RAID:

RAID (Redundant Array of Independent Disks) in Linux is a technology that combines multiple physical drives into a single logical unit to achieve one or more of these goals:

- **Improve performance (I/O throughput)**
- **Provide redundancy (data protection)**
- **Simplify capacity management**

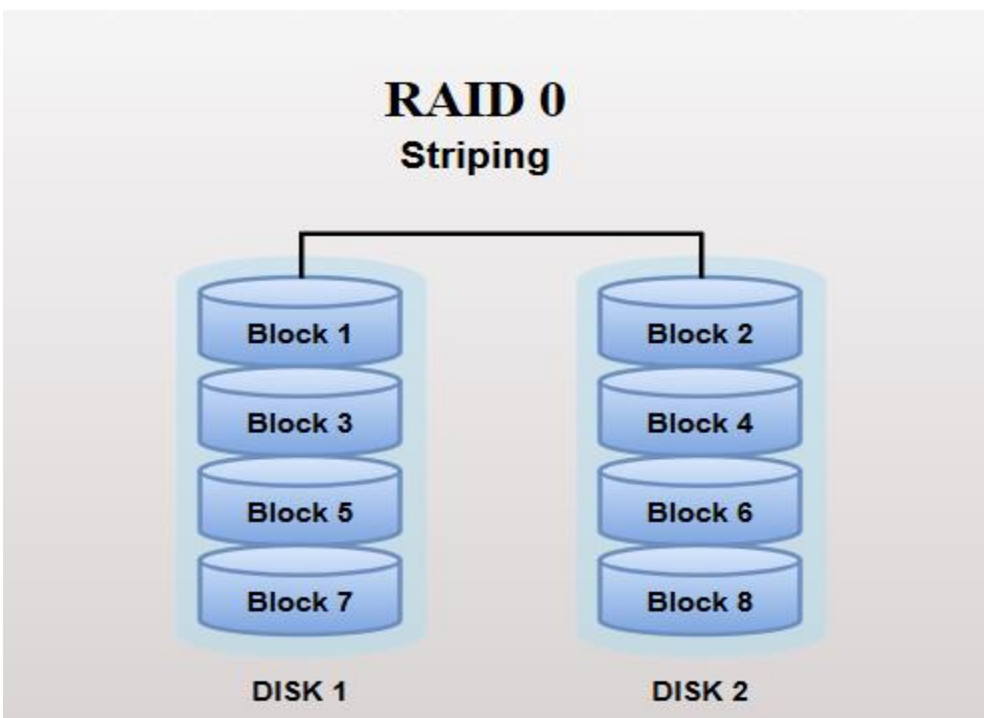
Why Use RAID?

Feature	Benefit
Redundancy	Prevents data loss during disk failure
Performance	Faster data read/write access via striping or parallel reads
Scalability	Easier expansion and management of disk space

Standard RAID Levels:

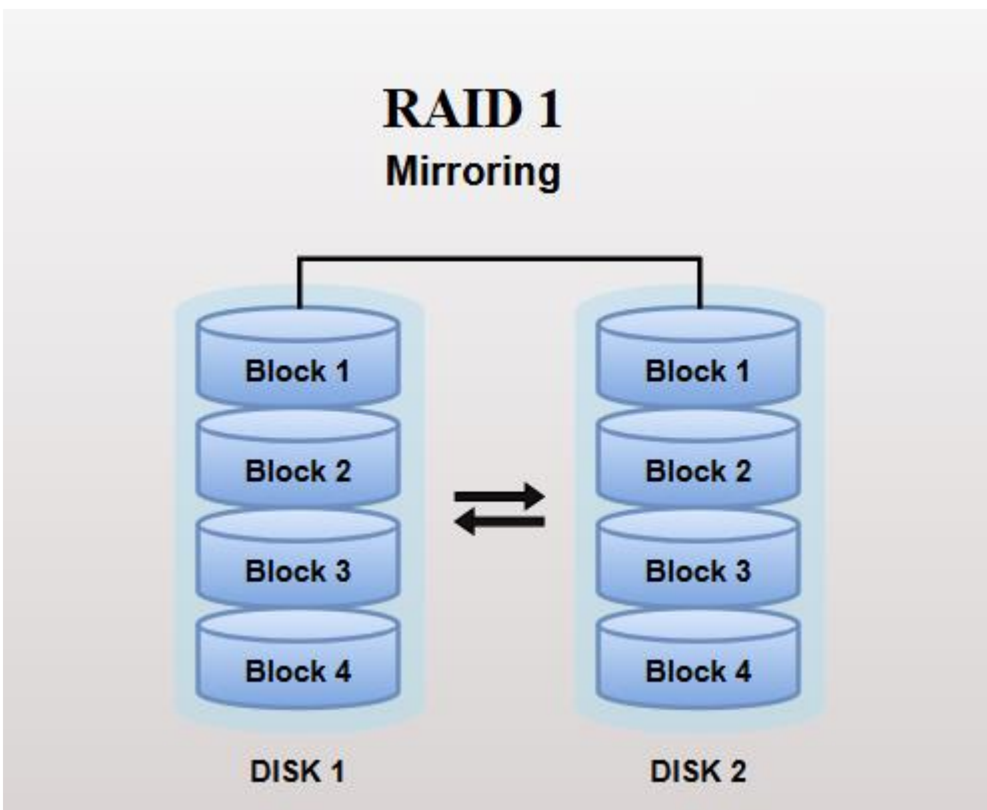
RAID 0 – Striping

- **Min disks:** 2
- **Pros:**
  - Full capacity (100%)
  - Excellent read/write performance
- **Cons:**
  - No redundancy — one disk failure = total data loss
- **Use case:** High-speed tasks like video editing or scratch space.
- **Diagram:** Data blocks A, B, C… are striped evenly across all drives.



RAID 1 – Mirroring

- **Min disks:** 2
- **Pros:**
  - Survives one disk failure
  - Read speeds boost via parallel reads
- **Cons:**
  - Only 50% usable capacity
  - Write performance limited to single disk speed
- **Use case:** Boot volumes, critical OS/data mirroring.
- **Diagram:** Each drive has an exact copy of data (A mirrored as A').

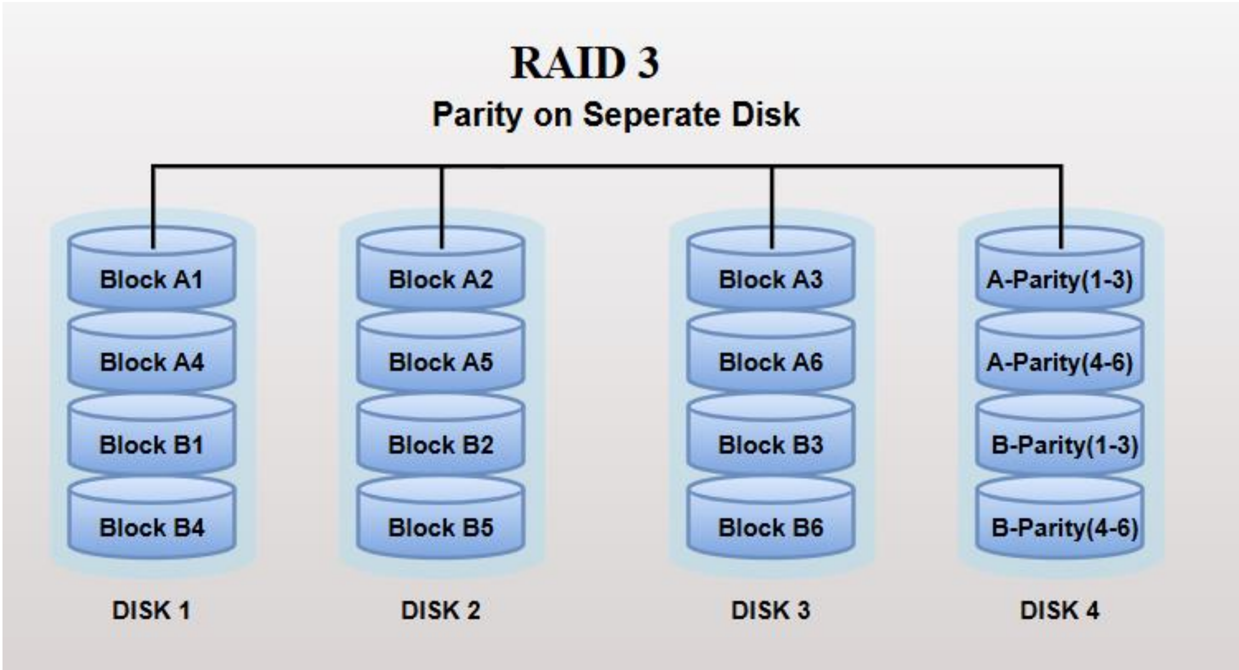


RAID 2 – Bit-Level Striping with Hamming Code

- **Min disks**: ≥ 3 (data + parity disks)
- **Diagram**: Each bit of a byte is written to its own disk; an extra disk holds Hamming ECC bits.
- **How it works**: Like RAID 0 but at bit granularity, with Hamming code redundancy.
- **Pros**: Very high sequential throughput; built-in error correction at bit level.
- **Cons**: Requires synchronized spindles, complex, inefficient, and redundant—modern drives have ECC; largely obsolete

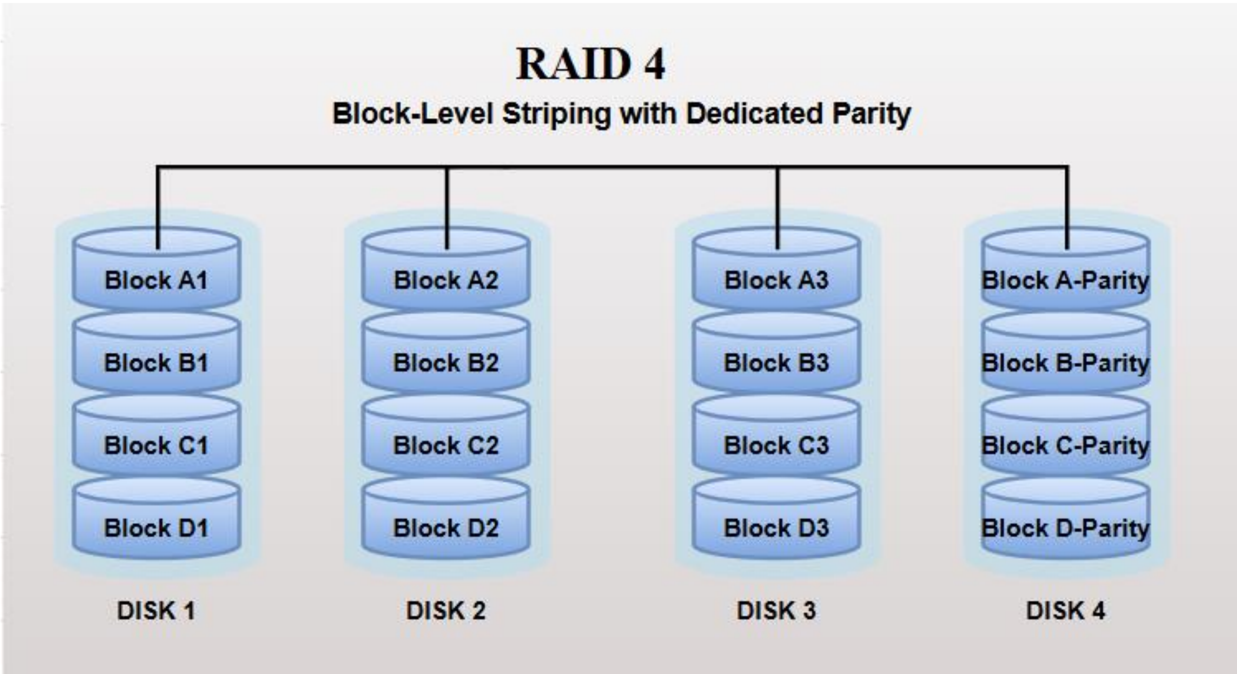
RAID 3 – Byte-Level Striping with Dedicated Parity

- **Min disks:** ≥ 3 (one disk solely for parity)
- **How it works:** A parity byte is written for each stripe, enabling recovery from a single disk failure.
- **Pros:** High-speed sequential reads/writes.
- **Cons:** Single parity disk creates write bottleneck; synchronous access only; unsuited to random I/O; largely superseded by RAID 5 .
- **Diagram:** Bytes (or words) striped across data disks; a separate disk holds parity.



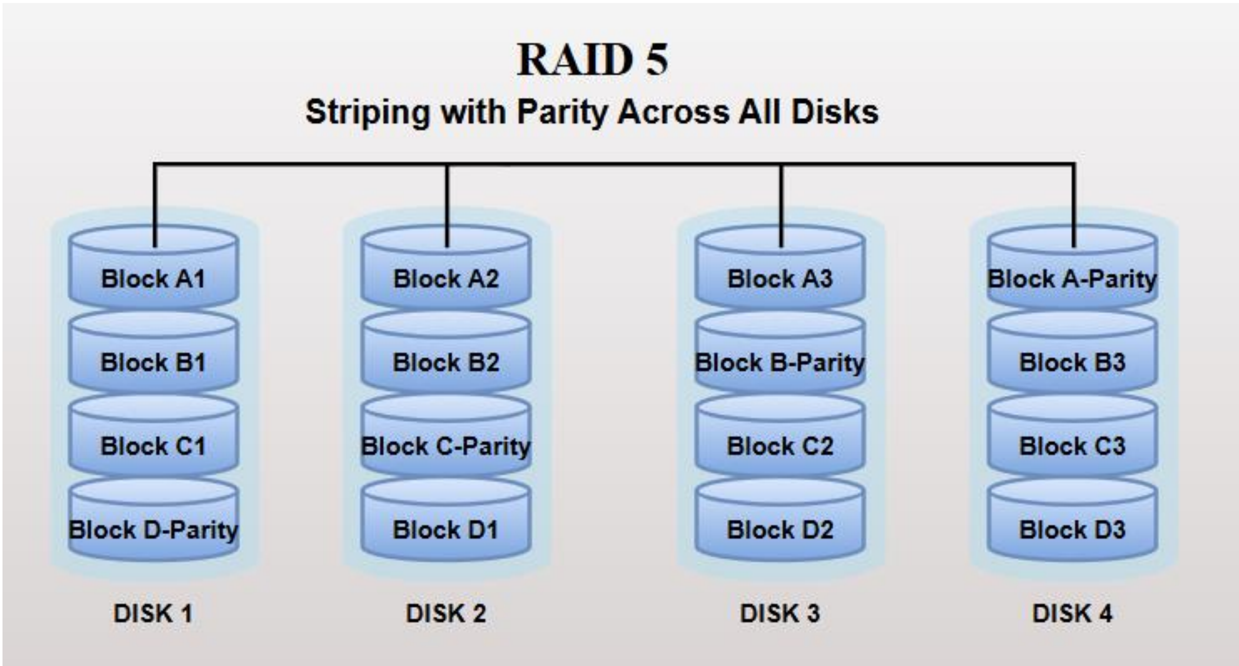
RAID 4 – Block-Level Striping with Dedicated Parity

- **Min disks:** 3
- **Pros:**
  - Maintains redundancy for one failed disk
- **Cons:**
  - Parity disk bottleneck slows writes
- **Note:** Rarely used in Linux—RAID 5 is preferred.
- **Diagram:** Data is striped across disks; a single disk holds parity (P).



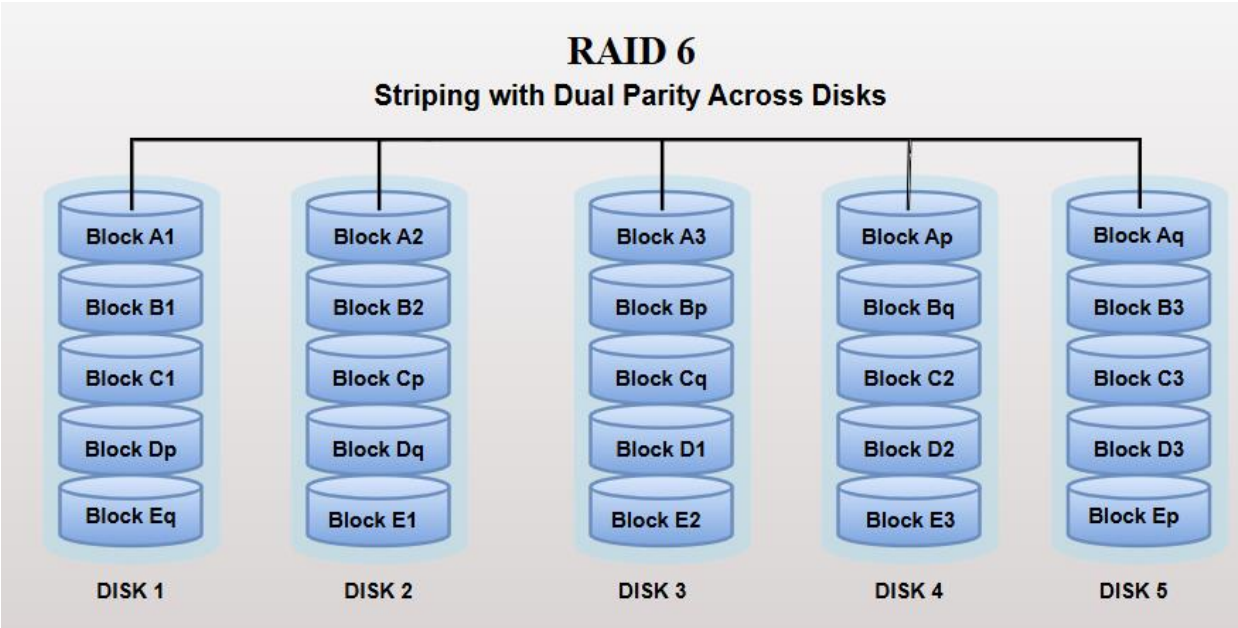
RAID 5 – Block-Level Striping with Distributed Parity

- **Min disks**: 3
- **Pros:**
  - Balanced capacity (≈ (n–1)/n)
  - Good read performance
  - Can survive one disk failure
- **Cons:**
  - Slower writes due to parity overhead
  - If second disk fails during rebuild, data loss occurs
- **Use case:** File servers, shared storage.
- **Diagram:** Data and parity blocks alternate across all disks.



RAID 6 – Block-Level Striping with Dual Parity

- **Min disks:** 4
- **Pros:**
  - Survives up to 2 disk failures
  - Read speed ≈ RAID 5
- **Cons:**
  - Extra parity slows writes more than RAID 5
- **Use case**: Large arrays where dual failure risk is significant.
- **Diagram:** Two parity blocks (P and Q) per stripe, distributed across disks.



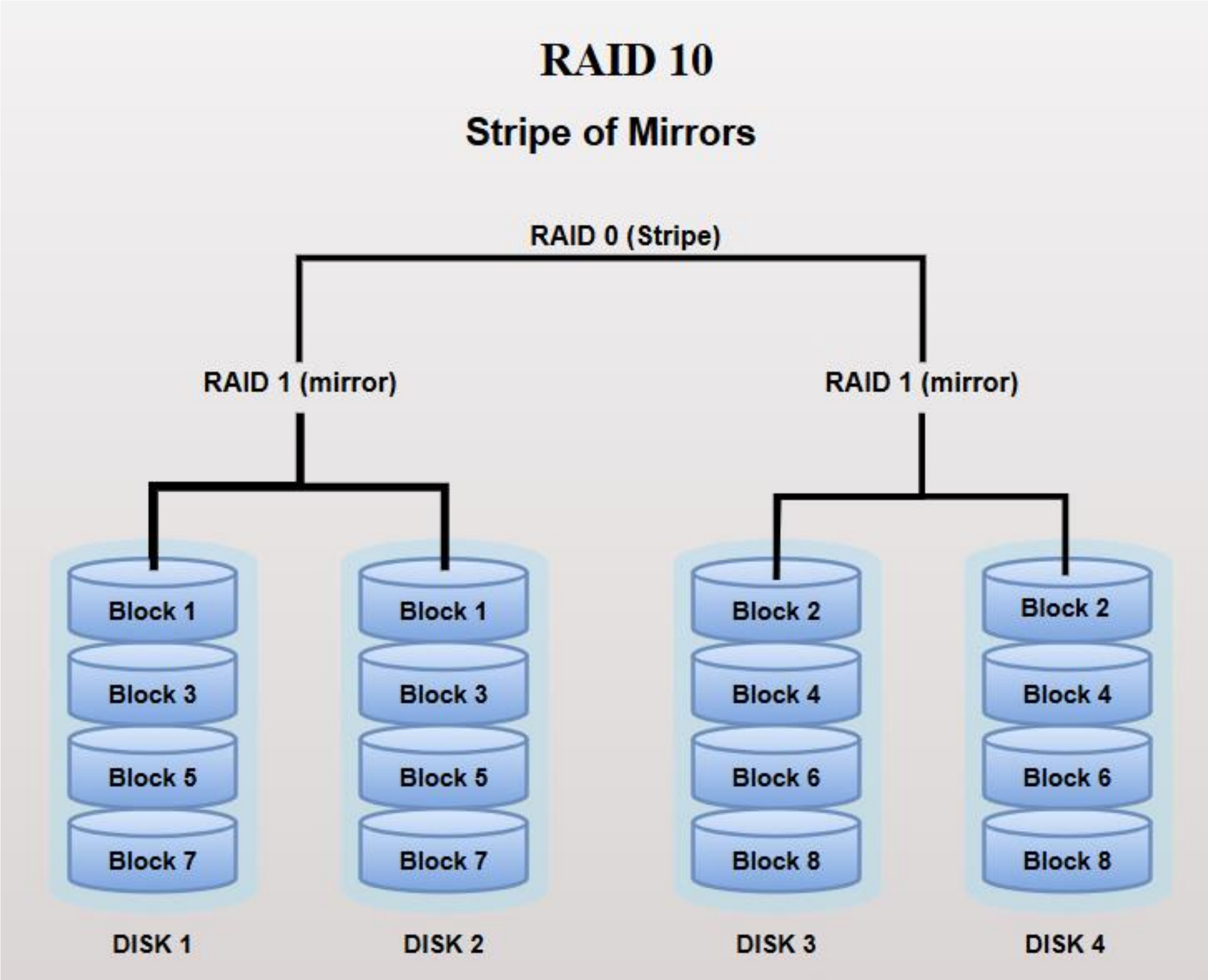
Nested RAID Levels (Hybrid RAID)

Nested RAID levels (also called **hybrid RAID levels**) are combinations of two standard RAID levels to achieve a balance of **performance, redundancy, and fault tolerance**. These configurations are commonly used in enterprise environments for mission-critical data. Below is a summary of the main nested RAID levels

RAID 10 (RAID 1 + RAID 0) – Stripe of Mirrors

- **Min disks:** 4
- **Pros:**
    - High performance and redundancy
    - Can tolerate multiple failures (unless both disks in a mirror fail)
- **Cons:**

    - 50% capacity
    - Higher cost per usable GB
- **Use case:** Databases, virtual machines, I/O-heavy workloads**.**
- **Diagram:** Mirror pairs are striped across (RAID 0 over RAID 1).

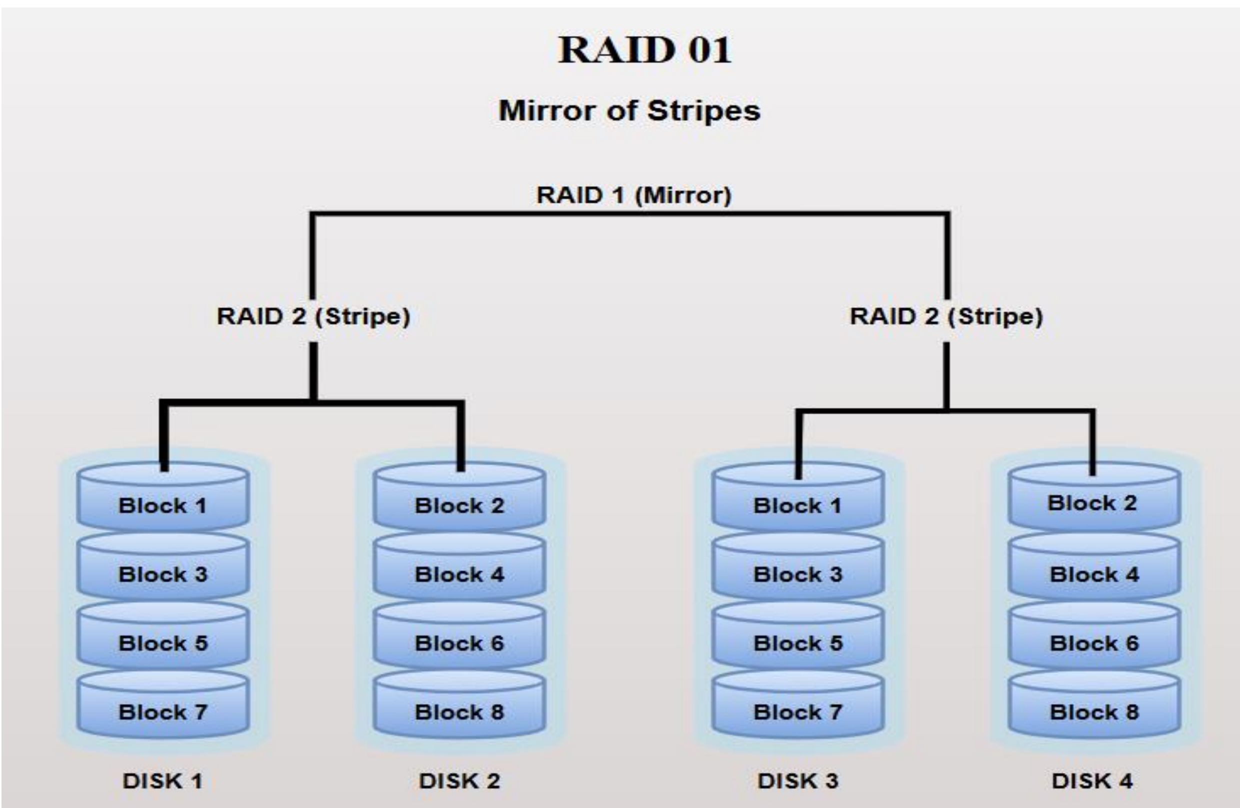


RAID 01 (RAID 0 + RAID 1) – Mirror of Stripes

- **Min disks:** 4
- **Diagram:** Striped disks are mirrored (RAID 1 over RAID 0).
- **Pros:**

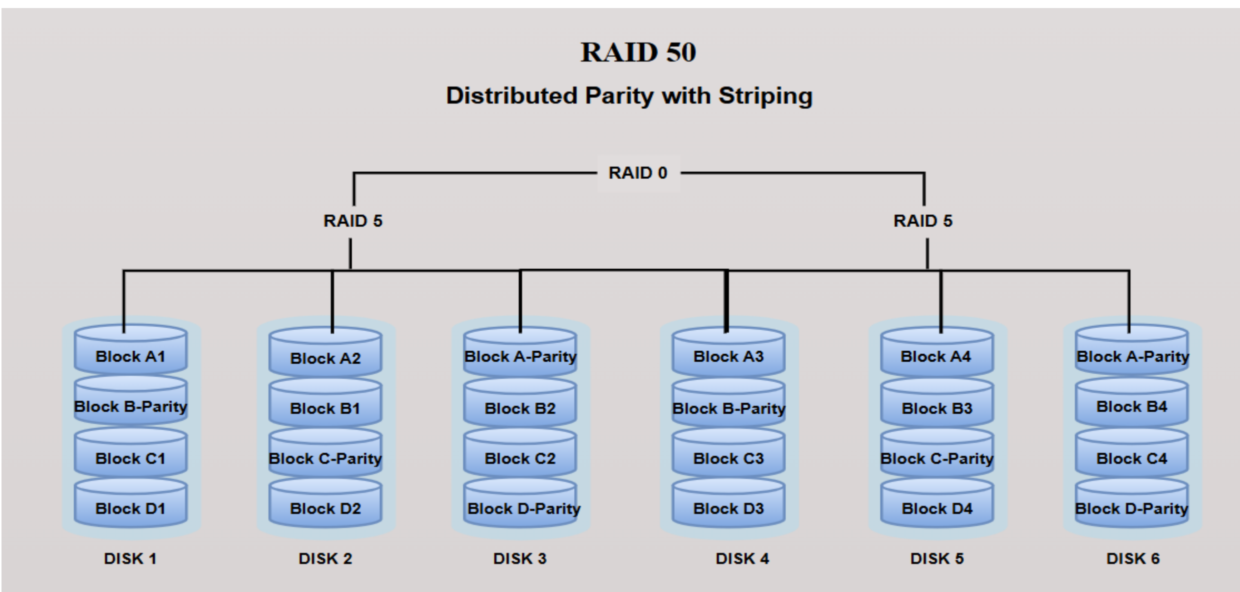
- High performance
- Redundancy available
- Simple to implement

- **Cons:**
  - Lower fault tolerance than RAID 10 (if one disk in a striped set fails, entire stripe may be lost)
  - 50% usable capacity
  - Higher cost per usable GB
- **Use Case:**
  - Environments where performance is more important than maximum fault tolerance
  - Non-critical applications with high read/write loads



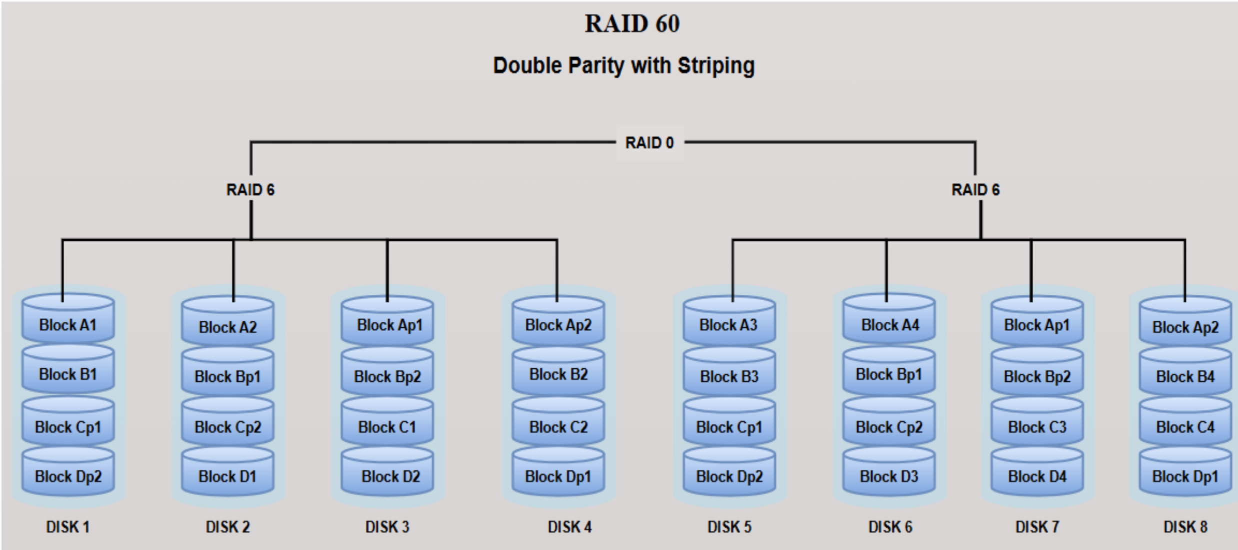
RAID 50 (RAID 5 + RAID 0) – Stripe of Parity Arrays

- **Min disks:** 6
- **Pros:**
  - Better performance and redundancy than standalone RAID 5
  - More usable capacity than RAID 10
  - Can tolerate one disk failure per RAID 5 set
  - Good balance between speed, storage, and fault tolerance
- **Cons:**
  - Complex setup
  - If two disks fail in the same RAID 5 set, data is lost
  - Slower rebuilds than RAID 10
- **Use Case:**
  - Large file or application servers
  - Data warehouses where performance and capacity are important
- **Diagram:** Multiple RAID 5 sets are striped (RAID 0 over RAID 5).



RAID 60 (RAID 6 + RAID 0) – Stripe of Dual-Parity Arrays

- **Min disks:** 8
- **Pros:**
  - Can survive two disk failures per RAID 6 set
  - More fault-tolerant than RAID 50
  - Balanced performance and high redundancy
  - Better protection for large arrays
- **Cons:**
  - Complex setup and maintenance
  - Slower write performance due to double parity
  - Longer rebuild times
- **Use Case:**
  - Critical backup servers
  - Archival and long-term storage
  - Systems where maximum fault tolerance is essential
- **Diagram:** Multiple RAID 6 sets are striped (RAID 0 over RAID 6).



Choosing the Right RAID

- Use **RAID 0** for speed when data isn't critical.
- Choose **RAID 1** for simple redundancy.
- Go with **RAID 5** for capacity + fault tolerance with minimal space loss.
- Pick **RAID 6** in large arrays where dual-drive failure is possible.
- Opt for **RAID 10** when you need top performance *and* reliability.