CSE2008: Operating Systems

L30 L31 L32 & L33: Storage Management

Dr. Subrata Tikadar

SCOPE, VIT-AP University

Recap

- Introductory Concepts
- Process Fundamentals
- IPC
- CPU Scheduling Algorithms
- Multithreading Concepts
- Synchronization
- Deadlock
- Memory Management
- Virtual Memory Concepts

Outline

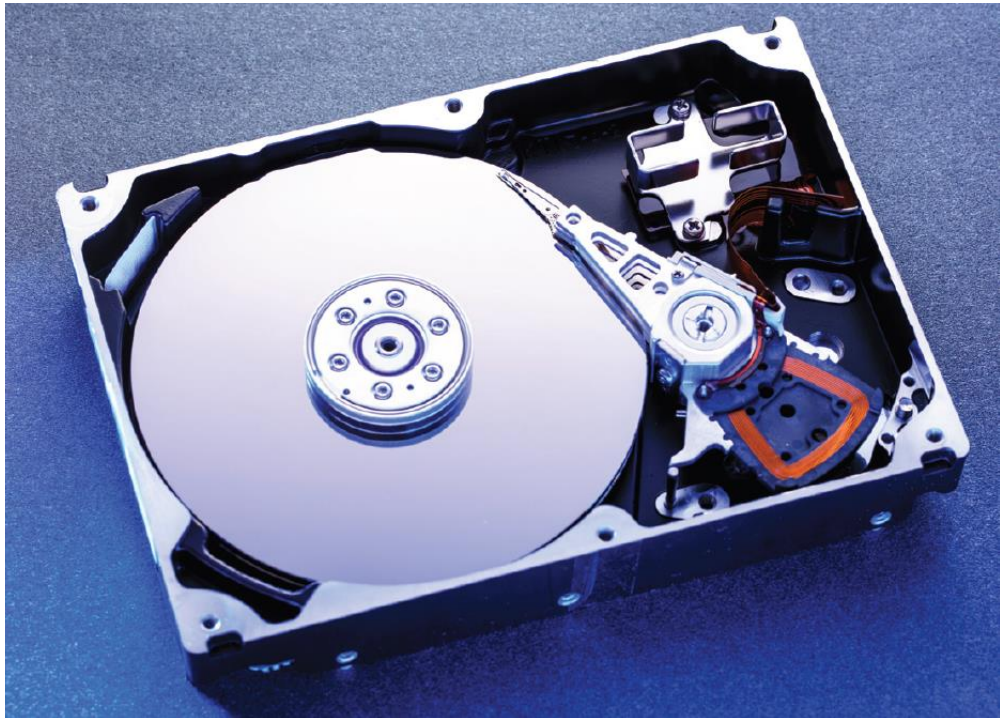
- Mass-Storage Structure
- Disk Scheduling Algorithms
- Storage Device Management
- RAID System Structure

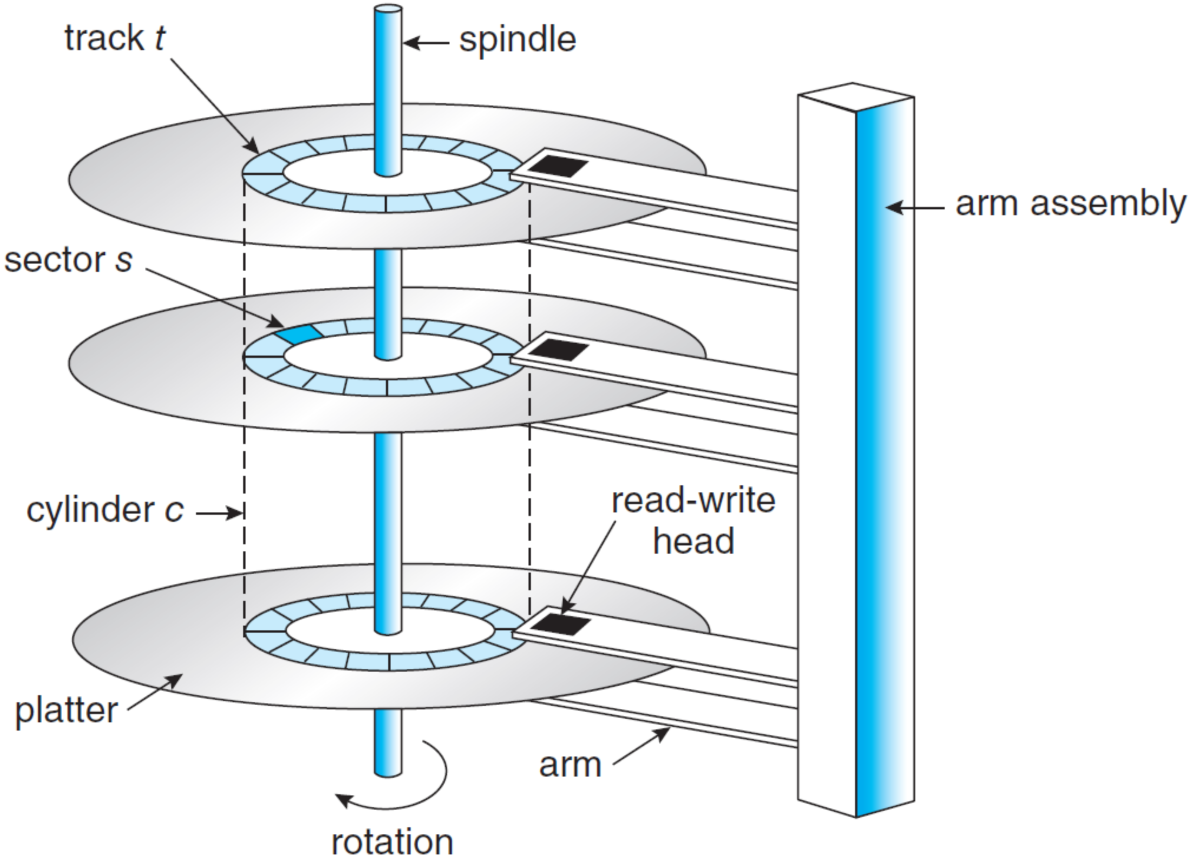
Overview of Mass-Storage Structure

- Overview of Hard Disk Drives
- Overview of Nonvolatile Memory Devices

Overview of Mass-Storage Structure

- Overview of Hard Disk Drives





Overview of Mass-Storage Structure

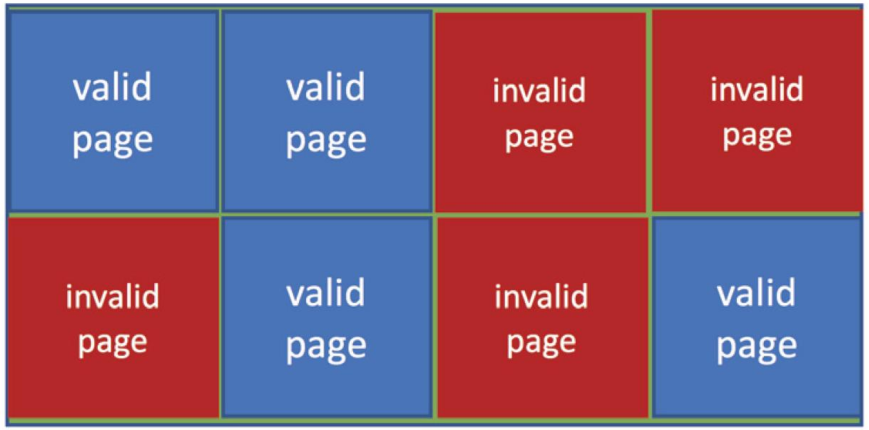
- Overview of Nonvolatile Memory Devices
  - Advantages
    - NVM devices can be more reliable than HDDs because they have no moving parts
    - Can be faster because they have no seek time or rotational latency.
    - Consume less power
  - Disadvantages
    - More expensive per megabyte than traditional hard disks
    - Have less capacity than the larger hard disks

Over time, however, the capacity of NVM devices has increased faster than HDD capacity, and their price has dropped more quickly, so their use is increasing dramatically. In fact, SSDs and similar devices are now used in some laptop computers to make them smaller, faster, and more energy-efficient.

Overview of Mass-Storage Structure

- Overview of Nonvolatile Memory Devices
  - NAND Flash Controller Algorithms

```
If some spaces are empty
    Write can be done to the empty spaces
Else (i.e., Full SSD)
    If invalid page references exists
            //Flash Translation Layer (FTL) tracks valid and invalid data
            Write could occur after erasing recorded invalid pages
    Else
            Garbage collection (through overprovisioning)
```



valid page	valid page	invalid page	invalid page
invalid page	valid page	invalid page	valid page

Note: *Overprovisioning also helps with wear leveling (so that SSD does not wear out quickly*)

Overview of Mass-Storage Structure

- Secondary Storage Connection Methods
  - By the system bus or an I/O bus
    - Advanced Technology Attachment (ATA)
    - **Serial ATA (SATA)**
    - eSATA
    - Serial Attached SCSI (SAS)
    - Universal Serial Bus (USB)
    - Fiber Channel (FC)
    - **NVM express (NVMe)** – specifically created for connecting NVM

Overview of Mass-Storage Structure

- Address Mapping
  - Storage devices are addressed as large one-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer.
    - In case of HDDs
      - Each logical block maps to a physical sector or semiconductor page. The one-dimensional array of logical blocks is mapped onto the sectors or pages of the device
    - In Case of NVM
      - The mapping is from a tuple (finite ordered list) of chip, block, and page to an array of logical blocks.

A logical block address (LBA) is easier for algorithms to use than a sector, cylinder, head tuple or chip, block, page tuple.

Disk Scheduling Algorithms

- Basic Objectives
  - Minimize access time
  - Maximize data transfer bandwidth

Example:

Access time (in case of HDDs and other mechanical storage that use platters**):**

Seek Time: Time for the device arm to move the heads to the cylinder containing the desired sector

Rotational Latency: The additional time for the platter to rotate the desired sector to the head

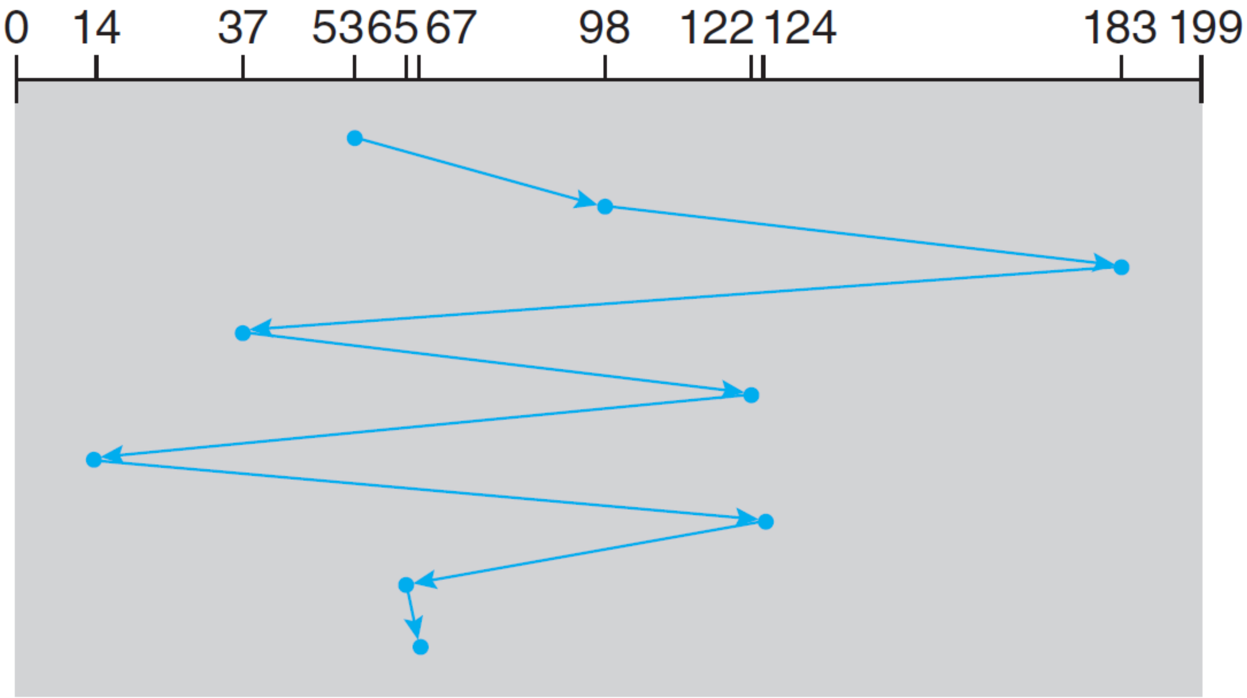
Bandwidth :

The total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer

Disk Scheduling Algorithms

- FCFS Scheduling

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

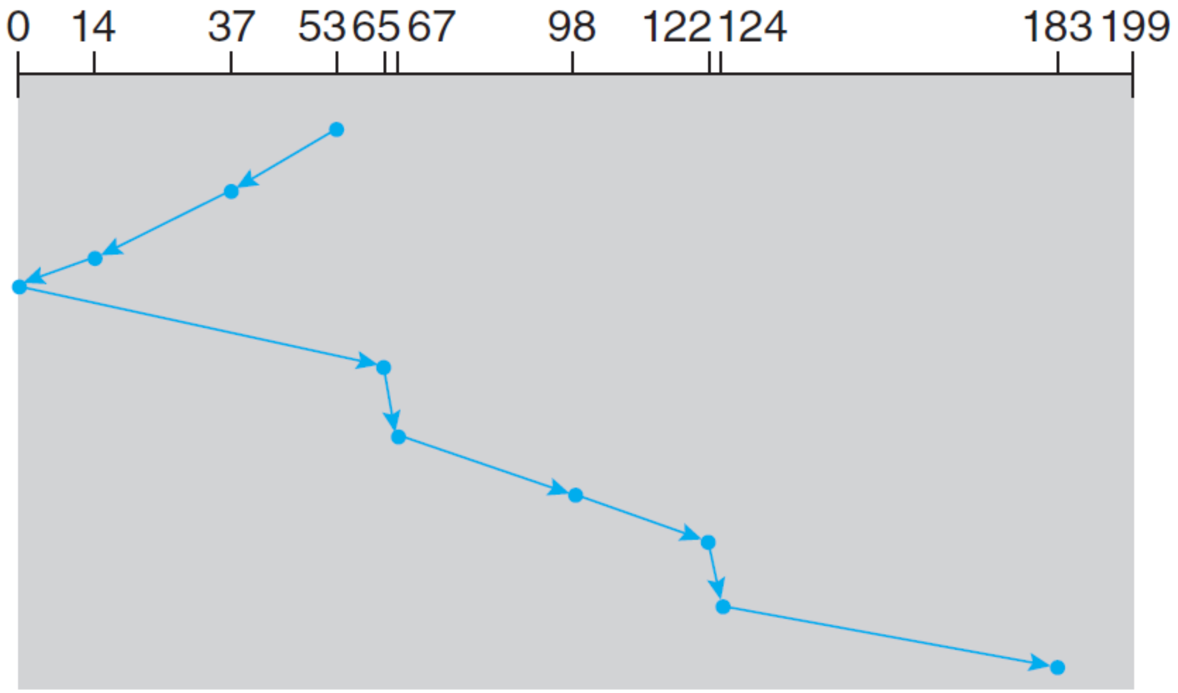


Disk Scheduling Algorithms

- SCAN Scheduling

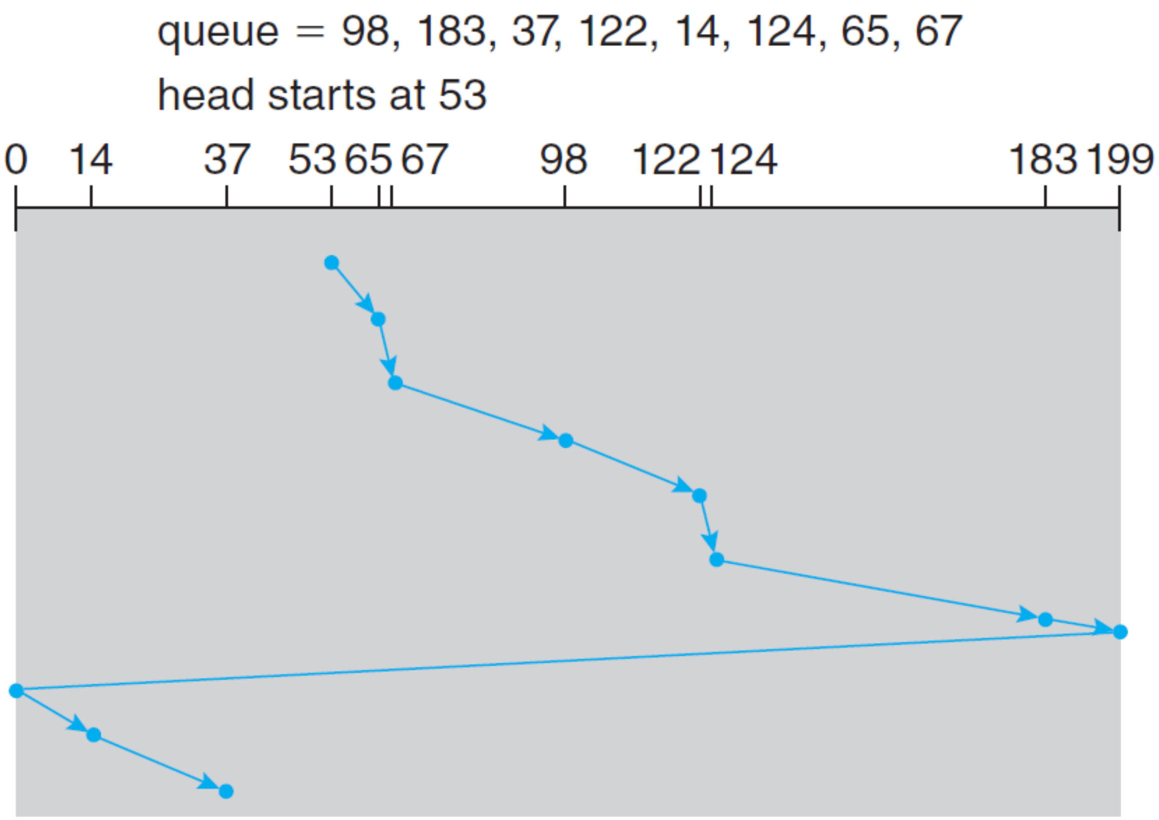
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



Disk Scheduling Algorithms

- C-SCAN Scheduling



queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

Disk Scheduling Algorithms

- NVM Scheduling
  - Simple FCFS policy
  - FCFS policy with merge adjacent requests

Storage Device Management

- Drive Formatting, Partitions, and Volumes
- Boot Block
- Bad Blocks

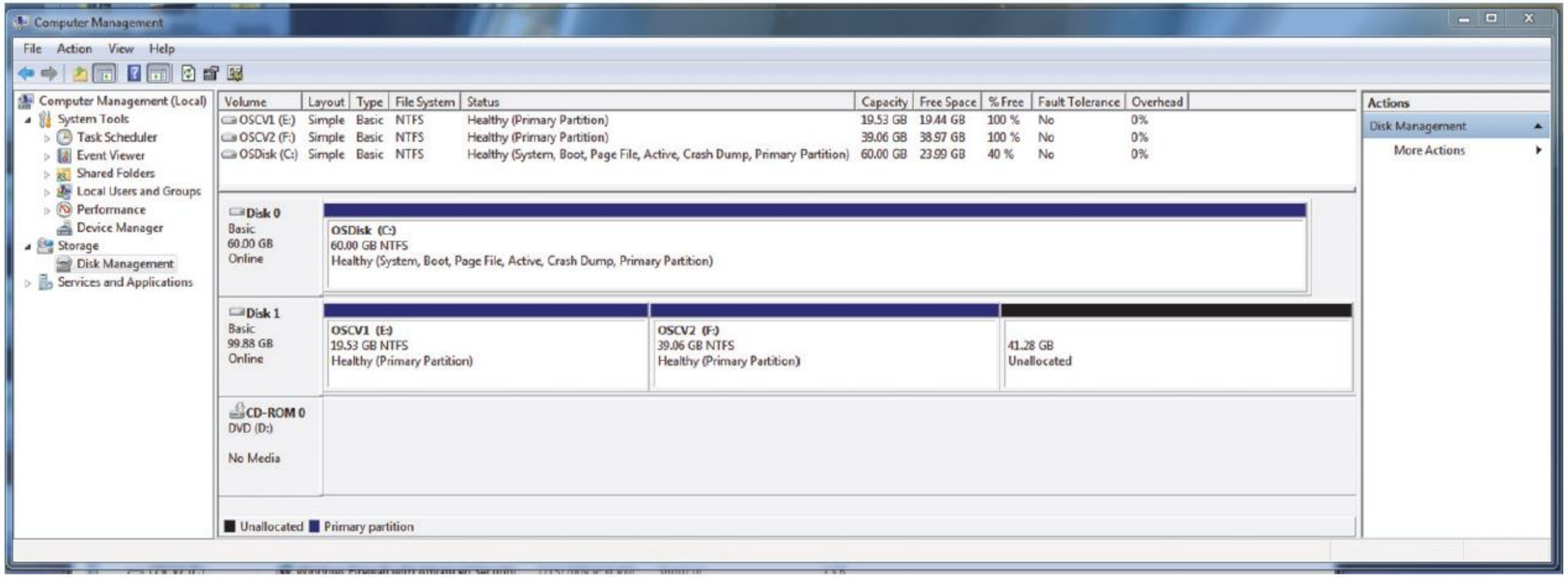
Storage Device Management

- Drive Formatting, Partitions, and Volumes

  Step1: Partition the device into one or more groups of blocks or pages

  Step2: Volume creation and management

  Step3: Logical formatting, or creation of a file system



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
OSCV1 (E:)	Simple	Basic	NTFS	Healthy (Primary Partition)	19.53 GB	19.44 GB	100 %	No	0%
OSCV2 (F:)	Simple	Basic	NTFS	Healthy (Primary Partition)	39.06 GB	38.97 GB	100 %	No	0%
OSDisk (C:)	Simple	Basic	NTFS	Healthy (System, Boot, Page File, Active, Crash Dump, Primary Partition)	60.00 GB	23.99 GB	40 %	No	0%

Disk 0
Basic
60.00 GB
Online

OSDisk (C:)
60.00 GB NTFS
Healthy (System, Boot, Page File, Active, Crash Dump, Primary Partition)

Disk 1
Basic
99.88 GB
Online

OSCV1 (E:)
19.53 GB NTFS
Healthy (Primary Partition)

OSCV2 (F:)
39.06 GB NTFS
Healthy (Primary Partition)

41.28 GB
Unallocated

CD-ROM 0
DVD (D:)

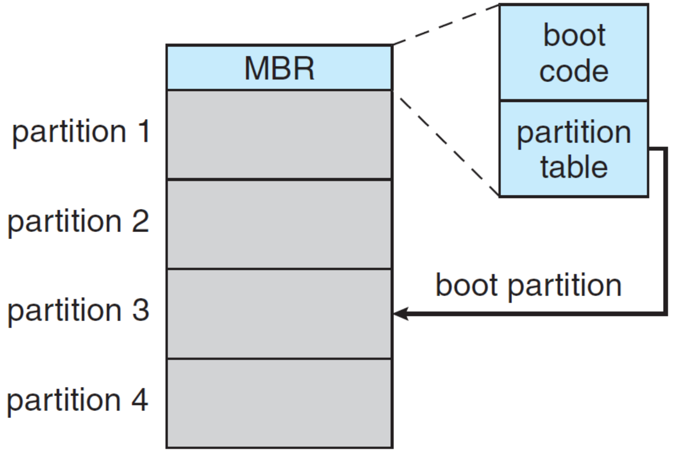
No Media

■ Unallocated ■ Primary partition

Storage Device Management

- Boot Block
  - The default Linux bootstrap loader is grub2 (https://www.gnu.org/software/grub/manual/grub.html/).
  - A device that has a boot partition is called a boot disk or system disk.

  - Booting from a storage device in Windows



Storage Device Management

- Bad Blocks
  - Can be handled (now-a-days)
    - Manually
      - ➢ Scan the disk to find bad blocks while the disk is being formatted
      - ➢ Discovered bad blocks are flagged as unusable so that the file system does not allocate them
    - Automatically (e.g., sector sparing or forwarding)
      - ➢ The controller maintains a list of bad blocks on the disk
      - ➢ The list is initialized during the low-level formatting at the factory and is updated over the life of the disk.
      - ➢ Low-level formatting also sets aside spare sectors not visible to the operating system.
      - ➢ The controller can be told to replace each bad sector logically with one of the spare sectors.

Storage Device Management

- Bad Blocks
  - Bad-sector transaction:
    - The operating system tries to read logical block 87
    - The controller calculates the ECC and finds that the sector is bad. It reports this finding to the operating system as an I/O error.
    - The device controller replaces the bad sector with a spare.
    - After that, whenever the system requests logical block 87, the request is translated into the replacement sector's address by the controller.

Swap-Space Management

- Swap-Space Use
- Swap-Space Location
- Example of Swap-Space Management

Swap-Space Management

- Swap-Space Use
  - Used in various ways by different OSs, depending on the memory-management algorithms in use

    Example:

    - Systems that implement swapping may use swap space to hold an entire process image, including the code and data segments.
    - Paging systems may simply store pages that have been pushed out of main memory

Swap-Space Management

- Swap-Space Location
  - Can reside in one of two places
    - Can be carved out of the normal file system
    - Can be in a separate partition.

Swap-Space Management

- Example of Swap-Space Management
  - The traditional UNIX kernel started with an implementation of swapping that copied entire processes between contiguous disk regions and memory. UNIX later evolved to a combination of swapping and paging as paging hardware became available.
  - In Solaris 1 (SunOS), when a process executes, text-segment pages containing code are brought in from the file system, accessed in main memory, and thrown away if selected for pageout.
  - Linux allows one or more swap areas to be established. A swap area may be in either a swap file on a regular file system or a dedicated swap partition.

RAID System Structure

- Background – Storage Attachment
  - Host-attached storage
  - Network-attached storage
  - Cloud storage

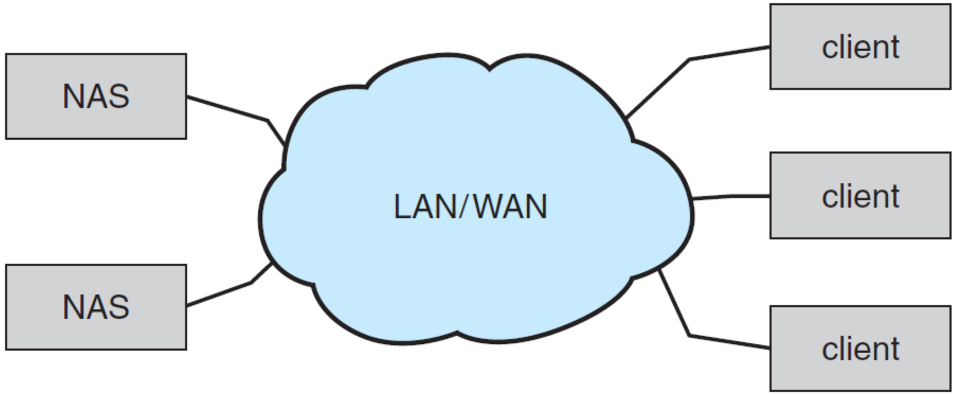
RAID System Structure

- Background – Storage Attachment
  - Host-attached storage
    - Accessed through local I/O ports (SATA, NVMe)
    - High-end workstations and servers generally need more storage or need to share storage, so use more sophisticated I/O architectures, such as fiber channel (FC), a high speed serial architecture that can operate over optical fiber or over a four-conductor copper cable
    - Example host-attached store device
      - HDDs; NVM devices; CD, DVD, Blu-ray, and tape drives

RAID System Structure

- Background – Storage Attachment
  - Network-attached storage (NAS)
    - Provides access to storage across a network
    - Can be
      - ○ A special-purpose storage system
      - ○ A general computer system that provides its storage to other hosts across the network
    - Clients access NAS via a remote-procedure call interface (RPCI)
      - ○ NFS for UNIX and Linux systems
      - ○ CIFS for Windows machines

Note: *Tends to be less efficient and have lower performance than some direct-attached storage options, although iSCSI (the latest NAS protocol) is improving over this.*



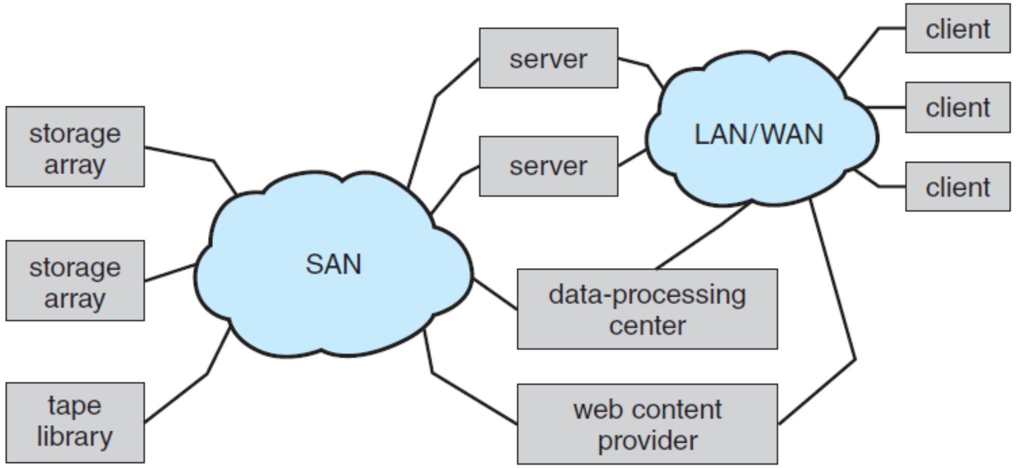
RAID System Structure

- Background – Storage Attachment
  - Cloud storage
    - Similar to NAS, cloud storage provides access to storage across a network
    - Difference (from NAS) in mainly two aspects
      1. The storage is accessed over the Internet or another WAN to a remote data center that provides storage for a fee (or even for free)
      2. Connection via CIFS/NFS/iSCSI protocol in case of NAS whereas it is API based in case of cloud storage (minimizes latency and failure)

Examples: Amazon S3, Dropbox, Microsoft OneDrive and Apple iCloud

RAID System Structure

- Background – Storage Attachment
  - Storage Area Networks and Storage Arrays
    - To address *the drawback of NAS (storage I/O operations consume bandwidth on the data network → increases latency of network communication)*, a private network (using storage protocols rather than networking protocols) is formed connecting servers and storage units (as shown in the RHS figure).
    - Advantage
      - Flexibility – multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts



- The storage arrays can be **RAID protected** or unprotected drives (Just a Bunch of Disks (JBOD))
- A storage array is a purpose-built device that includes SAN ports, network ports, or both

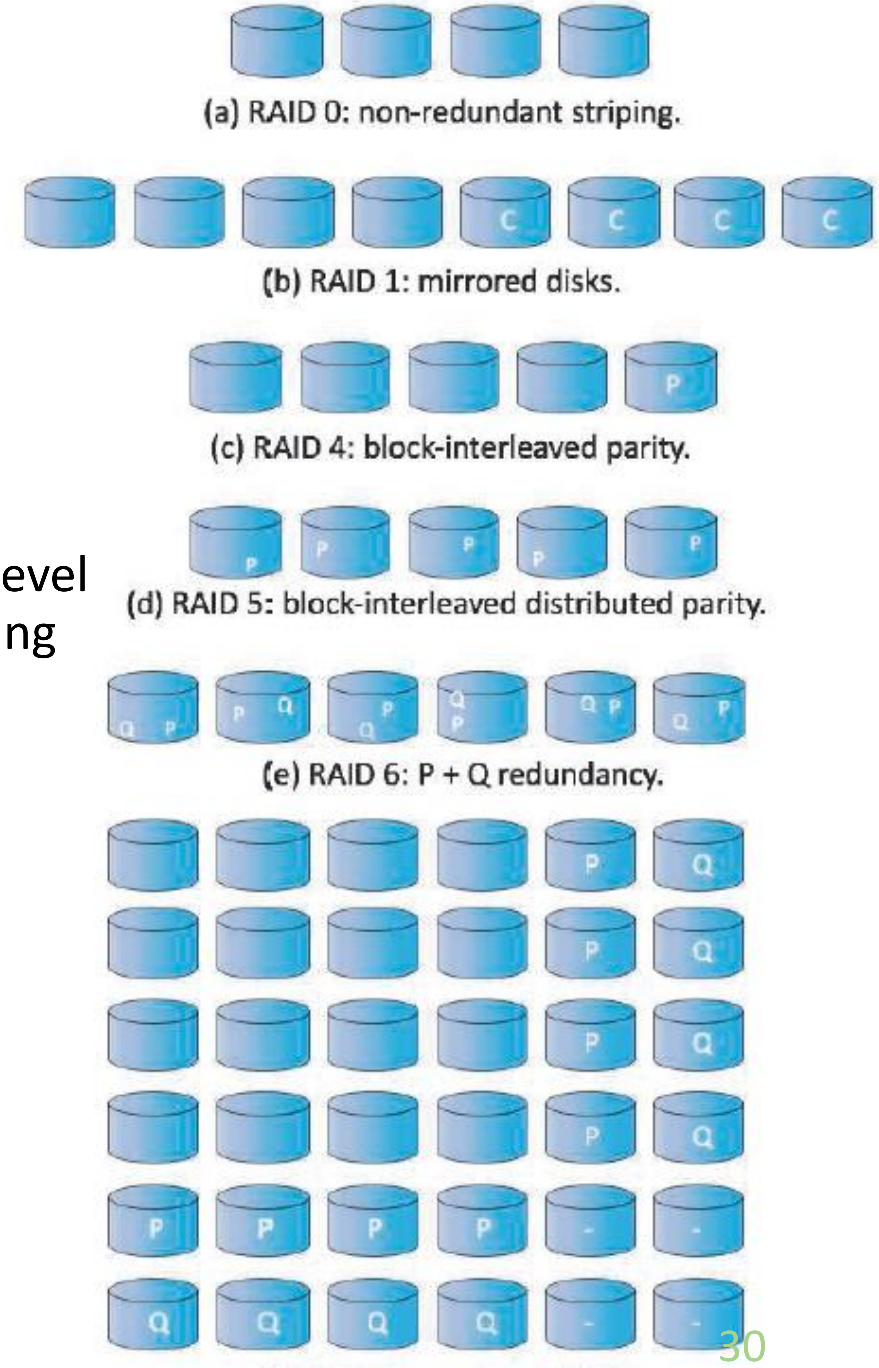
RAID System Structure

- Definition
  - In order to address the performance and reliability issues, a variety of disk-organization techniques have been proposed, which are collectively called *redundant arrays of independent disks* (RAIDs).

- Benefits
  - Having a large number of drives in a system presents opportunities for improving the rate at which data can be read or written, if the drives are operated in parallel
  - Offers the potential for improving the reliability of data storage, because redundant information can be stored on multiple drives → failure of one drive does not lead to loss of data

RAID System Structure

- RAID Levels
  - RAID level 0
    - RAID level 0 refers to drive arrays with striping at the level of blocks but without any redundancy (such as mirroring or parity bits), as shown in Figure (a)



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

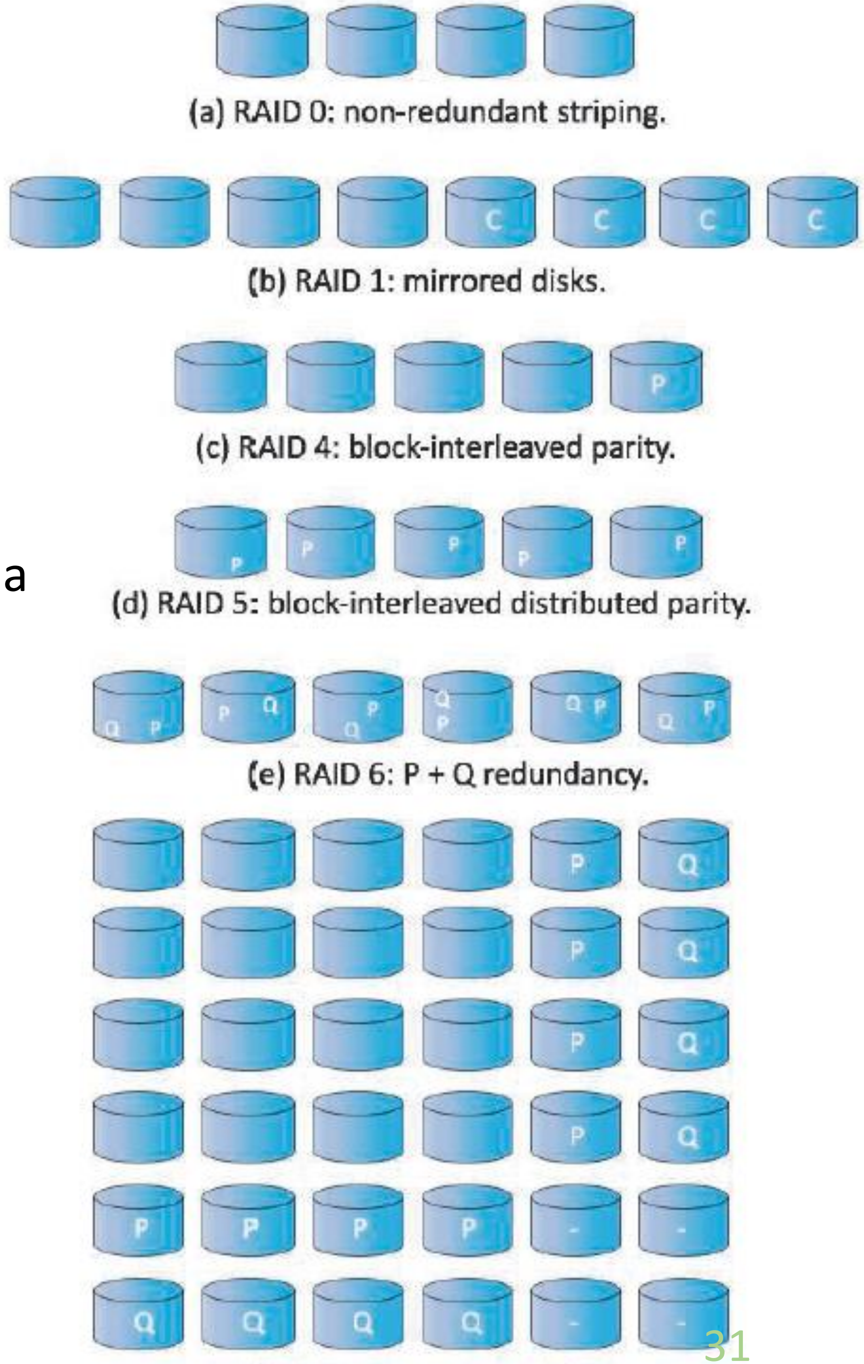
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
  - RAID level 1
    - RAID level 1 refers to drive mirroring. Figure (b) shows a mirrored organization



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

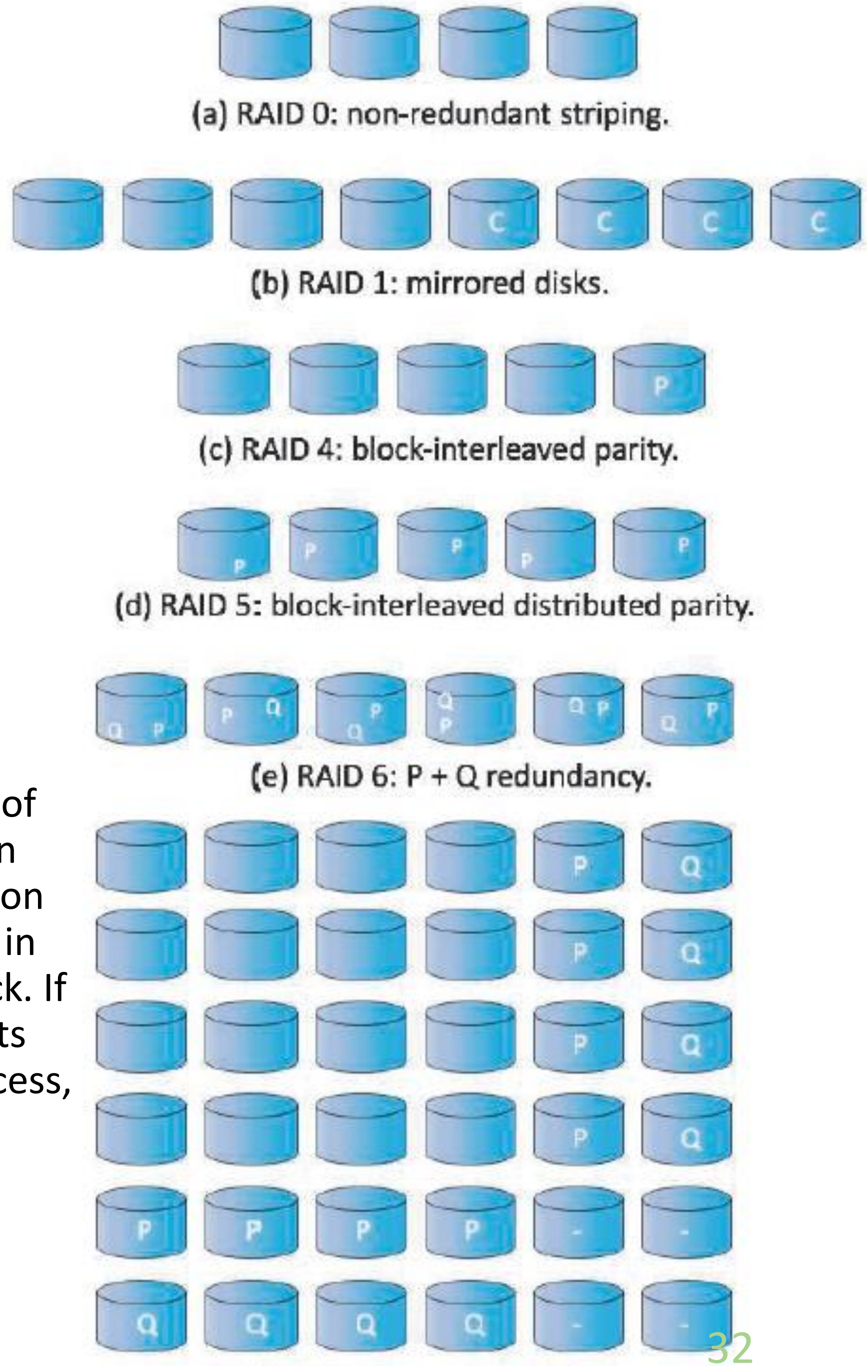
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
  - RAID level 4
    - RAID level 4 is also known as memory-style error-correcting code (ECC) organization. ECC is also used in RAID 5 and 6.
    - The idea of ECC can be used directly in storage arrays via striping of blocks across drives. For example, the first data block of a sequence of writes can be stored in drive 1, the second block in drive 2, and so on until the Nth block is stored in drive N; the error-correction calculation result of those blocks is stored on drive N + 1. This scheme is shown in Figure (c), where the drive labeled P stores the error-correction block. If one of the drives fails, the error-correction code recalculation detects that and prevents the data from being passed to the requesting process, throwing an error



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

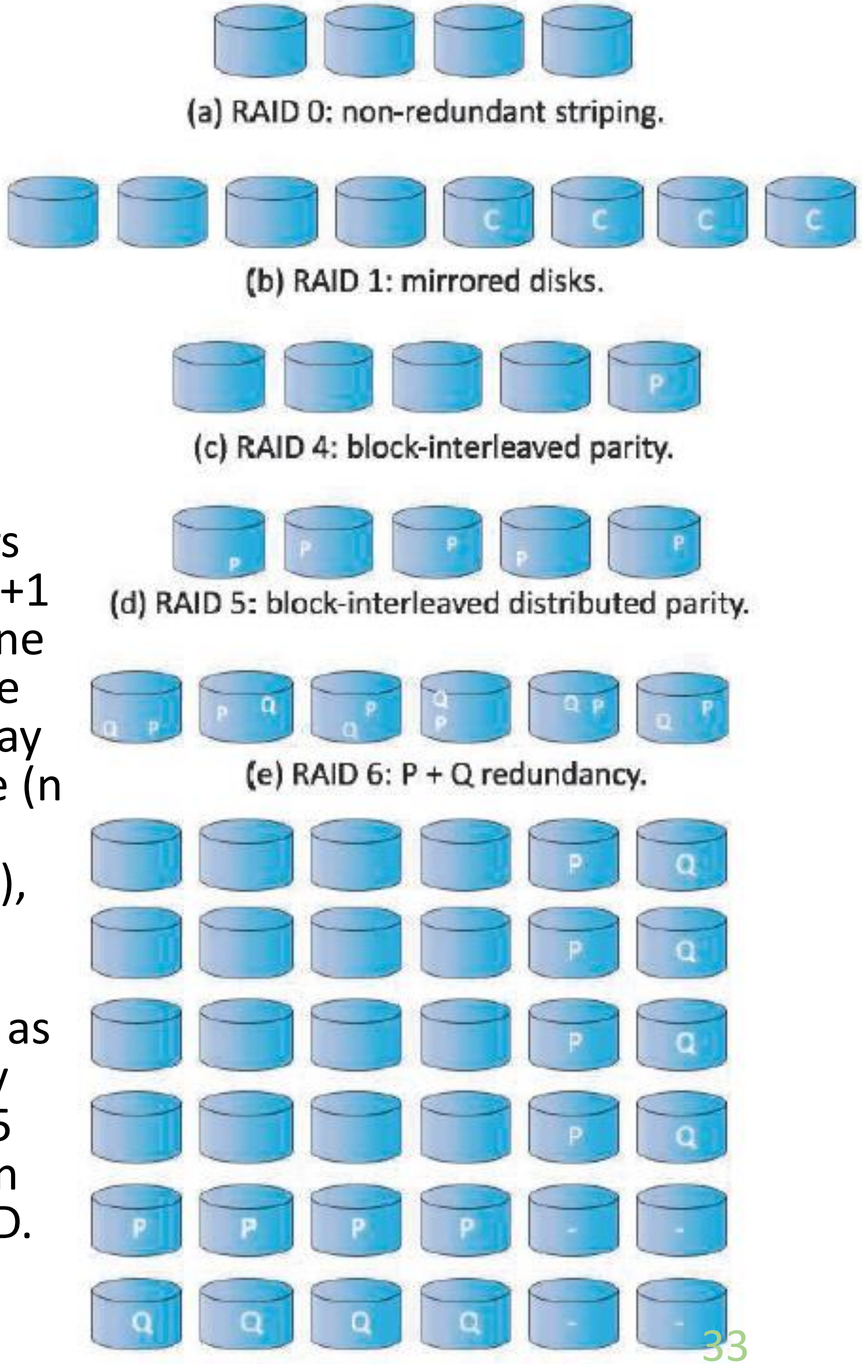
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
  - RAID level 5
    - RAID level 5, or block-interleaved distributed parity, differs from level 4 in that it spreads data and parity among all N+1 drives, rather than storing data in N drives and parity in one drive. For each set of N blocks, one of the drives stores the parity and the others store data. For example, with an array of five drives, the parity for the nth block is stored in drive (n mod 5) + 1. The nth blocks of the other four drives store actual data for that block. This setup is shown in Figure (d), where the Ps are distributed across all the drives. A parity block cannot store parity for blocks in the same drive, because a drive failure would result in loss of data as well as of parity, and hence the loss would not be recoverable. By spreading the parity across all the drives in the set, RAID 5 avoids potential overuse of a single parity drive, which can occur with RAID 4. RAID 5 is the most common parity RAID.



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

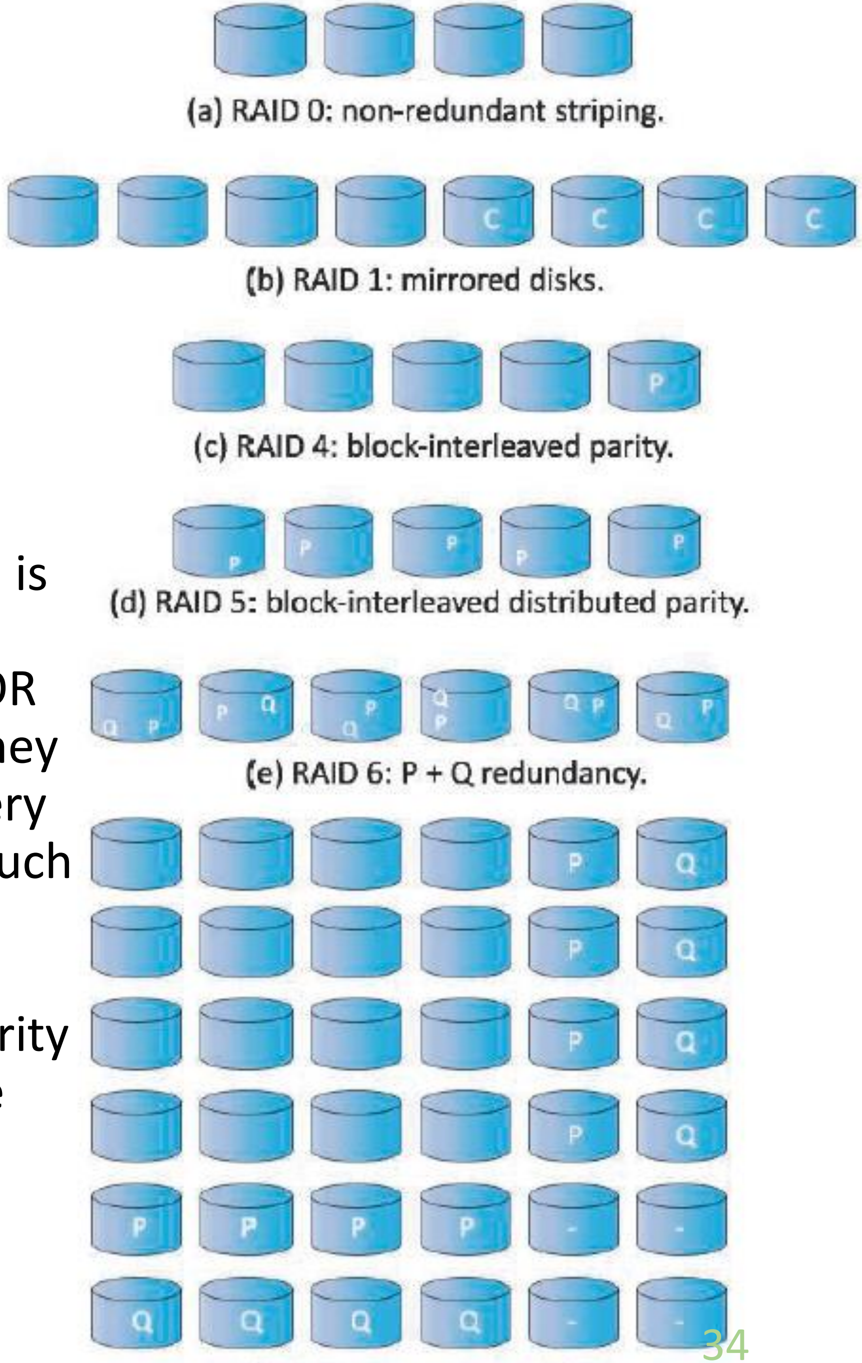
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
  - RAID level 6
    - RAID level 6, also called the P + Q redundancy scheme, is much like RAID level 5 but stores extra redundant information to guard against multiple drive failures. XOR parity cannot be used on both parity blocks because they would be identical and would not provide more recovery information. Instead of parity, error-correcting codes such as Galois field math are used to calculate Q. In the scheme shown in (e), 2 blocks of redundant data are stored for every 4 blocks of data—compared with 1 parity block in level 5—and the system can tolerate two drive failures.



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

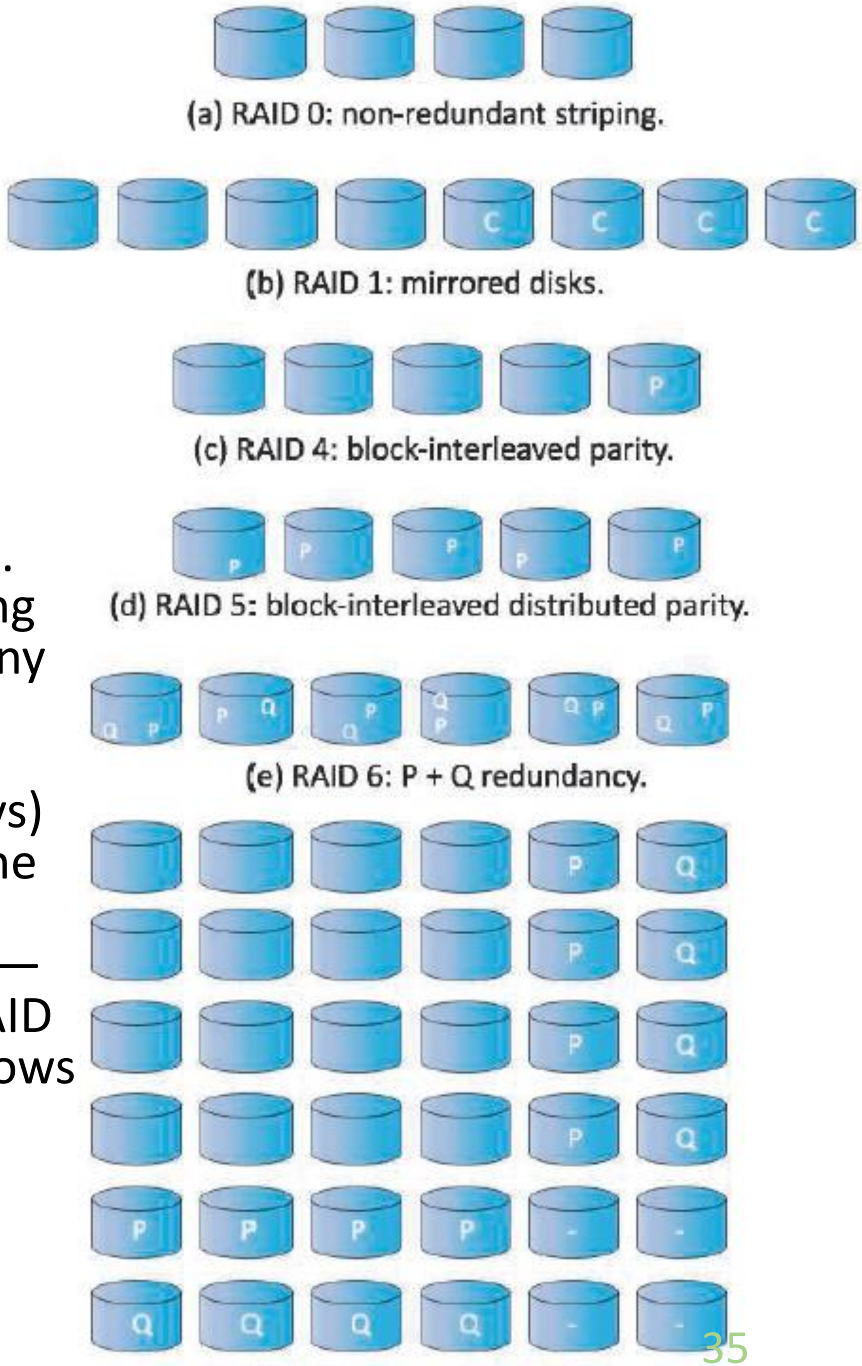
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
    - Multidimensional RAID level 6
        - Some sophisticated storage arrays amplify RAID level 6. Consider an array containing hundreds of drives. Putting those drives in a RAID level 6 stripe would result in many data drives and only two logical parity drives. Multidimensional RAID level 6 logically arranges drives into rows and columns (two or more dimensional arrays) and implements RAID level 6 both horizontally along the rows and vertically down the columns. The system can recover from any failure —or, indeed, multiple failures— by using parity blocks in any of these locations. This RAID level is shown in Figure (f). For simplicity, the figure shows the RAID parity on dedicated drives, but in reality the RAID blocks are scattered throughout the rows and columns.



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

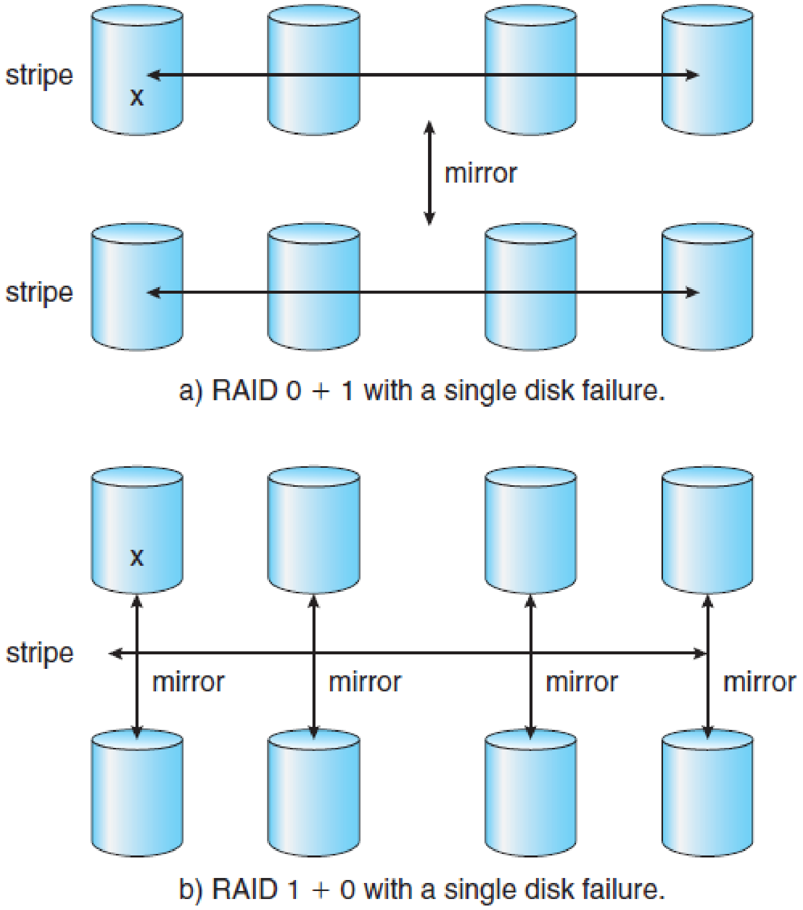
(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID System Structure

- RAID Levels
  - RAID levels 0 + 1 and 1 + 0
    - RAID level 0 + 1 refers to a combination of RAID levels 0 and 1. RAID 0 provides the performance, while RAID 1 provides the reliability. Generally, this level provides better performance than RAID 5. It is common in environments where both performance and reliability are important. Unfortunately, like RAID 1, it doubles the number of drives needed for storage, so it is also relatively expensive. In RAID 0 + 1, a set of drives are striped, and then the stripe is mirrored to another, equivalent stripe.
    - In RAID level 1 + 0, drives are mirrored in pairs and then the resulting mirrored pairs are striped. This scheme has some theoretical advantages over RAID 0 + 1. For example, if a single drive fails in RAID 0 + 1, an entire stripe is inaccessible, leaving only the other stripe. With a failure in RAID 1 + 0, a single drive is unavailable, but the drive that mirrors it is still available, as are all the rest of the drives.



a) RAID 0 + 1 with a single disk failure.

b) RAID 1 + 0 with a single disk failure.

RAID System Structure

- Selecting a RAID Level
  - Rebuild performance varies with the RAID level used. Rebuilding is easiest for RAID level 1, since data can be copied from another drive. For the other levels, we need to access all the other drives in the array to rebuild data in a failed drive. Rebuild times can be hours for RAID level 5 rebuilds of large drive sets.
    - Example:
      - RAID level 0 is used in high-performance applications where data loss is not critical
      - RAID level 1 is popular for applications that require high reliability with fast recovery
      - RAID 0 + 1 and 1 + 0 are used where both performance and reliability are important —for example, for small databases
      - Due to RAID 1's high space overhead, RAID 5 is often preferred for storing moderate volumes of data
      - RAID 6 and multidimensional RAID 6 are the most common formats in storage arrays

Reference

- Abraham Silberschatz , Peter B. Galvin, Greg Gagne, “Operating System Concepts”, Addison Wesley, 10th edition, 2018
  - Chapter 11: Section 11.1 – 11.8

Next

- File Systems

- Quiz (next week -anytime)
- Assignment (by the end of this week – submission: by the end of next week)

Thank You