

## Programból megvalósított RAID

A Linux adta lehetőségekkel és olcsó alkatrészekkel is nagyfokú adatbiztonságot érhetünk el.

**A** RAID a Redundant Array of Inexpensive Disks kifejezés rövidítése, amelyet magyarul nagyjából *olcsó lemezek hibátűrő tömbje*-ként adhatnánk vissza. Az eljárás lényege: az adatokat több lemezen, szétszórtan helyezük el, ezáltal próbálva meg teljesítménynövekedésre vagy nagyobb biztonságra szert tenni. Ezt több gyártó SCSI-vezérlőkártyája is biztosítja, de ugyanezt a rendszermag támogatásával programból is megtehetjük – így az adattöbbszörözést akár két IDE-merevlemezrel megvalósíthatjuk. Külön öröm, hogy a Linux programszintű RAID-vezérlőjének készítője, *Ingo Molnar* magyar származású (mingo@chiara.csoma.elte.hu). A RAID a fájlrendszerreteg alatt helyezkedik el, semmi köze sincs hozzá, ezért bármilyen fájlrendszerrel használni tudjuk.

### IDE vagy SCSI?

Többben is vannak, akik rendíthetetlenül állítják, hogy Linuxot csakis SCSI-s géppel lehet futtatni, IDE-merevlemez tenni linuxos kiszolgálóba főbenjáró bűn. Ez természetesen nem igaz. Az IDE kevés lemez esetén gyakran gyorsabb, mint a SCSI, viszont mondjuk egy nyolc lemezes kiszolgáló IDE-sínnel nem éppen bizalomgerjesztő. A túl hosszú IDE-kábelek majdnem biztos, hogy hibás adatokat eredményeznek. Az IDE másik gondja, hogy a szolgálalemez elérése nagyon lassúvá válhat, ami pedig RAID esetén, ahol gyakori ugyanazon adat több lemezre írása, nagy teljesítménycsökkenést idézhet elő. Ezért ha úgy döntesz, hogy IDE-s merevlemezekkel oldod meg a RAID-et, feltétlenül csak elsődleges merevlemezeket használj.

### Mire van szükség a RAID-hez?

*Legalább két merevlemezre.* A programból megvalósított RAID egy lemezzel is létrehozható, amennyiben több lemezterületet használsz. Ez azonban ismét teljesítménycsökkenéshez vezet, hiszen ugyanazt az adatot nem két lemez írja párhuzamosan, hanem egy lemez, csak két lemezterületre.

*2.4-es rendszermagra.* A 2.0-s és 2.2-es sorozatban is létezik RAID-támogatás, de kevesebb lehetőséget biztosít, és a fejlesztője szerint nem annyira megbízható, mint a 2.4-esben szereplő. Ha valamiért mégis ragaszkodsz régebbi rendszermagodhoz (én nem tenném, mert az új leírhatatlanul gyorsabb), és az újabb RAID előnyeit is élvezni szeretnéd, kénytelen leszel letölteni egy foltot, és újrafordítani a rendszermagot. A legfrissebb mag jelenleg a 2.4.14-es. Továbbá szükséged lesz a raid-tools csomagra. Ezt a legtöbb terjesztés tartalmazza, ha mégsem találnád, töltsd le az Internetről. A rendszermagot az <http://ftp.kernel.org/pub/linux> címről szerezheted be, a raid-tools ugyanitt, a *daemons/raid/alpha* alkönyvtárban található.

### Néhány magyarázatra szoruló fogalom

Egy tömbben raid- és tartaléklemezeket (spare disk) különböztetünk meg. A tartaléklemezeket a rendes működés során valójában sosem használjuk, csak akkor lépnek működésbe, ha az egyik raidlemez kiesik. Ekkor indul el az összehangolás, hogy a tömb az új lemez teljes értékű tagja lehessen. Ezalatt

már lehet használni, viszont ha az összehangolás közben még egy raidlemez leáll, a RAID-megvalósítások egyike sem biztosítja az adatok sértetlenségét. A RAID-et természetesen tartaléklemez nélkül is meg lehet oldani.

A tömbbel kapcsolatos tudnivalók egy beállítási állományban találhatóak. A tömb használatba vételéhez minden rendszerindítás után fel kell éleszteni, ezt pedig csak a beállítási állomány alapján lehet megtenni. Ez az állomány azonban nem mindig érhető el. Gondoljunk arra, ha a saját fájlrendszerünkben akarunk RAID-et készíteni, hogyan fog elindulni a rendszer? Erre való a *persistent superblock*. Ha bekapcsoljuk, minden hibátűrő lemezterület elején egy superblock található, amelyből a rendszermag megtudhatja a tömbbel kapcsolatos adatokat, ehhez tehát nincs szükség a beállítási állományra. Ha azonban tükrözött lemezről akarjuk a rendszert betölteni, mindenképpen szükségünk lesz rá! Mindenkinek ajánlom, hogy a persistent superblock mellett még egy beállítási fájlt is tartson meg, ugyanis bármikor szükség lehet rá.

### Hogyan fogjak hozzá?

Először ellenőrizd, hogy a RAID-támogatást modulként fordítottad-e. Ha igen, mindenekelőtt a következő parancsot add ki: `modprobe md`

Ezzel betöltötte a modult. Ha eddig mindent jól csináltál, `proc` fájlrendszeredben egy új állományt fogsz látni: `/proc/mdstat`. Ez az a fájl, amely az egyetlen segítség a RAID állapotának megfigyeléséhez. Minden lépés után nézd meg, hogyan változott.

`cat /proc/mdstat`

Ha most kiadod, ezt kell látnod:

```
Personalities:
read_ahead not set
unused devices: <none>
Ezután el kell döntened, hogy milyen RAID-megvalósítást akarsz kivitelezni. Ennek megfelelően kell szerkesztened a /etc/raidtab állományt, ami nem más, mint a fentebb már említett beállítási fájl. Az alábbi egy egyszerű tükrözést állít be:
# RAID-0 (mirroring) beáll tæs
# ezen az eszk znØven Ørj k el majd a t mb t
raiddev /dev/md0
# raid-megval s tæs: linear, 0, 1, 4 vagy 5
raid-level 1
# raidlemezek száma
nr-raid-disks 2
# tartaléklemezek száma
nr-spare-disks 1
# (kB); minden megval s tæsnel mæst jelent,
# lásd kØsibb
chunk-size 4
# persistent superblock: 0 a nem, 1 az igen
persistent-superblock 1
# a t mb elemei; itt kell megadni a raid- Øs
# a tartalØk-lemezter leteket
device /dev/hda1
```

```
raid-disk          0
device             /dev/hdc1
raid-disk          1
device             /dev/hde1
spare-disk         0
```

Ezekután lássuk az egyes RAID-megvalósításokat!

### Linear mode

A segítségével több lemezt egyként kezelhetünk. Ha az első lemez betelt, az írás a következőn onnan folytatódik, ahol az előzőn abbamaradt. A lemezeknek nem kell azonos méretűeknek lenniük. A tömb mérete az elemek méretének az összege. Ebben az esetben nincs adattöbbszörözés, egy lemez kiesésével is elveszítheted minden adatodat. A chunk size-nak itt nincs sok értelme, viszont nem lehet elhagyni. Legalább két raidlemezre igényel, tartaléklemek nincsenek.

### RAID-0 (stripe)

A csíkozás hasonló a linear mode-hoz, csak annyi a különbség, hogy a lemezek párhuzamosan töltődnek fel. A lemezek nem feltétlenül azonos méretűek, a tömb mérete az összes lemez együttes mérete. Nincs adattöbbszörözés. A chunk-size határozza meg, hogy egy adatsorból mekkora „falat” kerülhet egy lemezre. Abban az esetben, ha két lemezből álló RAID-0-s tömb van, amelynél a chunk size 4 KB, egy 8 KB-os állomány mentésekor annak egyik fele az első lemezre, a másik pedig a másodikra íródik. Az írási folyamat ilyen módon közel feleannyi időt vesz igénybe. Az olvasás hasonlóan zajlik. Láthatod, mennyire fontos, hogy egy RAID-tömb eleme ne egy szolgálomeghajtó legyen! Legalább két raidlemez szükséges a kiépítéshez, tartaléklemek nincsenek.

### RAID-1 (mirroring)

A tükrözés az első megvalósítás, amely valós hibátűrést jelent. A lemezek között egyszerű tükrözés történik és minden raidlemez tartalma megegyezik. Így ha egy kivételével az összes lemez kiesett, az adatok még mindig sértetlenek maradnak. A tömb mérete megegyezik a legkisebb lemezrész méretével. A chunk-size íráskor semmilyen szerepet nem játszik, hiszen az adatot az összes lemezre fel kell írni, olvasáskor viszont azt határozza meg, hogy egy lemezről mennyi adatot kell sorban beolvasni – így az olvasás párhuzamosan történhet. Legalább két lemezrész szükséges, a tartaléklemek pedig elhagyhatók.

### RAID-4

Nagyon ritka megvalósítás. Az adatok a raidlemezeken csíkozva helyezkednek el, mint a RAID-0-nál, és egy kiemelt lemez van, amely a párosságot (parity) tárolja. A tömb méretét úgy számolhatjuk, ha a párosságot tároló lemezt nem számítva az összes raidlemez számát a legkisebb méretével szorozzuk meg. Lemez elvesztésekor a párosságotok alapján a helyreállítás egy tartaléklemre indul meg. A chunk size a párosságotot tároló lemezen a párosságblokkok mérete. A kiépítéshez legalább három lemezrészre van szükség, és legalább egy tartaléklemez rendszerbe állítása ajánlott.

### RAID-5

A RAID egyik legnépszerűbb fajtája. Hasonló a RAID-4-hez, csak a párosságotok nem egy lemezen tárolódnak, hanem az összes lemezen szétszórva. A tömb méretét ugyanúgy számoljuk, mint a RAID-4-nél. A chunk size ugyanazt jelenti, mint a „kistestvérénél”. Használatához legkevesebb három lemezrész szükséges, valamint legalább egy tartaléklemez ajánlott.

### RAID-10 (mirrored stripe)

A tükrözött csíkozánum beépített megoldás, hanem két RAID-0-s tömb RAID-1-es tömbje. Ritkán használt kiépítés.

Ha a fentiek mérlegelése után elkészültél a beállítási állománnyal, először ellenőrizd, hogy a fájlban szereplő eszköz létezik-e. Az én SuSE 7.2-esem például 4 MD-eszközt tartalmaz (/dev/md0, /dev/md1, /dev/md2 és /dev/md3). Szinte soha nincs többre szükség, amennyiben mégis, az új RAID-eszközt az alábbi paranccsal hozhatod létre:

```
mknod /dev/md4 b 9 4
# a 4-es szám változhat; lásd még
man mknod
```

Ezután létrehozzuk vagy felélesztjük a tömböt:

```
mkraid /dev/md0
```

RAID-1-nél ilyenkor rögtön megindul az összehangolódás.

Ha kiíratod a /proc/mdstat fájl, a folyamatot megfigyelheted:

```
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdc1[1] hda1[0]
      292224 blocks [2/2] [UU]
      [=====>.....] resync = 45.8%
      ↳ (134272/292224) finish=2.6min speed=990K/sec
unused devices: <none>
```

Bármilyen történjék, ha az mkraid nem szakad meg, a tömb már működik is. A fenti példában az eszközt már formázhatod, a folyamat végét sem kell megvárni:

```
mke2fs /dev/md0
```

Ezekután erre az eszközre is úgy hivatkozhatasz, mint bármilyen másra. Ha a RAID-et ki szeretnéd kapcsolni, add ki a következő parancsot:

```
raidstop /dev/md0
```

Ha RAID-ről szeretnéd indítani, a tömböt persistent superblokkal kell létrehoznod. A RAID elemeinek lemezterület-típusa 0xFD legyen, így a rendszermag a tömböt már induláskor önműködően felismeri; arra viszont ügyelj, hogy a LILO hivatalos változata nem támogatja a RAID-ről történő rendszerbetöltést. Vannak terjesztések (például a RedHat), amelyek különböző foltokkal próbálnak segíteni ezen, én viszont azt ajánlom, inkább készíts egy külön /boot lemezterületet. Szükségtelen nagyra „növeszteni”, és az sem fontos, hogy „biztonságban” legyen.

### Csak a kezemet figyeljétek...

Felmerülhet az igény, hogy a csereterület (swap) csíkozd. Az adattöbbszörözésnek ebben az esetben nincs túl sok értelme, viszont előnyös lenne a virtuális memória gyorsabb elérése. A csereterület nagyon egyszerűen RAID-0-zható, még a *raidtools* csomag sem kell hozzá, a rendszermagban ugyanis hosszú ideje létezik hozzá támogatás. Ha a /etc/fstab valahogy így néz ki:

```
# fontos: a csereter lettek azonos
/dev/hda2 swap swap defaults,pri=1 0 0
# elsibbsögsek
/dev/hdc2 swap swap defaults,pri=1 0 0
```

akkor a rendszermag önműködően elvégzi a csíkozást, anélkül, hogy ezt külön mondani kellene neki. Ezzel jelentős teljesítménynövekedésre tehetünk szert.

Fülöp Balázs

(xut@freemail.hu) 17 éves gimnazista diák. Imádja a Túró Rudit, a Debian Linuxot és a teheneket. Az ELTE Radnóti Miklós Gyakorlóiskola tanulója immár ötödik éve. Kedvenc írója Slawomir Mrołlek. Leginkább a számítógépes hálózatok biztonsága érdekli.