Chapter 11: 대용량 저장장치 구조

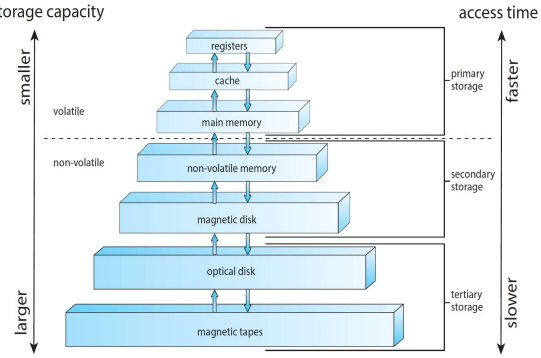
2020년

Chapter 12: 입출력 시스템 - Skip

대용량 저장 시스템

- Overview of Mass Storage Structure
- Storage Attachment
- Secondary Storage I/O Scheduling
- Storage Device Management
- Swap-Space Management
- RAID Structure

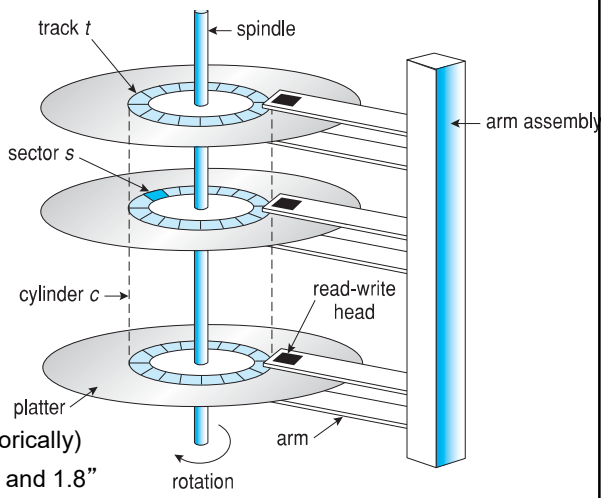
저장 장치 **계층**



Overview of Mass Storage Structure

- 자기 디스크(Magnetic disks) provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 250 times per second – RPM(revolution per minute)
  - **Transfer rate** is rate at which data flow between drive and computer
  - **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**탐색시간 seek time**) and time for desired sector to rotate under the disk head (**회전지연 rotational latency**)
  - **Head crash** results from disk head making contact with the disk surface
- Disks can be removable
- Drive attached to computer via **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fiber Channel**, **SCSI**, **SAS**, **Firewire**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

HDD Moving-head Disk Mechanism



- Platters range
  - from .85” to 14” (historically)
  - Commonly 3.5”, 2.5”, and 1.8”
- Range
  - from gigabytes through terabytes per drive

Disk Structure

- Disk drives are addressed as large **1-dimensional arrays of logical blocks**, where the logical block is the smallest unit of transfer
  - Low-level formatting creates **logical blocks** on physical media
- The 1-dimensional array of **logical blocks is mapped into the sectors of the disk sequentially**
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost
  - Logical to physical address should be easy
    - Except for bad sectors
    - Non-constant # of sectors per track via constant angular velocity
  - Each sector 512B or 4KB – smallest I/O the drive can do

Hard Disks

- HDDs rotate at 60 to 250 times per second
- **Transfer rate** is rate at which data flow between drive and computer
- **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
- **Head crash** results from disk head making contact with the disk surface
  - That's bad
- Disks can be removable
- Other types of storage media include CDs, DVDs, Blu-ray discs. magnetic tape

Hard Disk Performance

- **Access Latency** = **Average access time** = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead
- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead =
  - 5ms + 4.17ms + 0.1ms + transfer time =
  - Transfer time = 4KB / 1Gb/s * 8Gb / GB * 1GB / 1024^2KB = 32 / (1024^2) = 0.031 ms
  - Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms

Solid-State Disks(SSD)

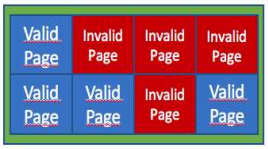
- Nonvolatile memory used like a hard drive
  - Many technology variations
- Can be more reliable than HDDs
- More expensive per MB
- Maybe have shorter life span
- Less capacity
- But much faster
- Busses can be too slow -> connect directly to PCI for example
- No moving parts, so no seek time or rotational latency

Nonvolatile Memory (NVM)

- **NVM** is electrical vs. mechanical (HDD)
- Commonly composed of a controller and flash NAND die semiconductor chips. Other forms include DRAM with battery backup, non-NAND die like 3D XPoint
- Flash-memory-based NVM is frequently used in disk-drive like container -> **solid-state disk (SSD)**
- Also can be in other formats like **USB drive**
- **More reliable than HDD** (no moving parts), can be faster (no seek time or latency), consumes less power
- Higher speed means new connection methods
  - Direct to PCIe bus (called **NVMe**)
- Read and written in "page" increment
- Cannot overwrite data, must erase it first
- Erase operation takes much longer than read or write
- NAND - ~100,000 program-erase cycles until cells no longer retain data

NVM Device Controller

- Several algorithms, usually implemented in NVM device controller
  - So operating system blissfully just reads and writes blocks and device deals with the physics
  - But can impact performance, so worth knowing about
- NAND block with valid and invalid pages
- NAND cannot be overwritten, therefore:
  - There are usually **pages containing invalid data**.
  - To track which logical blocks contain the valid data, the controller maintains a **Flash Translation Layer (FTL).**
  - This table maintains the mapping of which physical page contains the currently valid logical block.



Magnetic Tape

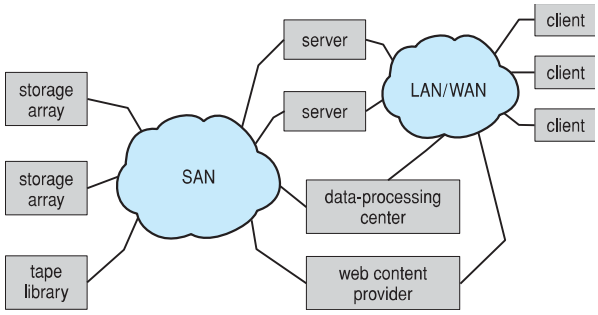
- Was early secondary-storage medium
  - Evolved from open spools to cartridges
- Relatively permanent and holds large quantities of data
- Access time slow
- Random access ~1000 times slower than disk
- Mainly used for backup, storage of infrequently-used data, transfer medium between systems

Secondary Storage Connection Methods

- Host-attached storage accessed through I/O ports talking to I/O busses
- Several bus technologies including **advanced technology attachment** (ATA), **serial ATA** (SATA), **eSATA**, **universal serial bus** (USB), **fibre channel** (FC), **serial attached SCSI** (SAS)
- Data transfers are carried out by special electronic processors: **controllers**
  - Host controller is in computer, device controller built into storage device
  - Talk to each other, usually via memory-mapped I/O ports
- Device controllers have built in caches
  - Data transfer from media into cache, then over bus to host (and vice versa)

Storage Area Network(SAN)

- Common in large storage environments
- Multiple hosts attached to multiple storage arrays - flexible

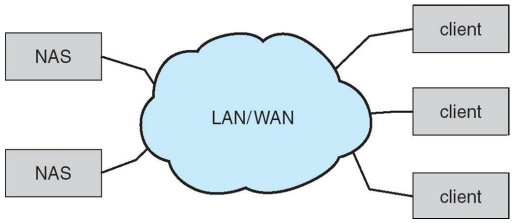


Storage Area Network (Cont.)

- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts also attach to the switches
- Storage made available via **LUN Masking** from specific arrays to specific servers
- Easy to add or remove storage, add new host and allocate it storage
  - Over low-latency Fibre Channel fabric
- Why have separate storage networks and communications networks?
  - Consider iSCSI, FCOE

Network-Attached Storage

- Network-attached storage (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
  - Remotely attaching to file systems
- NFS and CIFS are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network
- **iSCSI protocol** uses IP network to carry the SCSI protocol
  - Remotely attaching to devices (blocks)



Cloud Storage

- Similar to NAS
  - Provides storage across the network
  - Usually not owned by the company or user, but provided for a fee (based on time, storage capacity used, I/O done, etc.)
  - But across a WAN rather than a LAN
  - Frequently too slow, more prone to connection interruption than NAS so CIFS, NFS, iSCSI possibly but less used
  - Frequently use their own APIs, and apps that use those APIs to do I/O
    - Dropbox, Microsoft OneDrive, Apple iCloud, etc.

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Minimize seek time
- Seek time ≈ seek distance
- Disk **bandwidth** is 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 (Cont.)

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists

Disk Scheduling (Cont.)

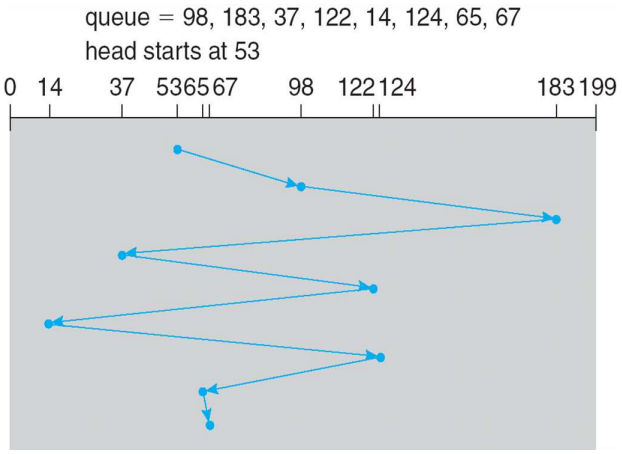
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying "depth")
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

Head pointer 53

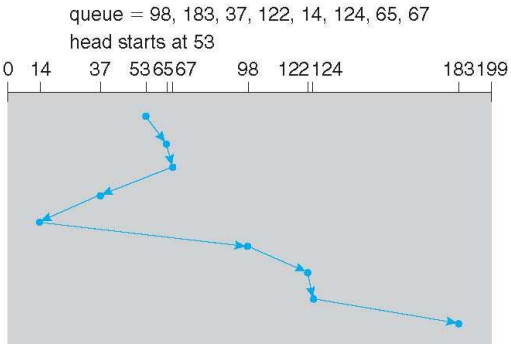
FCFS

Illustration shows total head movement of 640 cylinders



SSTF

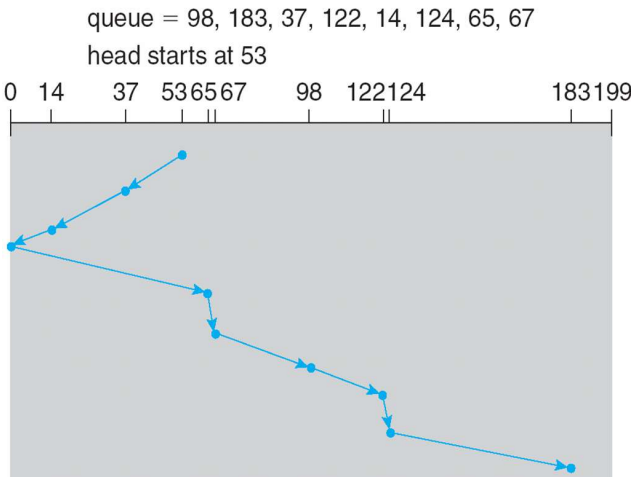
- **Shortest Seek Time First** selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders



SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- **SCAN algorithm** Sometimes called the **elevator algorithm**
- Illustration shows total head movement of 208 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

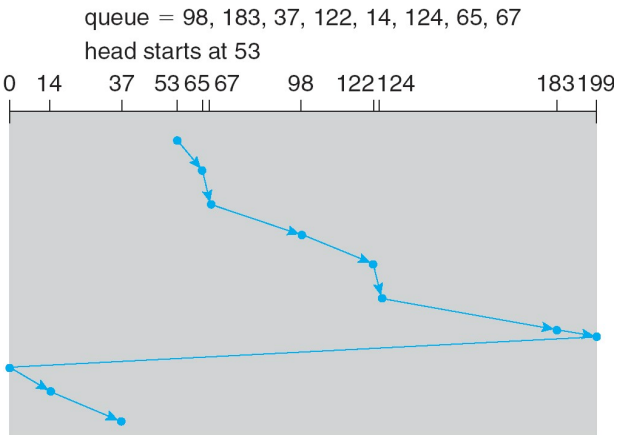
SCAN (Cont.)



C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Total number of cylinders?

C-SCAN (Cont.)



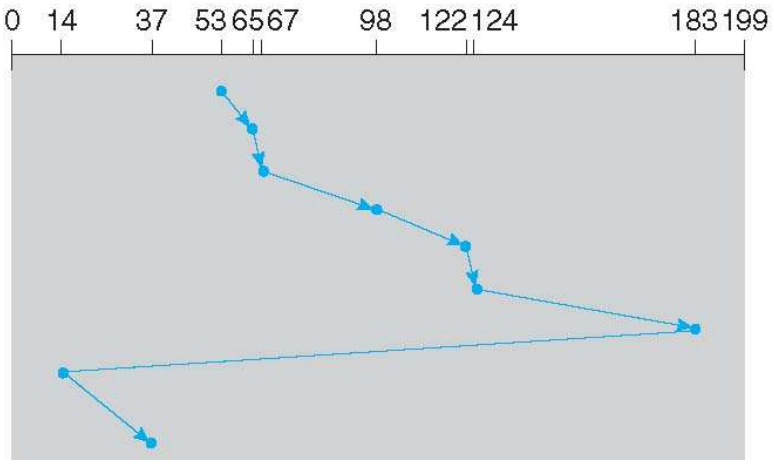
LOOK and C-LOOK

- **LOOK** is a version of SCAN. Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk
- **C-LOOK** a version of C-SCAN. Arm only goes as far as the last request in one direction, then reverses direction immediately, without first going all the way to the end of the disk

C-LOOK (Cont.)

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

head starts at 53



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
  - And metadata layout

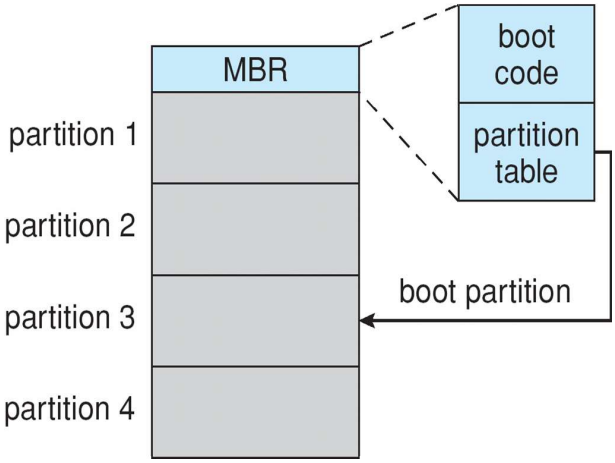
Disk Management

- **Low-level formatting**, or **physical formatting** — **Dividing a disk into sectors** that the disk controller can read and write
  - Each sector can hold header information, plus data, plus error correction code (**ECC**)
  - Usually 512 bytes of data but can be selectable
  - Bad Sector(배드 섹터) forwarding
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk
  - **Partition** the disk into one or more groups of cylinders, each treated as a **logical disk**
  - **Logical formatting** or "making a file system"
  - To increase efficiency most file systems group blocks into **clusters**
    - Disk I/O done in blocks
    - File I/O done in clusters

Disk Management (Cont.)

- Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot block initializes system
  - The bootstrap is stored in ROM
  - **Bootstrap loader** program stored in boot blocks of boot partition
- Methods such as **sector sparing** used to handle bad blocks

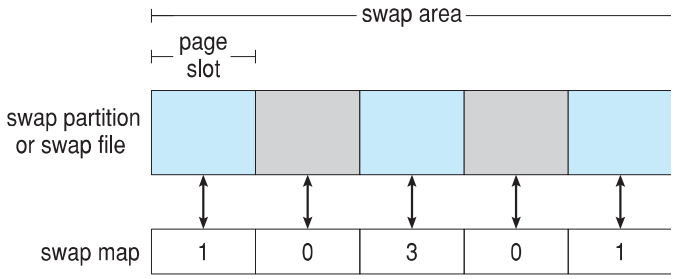
Booting from a Disk in Windows



Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
  - Less common now due to memory capacity increases
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management
  - 4.3BSD allocates swap space when process starts; holds text segment (the program) and data segment
  - Kernel uses **swap maps** to track swap-space use
  - Solaris 2 allocates swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first created
    - File data written to swap space until write to file system requested
    - Other dirty pages go to swap space due to no other home
    - Text segment pages thrown out and reread from the file system as needed
- What if a system runs out of swap space?
- Some systems allow multiple swap spaces

Data Structures for Swapping on Linux Systems



RAID 이해를 위하여-Reliability and Redundancy

- **Mean time to failure**. The average time it takes a disk to fail.
- **Mean time to repair.** The time it takes (on average) to replace a failed disk and restore the data on it.
- **Mirroring.** Copy of a disk is duplicated on another disk.
  - Consider disk with 100,000 hours of mean time to failure and 10 hour mean time to repair
    - Mean time to data loss is $100{,}000^2 / (2 * 10) = 500 * 10^6$ hours, or 57,000 years If mirrored disks fail independently,
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively

RAID Structure

- RAID – **redundant array of inexpensive disks**
- Use of many disks Increases the **mean time to failure.**
  - 100 disks with MTF of 100,000 hours.
  - 100,000/100 = 1,000 hours or 41.66 days.
- Solution is to have data redundancy over the 100 disks.
- **Disk striping.** Splitting the bits (or blocks) across multiple disks
  - **bit-level striping**. The bits of a byte are split across multiple disks.
  - **block-level striping**. The blocks of a file byte are split across multiple disks.
- For example, if we have an array of eight disks, we write bit "*i*" of each byte to disk "*i*". The array of eight disks can be treated as a single disk with sectors that are eight times the normal size and, more important, that have eight times the access rate.

RAID (Cont.)

- Disk **striping** uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - **Mirroring** or **shadowing** (**RAID 1**) keeps duplicate of each disk
  - Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability
  - **Block interleaved parity** (**RAID 4, 5, 6**) uses much less redundancy
- RAID within a storage array can still fail if the array fails, so automatic **replication** of the data between arrays is common
- Frequently, a small number of **hot-spare** disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them

RAID Levels

(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

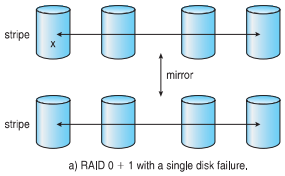
(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

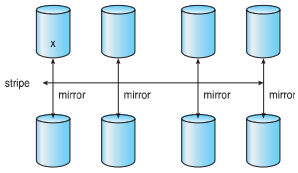
(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

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



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



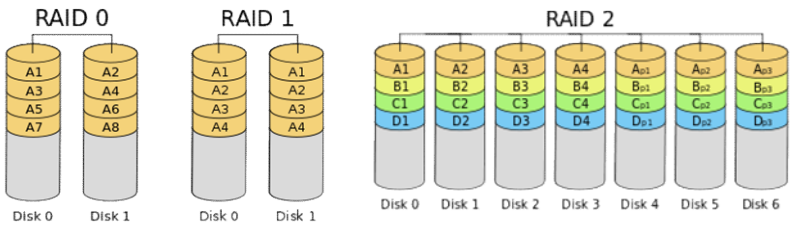
b) RAID 1 + 0 with a single disk failure.

RAID Levels 0, 1, 2

RAID 0 (also known as a stripe set or striped volume) splits ("stripes") data evenly across two or more disks, without parity information, redundancy, or fault tolerance.

RAID 1 consists of an exact copy (or mirror) of a set of data on two or more disks; a classic RAID 1 mirrored pair contains two disks.

RAID 2, which is rarely used in practice, stripes data at the bit (rather than block) level, and uses a Hamming code for error correction.



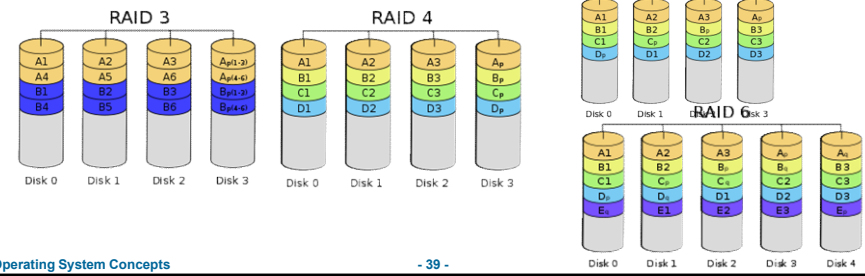
RAID Levels 3, 4, 5, 6

RAID 3, which is rarely used in practice, consists of byte-level striping with a dedicated parity disk.

RAID 4 consists of block-level striping with a dedicated parity disk.

RAID 5 consists of block-level striping with distributed parity. Unlike in RAID 4, parity information is distributed among the drives.

RAID 6 extends RAID 5 by adding another parity block; thus, it uses block-level striping with two parity blocks distributed across all member disks.



Extensions

- RAID alone does not prevent or detect data corruption or other errors, just disk failures
- Solaris ZFS adds **checksums** of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partitions
  - Disks allocated in **pools**
  - Filesystems with a pool share that pool, use and release space like `malloc()` and `free()` memory allocate / release calls