**5. tétel**

**Információtechnológiai alapok - Számítógépes architektúrák (1.1.1)**

Ismertesse a RAID technológiát! Fogalma, előnye, hátránya, alkalmazási területe. A felsorolt RAID szintek közül válasszon ki egyet, majd részletesen ismertesse azt! (RAID 0, 1, 5, 0+1, 1+0)

***Szempontok a tartalom rész értékeléséhez***

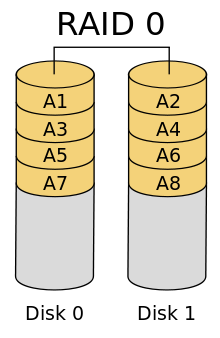
o A RAID technológia jellemzője, jelentősége

o Előnye és hátránya

o A választott RAID szint ismertetése ábrával

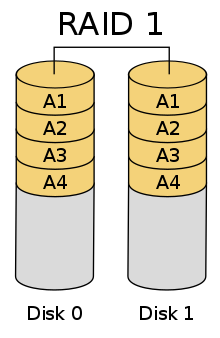
A **RAID** (angolul **Redundant Array of Inexpensive Disks** vagy **Redundant Array of Independent Disks**) tárolási technológia, mely segítségével az adatok elosztása vagy replikálása több fizikailag független merevlemezen, egy logikai lemez létrehozásával lehetséges. Minden RAID szint alapjában véve vagy az adatbiztonság növelését vagy az adatátviteli sebesség növelését szolgálja.

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 jelöli 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 vagy csíkozás)**[](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_0.svg)

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 lemezeken is csak a tömb legkisebb kapacitású lemezének méretét lehet használni (tehát egy 1 TB és egy 750 GB méretű lemez összefűzésekor mindössze egy 1,5 TB-os logikai meghajtót fogunk kapni, az 1 TB-os lemezen 250 GB szabad terület marad, amit más célokra természetesen felhasználhatunk).

A RAID 0 főleg olyan helyeken alkalmazható, ahol nem szempont az adatbiztonság vagy kevés merevlemez csatolható fel az operációs rendszer korlátozása miatt.

[](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_1.svg)**RAID 1 (tükrözés)**

RAID Level 1

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.

### [https://upload.wikimedia.org/wikipedia/commons/thumb/b/b5/RAID2_arch.svg/350px-RAID2_arch.svg.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID2_arch.svg)RAID 2

A RAID 2 használja a csíkokra bontás módszerét, emellett egyes meghajtókat hibajavító kód (ECC: Error Correcting Code) tárolására tartanak fenn. A hibajavító kód lényege, hogy az adatbitekből valamilyen matematikai művelet segítségével redundáns biteket képeznek. A használt eljárástól függően a kapott kód akár több bithiba észlelésére, illetve javítására (ez utóbbi persze több redundanciát igényel) alkalmas. A védelem ára a megnövekedett adatmennyiség. Ezen meghajtók egy-egy csíkjában a különböző lemezeken azonos pozícióban elhelyezkedő csíkokból képzett hibajavító kódot tárolnak. A módszer esetleges lemezhiba esetén képes annak detektálására, illetve kijavítására. Manapság nem használják, mivel a SCSI meghajtókban már minden egyes szektorban az adott szektorhoz tartozó ECC is eltárolódik.

### RAID 3[https://upload.wikimedia.org/wikipedia/commons/thumb/e/e7/RAID3.png/220px-RAID3.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID3.png)

A RAID 3 felépítése hasonlít a RAID 2-re, viszont nem a teljes hibajavító kód, hanem csak egy lemeznyi paritásinformáció tárolódik. 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 2 a diszkhibák ellen is védelmet nyújt, például egyes bájtok megsérülése esetén. (Vegyük észre, hogy csak az XOR-os paritásbit technikát használva az egyik meghajtón egy adott bájt megsérülése esetén csak azt vennénk észre, hogy a különböző meghajtókon az azonos csíkba tartozó részek XOR-a nem nullát adna, de nem tudnánk sem azt, hogy melyik meghajtón van a hiba, sem azt, hogy hogyan javítsuk ki. Ezért van szükség a szektoronkénti hibajavító kód alkalmazására.)

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.)

### [https://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/RAID_4.svg/220px-RAID_4.svg.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_4.svg)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 paritásmeghajtó 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[https://upload.wikimedia.org/wikipedia/commons/thumb/6/64/RAID_5.svg/220px-RAID_5.svg.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_5.svg)

A RAID 5 a paritás információt nem egy kitüntetett meghajtón, hanem „körbeforgó paritás” (rotating parity) használatával, egyenletesen az összes meghajtón elosztva tárolja, kiküszöbölvén a paritás-meghajtó jelentette szűk keresztmetszetet. Minimális meghajtó szám: 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; kisméretű csíkok esetén a RAID 3-hoz hasonló működést, míg nagyméretű 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 "a legkisebb kapacitású meghajtón tárolható adatmennyiség" \* ("meghajtók száma" - 1) lesz. (Pl. 4 db egyenként 1 TB -os HDD RAID 5-be fűzésének eredményeként egy 3 TB kapacitású logikai meghajtót látunk.)

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 hibás meghajtó egy-egy blokkját a következőképpen lehet visszaolvasni: Ah=(Aj1 XOR Aj2) XOR Aj3, ahol Ah: a fizikailag hibás meghajtó része és Aj1, Aj2, Aj3: a jó meghajtó része.

Az "olvasási sebesség"="A leglassabb meghajtó olvasási sebessége"\*("Meghajtók száma" - 1)

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

### [https://upload.wikimedia.org/wikipedia/commons/thumb/7/70/RAID_6.svg/220px-RAID_6.svg.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_6.svg)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.

### [https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/RAID_0%2B1.png/220px-RAID_0%2B1.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_0+1.png)RAID 0+1 vagy RAID 01

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. Fontos megjegyeznünk, hogy ez a megoldás legalább négy darab merevlemezt igényel.

### [https://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/RAID_10.svg/220px-RAID_10.svg.png](https://hu.wikipedia.org/wiki/F%C3%A1jl:RAID_10.svg)RAID 1+0 vagy RAID 10

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.

## **Használat**

***Az ismertetett RAID technikák közül a RAID 0-t, RAID 1-et és RAID 5-öt használják elterjedten, illetve az ezek kombinációjaként megvalósított RAID 10-et és RAID 50-et.*** A RAID 2-t a már korábban említett okok, míg a RAID 3-at és RAID 4-et a paritás meghajtó jelentette szűk keresztmetszet miatt nem alkalmazzák. A RAID 6 megvalósítása túl sok többletköltséget jelent, illetve az általa nyújtott többlet biztonság csak speciális alkalmazások esetében szükséges, ezért használata nem terjedt el.

## **Megvalósítás**

A RAID rendszerek megvalósítása történhet szoftver, illetve hardver támogatással. Szoftveres megvalósítás esetén vagy az operációs rendszer nyújt támogatást, vagy speciális driver programot használnak. A megvalósítás előnye, hogy nem igényel külön költséges hardver komponenseket, viszont hátránya, hogy a központi memóriát, illetve a CPU-t terheli, lerontván az egész rendszer teljesítményét. Hardveres megvalósítás esetén a szükséges feldolgozást a RAID vezérlő valósítja meg, ezáltal az operációs rendszer teljesítménye nem csökken le. A kezelés bonyolultsága miatt a RAID 5-öt a gyakorlatban csak hardver támogatással valósítják meg.

A Linux kernel támogatja a raid5 és raid6 szoftveres megvalósítását is.

A RAID tömbök mindig nagyobb valószínűséggel hibásodnak meg, mintha az azt alkotó lemezek valamelyikét használnánk, azonban míg az szóló lemezek meghibásodása a cég üzemmenetét megakasztja addig (RAID 0 kivételével) a cég tovább üzemelhet a hiba kijavításáig.