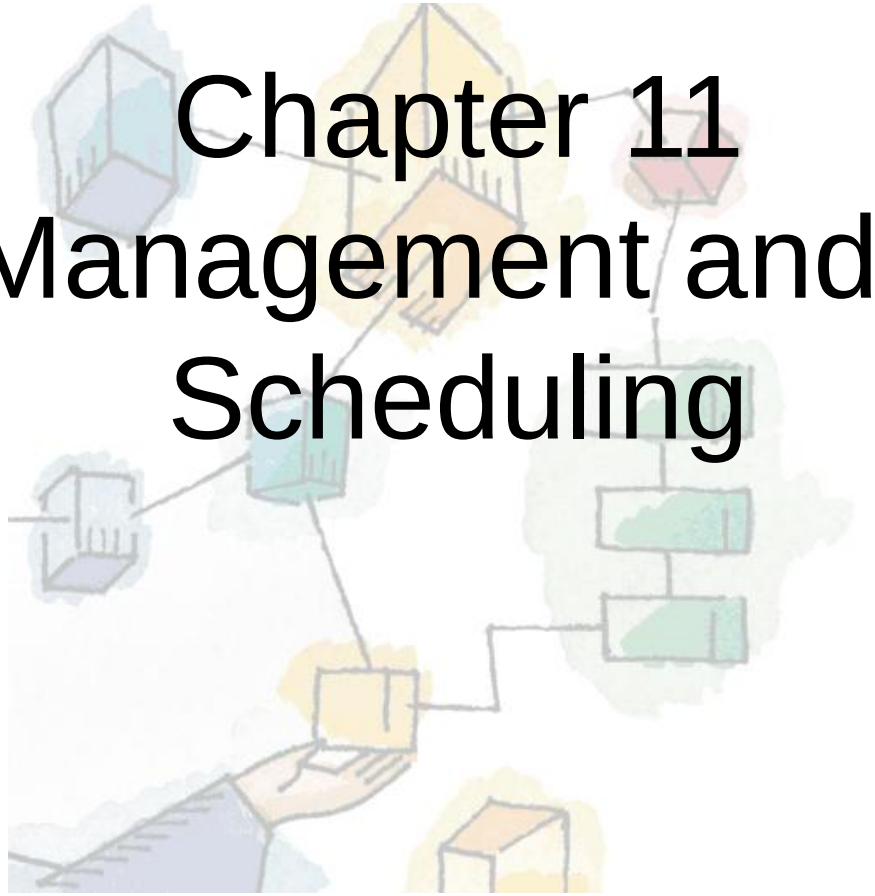
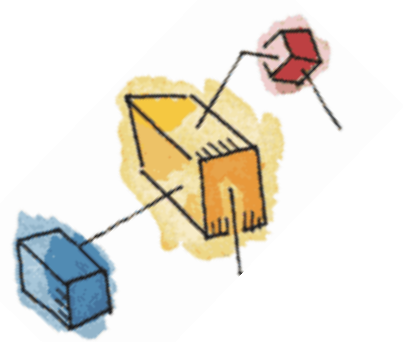
Operating Systems:
Internals and Design Principles, 6/E
William Stallings

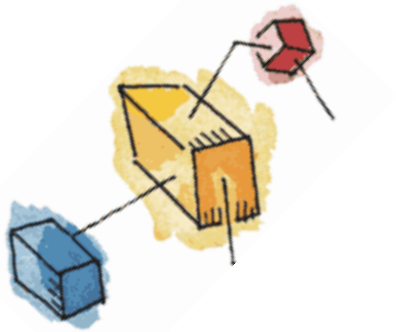


Chapter 11
I/O Management and Disk Scheduling



I/O Management

- Difficult area of OS design
  - Difficult to develop a consistent solution due to a wide variety of devices and applications
- Devices differ in a number of areas
  - Data Rate
  - Application
  - Complexity of Control
  - Unit of Transfer
  - Data Representation
  - Error Conditions



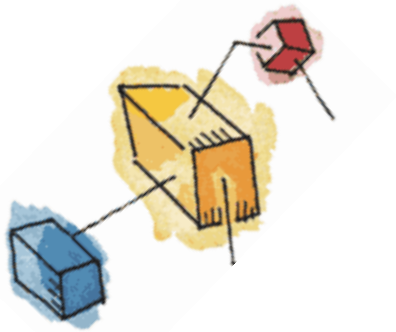
I/O Buffering

- Processes must wait for I/O to complete before proceeding

- It may be more efficient to perform input transfers in advance of requests being made and to perform output transfers some time after the request is made.

Block-oriented Buffering

- Information is stored in fixed sized blocks
- Transfers are made a block at a time
  - Can reference data by block number
- Used for disks and USB



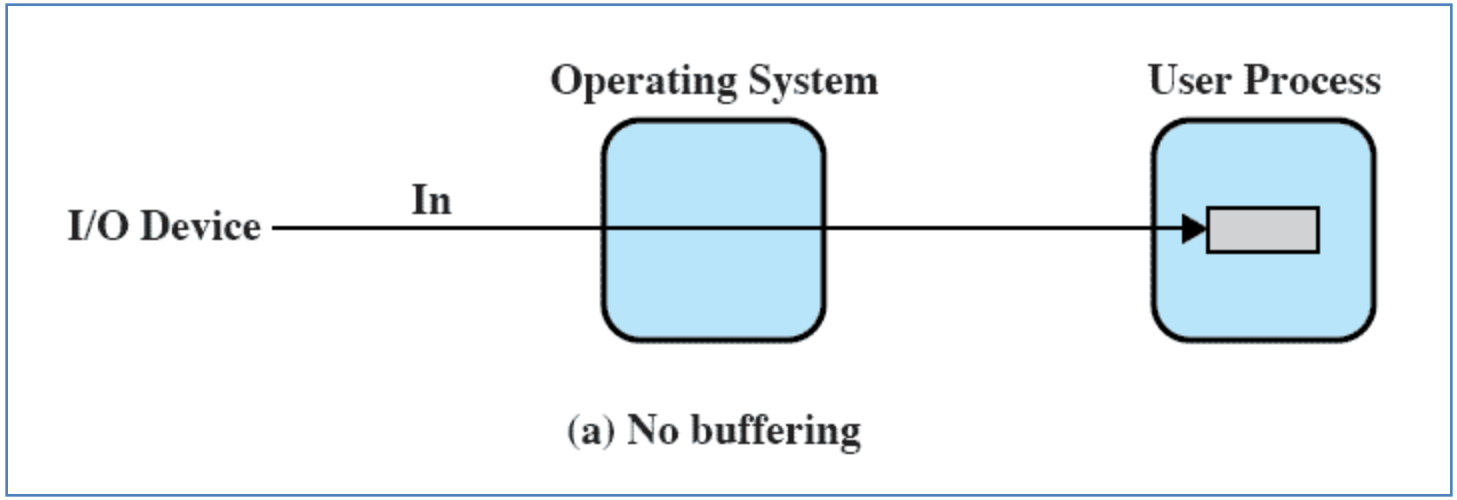
Stream-Oriented Buffering

- Transfer information as a stream of bytes

- Used for terminals, printers, communication ports, mouse and other pointing devices, and most other devices that are not secondary storage

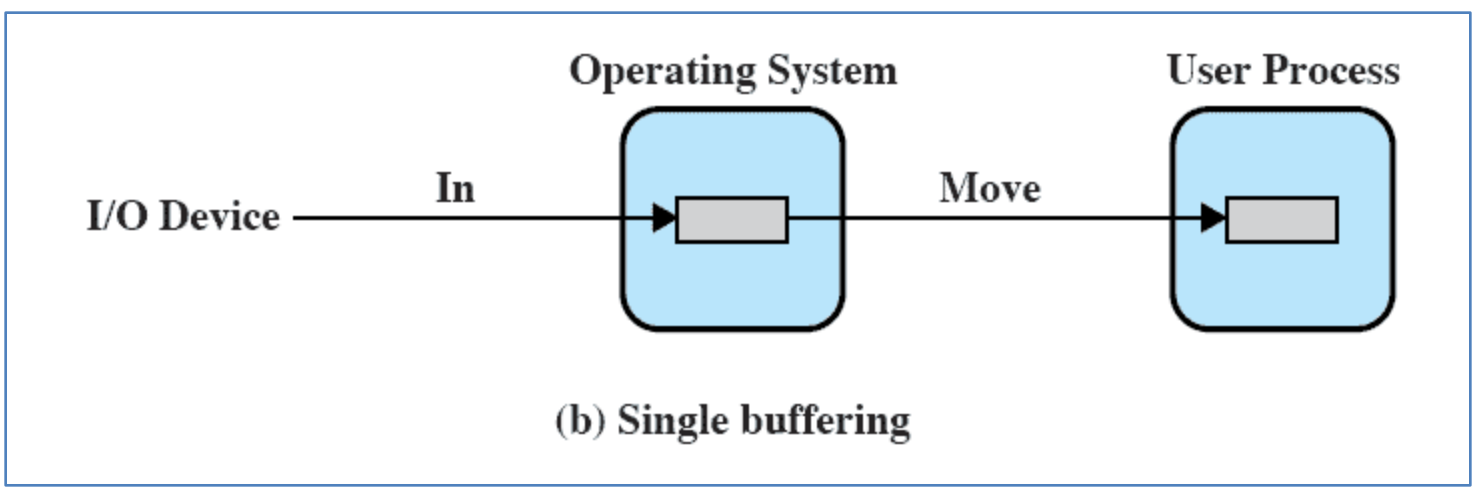
No Buffer

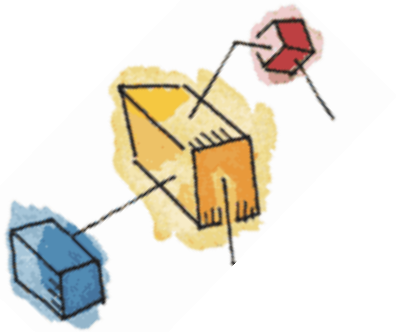
- Without a buffer, the OS directly access the device as and when it needs



Single Buffer

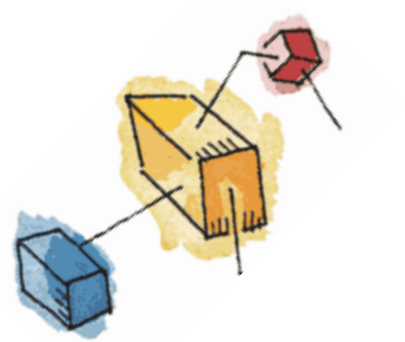
- Operating system assigns a buffer in main memory for an I/O request





Block Oriented Single Buffer

- Input transfers made to buffer
- Block moved to user space when needed
- The next block is moved into the buffer
  - *Read ahead* or *Anticipated Input*
- Often a reasonable assumption as data is usually accessed sequentially

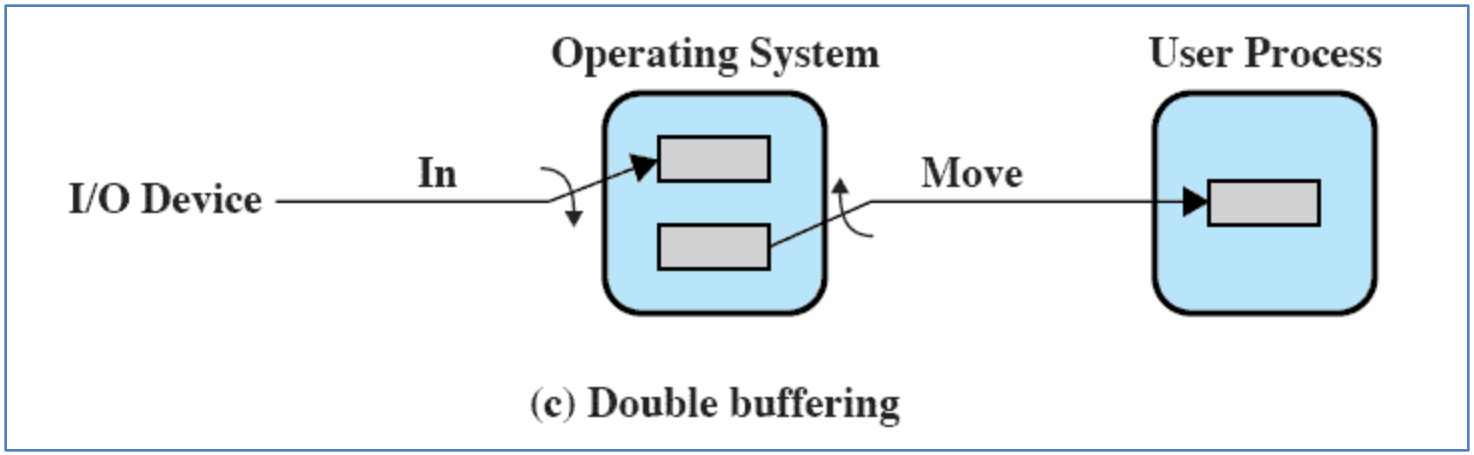


Stream-oriented Single Buffer

- Line-at-time or Byte-at-a-time
- Terminals often deal with one line at a time with carriage return signaling the end of the line
- Byte-at-a-time suites devices where a single keystroke may be significant
  - Also sensors and controllers

Double Buffer

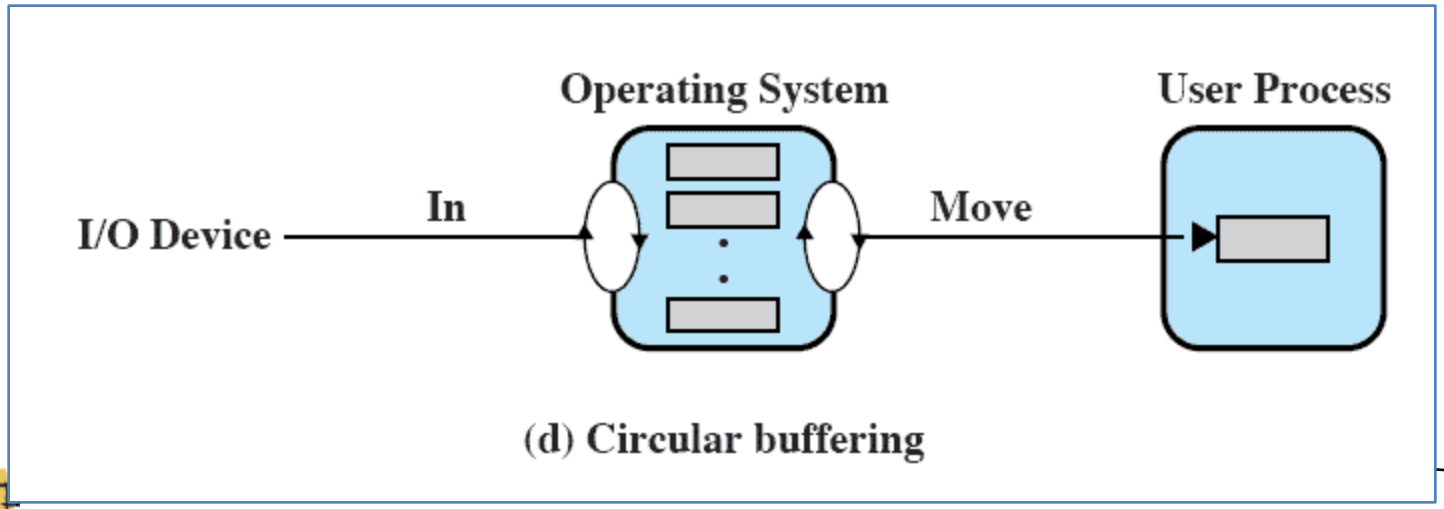
- Use two system buffers instead of one
- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



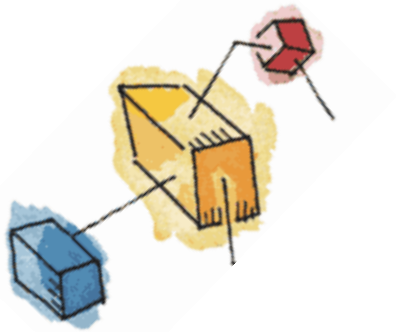
(c) Double buffering

Circular Buffer

- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



(d) Circular buffering



Disk Performance Parameters

- The actual details of disk I/O operation depend on many things
  - A general timing diagram of disk I/O transfer is shown here.

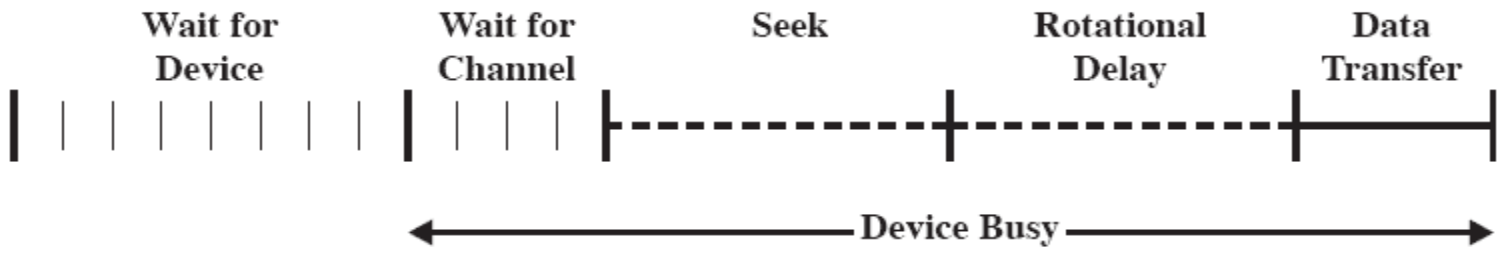
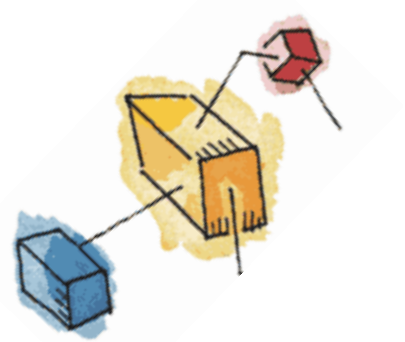


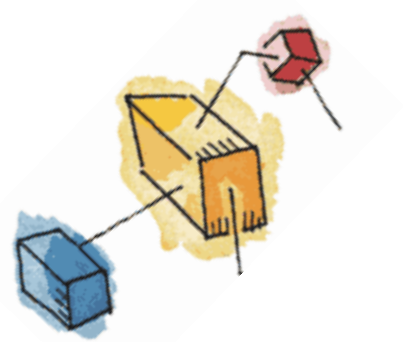
Figure 11.6 Timing of a Disk I/O Transfer



Positioning the Read/Write Heads

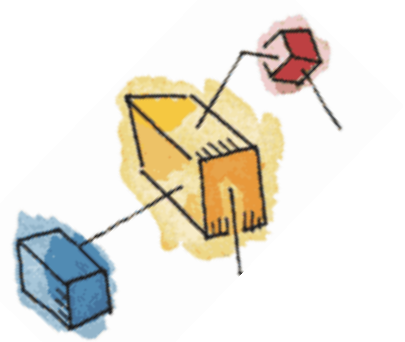
- When the disk drive is operating, the disk is rotating at constant speed.

- Track selection involves moving the head in a movable-head system or electronically selecting one head on a fixed-head system.



Disk Performance Parameters

- ***Access Time*** is the sum of:
  - ***Seek time:*** The time it takes to position the head at the desired track
  - ***Rotational delay*** or ***rotational latency:*** The time its takes for the beginning of the sector to reach the head
- ***Transfer Time*** is the time taken to transfer the data.



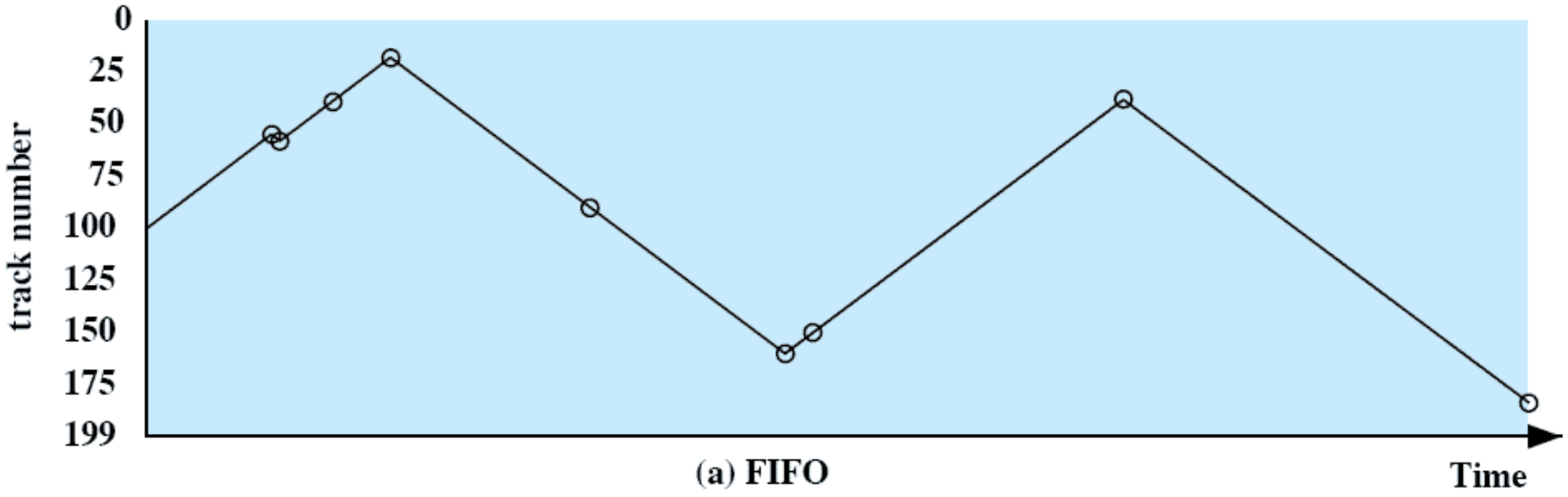
Disk Scheduling Policies

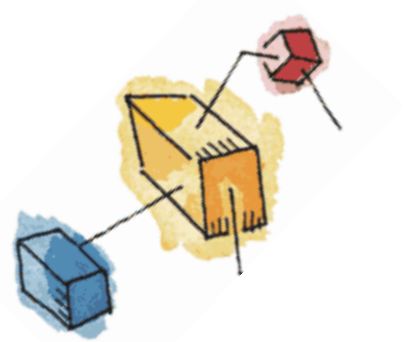
- To compare various schemes, consider a disk head is initially located at track 100.
  - Assume a disk with 200 tracks and that the disk request queue has random requests in it.

- The requested tracks, in the order received by the disk scheduler, are
  - 55, 58, 39, 18, 90, 160, 150, 38, 184.

First-in, first-out (FIFO)

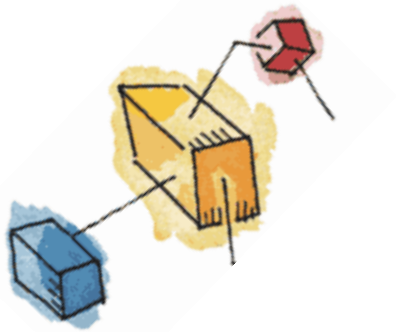
- 55, 58, 39, 18, 90, 160, 150, 38, 184
- Process request sequentially
- Fair to all processes, every request honored
- Can become random scheduling in performance if there are many processes competing for disk





Priority

- Goal is not to optimize disk use but to meet other objectives
- Short batch jobs may have higher priority
- Provide good interactive response time
- Longer jobs may have to wait an excessively long time

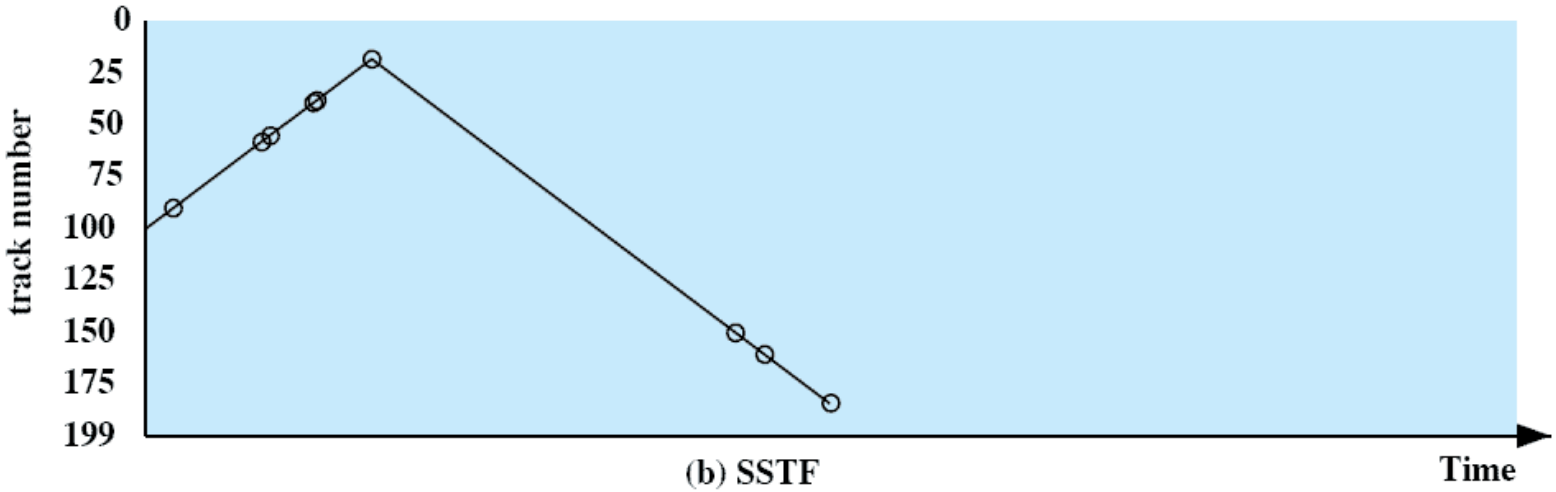


Last-in, first-out

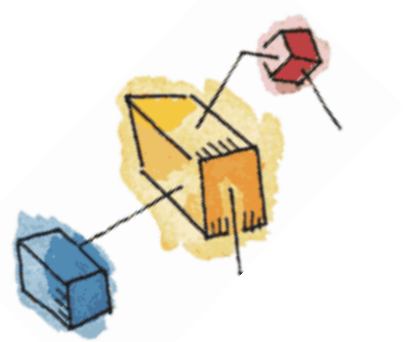
- Consider most recent request so there should be little arm movement (Principle of locality)
- Can provide increased throughput and reduced queue length
- Possibility of starvation since a job may never regain the head of the line

Shortest Service Time First

- Select the disk I/O request that requires the least movement of the disk arm from its current position
- Always choose the minimum seek time
- 55, 58, 39, 18, 90, 160, 150, 38, 184



(b) SSTF



SCAN

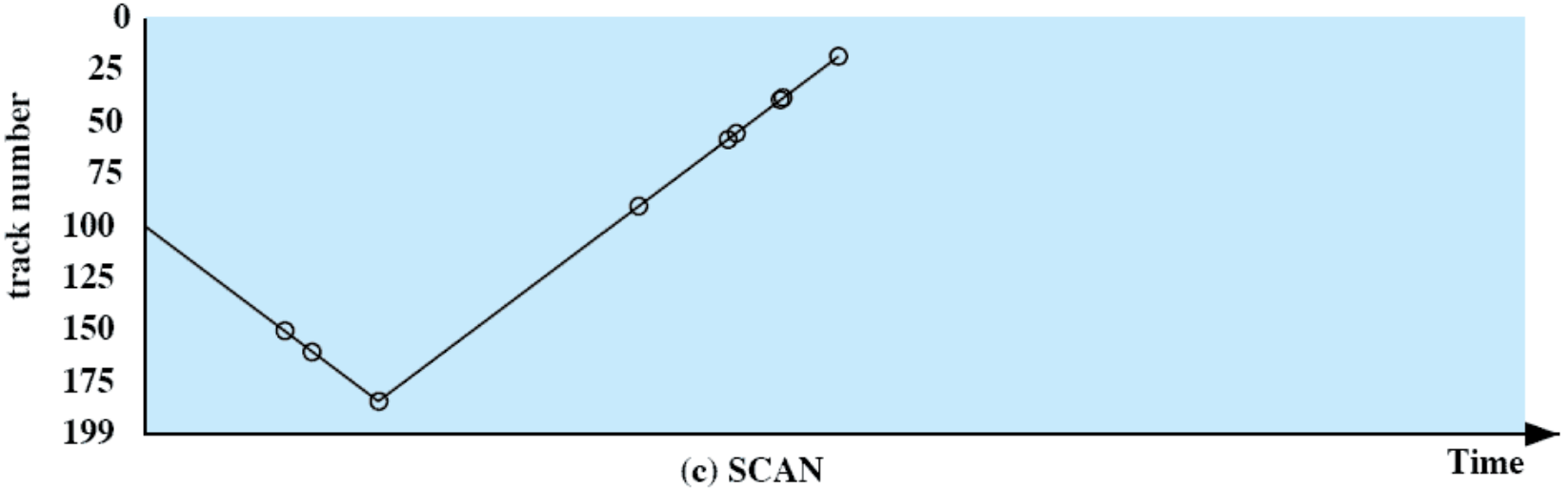
- Priority, LIFO and SSTF may leave some request unfulfilled, this is prevented by SCAN

- Arm moves in one direction only, satisfying all outstanding requests
  - until it reaches the last track in that direction or
  - there are no more requests in that direction
- then the direction is reversed

- Also known as the elevator algorithm

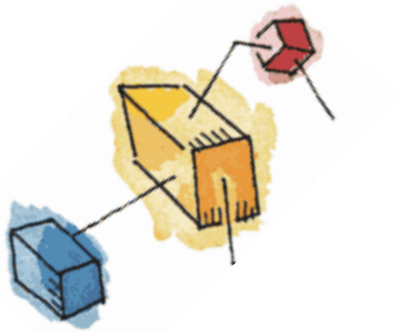
SCAN

- 55, 58, 39, 18, 90, 160, 150, 38, 184



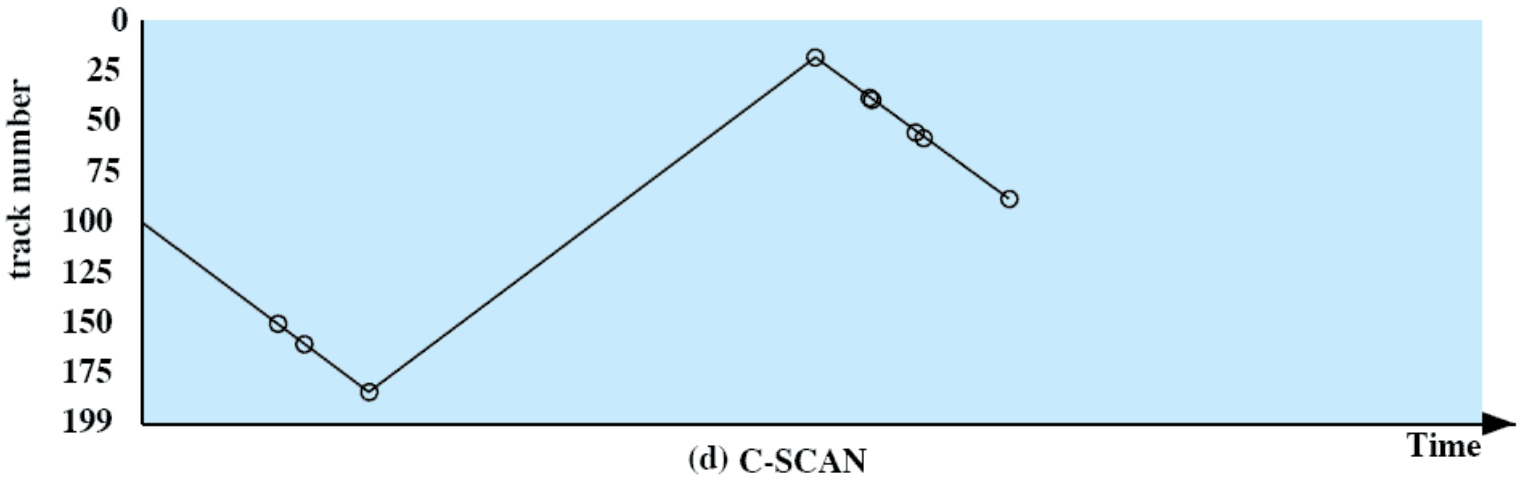
(c) SCAN

- SCAN is biased against the area most recently traversed

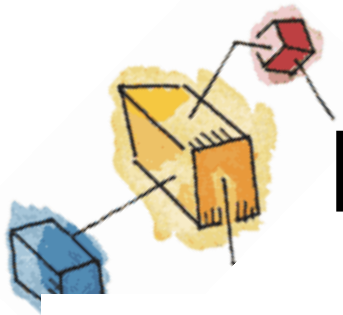


C-SCAN

- Restricts scanning to one direction only
- When the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again
- 55, 58, 39, 18, 90, 160, 150, 38, 184



(d) C-SCAN



Performance Compared

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	**Average seek length**	27.5	**Average seek length**	27.8	**Average seek length**	35.8