

Operating Systems

Prepared by Dr. Alaa Mohamed

Module 6 – Part2



I/O Management and Disk Scheduling



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

- Objectives:
 - I/O Devices
 - Organization of the I/O Function
 - Operating System Design Issues
 - Disk Scheduling

Categories of I/O Device

- Difficult area of OS design
 - Difficult to develop a consistent solution due to a wide variety of devices and applications
- Three Categories:
 - Human readable
 - Machine readable
 - Communications

- Devices used to communicate with the user
 - Printers and terminals
 - Video display
 - Keyboard
 - Mouse etc

- Used to communicate with electronic equipment
 - Disk drives
 - is a technology that enables the reading, writing, deleting and modifying of data on a computer storage **disk**. It is either a built-in or external component of a **disk** that manages the **disk's** input/output (I/O) operations.
 - USB keys
 - Sensors
 - Controllers

- Used to communicate with remote devices
 - Digital line drivers
 - are commonly used within digital systems, e.g. to communicate digital signals across circuit-board traces and cables, for example to connect a stereo music player to an amplified speaker system.
 - Modems
 - a device that converts signals produced by one type of device (such as a computer) to a form compatible with another (such as a telephone) and that is used specially to transmit and receive information between computers via landlines

- Devices differ in a number of areas
 - Data Rate
 - Application
 - Complexity of Control
 - Unit of Transfer
 - Data Representation
 - Error Conditions

- May be massive difference between the data transfer rates of devices

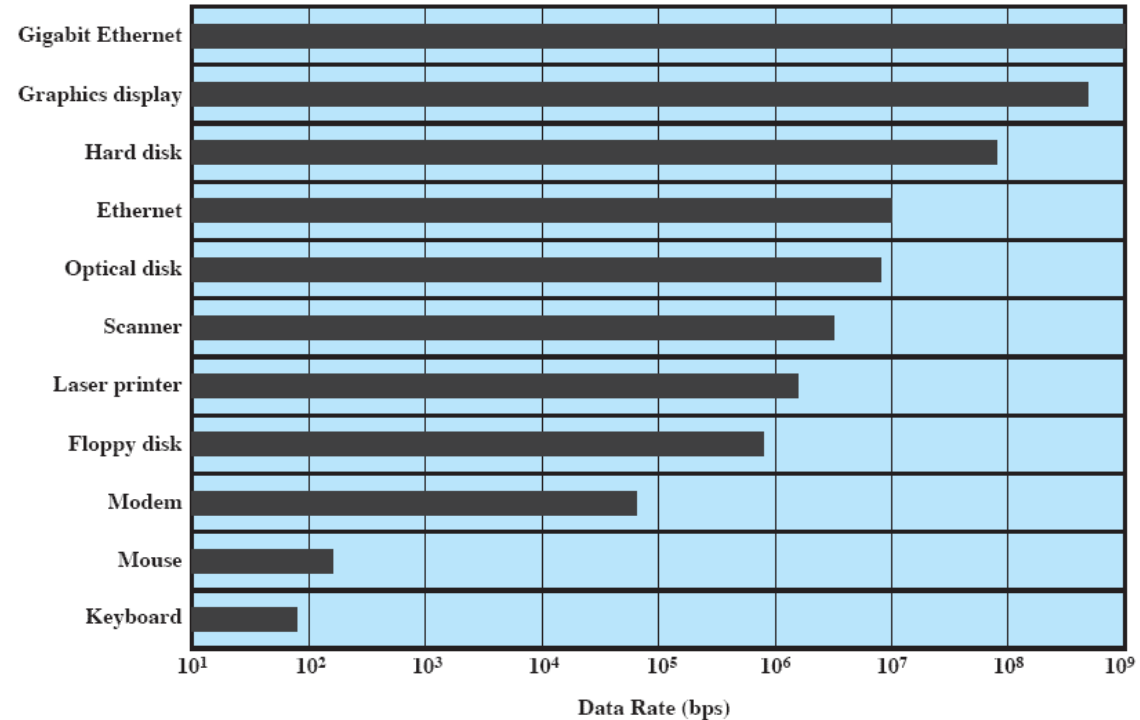


Figure 11.1 Typical I/O Device Data Rates

The use to which a device is put has an influence on the software and policies in the operating system and supporting utilities.

- Disk used to store files requires file management software
- Disk used to store virtual memory pages needs special hardware and software to support it
- Terminal used by system administrator may have a higher priority

- A printer requires a relatively simple control interface.
- A disk is much more complex.
- This complexity is filtered to some extent by the complexity of the I/O module that controls the device.

- Data may be transferred as
 - a stream of bytes or characters (e.g., terminal I/O)
 - or in larger blocks (e.g., disk I/O).

- Different data encoding schemes are used by different devices,
 - including differences in character code and parity conventions.

- The nature of errors differ widely from one device to another.
- Aspects include:
 - the way in which they are reported,
 - their consequences,
 - the available range of responses

- I/O Devices
- Organization of the I/O Function
- Operating System Design Issues
- Disk Scheduling

- Programmed I/O
- Interrupt-driven I/O
- Direct memory access (DMA)

Programmed I/O:

- Processor issues an I/O command, on behalf of a process, to an I/O module;
- that process then busy waits for the operation to be completed before proceeding.

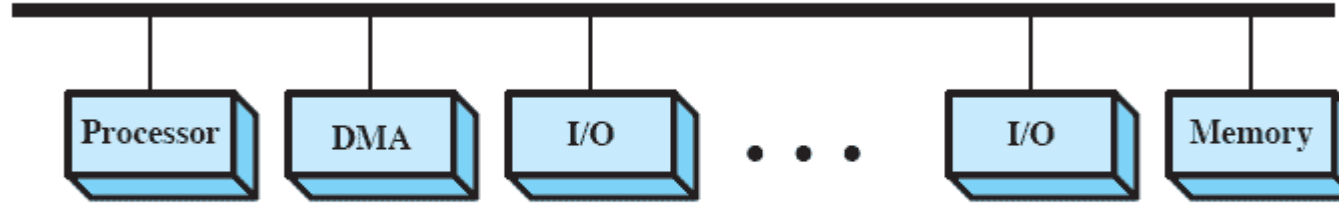
Interrupt-driven I/O:

- Processor issues an I/O command on behalf of a process.
- *If* the I/O instruction from the process is ***nonblocking***, then the processor continues to execute instructions from the process that issued the I/O command.
- *If* the I/O instruction is ***blocking***, then the next instruction that the processor executes is from the OS, which will put the current process in a blocked state and schedule another process.

Direct memory access (DMA):

- A DMA module controls the exchange of data between main memory and an I/O module.
- The processor sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.

DMA Configurations: Single Bus



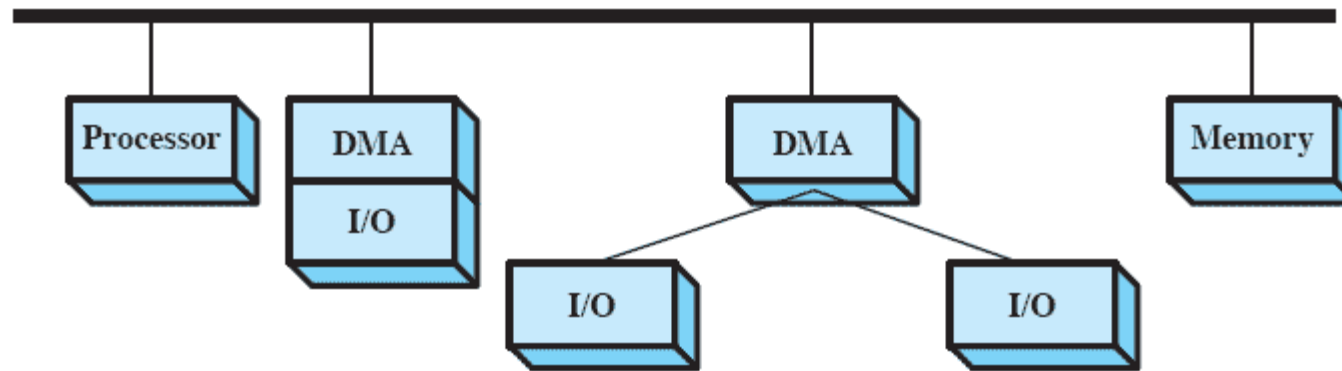
(a) Single-bus, detached DMA

- The DMA mechanism can be configured in a variety of ways.
- Some possibilities are shown here In the first example, all modules share the same system bus.
 - The DMA module, acting as a surrogate processor, uses programmed I/O to exchange data between memory and an I/O module through the DMA module.



- In the example provided by the above diagram, there is a **single bidirectional bus** connecting the processor, the memory, the DMA module and all the I/O modules.
 1. When a particular I/O module needs to read or write large amounts contiguous data it requests the processor for direct memory access.
 2. If permission is granted by the processor, the I/O module sends the read or write address and the size of data needed to be read or written to the DMA module.

3. Once the DMA module acknowledges the request, the I/O module is free to read or write its contiguous block of data from or onto main memory.
4. Even though in this situation the processor will not be able to execute while the transfer is going on (as there is a just a single bus to facilitate transfer of data),
5. DMA transfer is much faster then having each word of memory being read by the processor and then being written to its location.

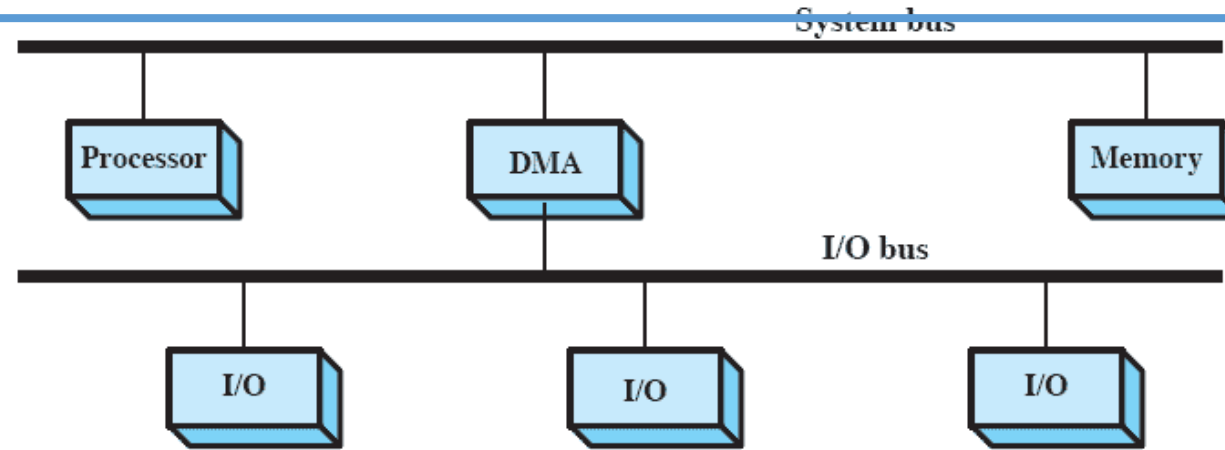


(b) Single-bus, Integrated DMA-I/O

- Direct Path between DMA and I/O modules
- The number of required bus cycles can be cut substantially by integrating the DMA and I/O functions.

- This means that there is a path between the DMA module and one or more I/O modules that does not include the system bus.
- The DMA logic may actually be a part of an I/O module, or it may be a separate module that controls one or more I/O modules.
- In this configuration the DMA and one or more I/O modules are integrated without the inclusion of system bus functioning as the part of I/O module or may be as a separate module controlling the I/O module.

DMA Configurations: I/O Bus



- This concept can be taken one step further by connecting I/O modules to the DMA module using an I/O bus
- This Reduces the number of I/O interfaces in the DMA module to one and provides for an easily expandable configuration.

- The system bus that the DMA module shares with the processor and main memory is used by the DMA module only to exchange data with memory and to exchange control signals with the processor.
- The exchange of data between the DMA and I/O modules takes place off the system bus.

- I/O Devices
- Organization of the I/O Function
- Operating System Design Issues
- Disk Scheduling

- Most I/O devices extremely slow compared to main memory
- One way to tackle this problem is multiprogramming which allows for some processes to be waiting on I/O while another process executes
- I/O cannot keep up with processor speed
 - Swapping used to bring in ready processes to keep processor busy
 - But this is an I/O operation itself, , Thus a major effort in I/O design has been schemes for improving the efficiency of the I/O

- For simplicity and freedom from error it is desirable to handle all I/O devices in a uniform manner
- Hide most of the details of device I/O in lower-level routines
- Difficult to completely generalize, but can use a hierarchical modular design of I/O functions

- A hierarchical philosophy leads to organizing an OS into layers
- Each layer relies on the next lower layer to perform more primitive functions
- It provides services to the next higher layer.
- Changes in one layer should not require changes in other layers

- I/O Devices
- Organization of the I/O Function
- Operating System Design Issues
- Disk Scheduling

- The actual details of disk I/O operation depend on many things such as, computer system, the operating system, and the nature of the I/O channel and disk controller hardware.
- A general timing diagram of disk I/O transfer is shown here.

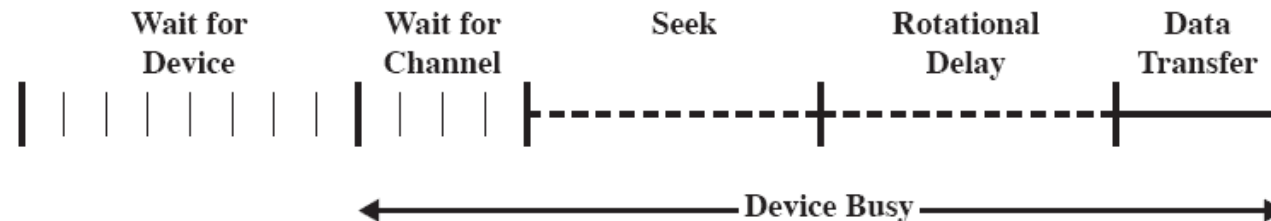


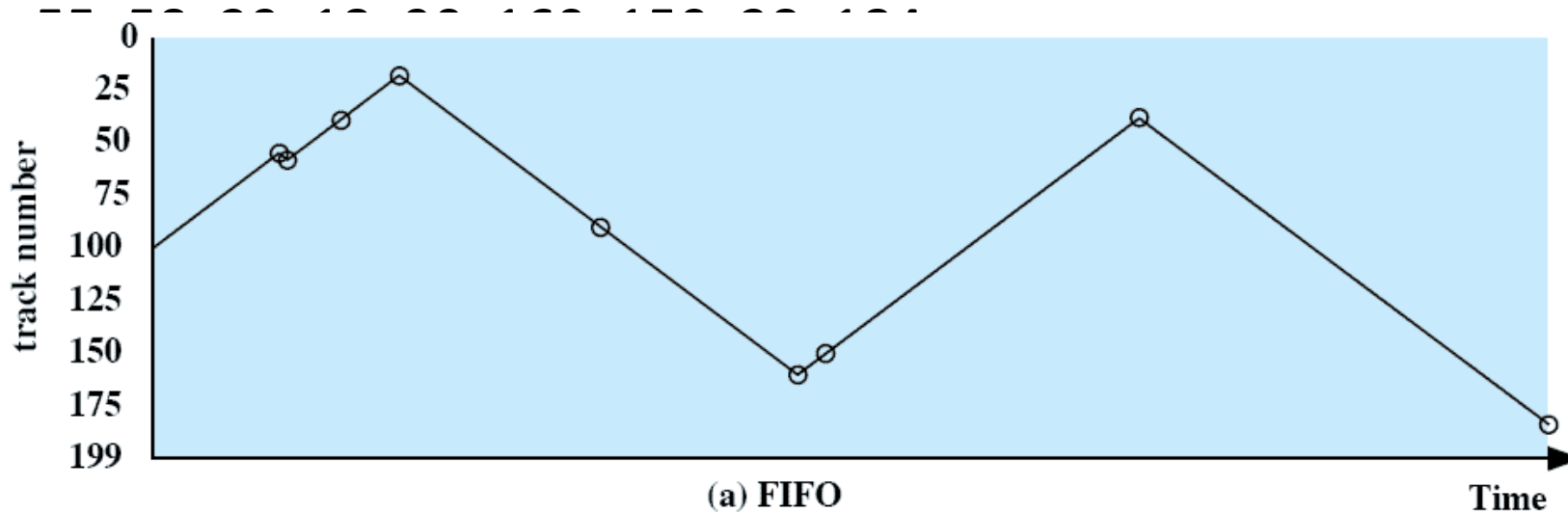
Figure 11.6 Timing of a Disk I/O Transfer

- 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.

- ***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.

- 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.

- Process request sequentially
- Fair to all processes
- this technique will often approaches random scheduling in performance if there are many processes

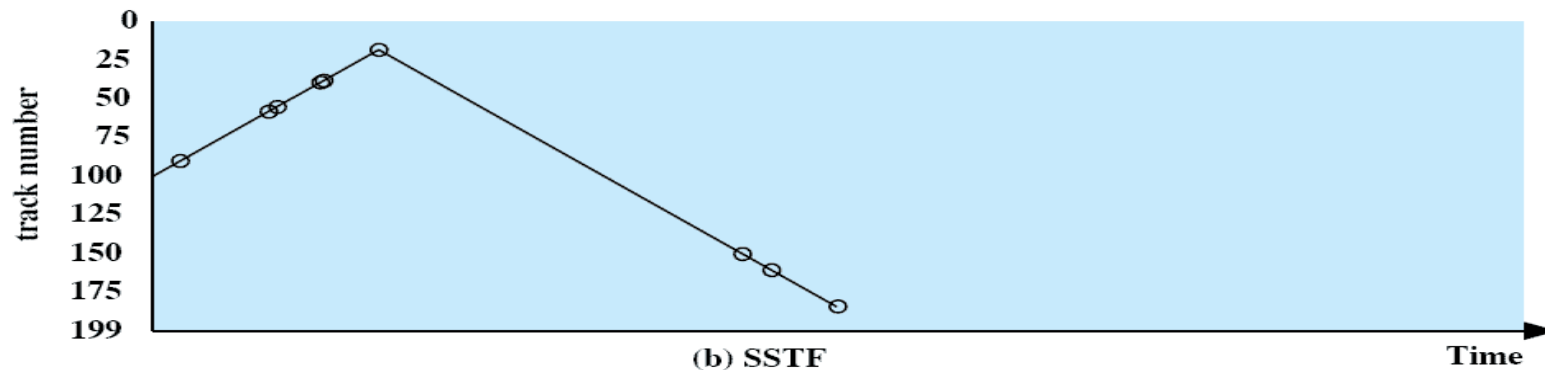


- Goal is not to optimize disk use but to meet other objectives
- Short batch jobs may have higher priority
- Provide good interactive response time
- Longer jobs may have to wait an excessively long time
- A poor policy for database systems

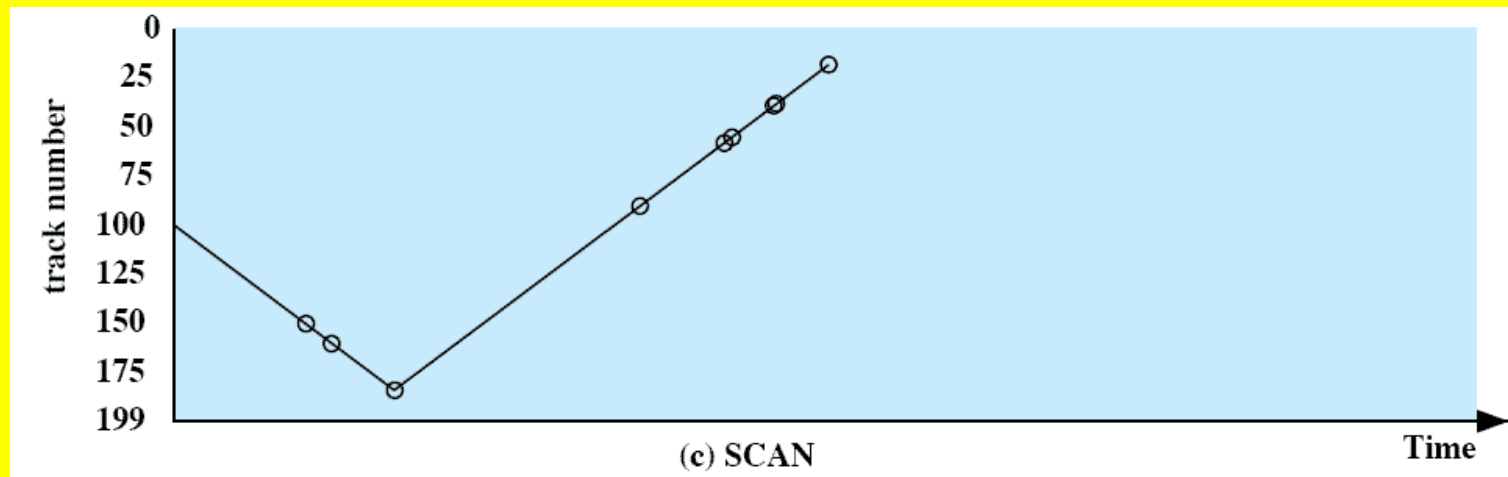
- Good for transaction processing systems
 - The device is given to the most recent user so there should be little arm movement
- Possibility of starvation since a job may never regain the head of the line

Shortest Service Time First (SSTF)

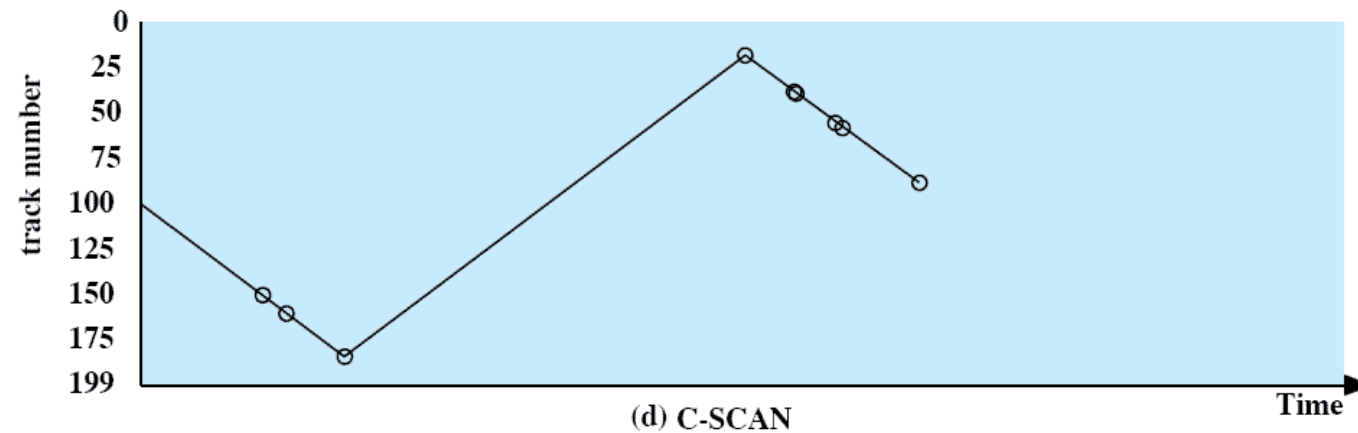
- Select the disk I/O request that requires the least movement of the disk arm from its current position
- Always choose the minimum seek time
- 55, 58, 39, 18, 90, 160, 150, 38, 184.



- Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction then the direction is reversed
- 55, 58, 39, 18, 90, 160, 150, 38, 184.



- 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.



- Segments the disk request queue into subqueues of length N
- Subqueues are processed one at a time, using SCAN
- New requests added to other queue when queue is processed

- Two subqueues
- When a scan begins, all of the requests are in one of the queues, with the other empty.
- All new requests are put into the other queue.
 - Service of new requests is deferred until all of the old requests have been processed.

Comparison of Disk Scheduling Algorithms

(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

End of Module 6- Part2

