

CSE 421/521 - Operating Systems
Fall 2012

LECTURE - XXI

MASS STORAGE & IO - II

Tevfik Koşar

University at Buffalo
November 13th, 2012

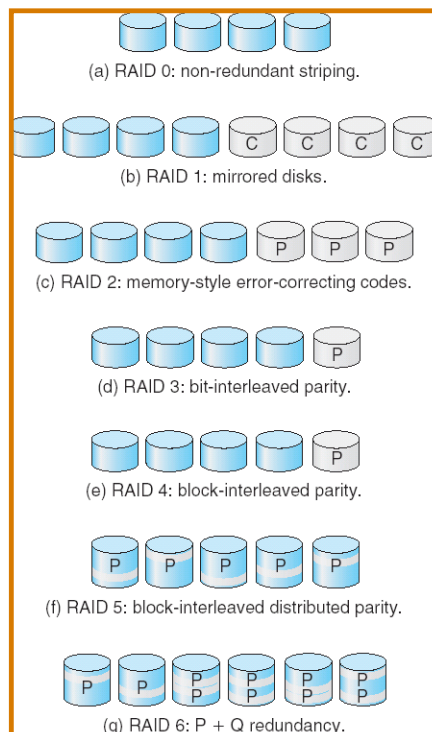
RAID Structure

- As disks get cheaper, adding multiple disks to the same system provides increased **storage space**, as well as increased **reliability** and **performance**.
- **RAID: Redundant Array of Inexpensive Disks**
 - multiple disk drives provides **reliability** via **redundancy**.
- RAID is arranged into six different levels.

RAID (cont)

- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.
 - **Data Striping**: splitting each bit (or block) of a file across multiple disks.
 - **Mirroring (shadowing)**: *duplicate each disk*
 - *Simplest but most expensive approach*
 - **Block interleaved parity** uses much less redundancy.

RAID Levels

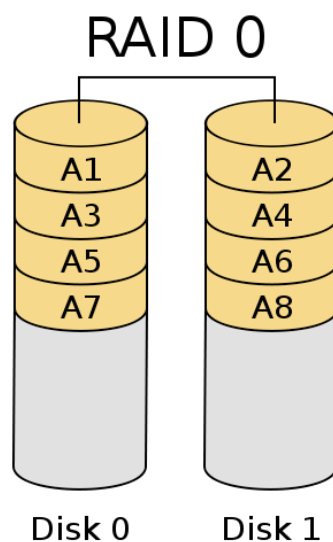


RAID Level 0

- Data is divided into blocks and is spread in a fixed order among all the disks in the array
- does not provide any fault tolerance
- also known as disk striping
- improves read and write performance via parallel access

5

RAID Level 0 - block level



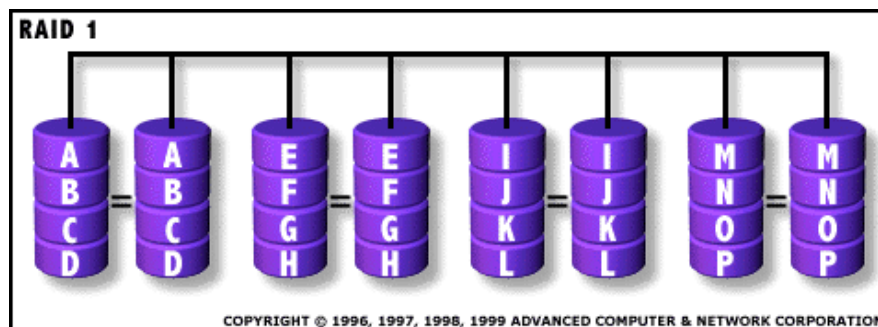
6

RAID Level 1

- All data written to the primary disk is written to the mirror disk
- provides a redundant, identical copy of all data
- provides fault tolerance
- also known as disk mirroring
- also generally improves read performance (but may degrade write performance).

7

RAID Level 1 - block level



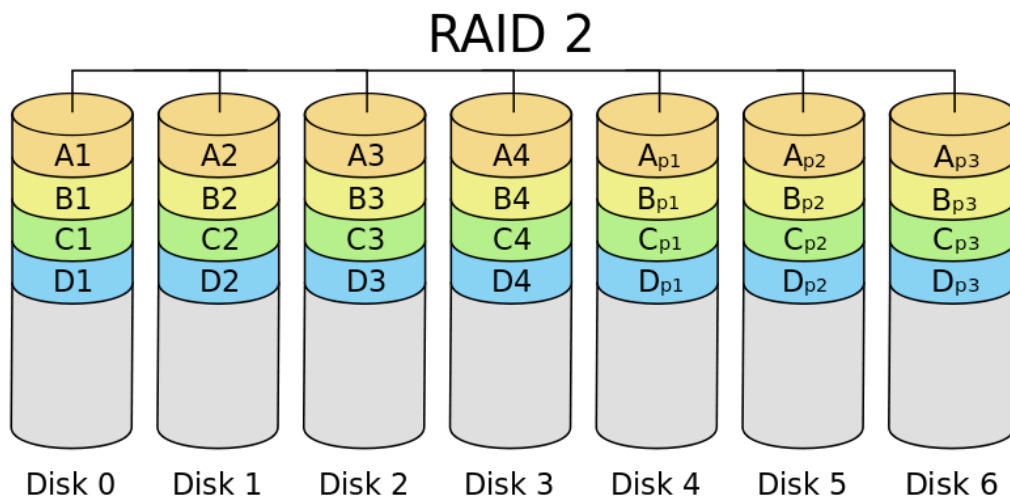
8

RAID Level 2

- uses memory-style error correcting code (ECC) that employs disk-striping strategy that **breaks a file into bits** and spreads it across multiple disks
- The error-correction method requires three extra disks for four data disks
- provides fault tolerance (Hamming Code [7,3])
 - can both detect & recover from single bit failures
 - can detect but not correct double bit failures
- but is not as efficient as other RAID levels

9

RAID Level 2 - bit level



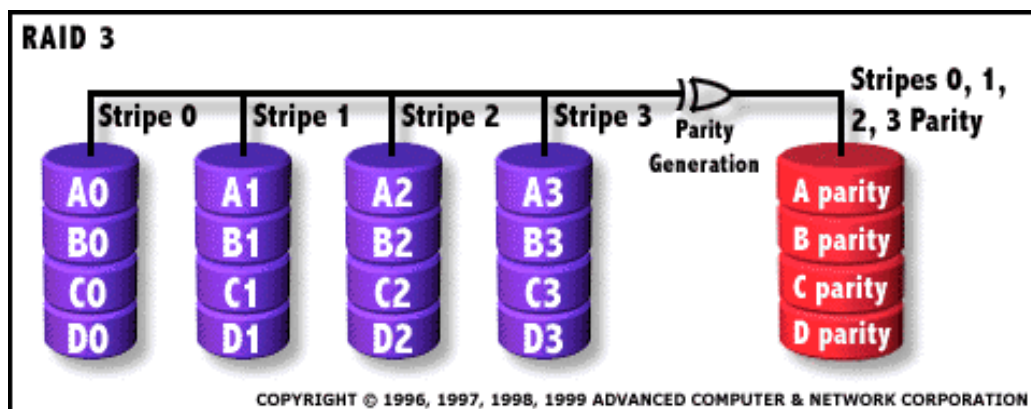
10

RAID Level 3

- similar to RAID level 2, but it requires only one disk for parity for 4 data disks
- **byte-level** striping
- suffers from a write bottleneck, because all parity data is written to a single drive
- but provides some read and write performance improvement.
- RAID 2 & 3 cannot serve multiple requests simultaneously

11

RAID Level 3 - byte level



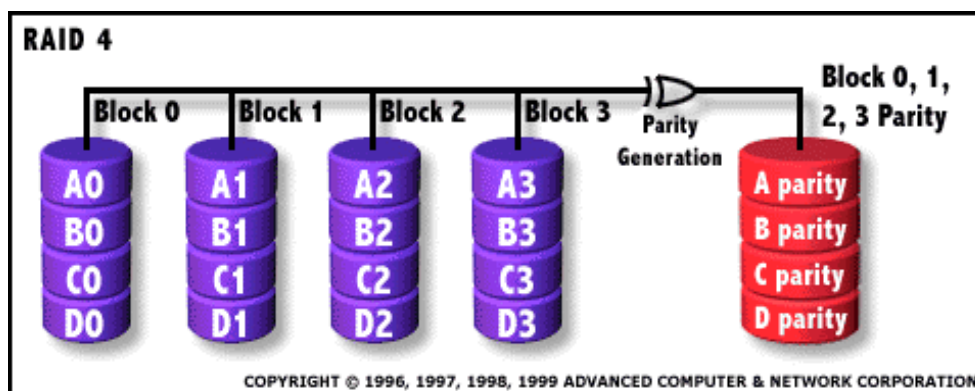
12

RAID Level 4

- Similar to RAID level 3, but it employs striped data in much larger **blocks** or segments
- Not as efficient as RAID level 5, because (as in RAID level 3) all parity data is written to a single drive
- RAID level 4 suffers from a write bottleneck (due to parity disk) and is not generally used.

13

RAID Level 4 - block level



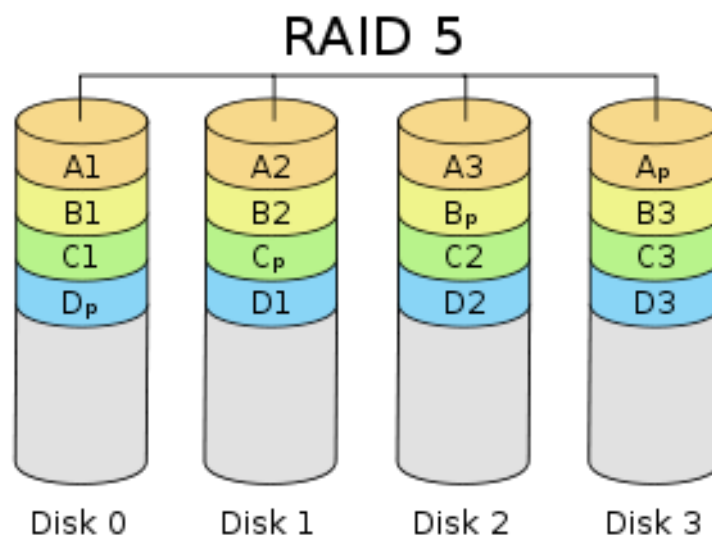
14

RAID Level 5

- known as **striping with parity**
- the most popular RAID level, replaced RAID 3 & 4
- similar to level 4 in that it stripes the data in large **blocks** across all the disks in the array
- It differs in that it **writes the parity across all the disks**
- The data redundancy is provided by the parity information
- The data and parity information are arranged on the disk array so that the two are always on different disks

15

RAID Level 5



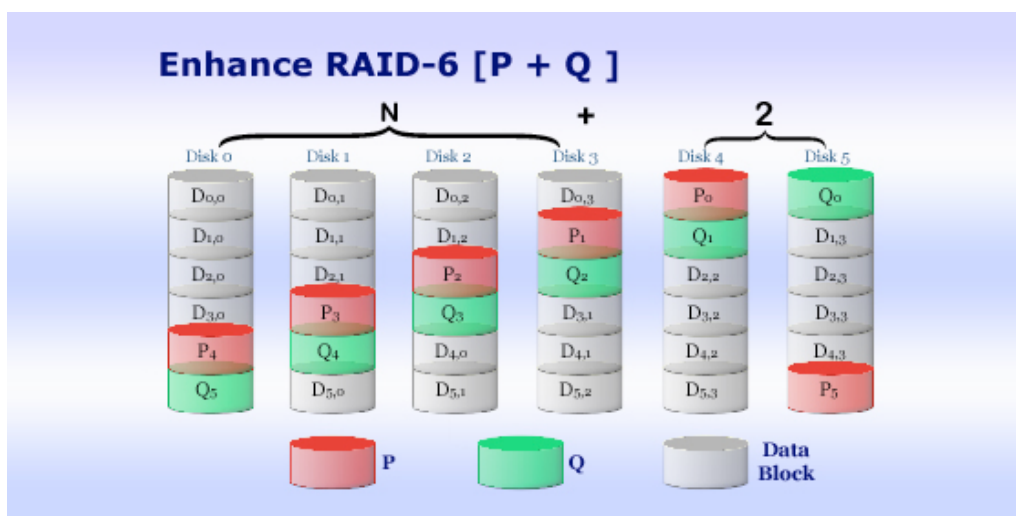
16

RAID Level 6

- stores extra redundant info to recover from multiple disk failures
- would need 2 additional disks for each 4 data disks
 - more reliability versus less data space
- uses Reed-Solomon error correcting code

17

RAID Level 6



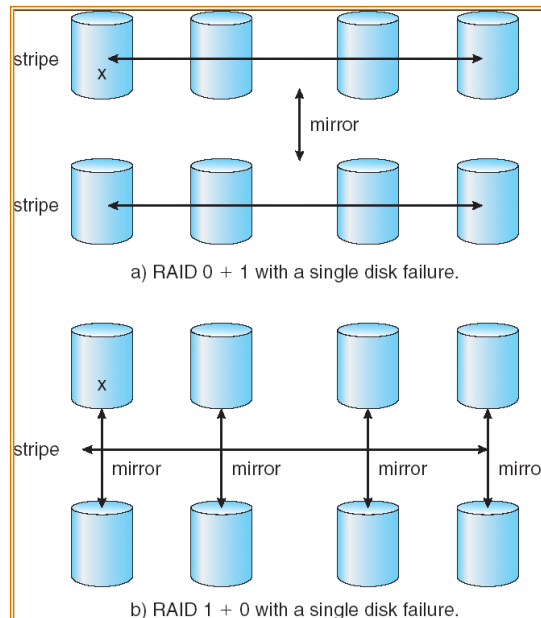
18

RAID (0+1) and (1+0)

- Combination of RAID 0 & 1
- better performance & reliability, but doubles the disk storage requirement

19

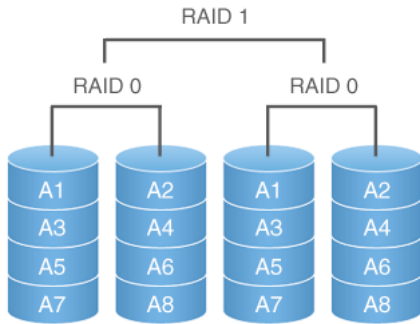
RAID (0+1) and (1+0)



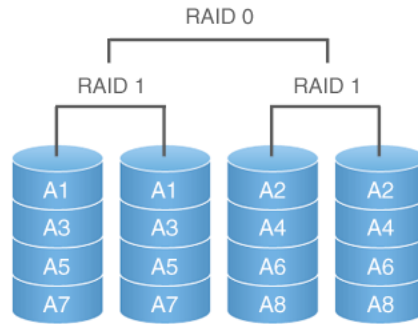
20

RAID (0+1) and (1+0)

RAID 0+1



RAID 1+0



Any Questions?



Acknowledgements

- “Operating Systems Concepts” book and supplementary material by A. Silberschatz, P. Galvin and G. Gagne
- “Operating Systems: Internals and Design Principles” book and supplementary material by W. Stallings
- “Modern Operating Systems” book and supplementary material by A. Tanenbaum
- R. Doursat and M. Yuksel from UNR