

## Biztonságos adattárolás – Szoftveres RAID kialakítása Linux alatt

Manapság egyre inkább előtérbe kerül az adatbiztonság kérdése otthoni felhasználóknál is. Konkrétan az adatok tárolásának biztonsága. Ezt ideig-óráig meg lehet valósítani szalagos vagy optikai egységgel, de belátható, hogy ezek az eszközök nem tesznek lehetővé valós idejű adatmentést. Erre találták ki a RAID-e.

© Kiskapu Kft. Minden jog fenntartva

**A** RAID rövidítés visszafejtésére általában kétféle „megoldást” talál az ember az interneten. Az egyik szerint „*Redundant Array of Independent Disks*” a betűszó jelentése, vagyis független lemezekből álló redundáns tömb, míg a másik megfejtés a „*Redundant Array of Inexpensive Disks*”, vagyis olcsó lemezekből álló redundáns tömb. Belátható, hogy mindkét írásmód jogos.

### RAID-ek típusai

A RAID-eknek számtalan típusa van, de otthoni felhasználó általában a RAID0-át használja, ha sebesség szükséges és RAID1-et, ha az adatbiztonság a fontos tényező. Céges környezetben használják még a RAID5-öt is, illetve ezen háromfajta RAID-ek keverékét.

### Linux kernel felkészítése és a szükséges csomagok

Ahhoz, hogy tudjunk RAID lemeztömböt Linux alatt kezelni, szükségünk van a kernelszintű támogatásra. 2.6-os kernel esetén keressük meg a *Device driver* pontot, és ebből válasszuk ki a *Multi-device support* pontot, ahol aztán már kedvünkre csemegézhetünk. Itt én csak a RAID1-et választottam ki, ugyanis így legalább nyugodtan alhatok, az adataim biztonságban lesznek. A RAID0-t én nem javasol-

nám senkinek, hiszen ha az egyik merevlemez meghalt a RAID-ből, akkor az egész tömbön tárolt adatoknak viszlát. Vagyis legyen inkább két fizikai eszközünk két fájlleíró táblával.

A RAID0-nak csak egy esetben van előnye: ha nagy adatátviteli sebességre van szükség, például videódigitalizáláskor.

A fordítás sikerességét úgy tudjuk ellenőrizni, hogy megnézzük az új kernellel létezik-e a */proc/mdstat* fájl. Ha igen, nyert ügyünk van. Ezzel megvagyunk, telepítsük fel az *mdadm* csomagot, ami a RAID eszközök létrehozásához és karbantartásához szükséges.

### Következő lépés

Tegyük fel, hogy van egy 80 gigabájtos merevlemezünk és ezt szeretnénk RAID1-be tudni. Jó, ha tudjuk, hogy a RAID1 tárolókapacitása mindig a legkisebb merevlemez méretével lesz egyenlő. Vagyis, jelen esetben egy másik 80 gigás merevlemezre lesz szükségünk. Amennyiben csak 120-ast találnánk, úgy

a merevlemez végén 40 gigabájt kárba vész – vagy két 120-ast veszünk.

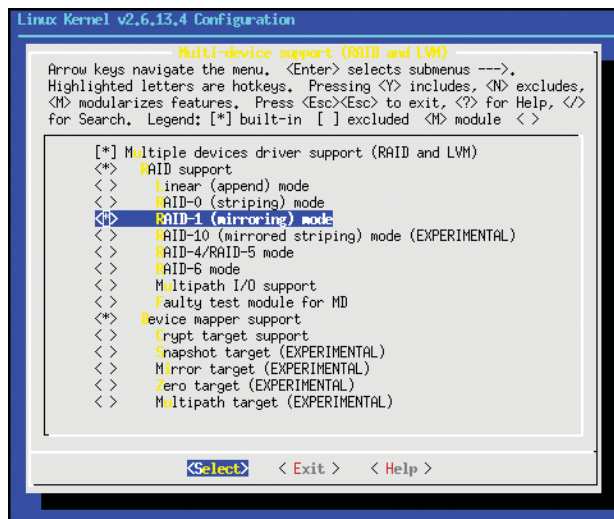
Na de haladjunk csak sorjában. Mentsük le a jelenlegi merevlemez partíciós tábláját:

```
sfdisk -d /dev/sda >
↳ particios.tabla
```

Majd írjuk ki a másik (jelenesetben üres) merevlemezre:

```
sfdisk /dev/sdb <
↳ particios.tabla
```

Ha ez megvan, módosítsuk a */dev/sdb* partíciós tábláján található partíciótipusokat *FD*-re (*Linux raid autodetect*), még a swap-et is.



1. lista

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/md6 / ext3 defaults,errors=remount-ro 0 1
/dev/md8 /home ext3 defaults 0 1
/dev/md9 none swap sw 0 0
```

**Barátunk, az mdadm**

Hozzuk létre sorban az md eszközöket:

```
mdadm -C /dev/md6 -l 1 -n 2
↳ /dev/sdb6 missing
```

Ezzel létrehoztunk egy egylemezes RAID1 tömböt. Látni fogjuk, miért így kell eljárni. Hozzuk létre a fájlrendszert:

```
mkfs.ext3 /dev/md6
```

Ezután másoljuk át a /dev/sda6 fájlrendszer tartalmát a /dev/md6 fájlrendszerbe. Ehhez az szükséges, hogy mindkettő fel legyen mountolva. Miután ezzel megvagyunk, hozzuk létre a /etc/raidtab-ot a /dev/md6-en, amiben az alábbi bejegyzések sorakoznak (ez csak egy partíciót ír le), igény szerint kell többszörözni:

```
raiddev /dev/md6
raid-level 1
nr-raid-disks 2
nr-spare-disks 0
chunk-size 4
persistent-superblock 1
device /dev/sda6
failed-disk 0
device /dev/sdb6
raid-disk 1
```

A /dev/sda6 alatti failed-disk bejegyzéstől ne ijedjünk meg, később majd ha minden rendben, eltávolítjuk. A RAID tömbön létre kell hozni a megfelelő /etc/fstab fájlt. Nálam így néz ki: 1. lista.

**(re)Boot**

Ezek után a GRUB-ot vagy a LILO-t konfiguráljuk. Ha kész, bootoljuk újra a rendszert a RAID-ról (jelenesetben /dev/sdb). Ekkor a mount parancs kiadása valami ilyesmit eredményez, ha valóban működik a RAID:

```
/dev/md6 on / type ext3
↳ (rw,errors=remount-ro)
proc on /proc type proc (rw)
/dev/md8 on /home type ext3
↳ (rw)
```

A sikeres bootolás után a forrás merevlemezen (/dev/sda) módosítuk a partíciók típusát a már ismert Linux raid autodetect típusra. Ha kész, akkor /etc/raidtab fájlban írjuk át a failed-disk sorokat raid-disk-re. Adjuk hozzá a forrás partíciókat a meglévő RAID-hez:

```
mdadm -a /dev/md6 /dev/sda6
```

Ha minden rendben akkor ezt írja:

```
mdadm: hot added /dev/sda6
```

Majd elkezdődik a szinkronizálás (cat /proc/mdstat):

```
md6 : active raid1 sda6[2]
↳ sdb6[0]
11719296 blocks [2/1]
↳ [U_]
[>.....]
↳ recovery = 3.0%
↳ (361216/11719296)
↳ finish=3.6min
↳ speed=51602K/sec
```

Innen megtudhatjuk, hogy hol tart a szinkronizálás. A szinkronizálás elég sokáig eltarthat nagyobb partíciók esetén. Ilyenkor érdemes elmenni ebédelni.

No igen, amiért a RAID-en újra létre kellett hozni a fájlrendszert. A dd nem működik jelen esetben, mert a RAID valamiért kb. Pár kilobájtal kisebb területet ad és hibát jelez a mount, ha esetleg dd-vel és nem szinkronizálással csináltuk volna a partíciók másolását.

**Jó tanácsok**

Akár RAID0-t, akár RAID1-et csinálunk, fontos, hogy ne ugyanazon az IDE kábel/csatornán legyen az eszköz. Ez felesleges lassulást eredményez. Az elérhető sebességről: nálam 2 darab 80 gigabájtos SATA merevlemez megy (52 és 62 megabájt per másodperc), a RAID1 rendszerben ez körülbelül 40-50 megabájt per másodpercre lassul. A sebesség tesztelésére a hdparm parancsot javaslom a -t opciót használva.

**Karbantartás**

A RAID tömbjeink állapotát a /proc/mdstat állományban figyelhetjük. Bármilyen hibajelenség van, arról a rendszer azonnal küld e-mail-t. Ilyenkor érdemes mihamarabb cserélni a hibás merevlemez. Amennyiben merevlemezcsere szükséges, úgy az alábbi dolgokat kell végig csinálni: partíciós tábla létrehozása az új merevlemezen (másoljuk le a régiről az sfdisk paranccsal), adjuk hozzá az újat az mdadm paranccsal, a többit majd elvégzi a szinkronizálás.



**Medve Zoltán**  
(e-medve@e-medve.hu)

2001-ben kezdett „Linuxolni”, de már korábban is ismerkedett a szabad szoftverek világával. Ha éppen nem a gép előtt ül, akkor fotóztat, olvasgat vagy bicajozik.

**KAPCSOLÓDÓ CÍMEK**

RAID a Wikipediában:  
↳ <http://en.wikipedia.org/wiki/RAID>