MODULE-4

Device management

#### Device Management in Operating System manages device communication via their respective drivers. It does the following activities for device management:

* Keeps tracks of all devices. The program responsible for this task is known as the I/O controller.
* Decides which process gets the device when and for how much time.
* Allocates the device in the most efficient way.
* De-allocates devices.
  + The main motive of Device Management is to ease the usage of hardware. A Device Manager controls the processing of drivers. The drivers function as a pathway between the hardware and software. Due to various kinds of hardware present in the market various hardware specific drivers are also present. This helps the operating system to maintain its rigidity without the need to modify itself for different hardware whenever a new hardware is installed.
  + Operating system maintains the system for easy and secure usage. It defines data as files. A driver translates the files into streams of bits to be kept on the storage devices at specific locations. The drivers are assigned high priority blocks by the operating system to fasten the use of hardware subsystems.
  + Input and output operations are done by managing queues. Queues are special storage facilities that inputs data translated into streams of bit by driver from any device and transfers them into the CPU at the threshold intake limit of the CPU. When a input from a process is suspended, the flow of input will continue but the queue will stop taking data from the device. When the sleeping process will become active again, the queue will start transmitting data once again. This procedure maintains an intelligent use of the processor time when a number of operations are simultaneously running and also allows external input devices like a keyboard or modem to manage various external users although there are instances when the personal computer cannot use input from these devices.

**device manager:** Device Manager is used to manage the hardware devices installed in a computer like [hard disk drives,](https://www.lifewire.com/what-is-a-hard-disk-drive-2618152) [keyboards](https://www.lifewire.com/what-is-a-keyboard-2618153), [sound cards](https://www.lifewire.com/what-is-a-sound-card-2618160), [USB devices](https://www.lifewire.com/universal-serial-bus-usb-2626039), and more.

Device Manager can be used for changing hardware configuration options, managing [drivers](https://www.lifewire.com/what-is-a-device-driver-2625796), disabling and enabling hardware, identifying conflicts between hardware devices, and much more.

Think of Device Manager as the master list of hardware that Windows understands. All the hardware on your computer can be configured from this centralized utility.

**device driver:**A device driver is a [program](https://searchsoftwarequality.techtarget.com/definition/program) that controls a particular type of [device](https://whatis.techtarget.com/definition/device) that is attached to your computer. There are device [driver](https://searchstorage.techtarget.com/definition/driver)s for printers, displays, CD-ROM readers, diskette drives, and so on. When you buy an [operating system](https://whatis.techtarget.com/definition/operating-system-OS), many device drivers are built into the product. However, if you later buy a new type of device that the operating system didn't anticipate, you'll have to install the new device driver. A device driver essentially converts the more general input/output instructions of the operating system to messages that the device type can understand.

**Technique for device management**

#### There are different kinds of devices

1. **dedicated devices**: From the name only we get that **dedicated devices** are allotted for a dedicated work. The devices that are assigned for only one job at a time are called Dedicated Devices. These dedicated devices run for the entire time serving only a single job when it is active. Examples of dedicated devices are as follows - tape drives, plotters and printers. These kind of dedicated devices are easy to use but they take a lot of time to execute a list of commands since they work each command at a time.

Disadvantages

With the advent of more efficient device management it is easy for the system to assign multiple jobs for the Central Processing Unit or the CPU. Intelligent multitasking reduces operating time and speeds up the system without heavy loading on the CPU.

As we know that dedicated devices are assigned for only one job at a time, it slows down the whole process.

Another disadvantage of the dedicated devices lie in the field that only a single user is allocated to them during the duration of the whole job’s execution. This prevents usage of multiple users which is not of good market value. Also during a single job another work cannot be assigned to a dedicated device.

A dedicated device will take the next job until and unless the previous job has been executed. All these leads to the formation of job queue which takes up important memory spaces in the Random **A**ccess **M**emory or RAM slowing down the whole system. For this reasons, all the other types of devices are preferred over **dedicated devices** since other devices canbe assigned multiple processes.

the Random Access Memory or RAM slowing down the whole system. For this reasons, all the other types of devices are preferred over dedicated devices since other devices canbe assigned multiple processes.

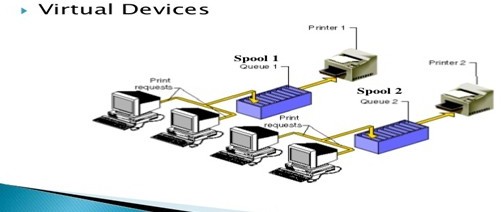
1. **shared devices:** Shared devices are the devices that can be **shared among several processes**. They can be assigned to more than one process at a time. These include **direct access storage devices** such as magnetic disks and optical discs.

* For example, consider a situation where two processes p1 and p2 want to access a disk. If the disk can be shared among them then, p1 will access the disk for some time and then p2 will access the disk for some time and this goes on until p1 and p2 finish their tasks.
* This sharing is achieved by interleaving the requests assigned to a device. The **device manager** here uses predefined policies to decide which process can access the device first and for how long it can access it. Then the device is assigned to the next process for some time as decided by the device manager.
* All the conflicts that may rise due to sharing of the devices are handled with the help of these predefined policies of the operating system.
* They are efficient than the dedicated devices which allow the allocation of a device to a single job only until that job completes its execution. Ex. Magnetic disk storage, optical disk storage, flash memory etc.

#### **irtual device:** In general Virtual devices are dedicated devices which are transformed into Shared devices through the technique of spooling. E.g. – printers and faxes use spooling technique by rerouting all the print requests to a disk. The printer receives output only when the output is complete and the printer is ready to print out the entire document.

So we get that a virtual device is the combination of the other kind of devices namely dedicated devices and shared devices. There is no general method to install the virtual devices.

Also a device ID is not exposed by the Virtual devices. Virtual devices take out the good points from a bunch of network specific appliances. Since disks are sharable devices, the spooling technique can convert the printer into virtual device or virtual printer, enhancing its performance manifolds.



**Advantages of Virtual Devices**

* + There will be no need to have a second system while testing the operating system.
  + All generic hardware found in other pc will be provided by the virtual machine.
  + More enhanced user experience.
  + Very good quality of service.
  + The most important advantage of all is that the virtual device manages to retain best of both the dedicated devices and shared devices. Virtual devices can complete several works at a time without creating pressure on the system.

**DMA :**Direct memory access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (random-access memory), independent of the central processing unit (CPU).

Stands for "Direct Memory Access." DMA is a method of transferring data from the computer's [RAM](https://techterms.com/definition/ram) to another part of the computer without processing it using the [CPU](https://techterms.com/definition/cpu). While most data that is input or output from your computer is processed by the CPU, some data does not require processing, or can be processed by another device. In these situations, DMA can save processing time and is a more efficient way to move data from the computer's memory to other devices.

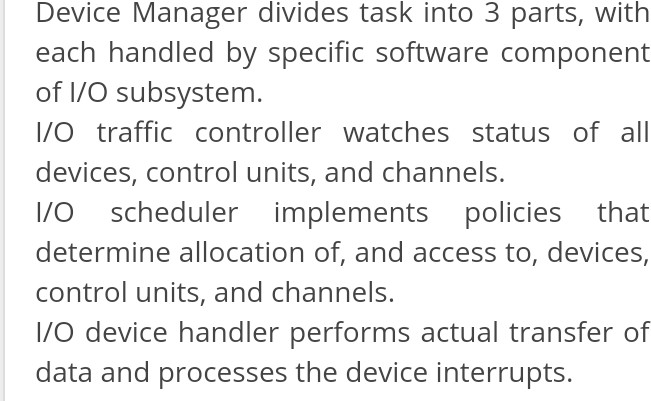
For example, a sound card may need to access data stored in the computer's RAM, but since it can process the data itself, it may use DMA to bypass the CPU. Video cards that support DMA can also access the system memory and process graphics without needing the CPU. [Ultra DMA](https://techterms.com/definition/ultradma) hard drives use DMA to transfer data faster than previous hard drives that required the data to first be run through the CPU.

In order for devices to use direct memory access, they must be assigned to a DMA channel. Each type of port on a computer has a set of DMA channels that can be assigned to each connected device. For example, a [PCI](https://techterms.com/definition/pci) controller and a [hard drive](https://techterms.com/definition/harddrive)controller each have their own set of DMA channels.

### DEVICE ALLOCATION CONSIDERATION

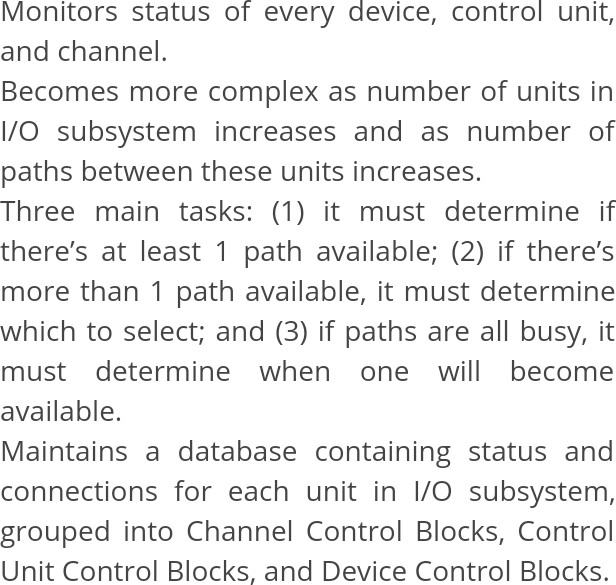
#### what are the function of the following?

1. i/o Traffic Controller
2. i/o Device Handler
3. i/o scheduler



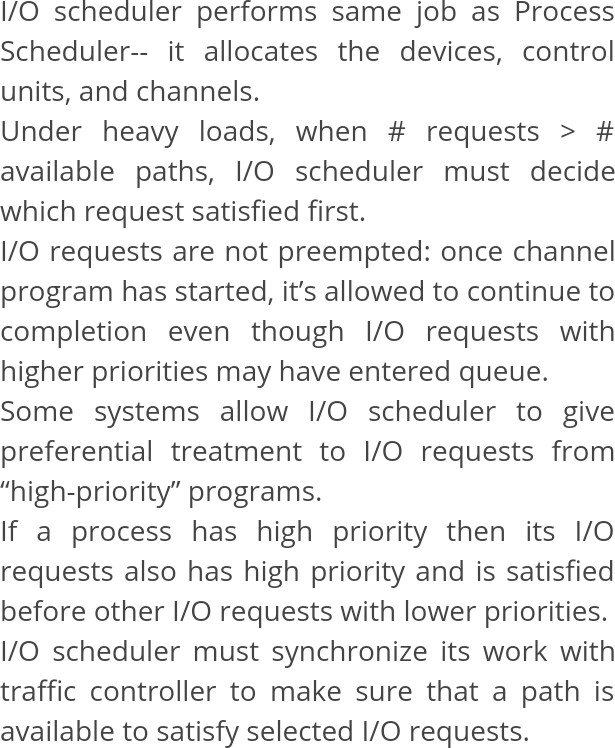
Followings are functions of the above described operating system modules,

1. i/o Traffic Controller: Keeps track of status of all devices, control units and channels.



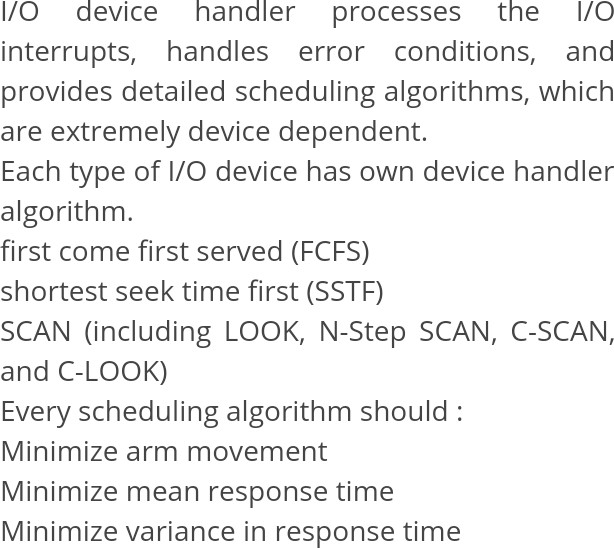
1. i/o scheduler: As the name indicates it just implements the scheduling algorithm used to allocate channel, control

unit and device access, to complete i/o request of a job.



1. i/o Device Handler: Performs the actual dynamic allocation , once the i/o scheduler has made the decision,

by constructing the program, issuing the start i/o instruction and processing the device interrupts. there is usually separate device handler for each type of device.



**Disk Scheduling Algorithms**

Disk scheduling is is done by operating systems to schedule I/O requests arriving for disk. Disk scheduling is also known as I/O scheduling.

Disk scheduling is important because:

Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by disk controller. Thus other I/O requests need to wait in waiting queue and need to be scheduled.

Two or more request may be far from each other so can result in greater disk arm movement.

Hard drives are one of the slowest parts of computer system and thus need to be accessed in an efficient manner.

There are many Disk Scheduling Algorithms but before discussing them let’s have a quick look at some of the important terms:

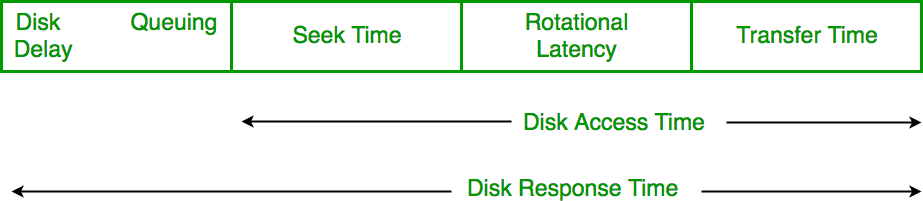
Seek Time:Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.

Rotational Latency: Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.

Transfer Time: Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.

Disk Access Time: Disk Access Time is:

Disk Access Time = Seek Time +Rotational Latency + Transfer Time



Disk Response Time: Response Time is the average of time spent by a request waiting to perform its I/O operation. Average Response time is the response time of the all requests. Variance Response Time is measure of how individual request are serviced with respect to average response time. So the disk scheduling algorithm that gives minimum variance response time is better.

**Disk Scheduling Algorithms**

FCFS: FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.

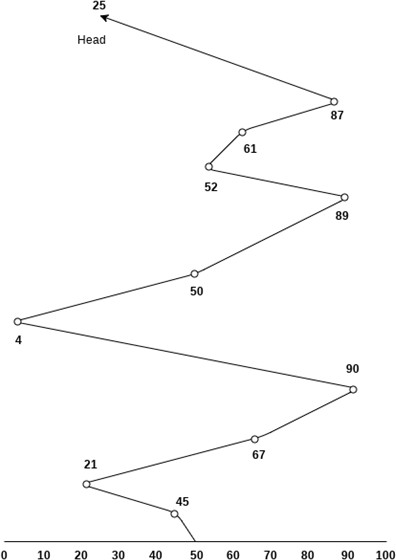
Example

Consider the following disk request sequence for a disk with 100 tracks 45, 21, 67, 90,

4, 50, 89, 52, 61, 87, 25

Head pointer starting at 50 and moving in left direction. Find the number of head movements in cylinders using FCFS scheduling.

## Solution



Number of cylinders moved by the head

= (50-45)+(45-21)+(67-21)+(90-67)+(90-4)+(50-4)+(89-50)+(61-52)+(87-61)+(87-25)

= 5 + 24 + 46 + 23 + 86 + 46 + 49 + 9 + 26 + 62

= 376

##### Advantages:

Every request gets a fair chance No indefinite postponement **Disadvantages:**

Does not try to optimize seek time

May not provide the best possible service

SSTF:

Shortest seek time first (SSTF) algorithm selects the disk I/O request which requires the least disk arm movement from its current position regardless of the direction. It reduces the total seek time as compared to FCFS.

It allows the head to move to the closest track in the service queue.

Disadvantages

* It may cause starvation for some requests.
* Switching direction on the frequent basis slows the working of algorithm.
* It is not the most optimal algorithm.

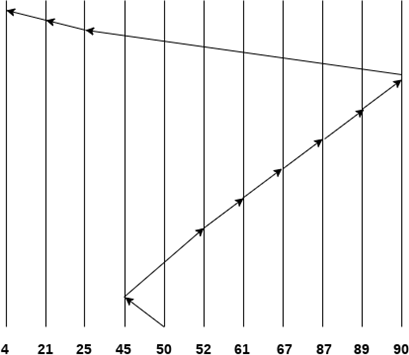
Example

Consider the following disk request sequence for a disk with 100 tracks

45, 21, 67, 90, 4, 89, 52, 61, 87, 25

Head pointer starting at 50. Find the number of head movements in cylinders using SSTF scheduling.

## Solution:



Number of cylinders = 5 + 7 + 9 + 6 + 20 + 2 + 1 + 65 + 4 + 17 = 136

# Scan Algorithm

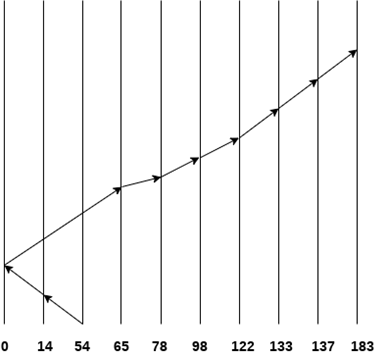
It is also called as Elevator Algorithm. In this algorithm, the disk arm moves into a particular direction till the end, satisfying all the requests coming in its path,and then it turns backand moves in the reverse direction satisfying requests coming in its path.

It works in the way an elevator works, elevator moves in a direction completely till the last floor of that direction and then turns back.

## Example

Consider the following disk request sequence for a disk with 100 tracks 98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using SCAN scheduling.



Number of Cylinders = 40 + 14 + 65 + 13 + 20 + 24 + 11 + 4 + 46 = 237

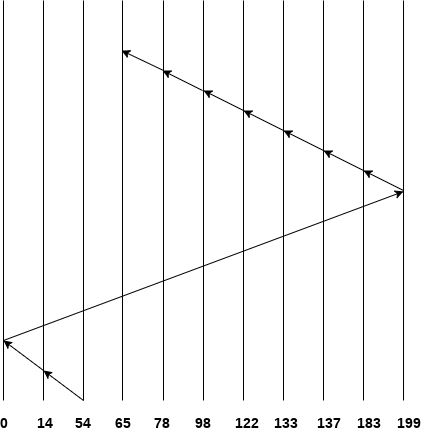
# C-SCAN algorithm

In C-SCAN algorithm, the arm of the disk moves in a particular direction servicing requests until it reaches the last cylinder, then it jumps to the last cylinder of the opposite direction without servicing any request then it turns back and start moving in that direction servicing the remaining requests.

## Example

Consider the following disk request sequence for a disk with 100 tracks 98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using C-SCAN scheduling.



No. of cylinders crossed = 40 + 14 + 199 + 16 + 46 + 4 + 11 + 24 + 20 + 13 = 387

**Advantages:**

High throughput

Low variance of response time Average response time **Disadvantages:**

Long waiting time for requests for locations just visited by disk arm

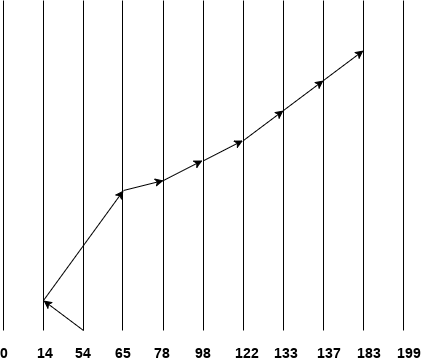
Look Scheduling

It is like SCAN scheduling Algorithm to some extant except the difference that, in this scheduling algorithm, the arm of the disk stops moving inwards (or outwards) when no more request in that direction exists. This algorithm tries to overcome the overhead of SCAN algorithm which forces disk arm to move in one direction till the end regardless of knowing if any request exists in the direction or not.

Example **Consider the following disk request sequence for a disk with 100 tracks**

98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using LOOK scheduling.



Number of cylinders crossed = 40 + 51 + 13 + +20 + 24 + 11 + 4 + 46 = 209

# C Look Scheduling

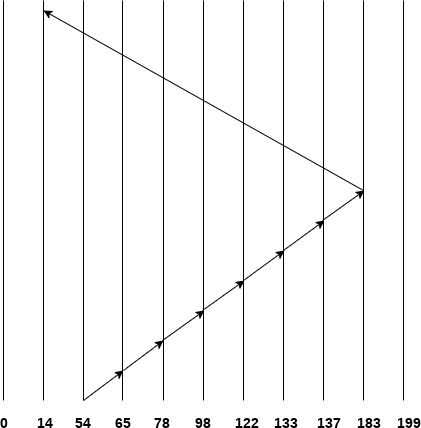
C Look Algorithm is similar to C-SCAN algorithm to some extent. In this algorithm, the arm of the disk moves outwards servicing requests until it reaches the highest request cylinder, then it jumps to the lowest request cylinder without servicing any request then it again start moving outwards servicing the remaining requests.

It is different from C SCAN algorithm in the sense that, C SCAN force the disk arm to move till the last cylinder regardless of knowing whether any request is to be serviced on that cylinder or not.

## Example

Consider the following disk request sequence for a disk with 100 tracks 98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using C LOOK scheduling.



Number of cylinders crossed = 11 + 13 + 20 + 24 + 11 + 4 + 46 + 169 = 298

**Analysis**

Disk scheduling algorithms are used to allocate the services to the I/O requests on the disk and improve its performance. Different qualities exists on these algorithms based on the given examples and computations. Several disadvantages also occur on these different algorithm and these are:

* The FCFS performs operations in order requested. No reordering of work queue since it processed disk requests according to its arrival. There is no starvation and all the requests are serviced but it doesn’t provide fastest service

.  The Shortest Seek Time First (SSTF) selects the disk I/O request that requires the least movement of the disk access arm from its current position regardless of direction. It also reduces the seek time compared to FCFS but in this algorithm, I/O requests at the edges of the disk surface may get starved.

* The SCAN algorithm go from the outside to the inside servicing requests and then back from the outside to the inside servicing requests. It also reduces variance compared to SSTF.
* The Circular SCAN (C-SCAN) moves from one end of the disk to the other, servicing requests. When other end is reached, it immediately returns to the beginning of the disk, without servicing any requests. This algorithm treats the cylinders as a circular list that wraps around from the last cylinder to the first one. It also provides a more uniform wait time than SCAN.
* In LOOK scheduling algorithm, the arm goes only as far as the final request in each direction. The direction reverses immediately, without going all the way to the end of the disk.
* The Circular LOOK (C-LOOK) algorithm is similar to C-SCAN. The disk head also goes as far as the last request in its direction then reverses its direction immediately without first going all the way to the end of the disk. When selecting a Disk Scheduling algorithm, performance depends on the number and types of requests. SSTF is common and has a natural appeal. SCAN gives better performance than FCFS and SSTF. 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 SCAN or C-LOOK is a reasonable choice for the default algorithm.

**What exactly Spooling is all about?**

SPOOL is an acronym for **simultaneous peripheral operations on-line**. It is a kind of buffering mechanism or a process in which data is temporarily held to be used and executed by a device, program or the system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.

In a computer system peripheral equipments, such as printers and punch card readers etc (batch processing), are very slow relative to the performance of the rest of the system. Getting input and output from the system was quickly seen to be a bottleneck. Here comes the need for spool.

Spooling works like a typical request queue where data, instructions and processes from multiple sources are accumulated for execution later on. Generally, it is maintained on computer’s physical memory, buffers or the I/O device-specific interrupts. The spool is processed in FIFO manner i.e. whatever first instruction is there in the queue will be popped and executed.

##### Applications/Implementations of Spool:

1. The most common can be found in I/O devices like keyboard printers and mouse. For example, In printer, the documents/files that are sent to the printer are first stored in the memory or the printer spooler. Once the printer is ready, it fetches the data from the spool and prints it.

Even experienced a situation when suddenly for some seconds your mouse or keyboard stops working? Meanwhile, we usually click again and again here and there on the screen to check if its working or not. When it actually starts working, what and wherever we pressed during its hang state gets executed very fast because all the instructions got stored in the respective device’s spool.

1. A batch processing system uses spooling to maintain a queue of ready-to-run jobs which can be started as soon as the system has the resources to process them.
2. Spooling is capable of overlapping I/O operation for one job with processor operations for another job. i.e. multiple processes can write documents to a print queue without waiting and resume with their work.
3. E-mail: an email is delivered by a MTA (Mail Transfer Agent) to a temporary storage area where it waits to be picked up by the MA (Mail User Agent)
4. Can also be used for generating Banner pages (these are the pages used in computerized printing in order to separate documents from each other and to identify e.g. the originator of the print request by username, an account number or a bin for pickup. Such pages are used in office environments where many people share the small number of available resources).