CMP325
Operating Systems
Lecture 26

Disk Geometry & Partitioning

Muhammad Arif Butt, PhD

Note:
Some slides and/or pictures are adapted from course text book and Lecture slides of

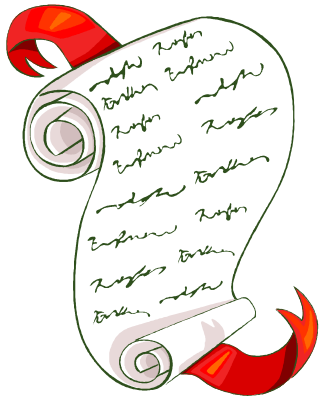
- Dr Syed Mansoor Sarwar
- Dr Kubiatowicz
- Dr P. Bhat
- Dr Hank Levy
- Dr Indranil Gupta

For practical implementation of operating system concepts discussed in these slides, students are advised to watch and practice following video lectures:

OS with Linux:
https://www.youtube.com/playlist?list=PL7B2bn3G_wfBuJ_WtHADcXC44piWLRzr8

System Programming:
https://www.youtube.com/playlist?list=PL7B2bn3G_wfC-mRpG7cxJMnGWdPAQTViW

Today's Agenda

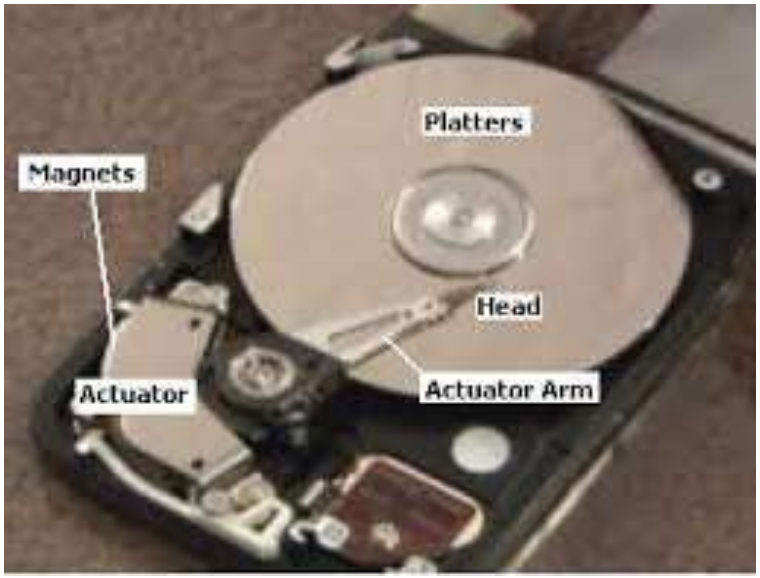


- Overview of Hard Disks
  - HDD
  - SSD
- How a Spinning HD works?
- HDD Address
  - CHS
  - LBA
- Partitioning a Hard Disk
  - Partition Tables
  - Partitioning Tools
- Disk Scheduling Algorithms

Overview of Hard Disks

Secondary Storage: Hard Disk

- A hard disk drive is a sealed unit that a Personal Computer uses for nonvolatile data storage. The two most common types of disks used in PCs these days are:
  - **HDD:** Traditional spinning/magnetic hard disk drive have one or more metallic platters with magnetic coating. The data is stored by generating flux transition. Two read write heads per platter on an arm that can move back and forth read/write data while the platters spin



Secondary Storage: Hard Disk

- **SSD:** A solid state disk is replacing the traditional HDDs. On SSDs, the data is stored on interconnected flash memory chips. These flash memory chips are of different type than is used in USB thumb drives and are faster, more reliable and of course more expensive

Secondary Storage: Hard Disk (cont…)

- **Pros & Cons:**
  - **Price:** SSDs are more expensive
  - **Capacity:** SSDs with 1TB are still expensive and rare
  - **Speed:** A system with SSD boot faster, runs applications faster, transfer files faster
  - **Fragmentation:** No fragmentation issues with SSDs
  - **Durability:** In SSDs, there are no read/write heads flying over the rotating platters just a few nanometers. So if you are rough on your equipment SSd is recommended for you
  - **Noise:** Even the quietest spinning disks emit a bit of noise when the platters spins t 7200 rpm. SSDs make no noise at all, since they are non mechanical

Hard Disk Interfaces

- The interface with which a hard disk is connected with the mother board (south bridge) of the system have evolved over time with attributes like:
  - Serial / Parallel
  - Data transfer rates
  - Cable length support
  - Pins in the connector
  - Number of devices that can be attached with a single port
  - Hot pluggable or not

Hard Disk Interfaces (cont…)

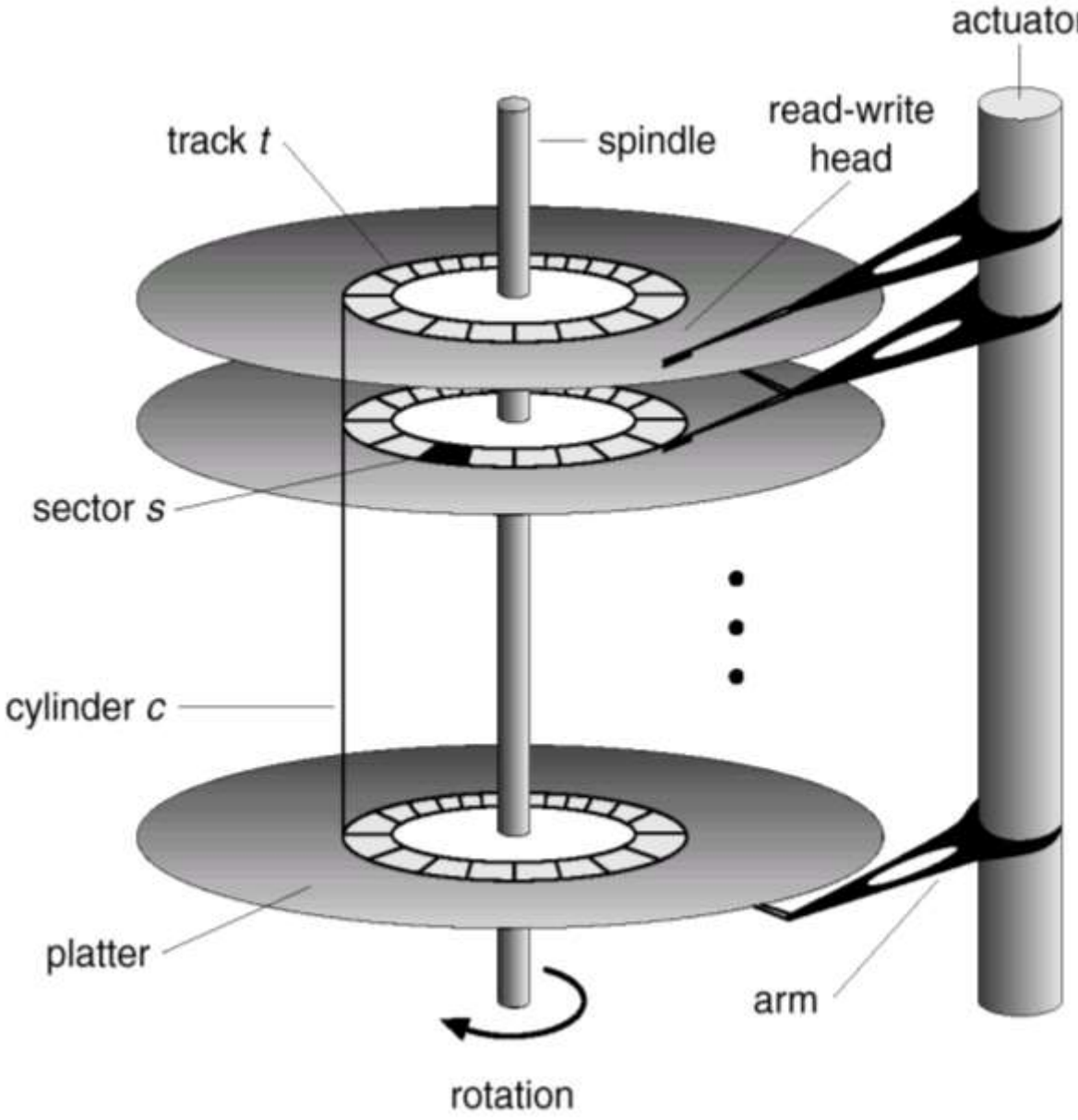
- **PATA:** Parallel Advance Technology Attachment has different versions. More or less all supports a cable length of up to 40 cm
  - PATA/IDE: Integrated Drive Electronics supports one or two disks and have a 16 pin interface with data transfer rate of 8MB/s
  - PATA2/EIDE: Support multi-word DMA modes and LBA
  - PATA/100: Supports data transfer of up to 100MB/sec
- **SATA:** Serial ATA is an advancement on PATA. It uses lesser number of pins (7), thus lesser issues with electromagnetic interference. Data transfer rate start from 150MB/s and go up to 3 Gbps. Cable length up to 1 meter. Hot pluggable
- **SCSI:** Small Computer System Interface has versions like SCSI1, SCSI2, WideSCSI, FastSCSI, UltraSCSI. It uses 8 pins to 68 pins connectors. Data transfer rate start from 4MB/s to 320 MB/s. Cable length up to 6 meter. Hot pluggable. Upto 15 devices can be connected with a single connector
- **SAS:** Serial SCSI is a bit interface to SCSI with data transfer up to 3Gbps

Multiple of Bytes (Hard Disk Sizes)

Decimal		Binary	
Value	Metric	Value	IEC
10^3	kB (kilobyte)	2^{10}	KiB (kibibyte)
10^6	MB (megabyte)	2^{20}	MiB (mebibyte)
10^9	GB (gegabyte)	2^{30}	GiB (gibibyte)
10^{12}	TB (terabyte)	2^{40}	TiB (tebibyte)
10^{15}	PB (petabyte)	2^{50}	PiB (pebibyte)
10^{18}	EB (exabyte)	2^{60}	EiB (exbibyte)
10^{21}	ZB (zetabyte)	2^{70}	ZiB (zebibyte)
10^{24}	YB (yottabyte)	2^{80}	YiB (yobibyte)

How a Spinning Hard Disk Works?

Magnetic Hard Disk Drive



A single stepper motor controls all the heads, so all heads move together.
All heads move together on the same corresponding track on all platters.

Magnetic Hard Disk

All magnetic storage devices read and write data by using electromagnetism. Electromagnetism means that when an electric current flows through a conductor, a magnetic field is generated around the conductor which can then create an influence on a magnetic material. To write, a drive head creates flux reversals on the medium to record data. To read from a medium, the head becomes a flux transition detector. The term flux describes a magnetic field that has a specific direction. A flux reversal or flux transition is a change in the polarity of the aligned magnetic particles on the surface of the storage medium. A bit cell or transition cell is a specific area of the medium in which the drive head creates flux reversal. The drive controller takes the data to be stored and encodes it as a series of flux reversals over a period of time, according to the pattern dictated by the encoding method it uses. Frequency Modulation (FM). Modified Frequency Modulation (MFM). Run Length Limited (RLL)

Magnetic Hard Disk (cont…)

- **Disk Platters:** A HDD contains one or more rigid, disk-shaped platters, usually constructed of aluminum. A blank disk platter have a number of tiny particles each having its own magnetic field, such that they cancel the effect of each other with the net result of no observable field polarity
- **Track:** Concentric set of rings that may exist on both sides of platters (for double sided). Each track is of the same width as the head. There are thousands of tracks per surface
- **Head:** It is the component that is responsible for reading/writing data. By movement of head, the track number change. While accessing a selected block, the time required by the head to reach the particular track or cylinder is called seek time (5ms to 12ms)

Magnetic Hard Disk (cont…)

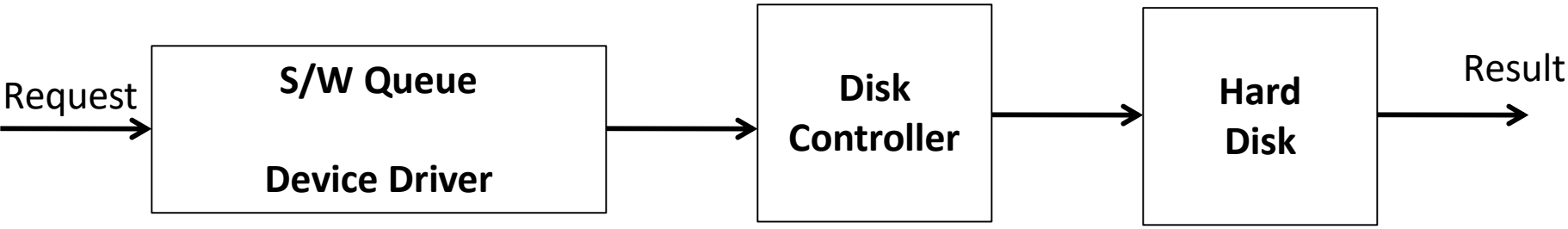
- **Cylinder:** Normally a HD has more than one platters, so the term track is replaced by cylinder. For each platter there are two heads. All heads are connected to a single actuator/spindle whose movement is controlled by a stepper motor. So all the heads move together on the same corresponding track on all platters. A cylinder is a collection of all the corresponding tracks under the heads on all the disk platters (e.g. if head0 is at track20, then all the rest of the heads will also be at track 20. So track 20 on all the platters makes cylinder 20)
- **Sectors:** A sector represents slices of the tracks and are the basic unit of data storage on hard disks. These can be of fixed or variable length. By rotation of disk platter sector number change. While accessing a selected block, the time required by the disk to rotate to bring the specified sector under head is called rotational delay. For a disk rotating at 7200 rpm, it takes 8 ms to have one complete rotation. So average rotational delay is 4 ms, assuming the data is half way around the disk platter

Multiple Zone Recording

- A bit near the center of a rotating disk travels past the read/write head **faster** than a bit on the outside. To compensate for the variation in speed, so that the head can read all the bits at the same rate; the space between bits of information is increased. The information can then be scanned at the same rate by rotating the disk at a fixed speed, known as the **Constant angular Velocity** (CAV). Because the density, in bits per linear inch, increases in moving from the outermost track to the innermost track, disk storage capacity in a straightforward CAV system is limited by the maximum recording density that can be achieved on the innermost track. To increase density, modern hard disk systems use a technique known as **Multiple Zone Recording**
- In **MZR** the surface is divided into a number of concentric zones. Within a zone, the number of bits per track is constant. Zones farther from the center contain more bits (more sectors), than zones closer to the center. As the disk head moves from one zone to another, the length (along the track) of individual bits changes, causing a change in timing for reads and writes

Hard Disk Latency

I/O controllers are h/w that actually controls actual device.
e.g. IDE, EIDE, SCSI, SATA, USB



Disk Latency =

Queuing Time + Controller Time + Seek Time + Rotational Delay + Transfer Time

- **Seek time**. While accessing a selected block, the time required by the head to reach the particular track or cylinder is called seek time
- **Rotational Delay**. While accessing a selected block, the time required by the disk to rotate to bring the specified sector under head is called rotational delay
- **Transfer Time**. Time taken to transfer the block of bits (sector) under the read/write head. It is a function of transfer size, rotation speed, recording density, diameter of platters (about 50 MB/s)

HDD Address
CHS vs LBA

504 MiB/ 528 MB Barrier of CHS Addressing

- The smallest addressable unit on disk is called block (bunch of sectors). Every block has a unique 3-D address: Cylinder#, Head#, Sector#. There two basic types of CHS schemes are:
  - The Logical CHS is used by the INT 13H interface, and allows up to **1024** cylinders, up to **256** heads and up to **63** sectors. This allows support of up to 8GiB drives
  - The Physical CHS is used at the device interface, and allows up to **65535** cylinders, up to **16** heads and up to **63** sectors. This allows support of up to 32GiB
- When a Physical CHS is used at the INT 13H interface it is limited to 1024 cylinders, 16 heads and 63 sectors. This is where the old 504MiB limit originated

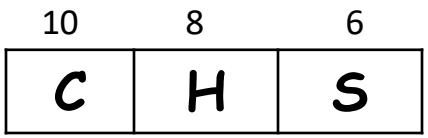
CHS vs LBA

- Most hard disk drives released after 1996 implement Logical Block Addressing (LBA) replacing the CHS scheme which exposed the physical details of the storage device to the operating system
- Large size disks are available now with thousands of cylinders, but the INT 13H interface is capable of accessing a maximum of 1024 cylinders. So to access disks of sizes greater than 504 MiB, Extended BIOS functions are used. Extended BIOS functions does not used CHS address rather use LBA index
- The IDE standard included 22-bit LBA as an option, which was further extended to 28-bit with the release of ATA-1 (1994) and to 48-bit with the release of ATA-6 (2003). The 48 bit LBA raise the addressing limit to 2^{48} x 512 Bytes (128PiB)

Logical Block Addressing

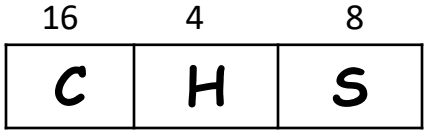
- Logical block addressing (LBA) is a technique that allows a computer to address a hard disk larger than 504 MiB. A logical block address is basically a n-bit value that maps to a specific cylinder-head-sector address on the disk

24 bit LBA:



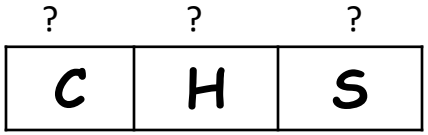
Max disk size support = (1024 x 256 x 63) x 512 = 7.8 GiB

28 bit LBA:



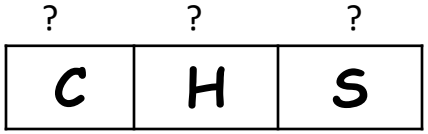
Max disk size support = (65536x 16x 255) x 512 = 127 GiB

48 bit LBA:



Max disk size support = ?

64 bit LBA:



Max disk size support = ?

Note: Zero Sector is not considered in any of the above schemes

Logical Block Addressing

Disk is divided into array of blocks called Logical Blocks. LBA is a 24/28/48/64 bit index. Each unique combination of the three parameters is assigned a unique index as shown below:

Q: How this scheme reduces Seek time?

	Cylinder	Head	Sector	LBA index
Side 0	0	0	1	0
	0	0	2	1
	0	0	3	2
	-	-	-	-
	0	0	63	62
Side 1	0	1	1	63
	0	1	2	64
	0	1	3	65
	-	-	-	-
	0	1	63	125
Side 2	0	2	1	126
	0	2	2	127
	0	2	3	128

LBA TRANSLATION

CHS to LBA translation

LBA address = (C * TH * TS) + (H * TS) + (S – 1)

- C is the selected cylinder#
- H is the selected head#
- S is the selected sector#
- TH is the total no of heads supported by disk
- TS is the total no of sectors supported by disk (63)

LBA to CHS Translation

Cylinder# = LBA / (TH * TS)

Head# = (LBA / TS) % TH

Sector# = (LBA % TS) + 1

% is the modulo operation, while / is integer division (discard fractional part

LBA TRANSLATION

- LBA and CHS equivalence is shown in this table with 16 Heads per cylinder
- Students to please use the formulae on the previous slide to confirm the calculations

LBA value	CHS tuple
0	0, 0, 1
1	0, 0, 2
2	0, 0, 3
62	0, 0, 63
63	0, 1, 1
945	0, 15, 1
1007	0, 15, 63
1008	1, 0, 1
1070	1, 0, 63
1071	1, 1, 1
1133	1, 1, 63
1134	1, 2, 1
2015	1, 15, 63
2016	2, 0, 1
16,127	15, 15, 63
16,128	16, 0, 1
32,255	31, 15, 63
32,256	32, 0, 1
16,450,559	16319, 15, 63
16,514,063	16382, 15, 63

Reading & Writing a Hard Disk

The operating system sends the LBA of the block to be read to the disk controller, which converts the LBA to appropriate CHS address and proceeds as follows:

- The stepper motor attached with the head assembly moves the head to specific cylinder
- Switch ON the specific head electronically
- Disk platters rotate until the specific sector comes under the head
- The data in the sector(s) is read and placed in a disk buffer to be read by the OS later

Partitioning a Hard Disk

DISK PARTITIONING

- Disk partitioning is the concept of dividing your hard disk into logical parts, making one hard drive appear as if it's actually multiple drives
- On Microsoft Operating Systems, disk partitions are represented by C:, D:, E:, …
- On all UNIX based systems, these partitions are represented as block special files, mostly located in /dev/ directory. (/dev/sda1, /dev/sda2,…). The first two characters identify the type of disk interface, i.e., **hd** for ATA and **sd** for SATA/SCSI. The third character represent the number of disk, i.e., **a** for 1st disk, **b** for 2nd and so on. Finally the number at the end represent the partition. So /dev/sdb3 means third partition of second disk which is SATA or SCSI

DISK PARTITIONING (cont…)

Reasons for partitioning a hard disk are:

- Better Organization of Data
- Multiple file systems (fat32, ntfs, ext3, ext4, zfs)
- No Crossing of Partition Limits
- Implement quotas (limiting amount of space that users or groups can take on a file system)
- Security (Mounting a file system on a partition that seldom change as read only. This improves security as well as reduces the recovery time of a file system after a system crash)

DISK PARTITIONING (cont…)

Recommended partitions on Linux system are:

- **Swap Partition(s):** Normally one, but can be more. It is the expansion of the computer's physical memory on HDD. It is hidden from the user and only accessible to the system itself. Swap space is normally set twice the size of the available physical memory
- **Data Partition(s):** There can be many data partitions but minimum there should be one and that is called the "root partition" (/). The root partition contains all the data to start up and run the system

DISK PARTITIONING (cont…)

Some candidates to be mounted on a separate data partition depending on the use of the machine (personal laptop, workstation, server) are

- /boot: A partition for kernel image
- /home: A partition for user's personal data
- /var: A partition to hold files that are unpredictable in size, e.g., log, spool, cache, mail
- /usr: A partition to hold UNIX System Resources, containing sharable and read only data

After system startup. These and all the other partitions that you make, appear under the root partition as an inverted tree. Data partitions can be created, resized or deleted. It is usually done before installation of an OS, however, can be done afterward as well

PARTITION TABLE

- Suppose we have made three partitions on our hard disk. The question is how does the operating system knows, where does a partition starts and where it ends or the other starts. Answer is partition table
- A partition table is maintained on disk, it is a structure which efficiently manages information about different partitions of a hard disk
- The different types of partition tables are:
  - Master Boot Record (MBR)
  - Globally Unique Identifier Partitioning Table (GPT)
  - Apple Partition Map (APM)
  - BSD disklabel
  - Sun
  - CGI

PARTITION TABLE (Cont…)

- On every hard disk there exist a **partition table** and this partition table can keep track of maximum 4 partitions called primary partitions. Thus we are restricted to only four primary partitions on one hard disk. To work around this, we have to make one of the primary partition as **extended partition**. An extended partition cannot hold any data but is used to maintain the list of **logical partitions** within it
- On a Linux machine with one primary and two logical partitions, the names are /dev/sda1, /dev/sda5, /dev/sda6

Zero sector

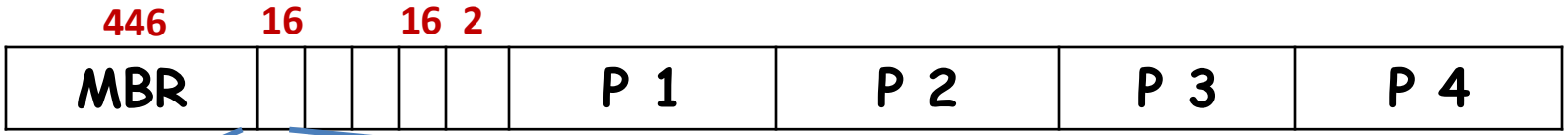
MBR	P 1	P 2	P 3	P 4

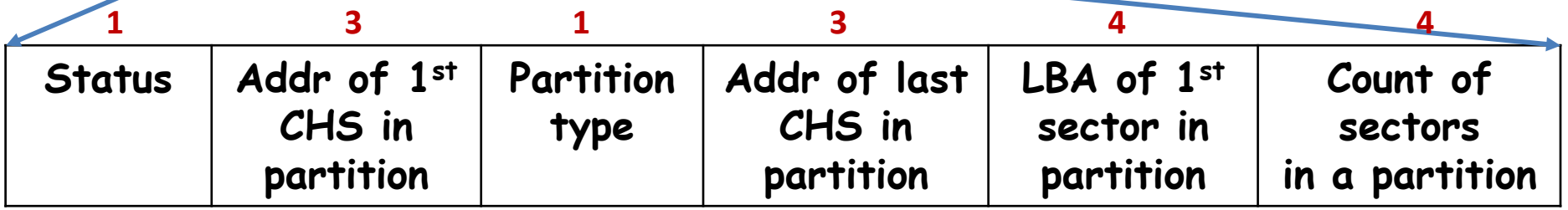
```
$ sudo fdisk -l /dev/sda

$ cat /proc/partitions
```

MBR PARTITION TABLE

- Master Boot Record partitioning table was introduced for Personal Computers back in 1983
- It is placed on the zero sector of HDD along with the stage-1 boot loader. Since it has only four entries (16B each), so it can keep track of only four primary partitions






```
$ sudo dd if=/dev/sda bs=512 count=1 | tail -c 66 | hexdump -C
```

Pitfalls of MBR

- You can have four primary partitions
- Partition size can be a maximum of 2 TiB
- Zero sector is the only place that holds partition information, if it is corrupted entire disk is unreadable

GPT

- Globally unique identifier Partitioning Table is the latest standard of laying out the partitions of a hard diskWith GPT, you can have 128 partitions with most OSs
- Size of partition can be upto 264 sectors/blocks. With one block of 512B, the partition size is limited upto 9.44 ZettaByte
- GPT stores a backup header and partition entries at zero sector as well as at the end of the disk, so it can be recovered if the primary tables are corrupted
- To boot from a hard disk using GPT, your BIOS should support Unified Extensible Firmware Interface (UEFI)

DISK PARTITIONING Tools

- **Partitioning Utilities.** In order to create partitions, you use a partitioning tool. There are many disk-partitioning utilities which fall under one of the following two categories:
  - **Destructive Resizing**. This technique wipes everything off your hard drive and starts fresh
  - **Non Destructive Resizing**. This technique dynamically shrink/expand the existing partitions and can also use the freed space to make a new partition
- Partitioning tools available on UNIX based systems are:
  - **`fdisk`**
  - **`gdisk`**
  - **`parted/gparted`**
  - **`cfdisk`**
  - **`sfdisk`**

DISK PARTITIONING Tools (cont…)

- **fdisk:** It is a dialog driven program for creation and manipulation of partition table. Older versions were used for only MBR partitions. But ver 2.8.2 understand MBR, GPT, BSD, Sun and SGI partition tables. It is able to optimize the disk layout for 4KB sector size, **fdisk(8)**
- **gdisk:** It is a text-mode menu driven program for creation and manipulation of partition table. It operates on GPT partitions and has the capability to transform MBR partitions to GPT partitions
- **parted/gparted:** A partition editor program and gparted is the gnome version of it. It supports multiple parition table formats including MBR, GPT, BSD, APM
- **cfdisk:** A curses-based program for partitioning any block device. Since version 2.25 supports MBR, GPT, Sun, SGI. It no longer provides any functionality for CHS addressing
- **sfdisk:** A script oriented tool for partitioning block devices

fdisk

- If fdisk is not installed on your machine you can use the following commands to do that:

```
# apt-get install fdisk
# fdisk --version
```

- If installed you can run the following command to check out the information about the existing partitions of your hard disk

```
# fdisk -l /dev/sda
```

- To practice changing the partition table, run the following command and enjoy

```
# fdisk /dev/sda
```

- fdisk doesn't write the changes to disk until you give the **w** command. So remember to quit without saving changes or you may destroy what will be difficult for you to recover

Disk Scheduling Algorithms

DISK SCHDULING

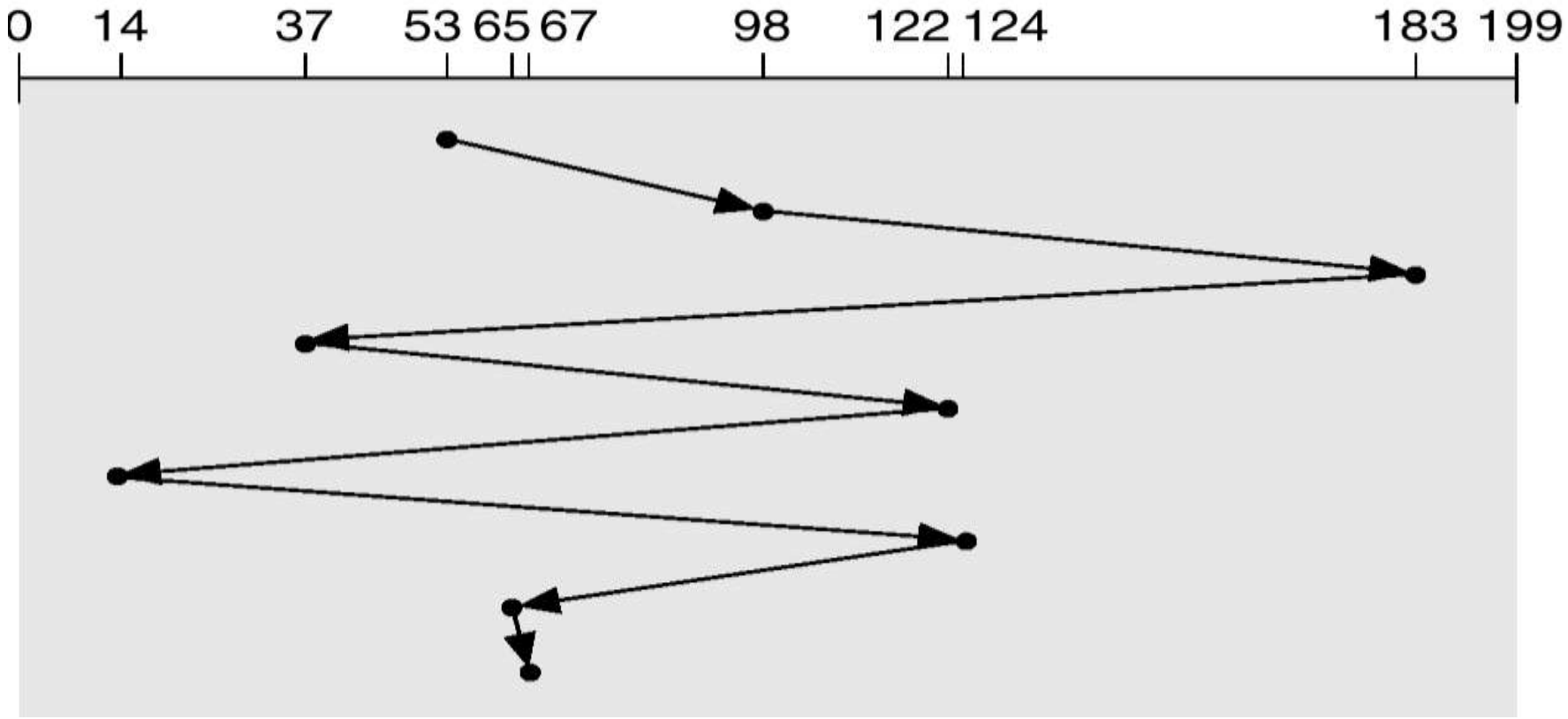
- **Disk bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.
- Minimize seek time
- Seek time ≈ seek distance
- We can improve access time and bandwidth by scheduling the servicing of disk I/O request in a good order.
- Several algorithms exists to schedule the servicing of disk I/O requests which are:

- FCFS
- SSTF
- SCAN
- C-SCAN
- LOOK
- C-LOOK

FCFS

Consider a disk with 200 sectors (0 – 199). Head is currently serving cylinder 53. Requests are for following cylinders.

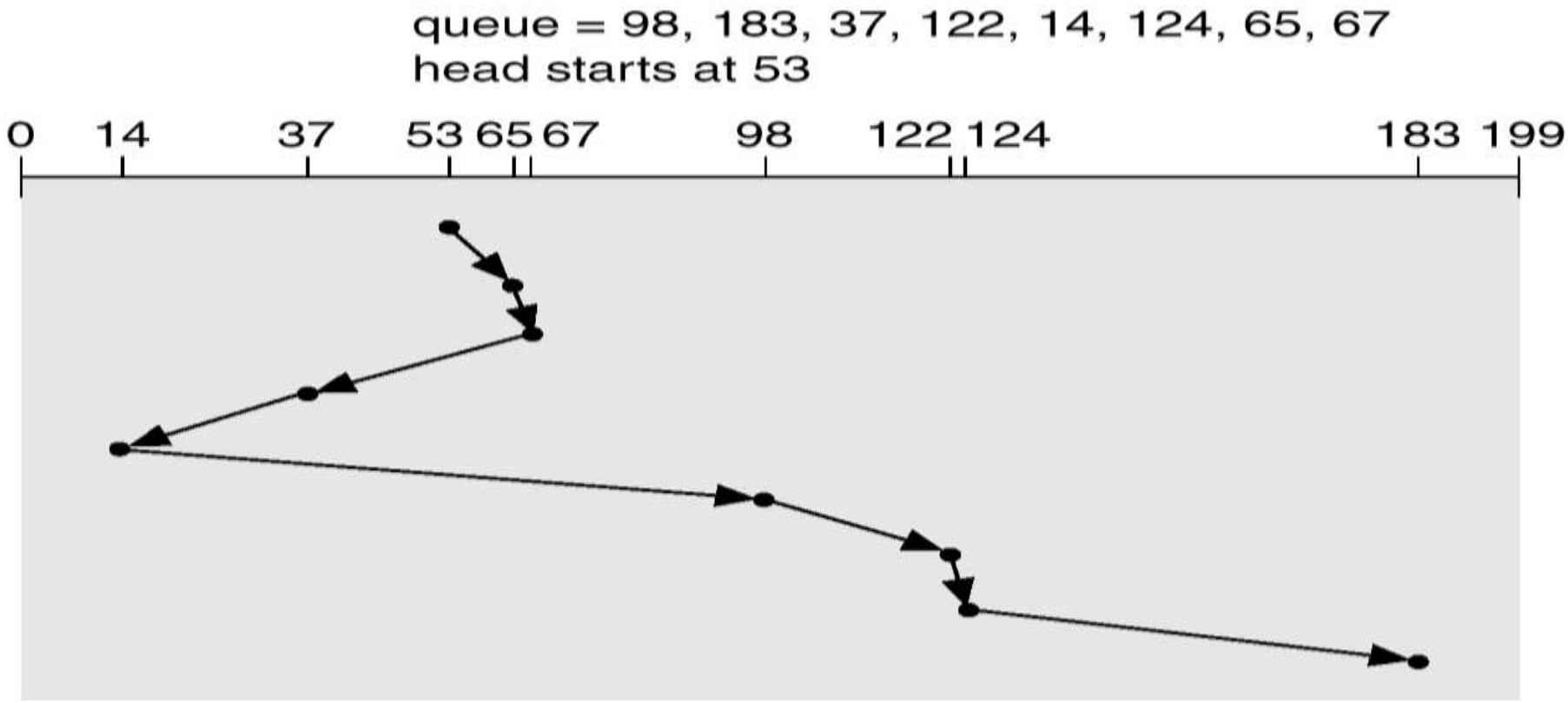
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



SSTF

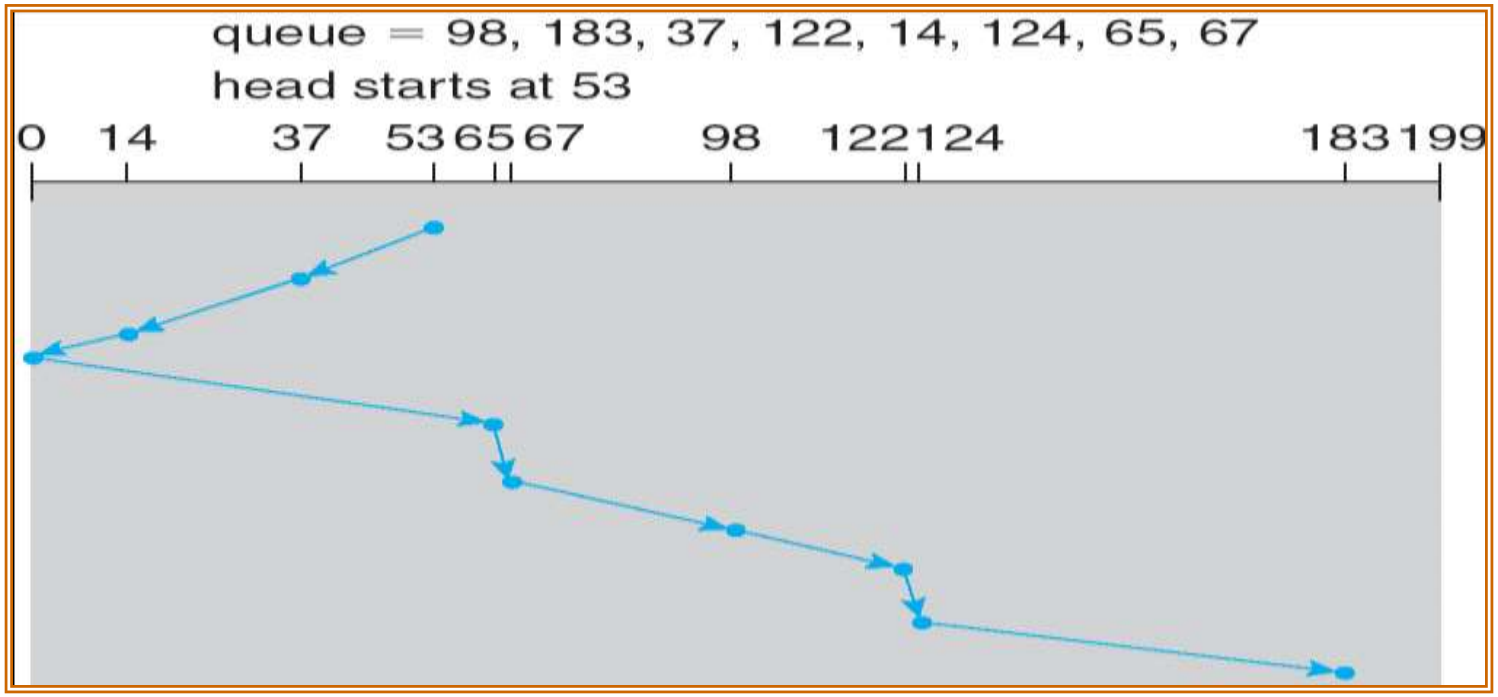
- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; **may cause starvation of some requests**.

Consider a disk with 200 sectors (0 – 199). Head is currently serving cylinder 53. Requests are for following cylinders.



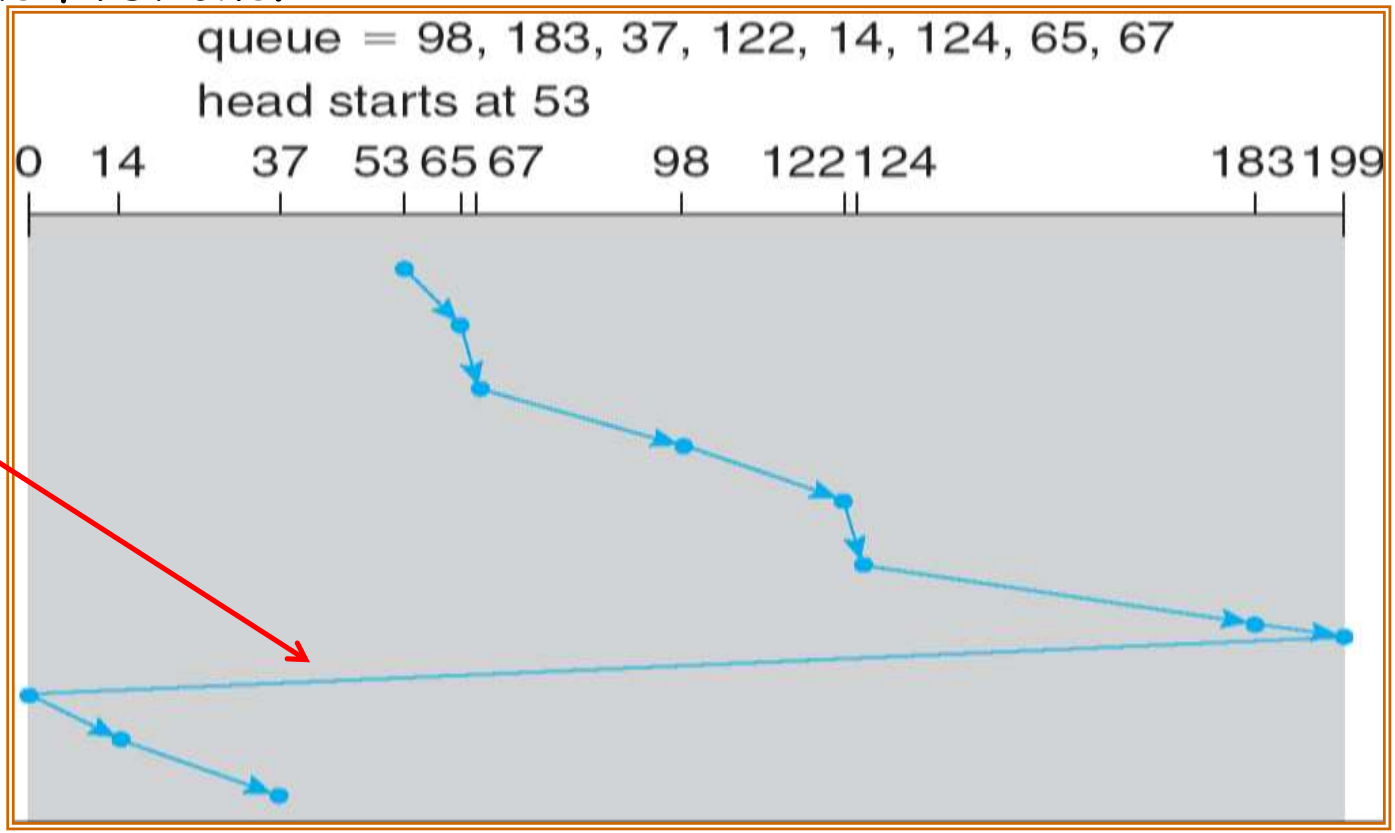
SCAN

- The disk arm s required to move in one direction only, satisfying all outstanding requests en route, until it reaches the last track in that direction. The service direction is then reversed and the scan proceeds in the opposite direction, again picking up all requests en route. (you can start either from left or right)
- Sometimes called the *elevator algorithm*.
- Inner tracks gets priority.
- Outer tracks requests have to wait
- Total head movement of 208 cylinders.



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. i.e. the requests are serviced in one direction only.
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

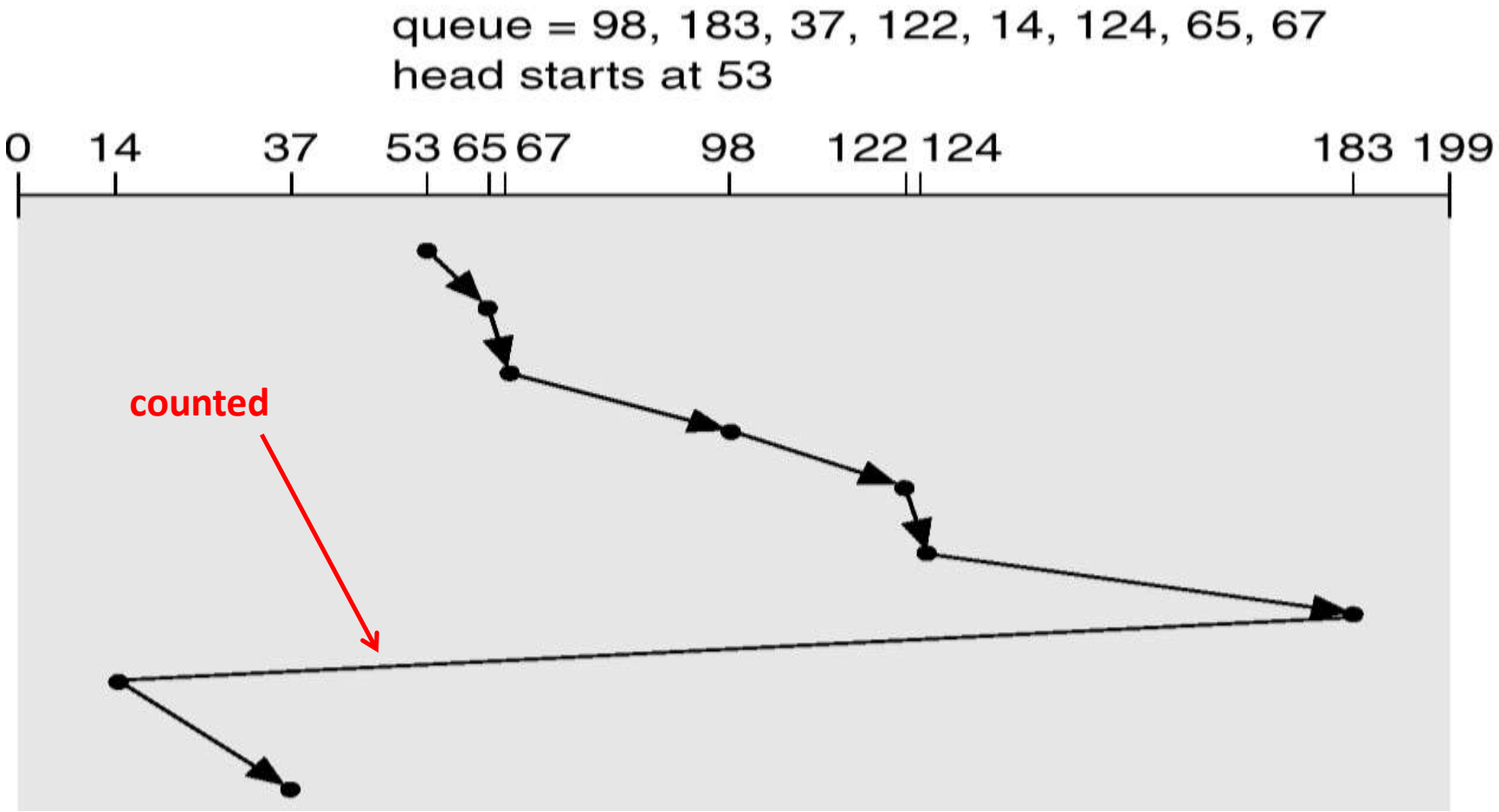


LOOK

- 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.
- Requests are served in both direction.
- Best algorithm

C-LOOK

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. Requests are served in one direction only.



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal because it increases performance over FCFS.
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk because they are less likely to have a starvation problem.
- Performance depends on the number and types of requests. If there is only one request in the queue at a time all algorithms behave like FCFS.
- Requests for disk service can be influenced by the file-allocation method. A process reading a contiguously allocated file will generate requests that will generate less head movements while a linked or indexed file will generate a greater head movement.

Selecting a Disk-Scheduling Algorithm

- Location of directories and index blocks are also important. For example whenever a file is opened it require a search in the directory structure. A directory entry can be in the first cylinder while the file can be on the last cylinder. Caching the directories and index blocks in main memory can help improve performance.
- 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.

We're done for now, but Todo's for you after this lecture…

- Go through the related video lectures # 16 and 17.
- Practice, practice and practice…