

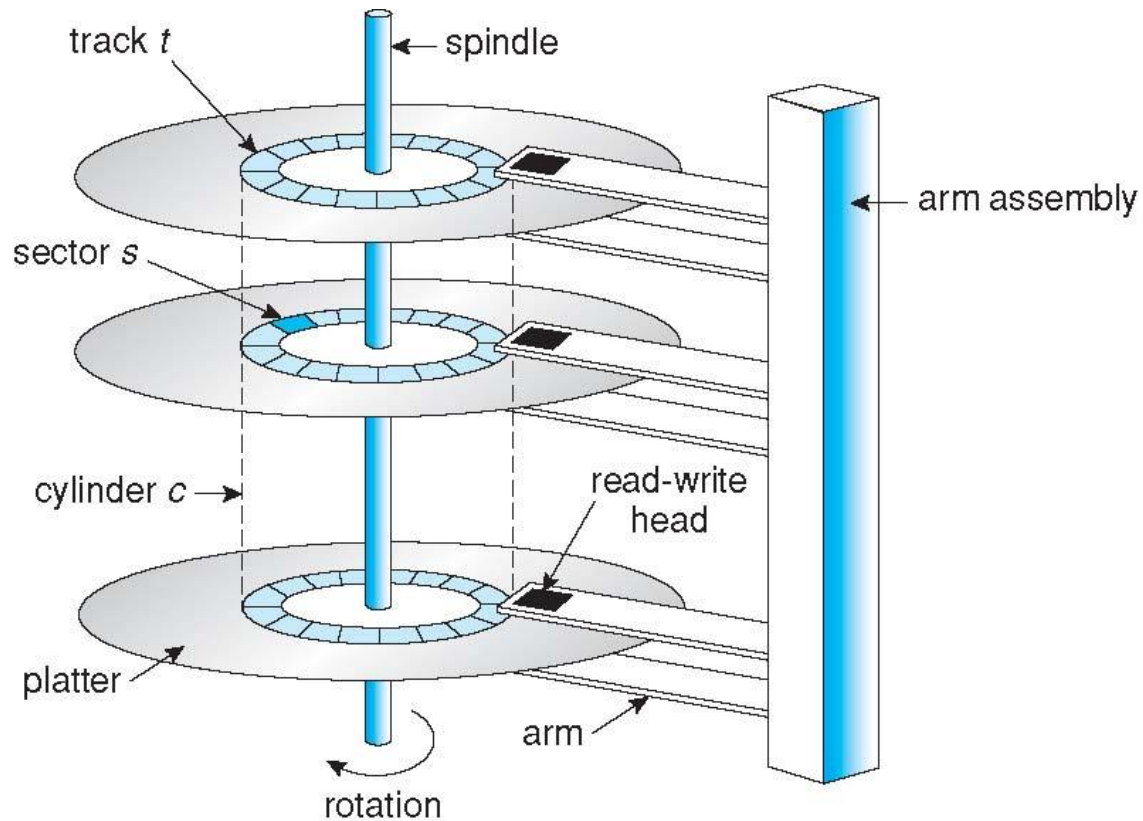


DISK MANAGEMENT

Topics

- ◉ Disk Structure
- ◉ Disk Scheduling
- ◉ Disk Management
- ◉ Swap-Space Management

Magnetic Head disk Mechanism



Magnetic Disk

- Disk speed – 2 parts – transfer rate + positioning time
- Transfer time – data flow between drive and main memory
- Seek Time – move the disk arm to the desired cylinder
- Latency Time – for the desired sector to rotate to disk head

Disk Structure

- ◉ Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer
- ◉ 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

Disk Structure

- 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 is not constant

Disk Structure

- Use **constant linear velocity** (CLV) , the density of bits per track is uniform
- From outer zone to inner – no of sectors decrease – drive increases its rotation of data to keep data rate constant
- Eg CD /DVD ROM
- If the speed remains constant – density of bits decrease from inner track to outer to keep the data rate constant
- Used in hard disks called constant **angular velocity** (CAV)

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 \approx 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

- I/O request includes
 - Is this operation input or output?
 - what disk address?
 - Which memory address?
 - number of sectors to transfer?
- OS maintains queue of requests, per disk or device
- When one request is completed, the OS must chose which pending request to service next
- It uses Disk scheduling algorithm to do the task

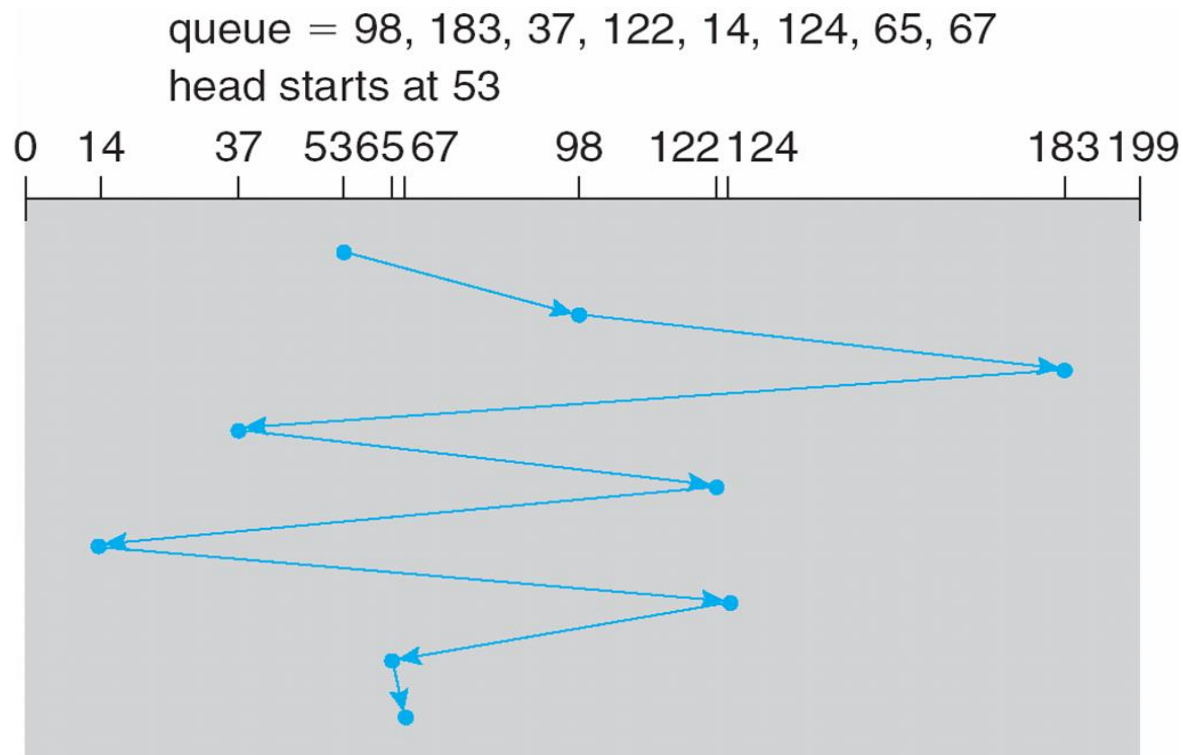
Disk Scheduling

- We illustrate scheduling algorithms with a request queue (0-199)

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

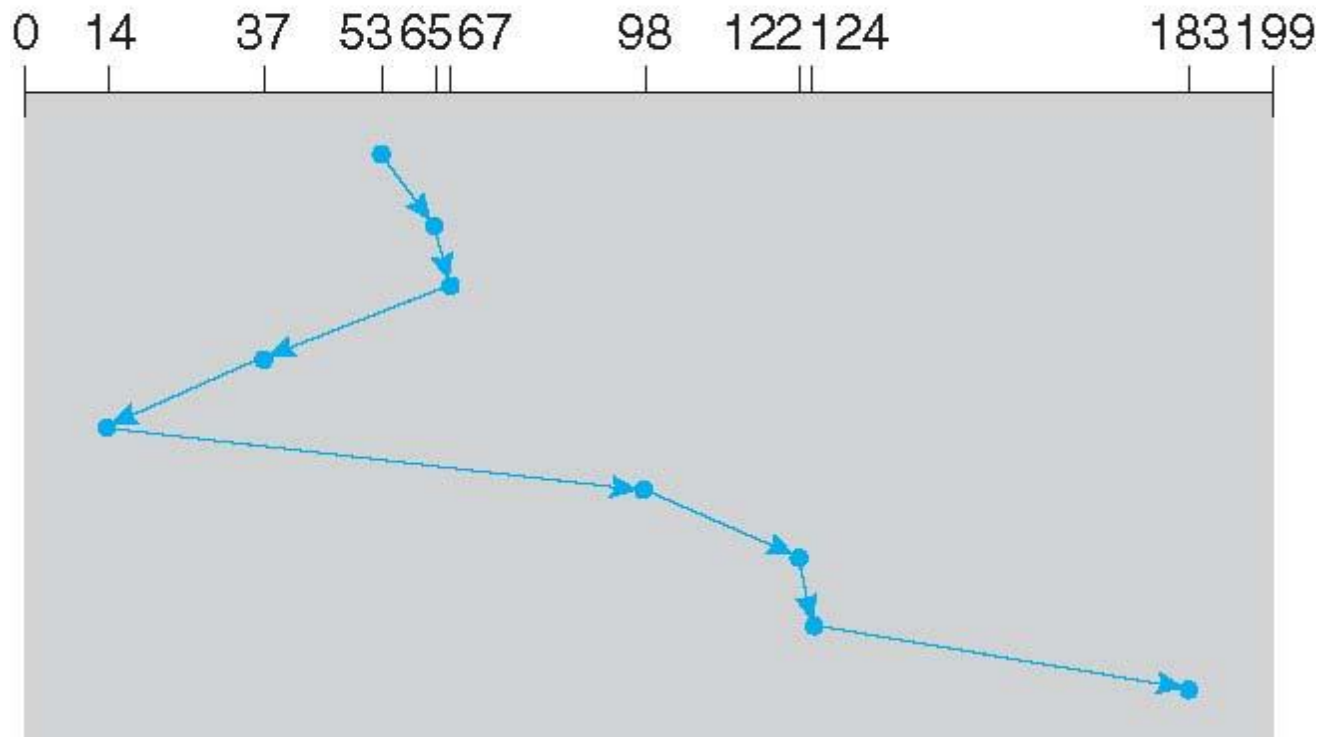
FCFS

- Illustration shows total head movement of 640 cylinders



SSTF(Shortest Seek time First)

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



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

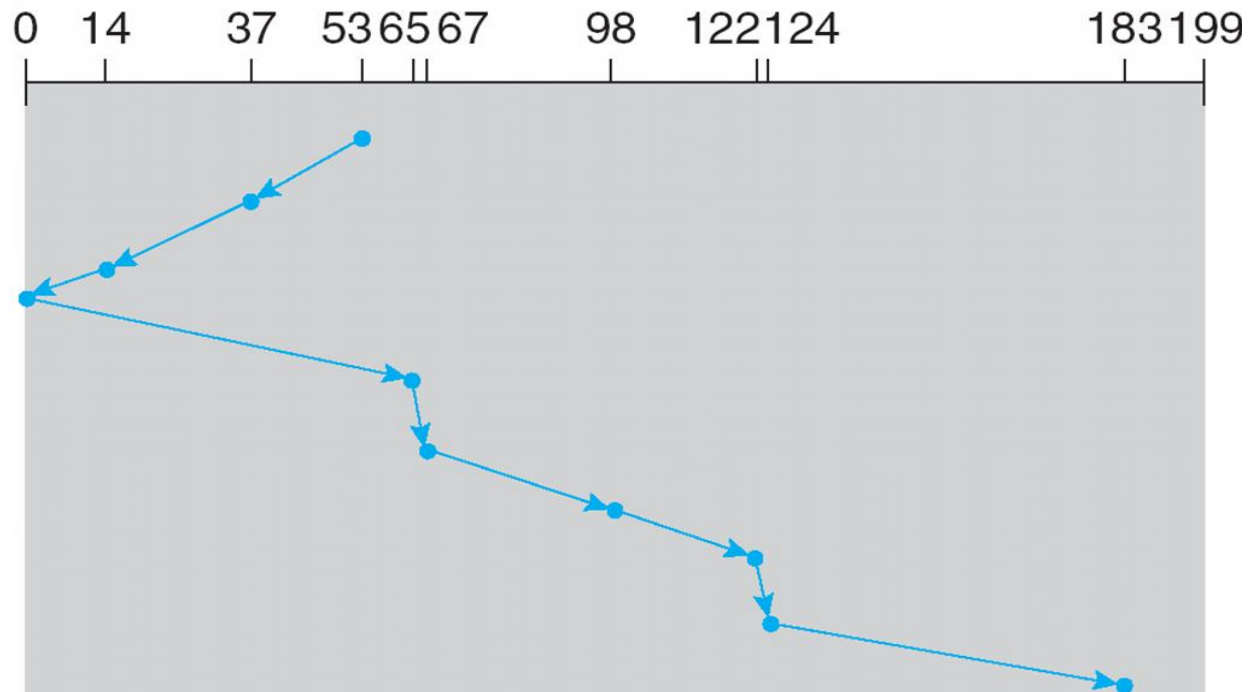
- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- **SCAN algorithm** Sometimes called the **elevator algorithm**
- But note that if requests are uniformly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

SCAN

- Illustration shows total head movement of 208 cylinders

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

head starts at 53



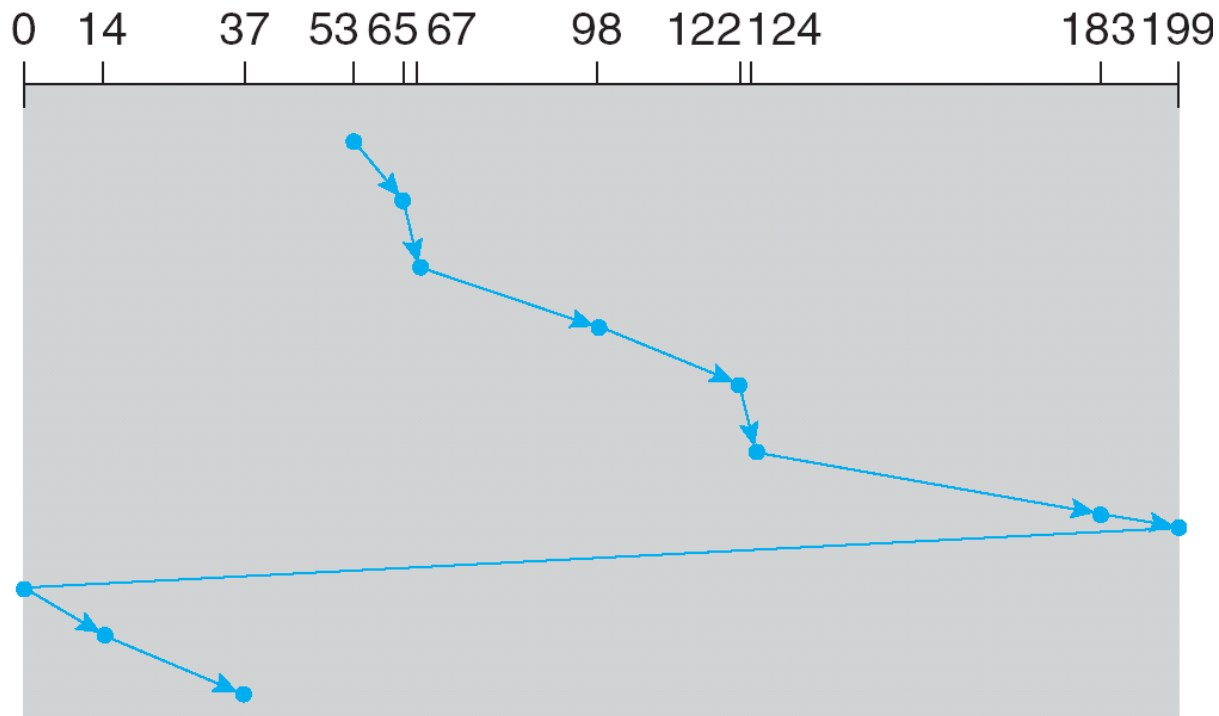
C-SCAN

- 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

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

head starts at 53



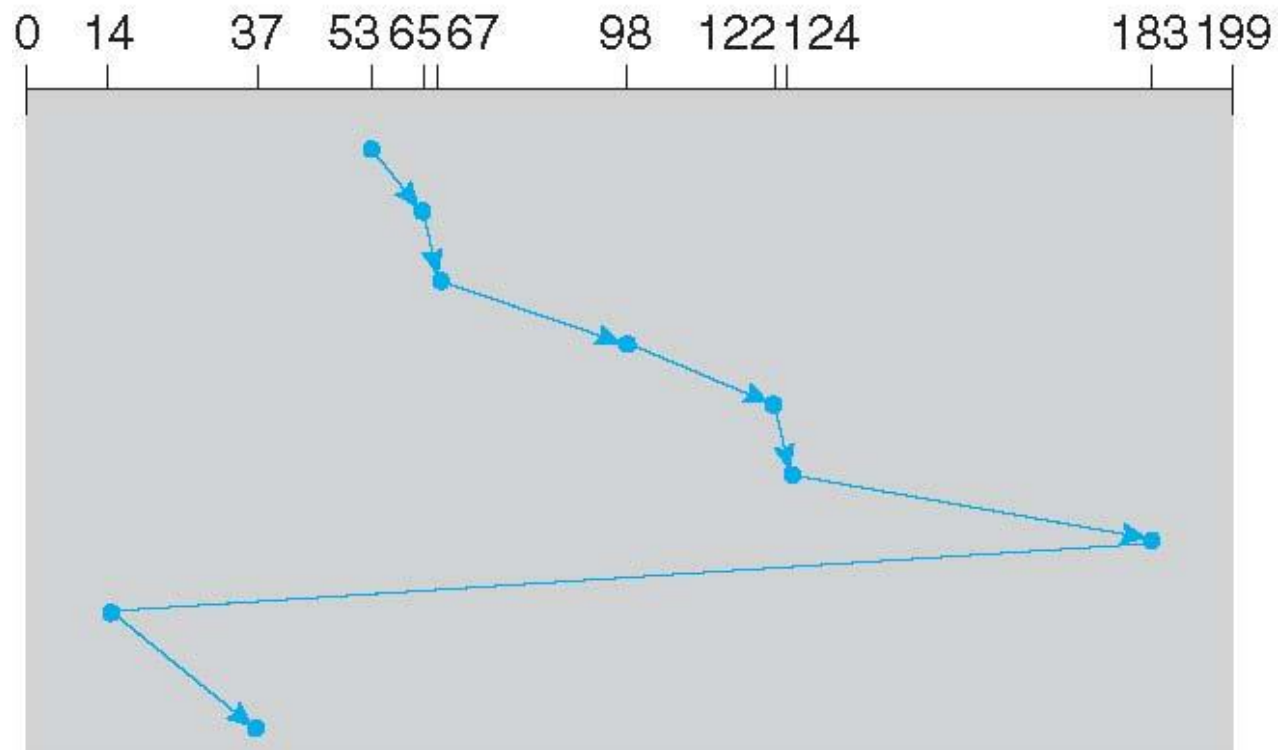
C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-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
- Total number of cylinders?

C-LOOK

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
 - sequential verses
 - linked or indexed file(blocks are widely scattered – greater head movement

Selecting a Disk-Scheduling Algorithm

- Location of directories and index block is important
- Opening a file -> Searching directory structure. Should be placed near files or center cylinder
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm

Selecting a Disk-Scheduling Algorithm

- What about rotational latency?
 - Difficult for OS to calculate
- Modern disk don't disclose the physical location of logical blocks
- Implementing scheduling in controller hardware built into the disk drive
- How does disk-based queuing effect OS queue ordering efforts?
- Demand paging takes priority over application I/O
- Writes are more imp than reads if cache is running out of free pages

Disk Management – Disk Formatting

- **Low-level formatting**, or **physical formatting** — Dividing a disk into sectors that the disk controller can read and write
- A special DS for each sector – header, trailer & data.
- Header & trailer contain info like sector no, error correction code (**ECC**)
- Usually 512 bytes of data but can be selectable
- ECC is matched each time we read data

Disk Management – 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”
- To increase efficiency most file systems group blocks into **clusters**
 - Disk I/O done in blocks
 - File I/O done in clusters

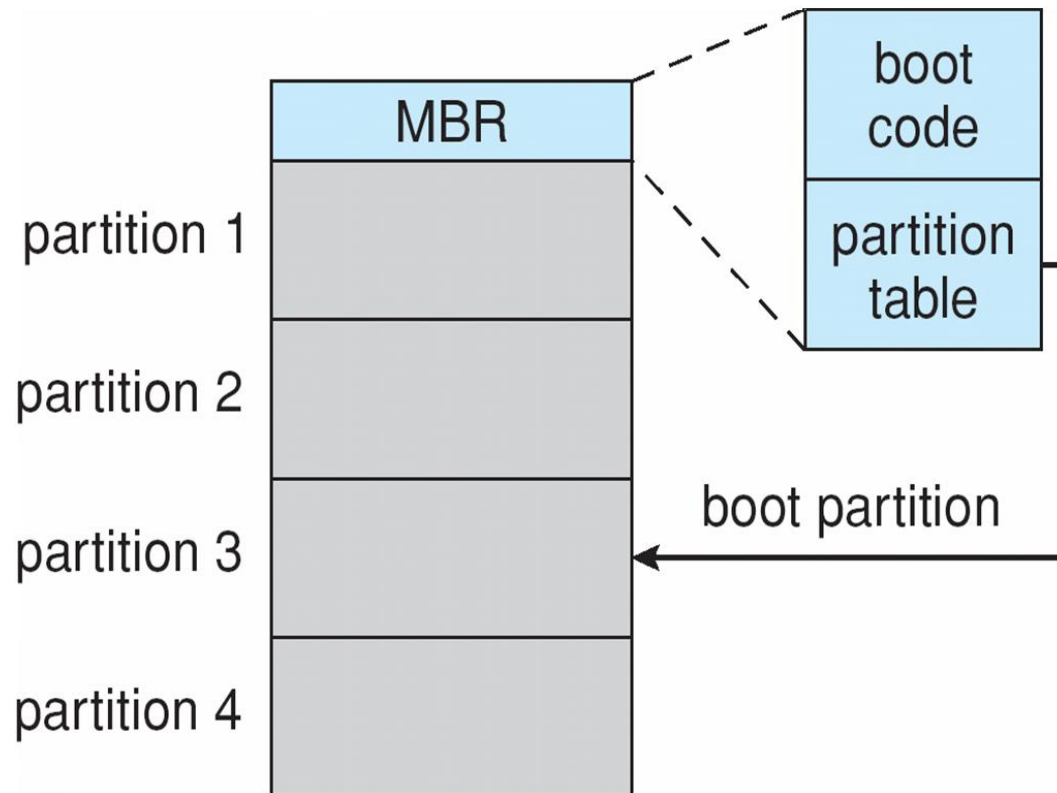
Disk Management – Disk formatting

- Some OS gives special programs to use a disk partition as large sequential array without any file system or DS
- This array – raw disk and I/O is termed as raw I/O
- Some Database prefer raw I/O as it enables them to control exact location
- Raw I/O bypasses file system services, buffer cache, file locking, prefetching, filenames, directories

Disk Management – Boot Block

- Boot block initializes system
- The bootstrap is stored in ROM – but need to change ROM to change OS, cannot be infected by virus
- **Bootstrap loader** program stored in boot ROM whose job is to bring full bootstrap program from the disk stored in boot block at fixed location
- A disk that has boot partition – **boot disk /system disk**
- Instructs the disk controller to load program – device drivers are not loaded.

Booting from a Disk in Windows 2000



Disk Management – Bad Blocks

- Methods such as **sector sparing** used to handle bad blocks
- Most disks are formatted to provide spare sectors in each cylinder and spare cylinder as well
- When a bad block is remapped then controller uses spare sector from same cylinder
- Methods like **sector slipping** is used to handle bad blocks
- Replacement of bad block is not automatic as data is usually lost –soft error(spare/slip) hard error(data is lost)

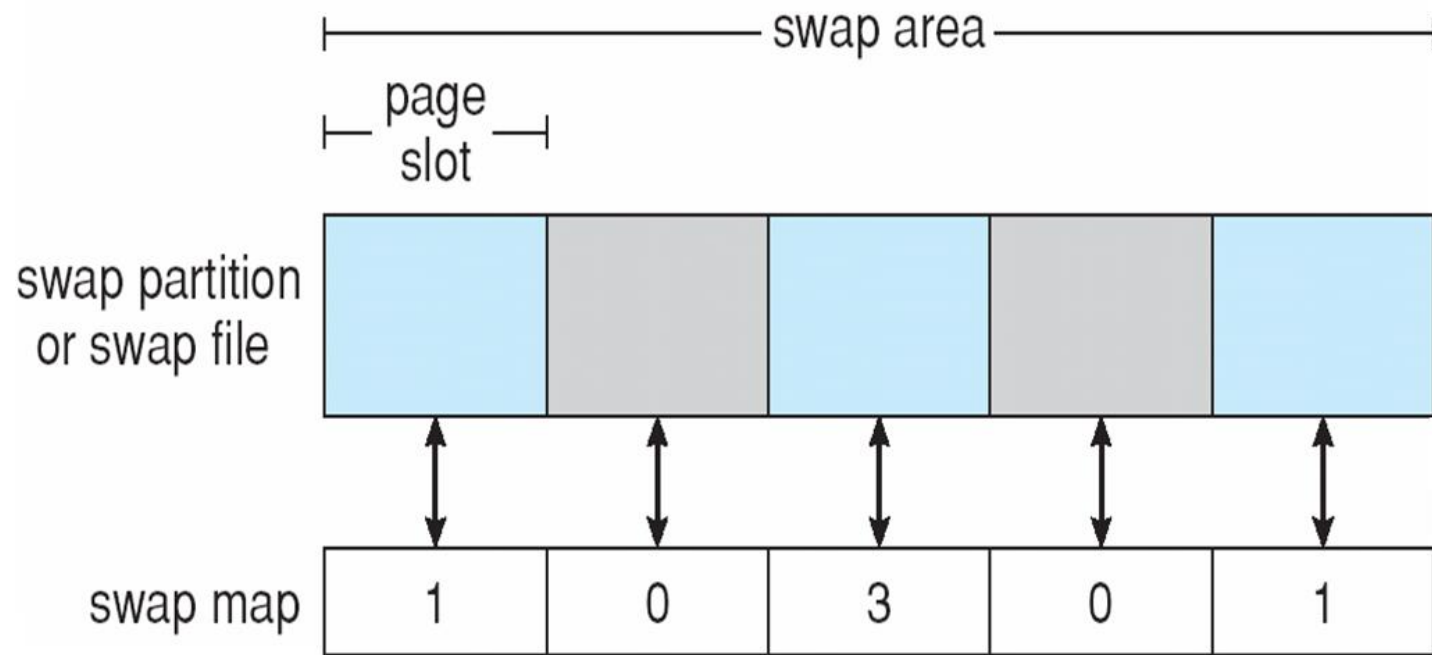
Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
 - Less common now due to memory capacity increase
- 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

- 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

Data Structures for Swapping on Linux Systems



Swap-Space Management

- 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 till write to file system is 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

Swap-Space Management

- A separate swap-space storage manager is used to allocate and de allocate the blocks from raw partition.
- Fragmentation is short lived – more concentration on faster access.
- Swap space is reinitialized at boot time
- Life of data in swap space is shorter