I/O Devices

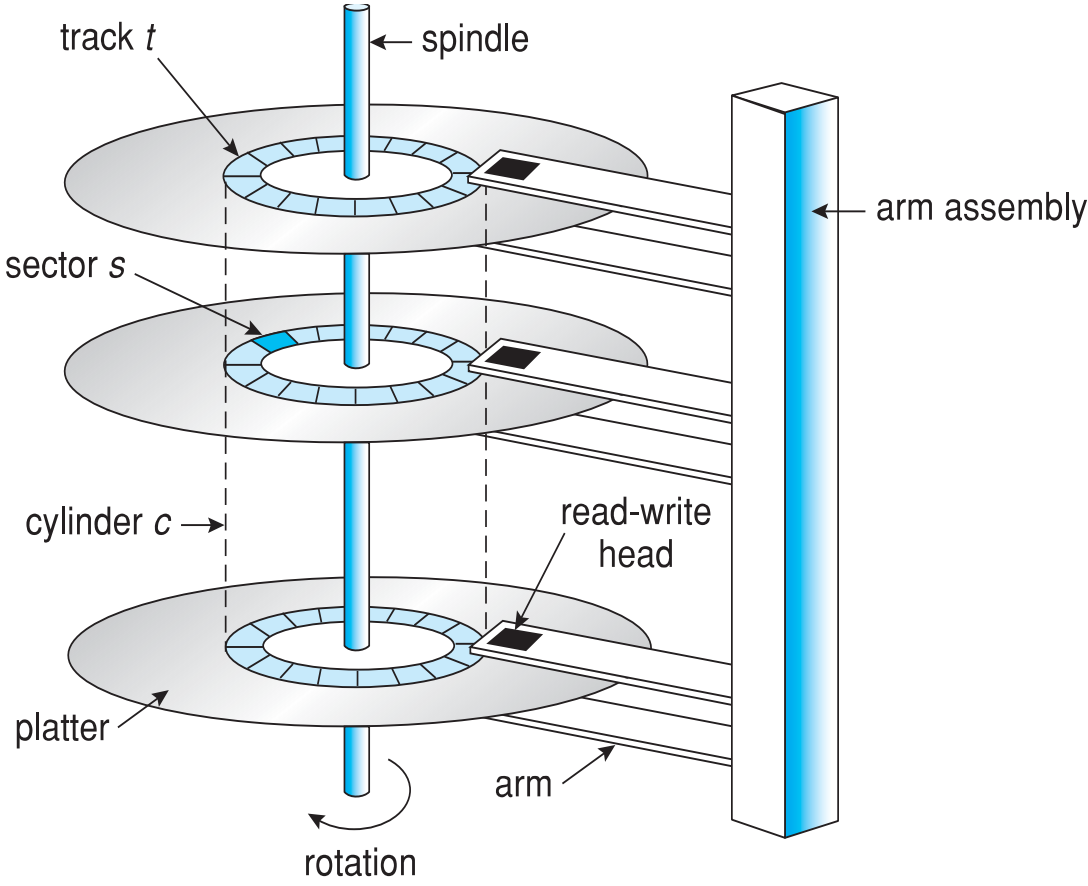
Operating Systems

Input/Output

- So far…
  - Principles of I/O Hardware
  - Principles of I/O Software
  - I/O Software Layers
- Now, some real I/O devices...
  - Disks
  - Clocks
  - Keyboards
  - Displays

Magnetic Disks

- **Magnetic disks** provide bulk of secondary storage of modern computers
  - Drives 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



Magnetic Disks

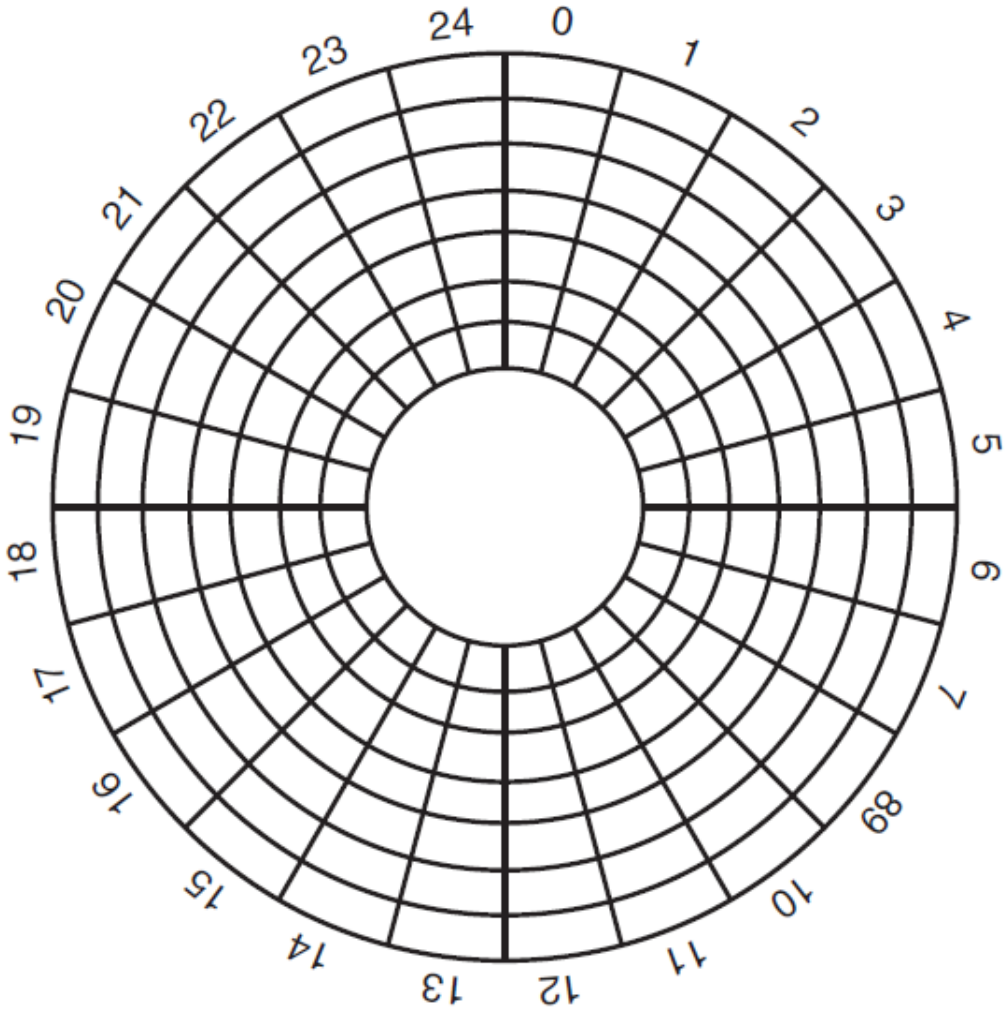
- Disks can be removable
- Drive attached to computer via **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fibre Channel**, **SCSI, SAS, Firewire**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

Solid-State Disks

- 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

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

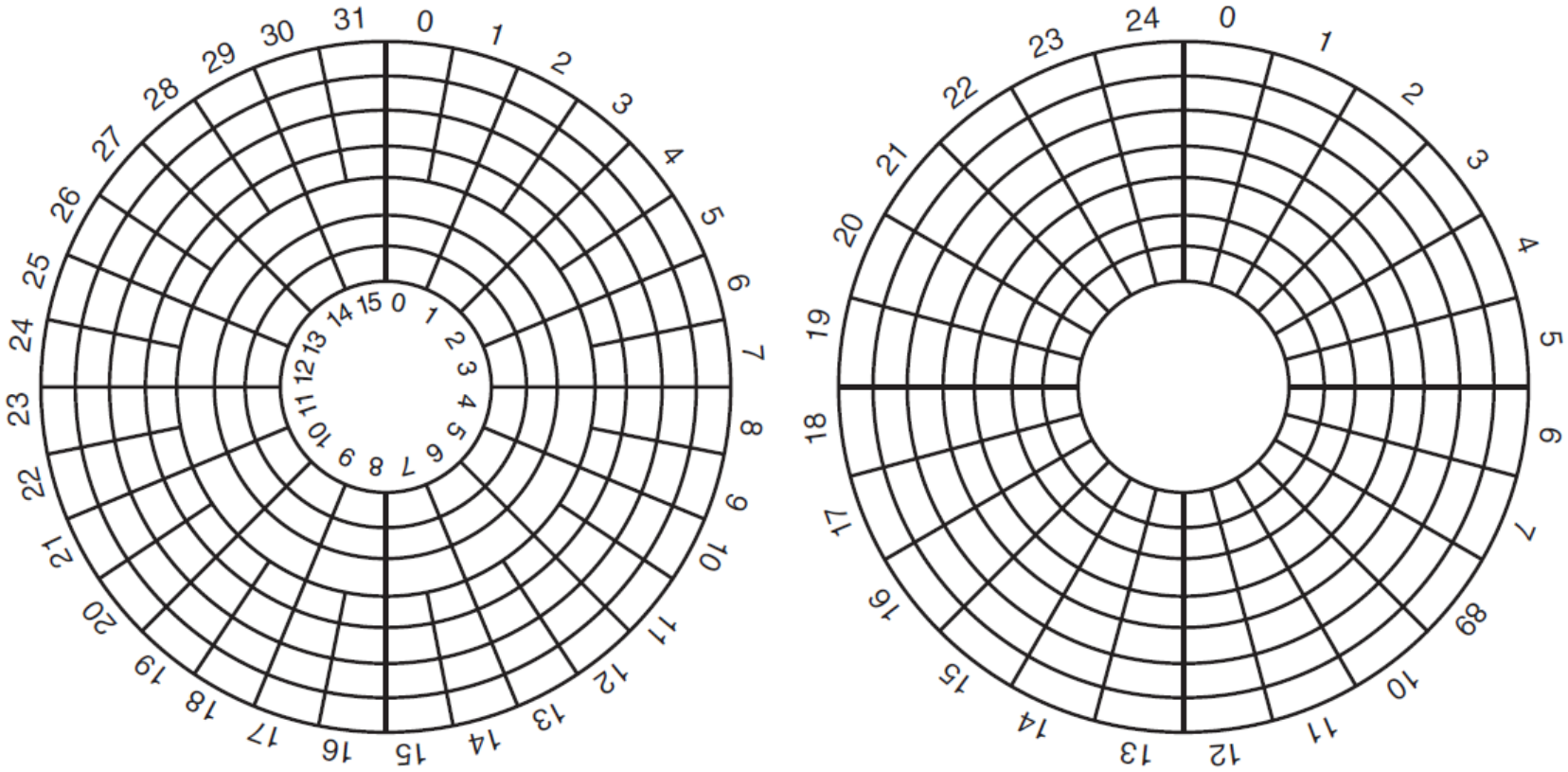


Disk Structure

- 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

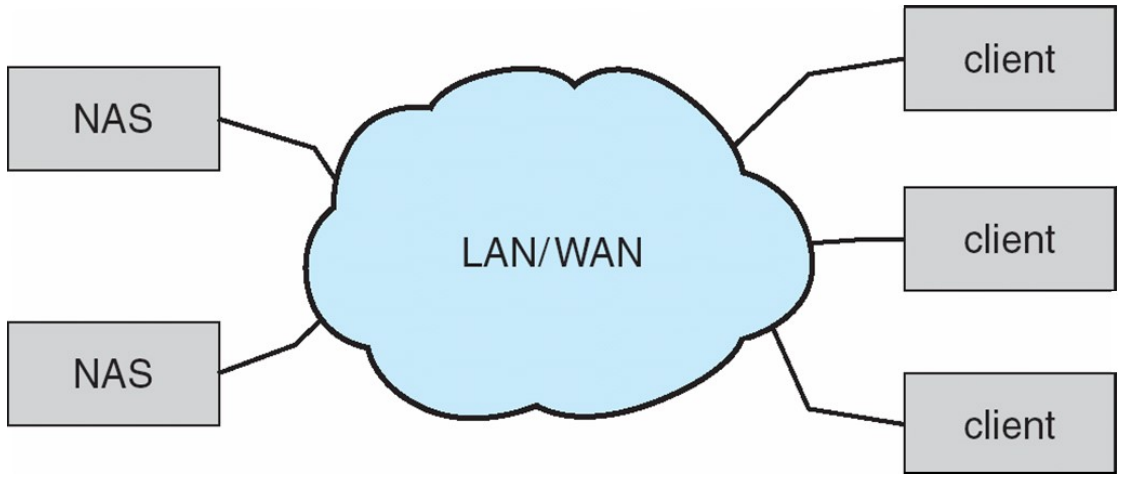
Modern disks with multiple zones

(a) Physical geometry of a disk with two zones. (b) A possible virtual geometry for this disk.



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)

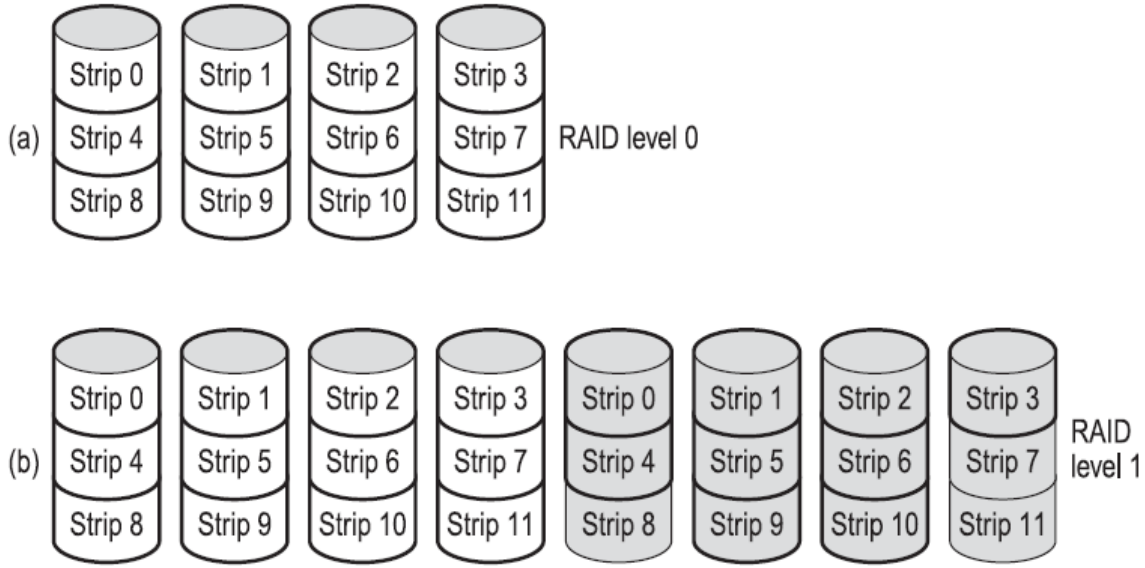


RAID

- RAID – redundant array of inexpensive disks
  - multiple disk drives provides reliability via **redundancy**
  - used to improve disk performance, reliability, or both
- Increases the **mean time to failure**
  - **Mean time to repair –** exposure time when another failure could cause data loss
  - **Mean time to data loss** based on above factors
- If mirrored disks fail independently, consider disk with 1300,000 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!

RAID

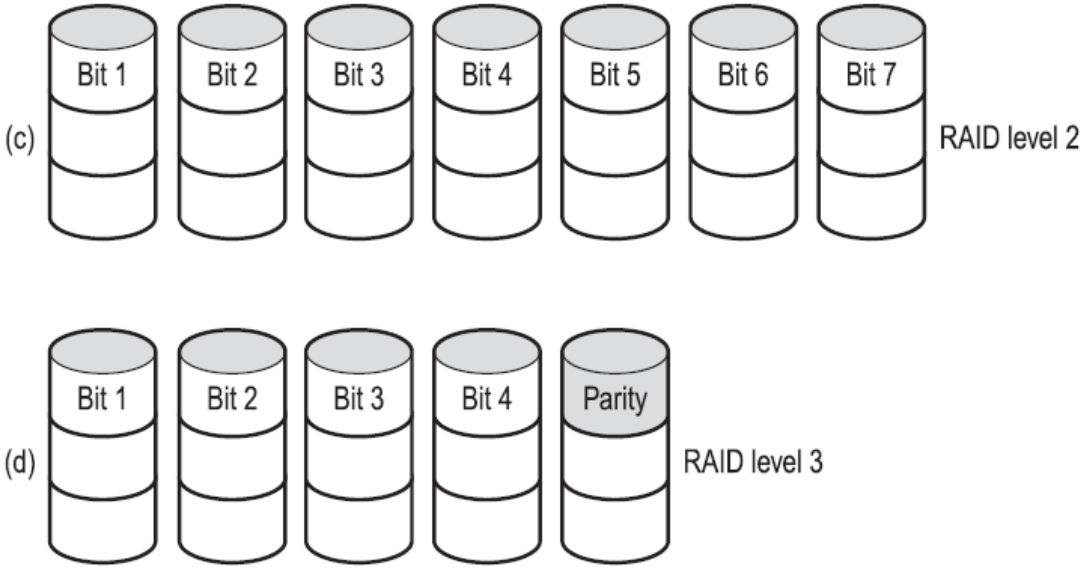
- RAID level 0 – striping
  - Data is split up into blocks that get written across all the drives in the array.
  - Offers great performance, both in read and write operations.
  - Not fault-tolerant. If one drive fails, all data in the RAID 0 array are lost.
- RAID level 1 – mirroring
  - Data is stored twice by writing them to both the data and mirror drives.
  - If one drive fails, data can just be copied to the replacement drive.
  - Effective storage capacity is only half of the total drive capacity.



https://www.youtube.com/watch?v=wTcxRObq738

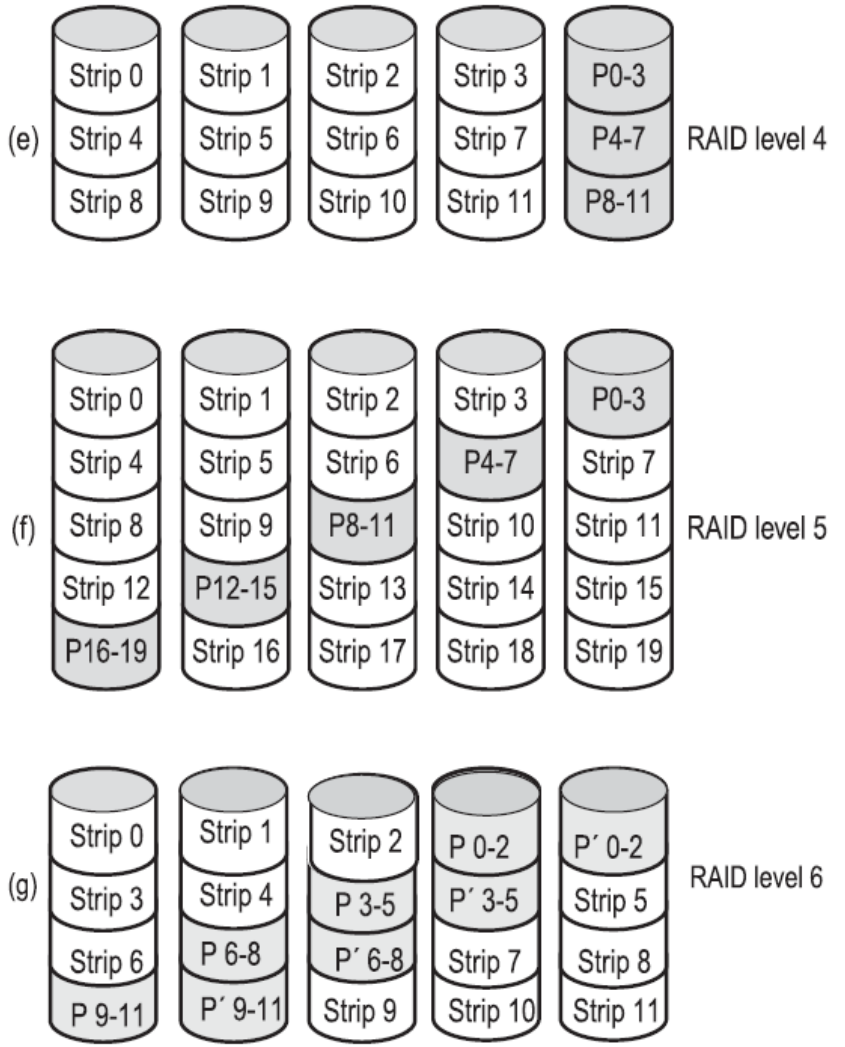
RAID

- RAID level 2
  - Data is striped as sequential bits on different drives.
  - Hamming-code parity is calculated across corresponding bits and stored on at least one parity drive.
  - https://www.youtube.com/watch?v=1A_NcXxdoCc
  - https://www.youtube.com/watch?v=wbH2VxzmoZk
- RAID level 3
  - Here a single parity bit is computed for each data word and written to a parity drive.
  - RAID 2 and RAID 3 are rarely used in practice.
  - https://www.youtube.com/watch?v=DdMcAUlxh1M



RAID

- RAID level 4 – striping with dedicated parity
  - Consists of block-level striping with a dedicated parity disk.
- RAID level 5 – striping with distributed parity
  - Consists of block-level striping with parity information is distributed among the drives.
  - Most common secure RAID level used.
  - If a drive fails, you still have access to all data, even while the failed drive is being replaced and the storage controller rebuilds the data on the new drive.
- RAID level 6 – striping with double parity
  - 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.
- https://www.youtube.com/watch?v=1P8ZecG9iOI

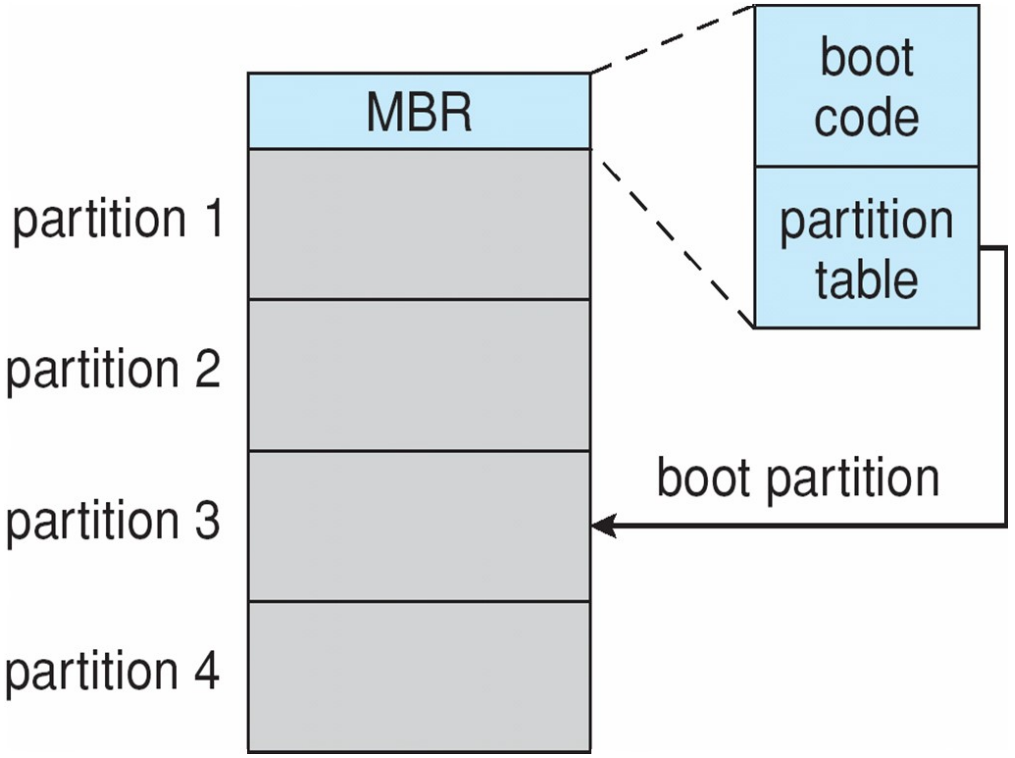


Disk Formatting

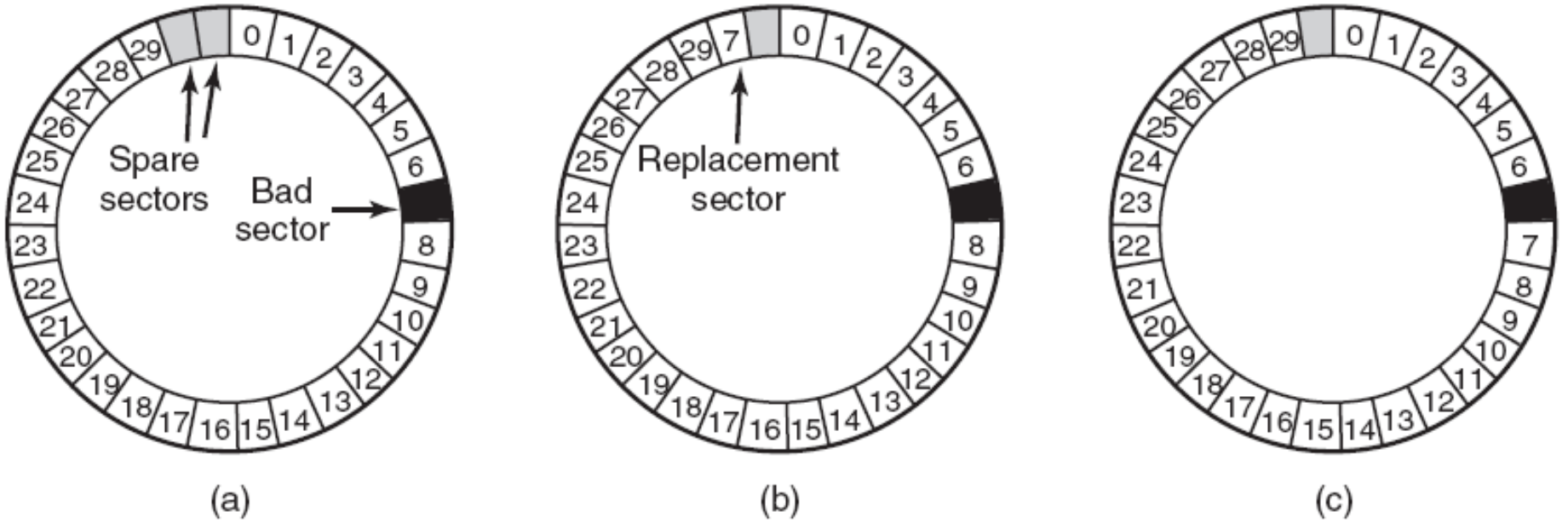
- After manufacturing, there is no information on the disk.
- Before the disk can be used, each platter must receive a **low-level format** done by software.
- The format consists of a series of concentric tracks, each containing some number of sectors, with short gaps between the sectors.

Disk Formatting

- 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”
- Boot block initializes system
  - **Bootstrap loader** program stored in boot blocks of boot partition
  - **MBR** (**Master Boot Record**) contains some boot code and the partition table at the end.



Error Handling



(a) A disk track with a bad sector.

(b) Substituting a spare for the bad sector.

(c) Shifting all the sectors to bypass the bad one.

Disk Scheduling Algorithms

- The time required to read or write a disk block is determined by:
  1. Seek time (the time to move the arm to the proper cylinder).
  2. Rotational delay (how long for the proper sector to appear under the reading head).
  3. Actual data transfer time.
- The goal of disk scheduling is:
  - 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 Algorithms

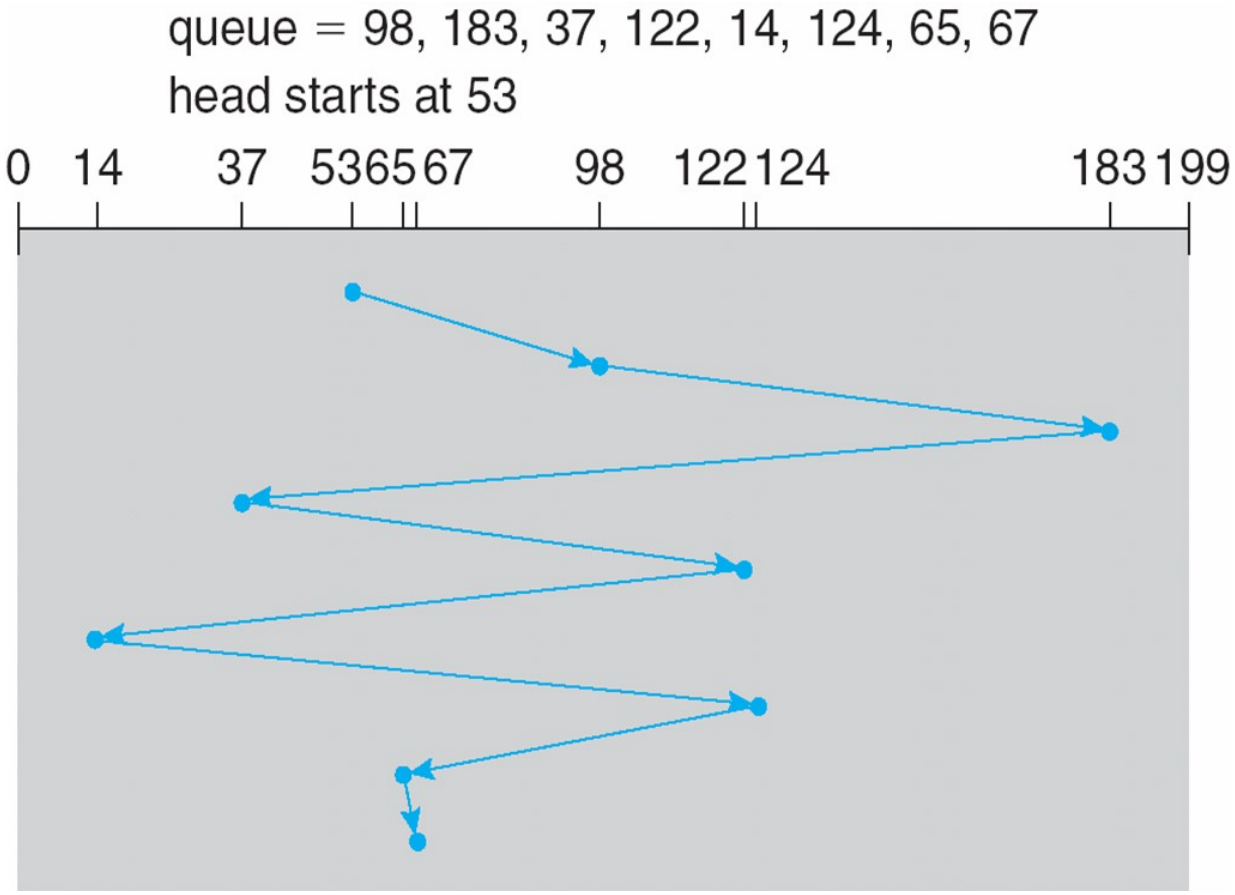
- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- OS maintains queue of requests, per disk or device
- We illustrate scheduling algorithms with a request queue (0-199)

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

  Head pointer 53

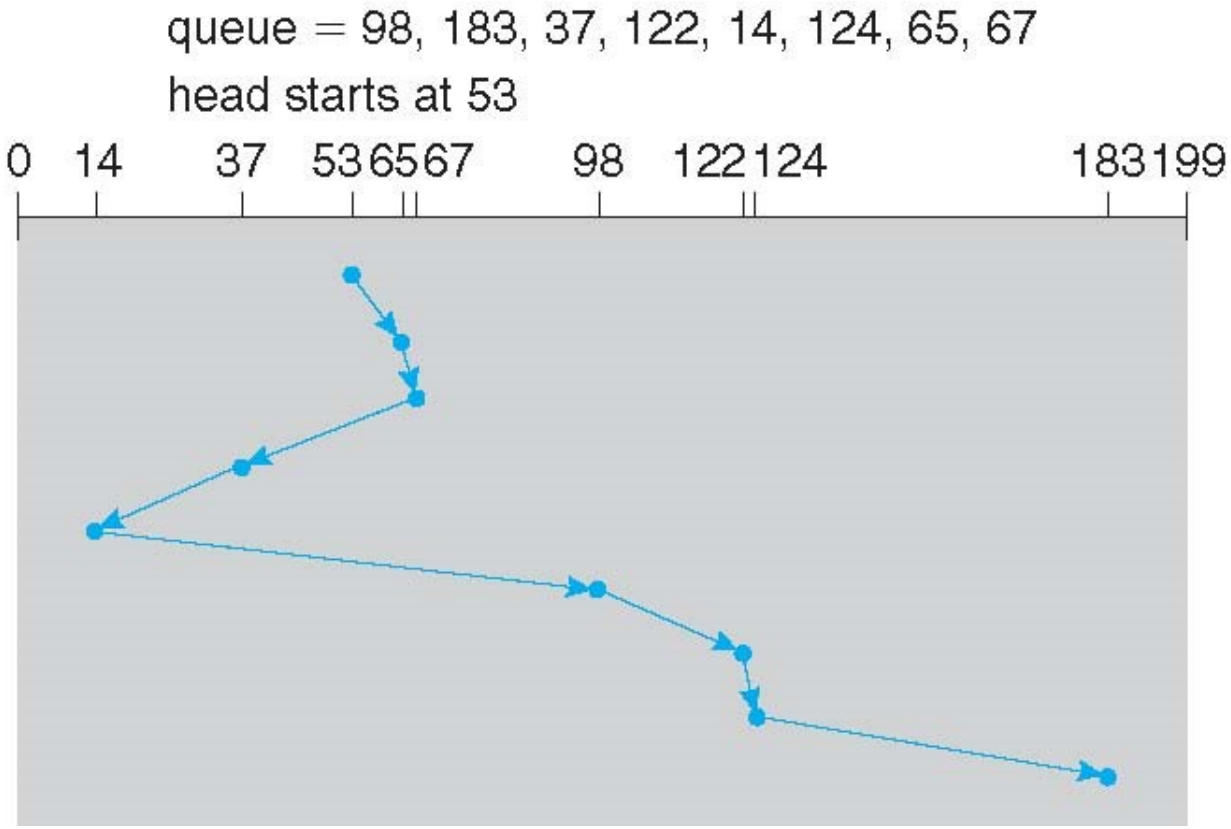
FCFS (First-Come, First-Served)

- Illustration shows total head movement of 640 cylinders



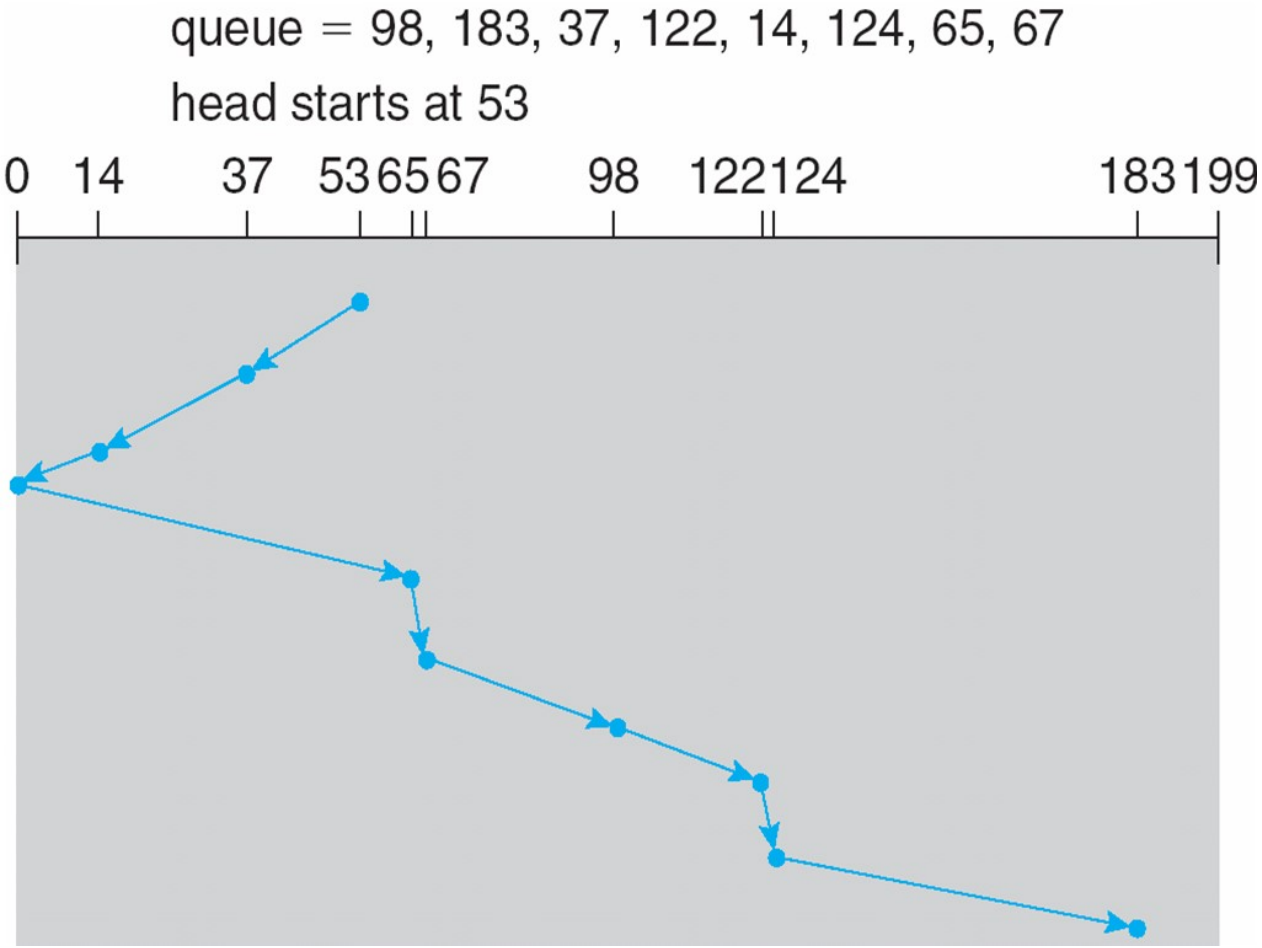
SSF (Shortest Seek First)

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



Elevator algorithm

- 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.
- Illustration shows total head movement of 236 cylinders.

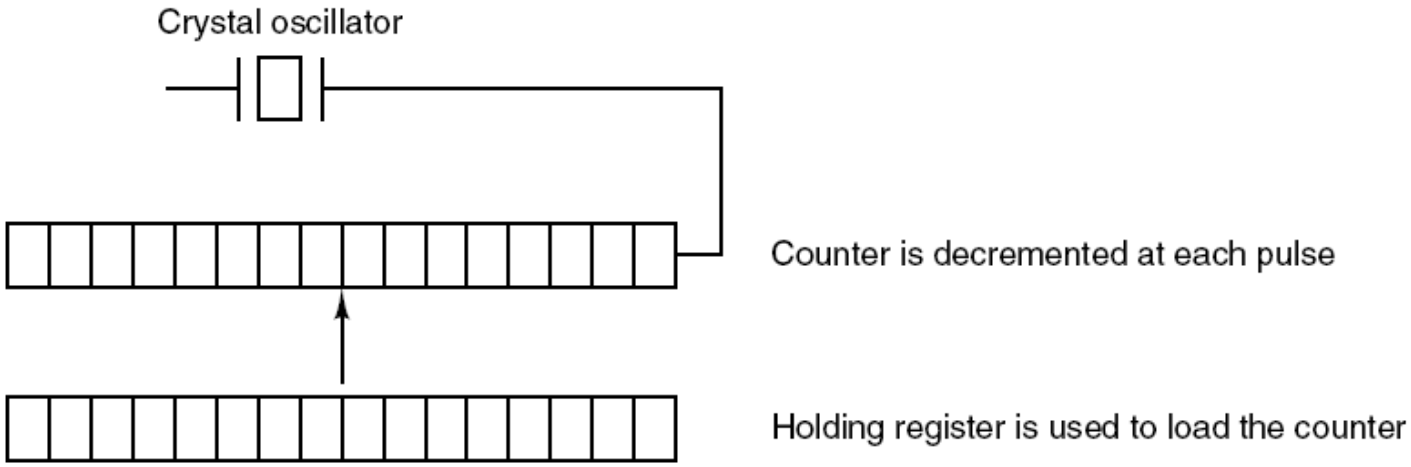


Clocks

- **Clocks** (also called **timers**) are essential to the operation of any multiprogrammed system for a variety of reasons.
- They maintain the time of day and prevent one process from monopolizing the CPU, among other things.
- The clock software can take the form of a device driver, even though a clock is neither a block device, like a disk, nor a character device, like a mouse.

Clock Hardware

- Two types of clocks are commonly used in computers, and both are quite different from the clocks and watches used by people.
  - The simpler clocks are tied to the 110- or 220-volt power line and cause an interrupt on every voltage cycle, at 50 or 60 Hz. These clocks used to dominate, but are rare nowadays.
  - The other kind of clock is built out of three components: a crystal oscillator, a counter, and a holding register.



Clock Hardware

- The advantage of the programmable clock is that its interrupt frequency can be controlled by software.
- To prevent the current time from being lost when the computer's power is turned off, most computers have a battery-powered backup clock, implemented with the kind of low-power circuitry used in digital watches.
- There is also a standard way for a networked system to get the current time from a remote host.

Clock Software

- Typical duties of a clock driver
  1. Maintaining the time of day.
  2. Preventing processes from running longer than they are allowed to.
  3. Accounting for CPU usage.
  4. Handling alarm system call made by user processes.
  5. Providing watchdog timers for parts of the system itself.
  6. Doing profiling, monitoring, statistics gathering.

Keyboard Software

- User input comes primarily from the keyboard and mouse (or somtimes touch screens).
- An interrupt is generated whenever a key is struck and a second one is generated whenever a key is released.
- At each of these keyboard interrupts, the keyboard driver extracts the information about what happens from the I/O port associated with the keyboard.
- Everything else happens in software and is pretty much independent of the hardware.

The X Window System

- Most UNIX systems use the X Window System as the basis of the user interface.
- It consists of programs that are bound to special libraries that issue drawing commands and an X server that writes on the display.
- It runs on a client-server model, which can potentially run on different computers.
- On modern personal computers, both parts can run on the same machine.
- On Linux systems, the popular Gnome and KDE desktop environments run on top of X.

Graphical User Interfaces

- Many personal computers use GUIs for their output. These are based on the WIMP paradigm: windows, icons, menus, and a pointing device.
- GUI-based programs are generally event driven, with keyboard, mouse, and other events being sent to the program for processing as soon as they happen.
- In UNIX systems, the GUIs almost always run on top of X.

Power Management

- Power management is a major issue for phones, tablets, and notebooks because battery lifetimes are limited and for desktop and server machines because of an organization's energy bills.
- The operating system plays a key role in energy management.
- It controls all the devices, so it must decide what to shut down and when to shut it down.
- Programs can also help out by sacrificing some quality for longer battery lifetimes.

Thin Clients

- Thin clients have some advantages over standard PCs, notably simplicity and less maintenance for users.
- With a centralized system, only one or a few machines have to be updated and those machines have a staff of experts to do the work.
- We are starting to see a shift from PC-centric computing to Web-centric computing.
  - Web email, web applications, etc.