

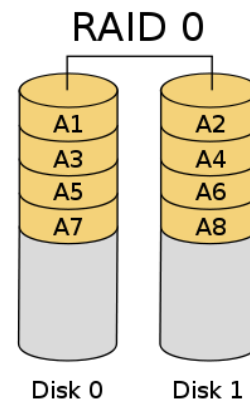


## RAID szintek

A RAID-ben eredetileg 5 szintet definiáltak (RAID 1-től RAID 5-ig). Az egyes szintek nem a fejlődési, illetve minőségi sorrendet tükrözik, hanem egyszerűen a különböző megoldásokat. A kezdeti 5 szinthez később hozzávették a RAID 6-ot. RAID 0-ként szokták említeni azt a változatot, ahol a lemezeket összefűzzük, azaz redundancia nélkül kapcsoljuk össze. Ezeken kívül használják még több RAID tömb egymásra építését is, a legelterjedtebb a RAID 10 (vagy RAID 1+0), RAID 01 (vagy RAID 0+1), illetve a RAID 50 (vagy RAID 5+0). A RAID alapötlete a lemezegységek csíkokra (stripes) bontása. Ezek a csíkok azonban nem azonosak a lemez fizikai sávjaival (tracks), amit az angol és magyar elnevezés különbözősége is jelez.

### RAID 0 (összefűzés)

A RAID 0 az egyes lemezek egyszerű összefűzését jelenti, viszont semmilyen redundanciát nem ad, így nem biztosít hibatűrést, azaz egyetlen meghajtó meghibásodása az egész tömb hibáját okozza. Mind az írási, mind az olvasási műveletek párhuzamosítva történnek, ideális esetben a sebesség az egyes lemezek sebességének összege lesz, így a módszer a RAID szintek közül a legjobb teljesítményt nyújtja (a többi módszernél a redundancia kezelése lassítja a rendszert). A megoldás lehetővé teszi különböző kapacitású lemezek összekapcsolását is, viszont a nagyobb kapacitású lemezekben is csak a tömb legkisebb kapacitású lemezének méretét lehet.

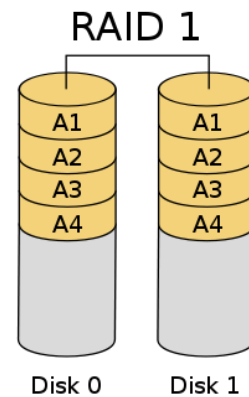


A RAID 0 főleg olyan helyeken alkalmazható, ahol nem szempont az adatbiztonság vagy kevés merevlemez csatolható fel az operációs rendszer korlátozása miatt. A másik pozitív tulajdonsága viszont továbbra is csábító lehet olyan, kifejezetten csak játékokra épített rendszereknél, ahol ezzel tetemes teljesítménynövekedést érhetünk el. Ilyen célú alkalmazásra mégsem ajánlott, mivel az egyszer már összekapcsolt diszkek különálló alkalmazásra csak újraszervezés után, a teljes adattartalom eltávolításával és újraformázással lehetséges.



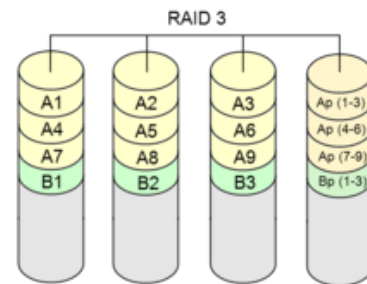
## RAID 1 (tükrözés)

A RAID 1 eljárás alapja az adatok tükrözése (disk mirroring), azaz az információk egyidejű tárolása a tömb minden elemén. A kapott logikai lemez a tömb legkisebb elemével lesz egyenlő méretű. Az adatok olvasása párhuzamosan történik a diszkekről, felgyorsítván az olvasás sebességét; az írás normál sebességgel, párhuzamosan történik a meghajtókon. Az eljárás igen jó hibavédelmet biztosít, bármely meghajtó meghibásodása esetén folytatódhat a működés. A RAID 1 önmagában nem használja a csíkokra bontás módszerét.



## RAID 3

Egy adott paritáscsík a különböző lemezeken azonos pozícióban elhelyezkedő csíkokból XOR művelet segítségével kapható meg. A rendszerben egy meghajtó kiesése nem okoz problémát, mivel a rajta lévő információ a többi meghajtó (a paritást tároló meghajtót is beleértve) XOR-aként megkapható. Az alapvető különbség a RAID 2-ben alkalmazott hibajavító kóddal szemben, hogy itt feltesszük, hogy a meghajtó meghibásodását valamilyen módon (például többszöri sikertelen olvasás hatására) észleljük, majd a meghibásodott diszken lévő információt a többi diszken lévő adatok segítségével állítjuk elő.



A RAID 3-nál kisméretű csíkokat definiálnak, így az egyes fájlok olvasása és írása párhuzamosan történhet az egyes meghajtókon, viszont a módszer nem támogatja egyszerre több kérés párhuzamos kiszolgálását (single-user mode). (Természetesen a paritáscsíkot minden egyes íráskor módosítani kell, amihez szükséges a korábbi tartalom kiolvasása. Viszont például fájltranszfer esetén, pont a kisméretű csíkok miatt, az azonos pozícióban lévő csíkokat általában az összes diszken felülírják, így ez esetben a probléma kevésbé jelentkezik.)

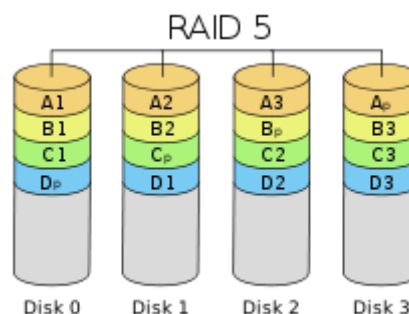


## RAID 4

A RAID 4 felépítése a RAID 3-mal megegyezik. Az egyetlen különbség, hogy itt nagyméretű csíkokat definiálnak, így egy rekord egy meghajtón helyezkedik el, lehetővé téve egyszerre több (különböző meghajtókon elhelyezkedő) rekord párhuzamos írását, illetve olvasását (multi-user mode). Problémát okoz viszont, hogy a paritás-meghajtó adott csíkját minden egyes íráskor frissíteni kell (plusz egy olvasás és írás), aminek következtében párhuzamos íráskor a paritás-meghajtó a rendszer szűk keresztmetszetévé válik. Ezenkívül valamely meghajtó kiesése esetén a rendszer olvasási teljesítménye is lecsökken, a paritás-meghajtó jelentette szűk keresztmetszet miatt.

## RAID 5

A RAID 5 a paritás információt nem egy kitüntetett meghajtón, hanem „körbeforgó paritás” (rotating parity) használatával, egyenletesen az összes meghajtón elosztva tárolja, kiküszöbölve a paritás-meghajtó jelentette szűk keresztmetszetet. Minimális meghajtószám: 3. Mind az írási, mind az olvasási műveletek párhuzamosan végezhetőek. Egy meghajtó meghibásodása esetén az adatok sértetlenül visszaolvashatóak, a hibás meghajtó adatait a vezérlő a többi meghajtóról ki tudja számolni. A csíkméret változtatható; kis méretű csíkok esetén a RAID 3-hoz hasonló működést, míg nagy méretű csíkok alkalmazása esetén a RAID 4-hez hasonló működést kapunk. A hibás meghajtót ajánlott azonnal cserélni, mert két meghajtó meghibásodása esetén az adatok elvesznek!



A tárolható adatmennyiség:

$x_{\min} * (y-1)$  ahol:

x: lemezméret

y: lemezek száma

Az írási sebességnél fontos figyelembe venni a paritás adatok előállítására szükséges számítási kapacitás igényt! Szoftveres megoldásnál ez jelentős processzorterhelést, illetve az írási sebesség csökkenését eredményezheti, ezért ajánlott a hardveres megoldás, ahol a célhardver látja el ezeket a feladatokat.

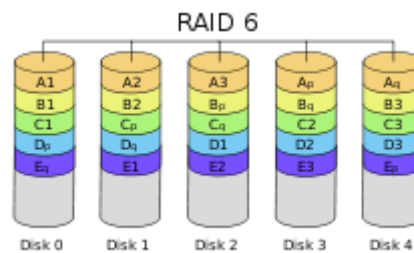
A RAID 5 vezérlők a hibás meghajtó helyére betett új, üres meghajtót automatikusan fel tudják tölteni az eredeti adatokkal.



*A tömb egyetlen meghajtójáról nem állítható vissza a teljes adattartalom, viszont egy-egy adatblokknyi igen. Mivel akár ez is tartalmazhat értékes információt, így a már nem használt vagy hibás adathordozót érdemes megsemmisíteni.*

### RAID 6

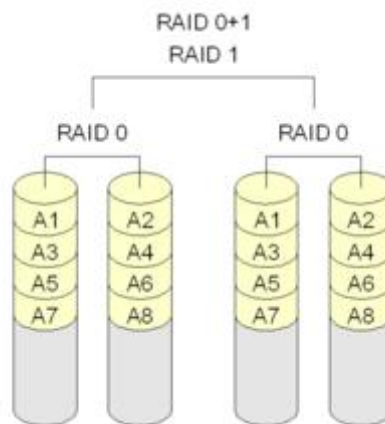
A RAID 6 tekinthető a RAID 5 kibővítésének. Itt nemcsak soronként, hanem oszloponként is kiszámítják a paritást. A módszer segítségével kétszeres meghajtó meghibásodás is kiküszöbölhetővé válik. A paritáscsíkokat itt is az egyes meghajtók között, egyenletesen elosztva tárolják, de ezek természetesen kétszer annyi helyet foglalnak el, mint a RAID 5 esetében.



### Emeletes RAID tömbök

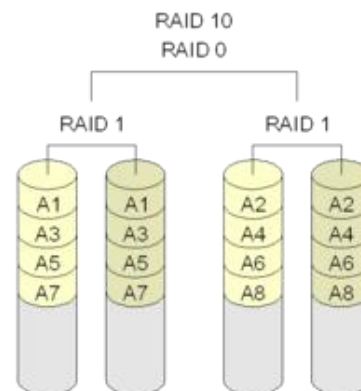
#### RAID Level 0+1

Ez egy olyan hibrid megoldás, amelyben a RAID 0 által hordozott sebességet a RAID 1-et jellemző biztonsággal ötvözhetjük. Hátránya, hogy minimálisan 4 eszközre van szükségünk, melyekből 1-1-et összefűzve, majd páronként tükrözve építhetjük fel a tömbünket, ezért a teljes kinyerhető kapacitásnak mindössze a felét tudjuk használni. Mivel a tükrözés (RAID 1) a két összefűzött (RAID 0) tömbre épül, ezért egy lemez meghibásodása esetén az egyik összefűzött tömb mindenképp kiesik, így a tükrözés is megszűnik.



#### RAID Level 1+0

Hasonlít a RAID 01 megoldáshoz, annyi különbséggel, hogy itt a lemezeket először tükrözzük, majd a kapott tömböket fűzzük össze. Ez biztonság szempontjából jobb megoldás, mint a RAID 01, mivel egy diszk kiesése csak az adott tükrözött tömböt érinti, a rá épült RAID 0-t nem; sebességben pedig megegyezik vele.





## JBOD

"Just a Box Of Disks", "Just a Bunch Of Drives"

Minden egyes fizikai lemezt külön-külön címez meg a rendszer, de az Operációs rendszer számára csak egy lemeznek tűnik az összes. Általában Hardveres RAID kártyáknál alkalmazzák.

## Hardveres vagy szoftveres?

A mai napig nem tudták egyértelműen kijelenteni melyik a jobb. Hardveres RAID esetén a fenti lemezkonfigurációkat egy erre kitalált hardver (ált. PCI, PCI-X, PCI-e) állítja össze és vezérli, míg szoftveres RAID esetén maga az Operációs rendszer (illetve annak egy modulja) látja el ezt a feladatot.

### Pro/Kontra:

Szoftver RAID	Hardver RAID
<b>Olcsó</b>	<b>Egy komoly RAID kártya 100e forint is lehet</b>
<b>RAID0, RAID1 esetén előnyös</b>	<b>Paritás számolás esetén nem dolgozik a CPU</b>
<b>Kisvállalati környezetbe ajánlott</b>	<b>Cluster rendszerek esetén javallott (adatbázis)</b>
<b>Nem kell egy gyártóhoz kötnünk magunkat</b>	<b>Nagyobb teljesítmény</b>
<b>különböző lemezméretekhez igazodás</b>	<b>kártya-meghibásodás</b>
<b>Monitorozás</b>	

### 1. feladat

A mérésvezető által bemutatott módon telepítsük fel a Debian Operációs rendszert RAID tömbök kezelésével.

*Néhány példa mdadm parancs használatára:*

```
mdadm --create -l1 -n2 -x1 /dev/md0 /dev/sd[abc]1
```

RAID 1 tömb létrehozása 1 spare (tartalék) diszkkal.

```
mdadm --create -l5 -n3 /dev/md0 /dev/sda1 missing /dev/sdb1
```

RAID 5 létrehozása 1 eszköz hiányzik.

```
mdadm --fail /dev/md0 /dev/sdb1
```

RAID kötet hibásnak jelölése

```
mdadm --remove /dev/md0 /dev/sdb1
```

Majd törlése