14. Secondary Storage

CS 4352 Operating Systems

Adopted from slides of Drs. Silberschatz, Huang, & Bryant

Secondary Storage

- Hard disk drives (HDDs)

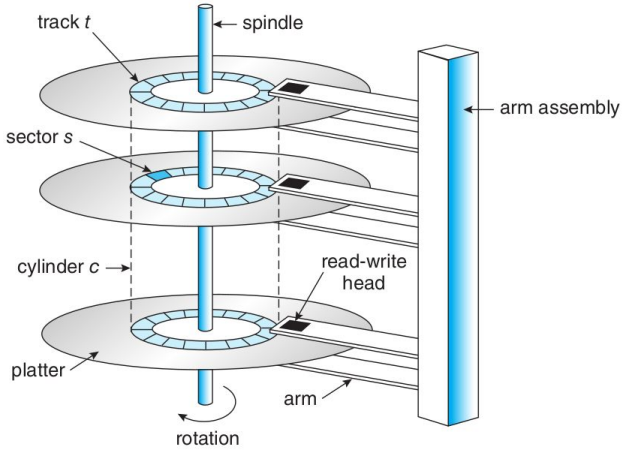
- Nonvolatile memory (NVM) devices

Basic HDD Interface

- HDD interface presents a linear array of sectors
  - Historically 512 Bytes
  - Written atomically

- HDD controller maps logical sector numbers to physical sectors
  - OS doesn’t know logical to physical sector mapping

Hard Disk Drives (HDDs)



- Platter (aluminum coated with a thin magnetic layer)
  - A circular hard surface
  - Data is stored persistently by inducing magnetic changes to it.
  - Each platter has 2 sides, each of which is called a surface.
- Spindle
  - Spindle is connected to a motor that spins the platters around
  - The rate of rotations is measured in RPM (Rotations Per Minute).
    - Typical modern values : 7,200 RPM to 15,000 RPM
- Track
  - Concentric circles of sectors
  - Data is encoded on each surface in a track
  - A single surface contains many thousands and thousands of tracks
- Cylinder
  - A stack of tracks of fixed radius
  - Heads record and sense data along cylinders
  - Generally only one head active at a time

The First Commercial Disk Drive

1956 IBM RAMDAC computer included the IBM Model 350 disk storage system

5M (7 bit) characters

50 x 24” platters

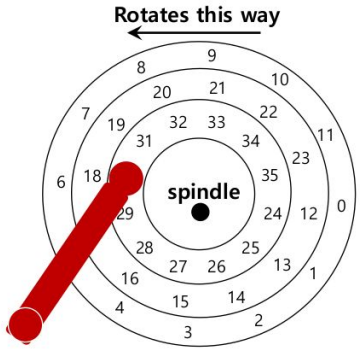
Access time = < 1 second

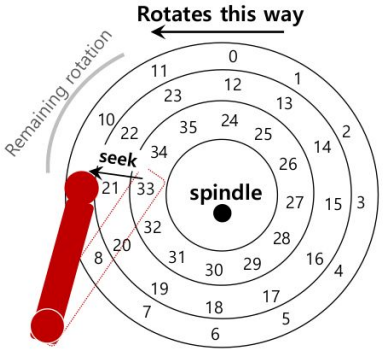
HDD Performance

- HDDs spin platters of magnetically-coated material under moving read-write heads
  - 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
- Average access time = average seek time + average rotational latency
  - For fastest disk, e.g., 3ms + 3ms = 6ms
  - For slow disk, e.g., 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

Multiple Tracks: Seek Time

- Seek: Move the disk arm to the correct track
  - Seek time: Time to move head to the track which contains the desired sector
  - One of the most costly disk operations
- Acceleration -> Coasting -> Deceleration -> Settling
  - Acceleration: The disk arm gets moving
  - Coasting: The arm is moving at full speed
  - Deceleration: The arm slows down
  - Settling: The head is carefully positioned over the correct track
- Seeks often take several milliseconds!
  - Settling alone can take 0.5 to 2ms.
  - Entire seek often takes 4 - 10 ms.





Rotate

- Depends on rotations per minute (RPM)
  - 7200 RPM is common, 15000 RPM is high end
- With 7200 RPM, how long to rotate around?
  - 1 / 7200 RPM = 1 minute / 7200 rotations = 1 second / 120 rotations = 8.34 ms / rotation
- Average rotational latency?
  - 8.34 ms / 2 = 4.17 ms

Transfer

- The final phase of I/O
  - Data is either read from or written to the surface
- Pretty fast — depends on RPM and ector density
  - 1Gbps is typical for maximum transfer rate
- How long to transfer 512-bytes?
  - 512 bytes * 8 / 1Gbps = 4 μs

Example

- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead, the average I/O time =
  - 5ms + rotational latency + 0.1ms + transfer time =
    - Rotational latency = 4.17ms
    - Transfer time = 0.031ms
  - Average I/O time for 4KB block = 9.27ms + 0.031ms = 9.301ms
- So …
  - seeks are slow
  - rotations are slow
  - transfers are fast

HDD Scheduling

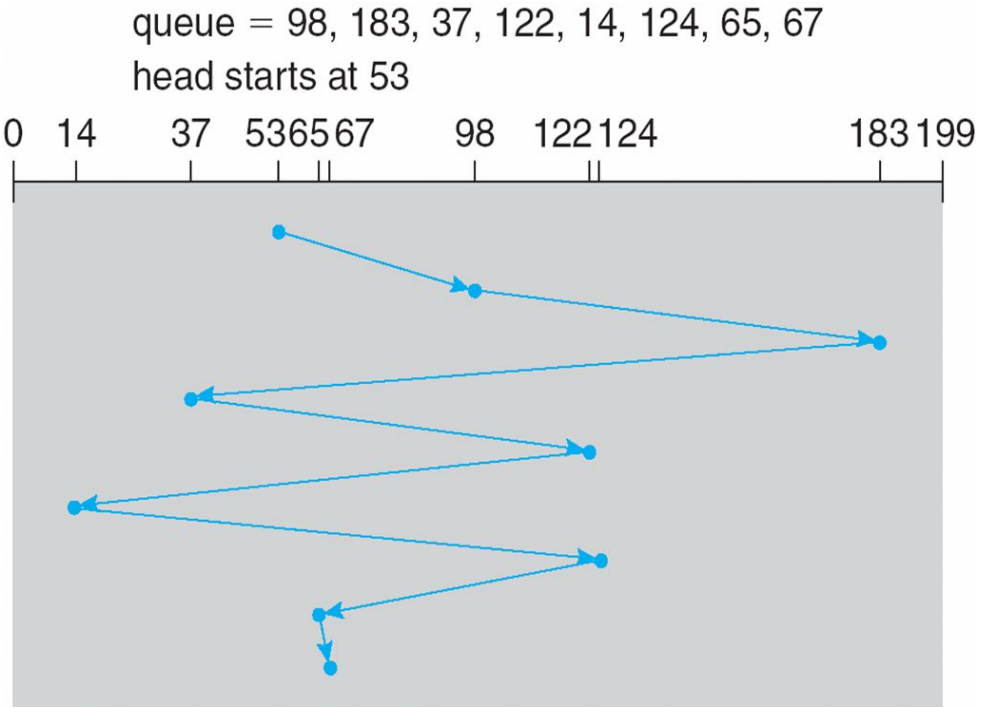
- The OS is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
  - Minimize seek time
    - Seek time is proportional to seek distance
  - Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer
- OS maintains queues of HDD I/O requests
  - In the past, operating system was responsible for HDD head scheduling
  - Now, built into the storage devices, controllers
  - Some of the algorithms they use described next

FCFS (First Come First Served)

- “First Come First Served”
  - Process disk requests in the order they are received
- Advantages
  - Easy to implement
  - Good fairness
- Disadvantages
  - Cannot exploit request locality
  - Increases average latency, decreasing throughput

FCFS Example

- Illustration shows total head movement of 640 cylinders

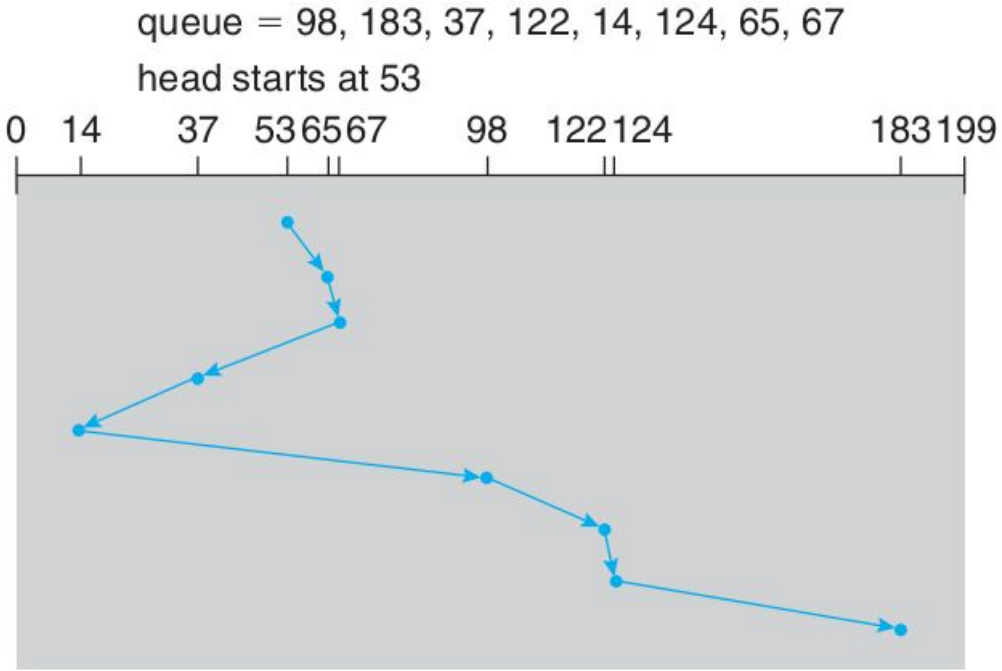


SSTF (Shortest Seek Time First)

- Order the queue of I/O request by track
  - Pick requests on the nearest track to complete first
- Advantages
  - Exploits locality of disk requests
  - Higher throughput
- Disadvantages
  - Starvation
  - Don’t always know what request will be fastest

SSTF Example

- Illustration shows total head movement of 208 cylinders

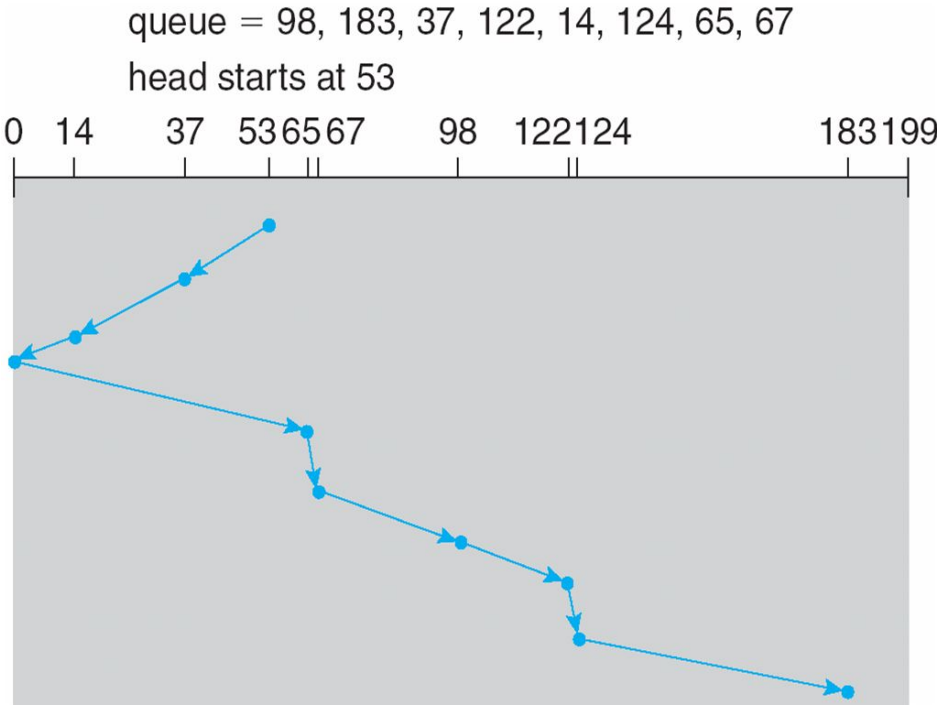


SCAN

- 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
- Advantages
  - Takes advantage of locality
  - Bounded waiting
- Disadvantages
  - Cylinders in the middle get better service
  - Might miss locality SSTF could exploit

SCAN (Cont.)

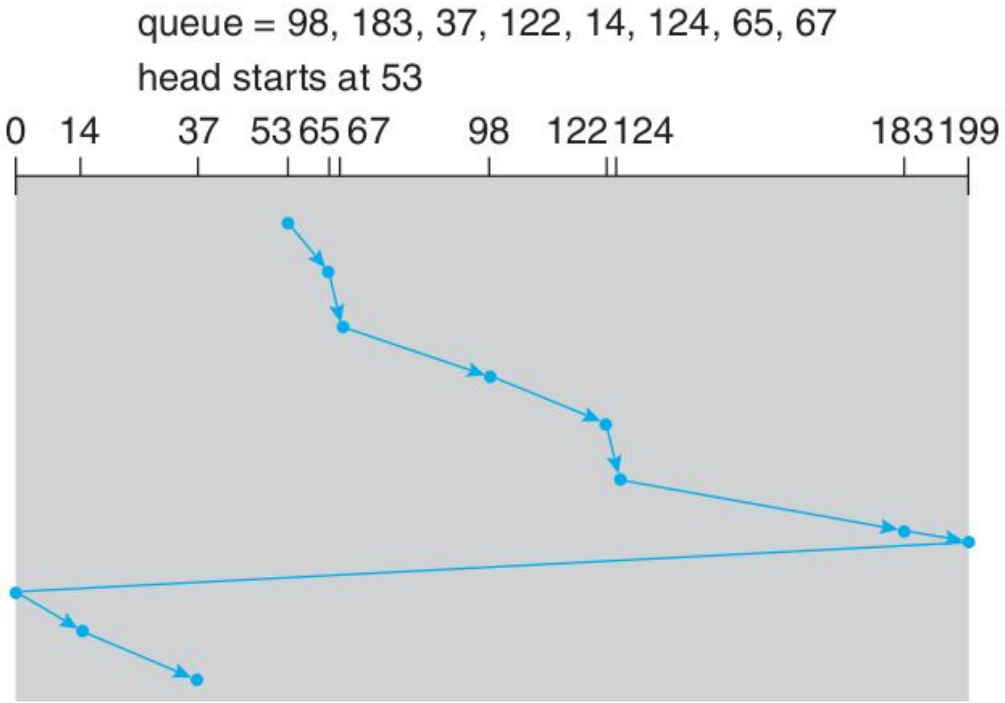
- Illustration shows total head movement of 208 cylinders



C-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
  - Very commonly used algorithm in Unix
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Provides a more uniform wait time than SCAN

C-SCAN Example



Nonvolatile Memory (NVM) Devices

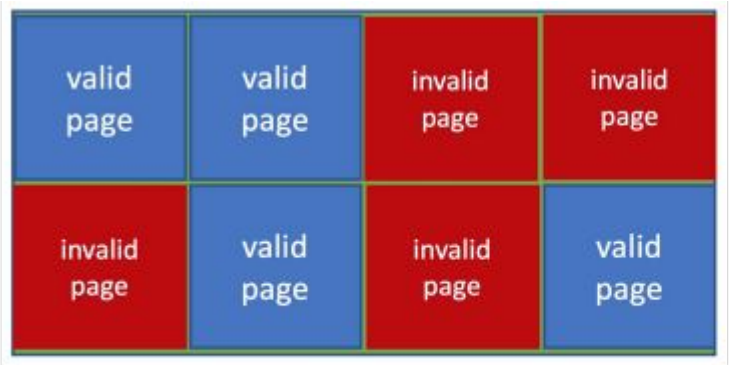
- If disk-drive like, then called solid-state disks (SSDs)
- Other forms include USB drives (thumb drive, flash drive)
- Pros and Cons
  - Can be more reliable than HDDs
  - More expensive per MB
  - Maybe have shorter life span – need careful management
  - Less capacity
  - But much faster
  - No moving parts, so no seek time or rotational latency

Basic SSD Interface

- The smallest unit of an SSD is a page, which is composed of several memory cells, and is usually 4 KB in size
  - Several pages on the SSD are summarized to a block
- A block is the smallest unit of access on a SSD
  - Currently, 128 pages are mostly combined into one block; therefore, a block contains 512 KB
- Read and written in “page” increments but can’t overwrite in place
  - Must first be erased, and erases happen in larger ”block” increments
  - Can only be erased a limited number of times before worn out – ~ 100,000
  - Life span measured in drive writes per day (DWPD)
    - A 1TB NAND drive with rating of 5DWPD is expected to have 5TB per day written within warranty period without failing

NAND Flash Controller Algorithms

- With no overwrite, blocks end up with mix of valid and invalid data
- To track which pages are valid, controller maintains flash translation layer (FTL) table
  - Also implements garbage collection to keep track of invalid page space
  - Allocates overprovisioning to provide working space for GC
- Each cell has lifespan, so wear leveling needed to write equally to all cells



NAND block with valid and invalid pages

NVM Scheduling

- No disk heads or rotational latency but still room for optimization
- In RHEL 7 NOOP (simple FIFO) is used but requests of adjacent addresses are combined
  - NVM best at random I/O, HDD at sequential
  - Input/Output operations per second (IOPS) much higher with NVM (hundreds of thousands vs hundreds)
  - But write amplification (one write, causing garbage collection and many read/writes) can decrease the performance advantage

Drive Attachment

- Host-attached storage accessed through I/O ports talking to I/O busses
  - Advanced technology attachment (ATA), serial ATA (SATA), eSATA, serial attached SCSI (SAS), universal serial bus (USB), and fibre channel (FC).
  - Most common is SATA
  - Because NVM much faster than HDD, new fast interface for NVM called NVM express (NVMe), connecting directly to PCI bus
- Data transfers on a bus carried out by special electronic processors called controllers (or host-bus adapters, HBAs)
  - Host controller on the computer end of the bus, device controller on device end
  - Computer places command on host controller, using memory-mapped I/O ports
  - Host controller sends messages to device controller
  - Data transferred via DMA between device and computer DRAM

Error Detection and Correction

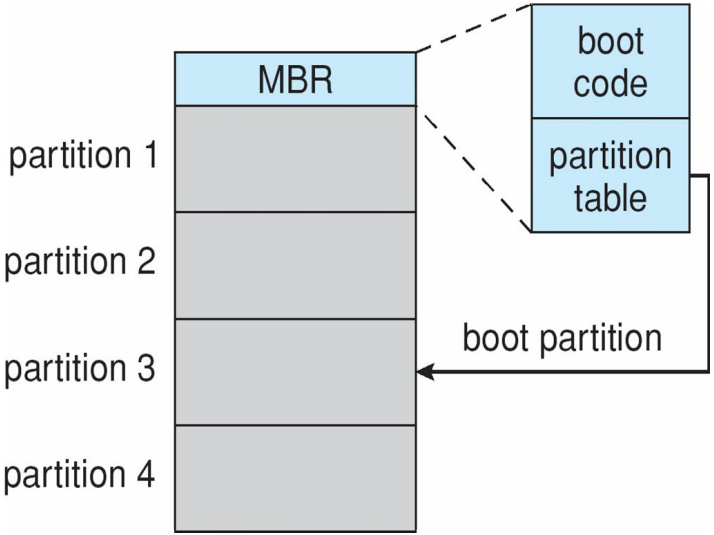
- Needed for many parts of computing (memory, networking, storage)
- Error detection determines if there a problem has occurred (for example a bit flipping)
  - If detected, can halt the operation
  - Detection frequently done via parity bit
  - Another error-detection method common in networking is cyclic redundancy check (CRC) which uses hash function to detect multiple-bit errors
- Error-correction code (ECC) not only detects, but can correct some errors
  - Soft errors correctable, hard errors detected but not corrected

Storage Device Management

- 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 logical disks
  - Logical formatting or “making a **file system**”
  - Root partition has an OS, other partitions can hold other Oses, other file systems, or be raw
    - Mounted at boot time
    - Other partitions can mount automatically or manually
- At mount time, file system consistency is checked
  - Is all metadata correct?
    - If not, fix it, try again
    - If yes, add to mount table, allow access

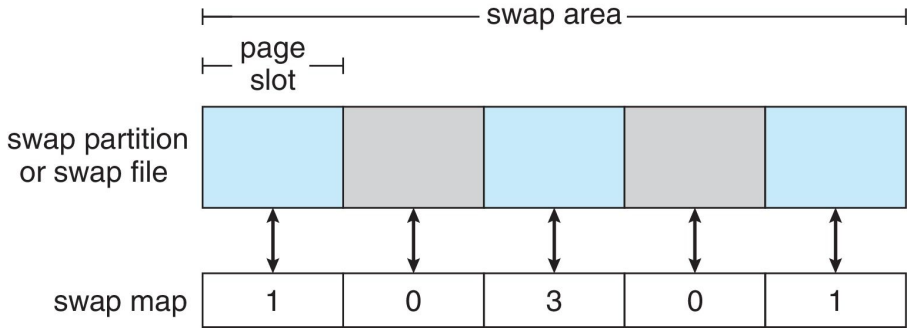
Device Storage Management (Cont.)

- Raw disk access is for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot block initializes system
  - Boot block can point to boot volume or boot loader set of blocks that contain enough code to know how to load the kernel from the file system
- Methods such as sector sparing used to handle bad blocks



Swap-Space Management

- Used for moving entire processes (swapping), or pages (paging), from DRAM to secondary storage when DRAM not large enough for all processes
- Operating system provides swap space management
  - Secondary storage slower than DRAM, so important to optimize performance
  - Usually multiple swap spaces possible – decreasing I/O load on any given device
  - Best to have dedicated devices
  - Can be in raw partition or a file within a file system (for convenience of adding)
- Data structures for swapping on Linux systems:



Homework

Read Chapters 13, 14, and 15

Next Lecture

We start looking at file systems