrehozásának a dátuma, a jogosultságok és a legfontosabb: hogy hol helyezkedik el a lemezen. Ha a metaadatok nem tudnak rendesen kiíródni, az is előfordulhat, hogy az egész fájl örökre eltűnik.

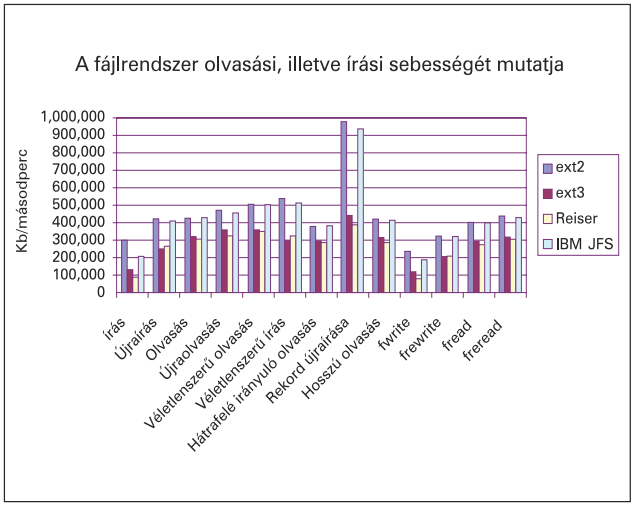
Azért nincs minden veszve, mert a legtöbb esetben (ha nem is teljes mértékben) helyreállítható a károsodás. Erre szolgál az `fsck` nevű program is, ami minden szabálytalan leállítás után a rendszer indulásakor önműködően elindul. Ez egy fájlrendszerelemző program, ami az esetleges hibákat próbálja meg felderíteni és helyreállítani a fájlrendszerben. Amennyiben a metaadatok megsérültek, általában vissza lehet állítani őket, ha volt róluk másolat. Ha nem volt, akkor a sérült állomány tartalma véglegesen elveszett.

Ezekután kétségtelen, hogy aki létfontosságú adatokat `ext2`-n tárol, az a tűzzel játszik (hacsak nem tart otthon szünetmentes tápot – bár hirtelen leállást más is okozhat, nem csak áramszünet). A másik nagy gond az `ext2`-vel az, hogy az `fsck`-val történő ellenőrzés időigényes, bizonyos esetekben több óráig is eltarthat. Ez különösen kellemetlen lehet egy hálózati kiszolgáló esetében, ugyanis amíg az ellenőrzés tart, addig nem tudja ellátni a feladatait.

A naplózó fájlrendszerek áttekintése

Mi lehet a megoldás? Használjunk úgynevezett naplózó (journaling) fájlrendszereket.

Az ezek mögött meghúzódó alapgondolat az, hogy mielőtt az írási műveletet végrehajtanánk a lemezen, egy csakis erre a célra fenntartott területre (a naplóba) beírjuk, hogy mit kellene elvégeznünk. A napló tulajdonképpen egy adatbázis, amelybe szekvenciálisan írjuk be a lemezen elvégzendő változtatásokat.



Mit nyerünk ezzel? Tegyük fel, egy hirtelen leállás történik. A rendszer újraindítása után a fájlrendszer ellenőrzi a naplót. Ha nem talál benne új bejegyzést, akkor két dolog lehetséges: vagy minden írási művelet befejeződött, vagy a leállás még az új művelet naplóba történő felvétele előtt bekövetkezett. Az utóbbi esetben az új adatok sajnos véglegesen elvesztek, de a régiek sértetlenül megmaradnak.

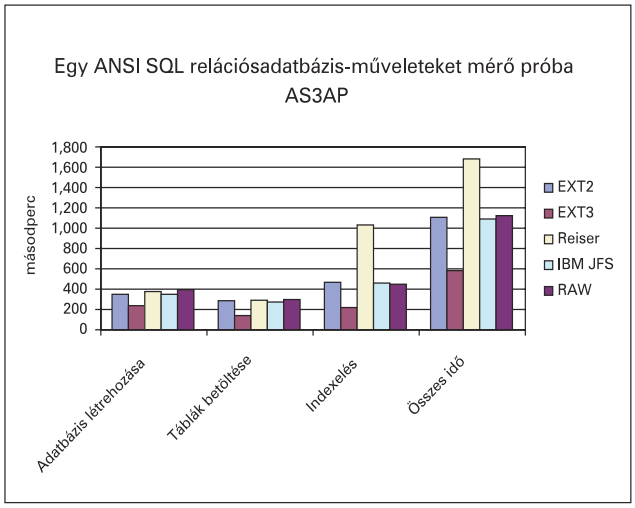
Amennyiben új bejegyzést talál a naplóban, akkor a leállás minden bizonnyal a lemezre való írás pillanatában következett be. Ilyenkor azonban nem kell számolnunk az `ext2` esetében előforduló adatvesztés lehetőségével, mivel a napló segítségével a fájlrendszer a félbe maradt műveletet be tudja fejezni. Ez az egész folyamat nem tarthat tovább pár másodpercnél, tehát az órákig tartó „fsckázásnak” is búcsút inthetünk.

A naplózó fájlrendszerek másik előnye, hogy sok esetben gyorsabbak, mint némely hagyományos fájlrendszer. Az írási kérések ugyanis rendszerint a lemez különböző helyein lévő blokkokra érkeznek, így a fejet állandóan pozícionálni kell. Ez komoly visszaesést jelent a teljesítményben. A naplózó fájlrendszerek esetében először mindig szekvenciálisan a naplóba írunk, így a fejet több írási művelet esetén is csak egyszer kell pozícionálni. A naplóban feltüntetett tranzakciókat pedig ráérünk elvégezni akkor, amikor a lemeznek nincs más feladata.

A naplózó fájlrendszerekre jellemző még, hogy nem annyira szétszórtan tárolják az állományokat, mint más fájlrendszerek. Ezenkívül különböző egyszerűsítéseket és javításokat is végeznek. Példaként említhetjük, hogy az AIX fájlrendszere például a kis könyvtárak tartalmát a fájlleíróban (i-node) tárolja.

A naplózó fájlrendszer tehát jó dolog. Nézzük, milyen kínálat áll belőlük rendelkezésre Linux alá. Használhatjuk az IBM által kifejlesztett, az OS/2 megvalósításra épülő JFS-t (lásd még a 22–25. oldalon). Ezzel a legnagyobb gond az, hogy a többihez képest viszonylag kevés segédprogram lelhető fel hozzá.

Az XFS szintén egy másik rendszerből átültetett naplózó fájlrendszer. Ez az SGI Irix nevű operációs rendszeréből származik, és számos alkalmazás támogatja. A ReiserFS egy szintén nagyon megbízható fájlrendszer, fejlesztése folyamatosan zajlik (bővebben a 34–36. oldalon olvashatunk róla).
A fenti fájlrendszerek fejlesztésével (illetve azoknak Linuxra történő átültetésével) párhuzamosan egy másik naplózó fájlrendszer is megjelent, az `ext2`-vel teljesen együttműködő `ext3` fájlrendszer.



Az `ext3` kifejlesztésére több okból is szükség volt. Először is a Linux-felhasználók többsége `ext2`-t használt, ezen tárolták több gigabájtnyira duzzadt adataikat. Ahhoz, hogy egy másik, korszerűbb fájlrendszerre térjenek át, meg kellett oldaniuk adataiknak egy másik lemezrészre történő mentését, majd az átformázás után az adatok visszamásolásának is zökkenőmentesen kellett mennie. Aki már csinált ilyet, tudja, hogy elég körülményes művelet. (Nem is beszélve arról, ha az adatok mellett a kérdéses lemezrész magának a rendszernek is az otthona.) Másrészt az `ext2` akkorra már kiforrott fájlrendszer volt, a felhasználók többsége bízott benne.
Így nem csoda, ha örömmel fogadtak egy olyan naplózó fájlrendszert, amelyik megbízhatóan együttműködik az `ext2`-vel. Ez azt jelenti, hogy a felhasználó egy egyszerű művelet segítségével – a rendszer működése közben – régi fájlrendszerét `ext3`-ra alakíthatja át, anélkül, hogy adatai ideiglenes tárolásáról gondoskodnia kellene. Sőt az `ext3` bármikor visszaalakítható `ext2`-vé.
A másik nem elhanyagolható pozitív tulajdonsága az `ext3`-nak az, hogy megszabadulhatunk az `fsck`-val történő állandó lemezellenőrizgetésektől. Egy hirtelen leállás utáni ellenőrzés, illetve az esetleges hibák kijavítása egy több gigás lemezrész esetében akár órákat is igénybe vehet. Az `ext3` használatakor ez a művelet azonban nem tarthat tovább, mint pár másodperc. Az igazsághoz hozzátartozik az is, hogy az `ext3` igazából nem egy teljesen új fájlrendszer. Valójában nem más, mint az eredeti `ext2` kibővítése a naplózó szolgáltatással. A napló itt nem más, mint egy különleges rejtett állomány a gyökérben.
Ennek a megoldásnak az az ára, hogy az `ext3` nem tartalmaz olyan lemezgyorsító szolgáltatásokat, mint a többi korszerű fájlrendszer. Elméletileg azonban így is gyorsabb az `ext2`-nél, mert hatékonyá teszi a fej mozgását. (A különböző sebességmérő próbák ez ügyben azonban nem mindig szolgálnak meggyőző bizonyítékkal.)

Átállás ext2-ről ext3-ra

A továbbiakban az `ext3`-as fájlrendszerrel fogunk foglalkozni, mivel erre pár másodperc alatt bárki „áttérhet”, és a Red Hat is ezt a fájlrendszert támogatja. Megnézzük, miképp formázhatunk egy üres lemezrészt, illetve hogyan alakíthatjuk át meglévő `ext2`-es fájlrendszerünket `ext3`-sá.
Mindenekelőtt a rendszermagnak támogatnia kell az `ext3` fájlrendszert. A 2.4.15-ös rendszermagtól kezdve minden változatban megtalálhatjuk, ha ennél régebbi rendszermagunk van, akkor külön foltot kell letöltenünk. Ezt a
➲ http://www.zip.com.au/~akpm/linux/ext3/ címről tehetjük meg. A foltot másoljuk be a rendszermag forrását tartalmazó könyvtárba, majd ugyaninnen adjuk ki a következő parancsot:

```
gunzip < ext3-2.4-x-y.gz | patch -p1
```

Az x az `ext3` folt változatszámát, az y pedig a rendszermag változatát jelöli.
Vigyázzunk arra, hogyha a gyökérlemez is `ext3`-as, akkor semmiképp se fordítsuk a támogatást modulba. A másik dolog, amire fel szeretnénk hívni a figyelmet: lehetőleg ne használjuk a rendszermagban található általános tárkorlát-támogatást (quota) az `ext3`-as fájlrendszerrel, mert rendszeromlást okozhat. Ha tárkorlátot szeretnénk használni, telepítsük az ***Alan Cox***-féle `-ac` foltokat, amivel efféle baleset elvileg nem fordulhat elő.
A rendszermagoldali támogatáson kívül szükségünk lesz még két csomagra is. Az első a `util-linux`, ami minden bizonnyal már telepítve is van. Arra figyeljünk, hogy ne legyen 2.11-nél idősebb változatú. A másik csomag az `e2fsprogs` névre hallgat, ez is megtalálható az összes terjesztésben.
Az `ext3` lemezrészt az `mke2fs` program segítségével hozhatunk létre a következő módon:

```
mke2fs -j /dev/eszk z
```

Egy meglévő `ext2`-es lemezrész átalakításához a `tune2fs`-t használhatjuk:

```
tune2fs -j /dev/eszk z
```

Ez a művelet semmiféle adatvesztést nem okozhat.
Az `ext2` lemezrész átalakítása után ne felejtsük el a */etc/fstab* állományt módosítani: a megfelelő bejegyzésnél a fájlrendszer típusa oszlopot (ez a harmadik) változtassuk `ext3`-ra.
Minden mást úgy hagyhatunk, ahogy volt. Nem árt még leállítanunk az önműködő `fsck`-ellenőrzést, mivel az `ext3` ezt nem igényli:

```
tune2fs -i 0 -c 0 /dev/eszk z
```

(a `-i` kapcsoló azt mondja meg, hogy az `fsck` milyen gyakran fusson le másodpercként. A `-c` kapcsolóval pedig azt állíthatjuk be, hogy hány befűzés után fusson le az `fsck` önműködően).
Ha valaki valamilyen okból kifolyólag a fájlrendszerét vissza szeretné állítani `ext2`-esre, a */etc/fstab* visszaírása után a fent említett módon állítsa be az `fsck`-ellenőrzések gyakoriságát.

Garzó András (garzoand@interware.hu)
Körülbelül három éve foglalkozik Linux- és más Unix-rendszerekkel. Legjobban az operációs rendszerek lelkivilága érdekli, de nyitott egyéniség. Kedvenc étele a palacsinta, és van egy Richard nevű macskája. Minden észrevételt, megjegyzést, levelet szívesen fogad.