UNIT 5

INPUT/OUTPUT Management System

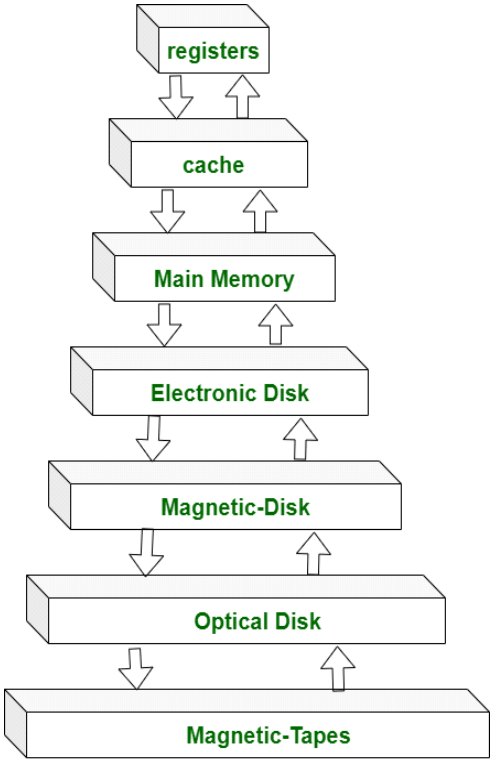
Mass-Storage Management

main memory is too small to accommodate all data and programs, and because the data that it holds are lost when power is lost, the computer system must provide secondary storage to back up main memory. Most modern computer systems use disks as the principal on-line storage medium for both programs and data. Most programs—including compilers, assemblers, word processors, editors, and formatters—are stored on a disk until loaded into memory and then use the disk as both the source and destination of their processing. Hence, the proper management of disk storage is of central importance to a computer system.

There are two types of storage devices:-

- **Volatile Storage Device –**
  It looses its contents when the power of the device is removed.
- **Non-Volatile Storage device –**
  It does not loses its contents when the power is removed. It holds all the data when the power is removed.

this fig Hierarchy of storage is shown –



Storage Device Hierarchy

In this hierarchy all the storage devices are arranged according to speed and cost. The higher levels are expensive, but they are fast. As we move down the hierarchy, the cost per bit generally decreases, where as the access time generally increases.

The storage systems above the Electronic disk are Volatile, where as those below are Non-Volatile.

The Basic mass storage devices are:

1.Magnetic Disk 2.Magnetic tapes

1.Magnetic Disk :

It is coated magnetically and has tracks, spots, and sectors for storing data.

Basic Common Examples of Magnetic Disks are:

Floppy Disks

Hard Disks

Zip Disks

It is a collection of platters.

sectors are used to store data.

Read/Write operations are used to perform operations on sector.

3 Performance parameters are used to find the performance of magnetic disk.

- **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.

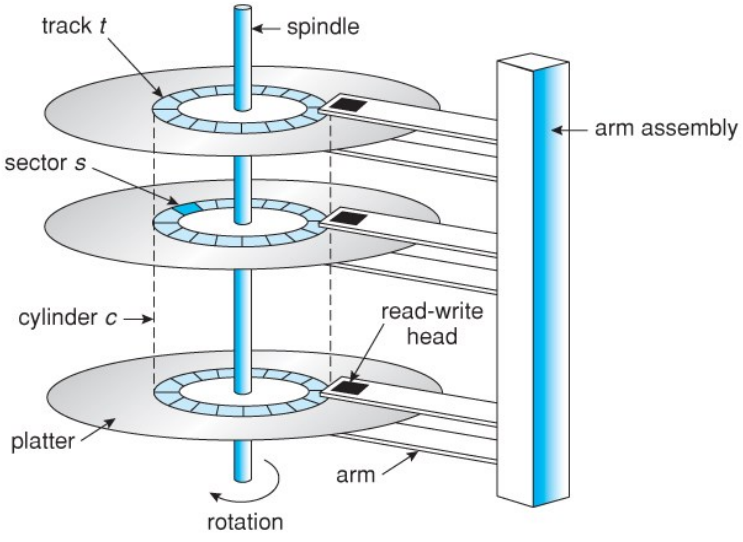
2.Magnetic tapes:

Magnetic tapes were once used for common secondary storage before the days of hard disk drives, but today are used primarily for backups.

Accessing a particular spot on a magnetic tape can be slow, but once reading or writing commences, access speeds are comparable to disk drives.

Capacities of tape drives can range from 20 to 200 GB, and compression can double that capacity.

Disk structure



Traditional magnetic disks have the following basic structure:

One or more platters in the form of disks covered with magnetic media. Hard disk platters are made of rigid metal, while "floppy" disks are made of more flexible plastic.

Each platter has two working surfaces. Older hard disk drives would sometimes not use the very top or bottom surface of a stack of platters, as these surfaces were more susceptible to potential damage.

Each working surface is divided into a number of concentric rings called tracks. The collection of all tracks that are the same distance from the edge of the platter, (i.e. all tracks immediately above one another in the following diagram) is called a cylinder.

Each track is further divided into sectors, traditionally containing 512 bytes of data each, although some modern disks occasionally use larger sector sizes. (Sectors also include a header and a trailer, including checksum information among other things. Larger sector sizes reduce the fraction of the disk consumed by headers and trailers, but increase internal fragmentation and the amount of disk that must be marked bad in the case of errors.)

The data on a hard drive is read by read-write heads. The standard configuration (shown below) uses one head per surface, each on a separate arm, and controlled by a common arm assembly which moves all heads simultaneously from one cylinder to another. (Other configurations, including independent read-write heads, may speed up disk access, but involve serious technical difficulties.)

The storage capacity of a traditional disk drive is equal to the number of heads (i.e. the number of working surfaces), times the number of tracks per surface, times the number of sectors per track, times the number of bytes per sector. A particular physical block of data is specified by providing the head-sector-cylinder number at which it is located.

Disk Attachment

Introduction:-Computers access disk storage in two ways. One way is via I/O ports (or host-attached storage); this is common on small systems. The other way is via a remote host in a distributed file system; this is referred to as network-attached storage.

Host-Attached Storage:-Host-attached storage is storage accessed through local I/O ports. These ports use several technologies. The typical desktop PC uses an I/O bus architecture called IDE or ATA. This architecture supports a maximum of two drives per I/O bus. A newer,

similar protocol that has simplified cabling is SATA. High-end workstations and servers generally use more sophisticated I/O architectures, such as SCSI and fiber channel (FC).

SCSI is a bus architecture. Its physical medium is usually a ribbon cable having a large number of conductors. The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets). A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target. A typical use of logical unit addressing is to direct commands to components of a RATD array or components of a removable media library (such as a CD jukebox sending commands to the media-changer mechanism or to one of the drives).

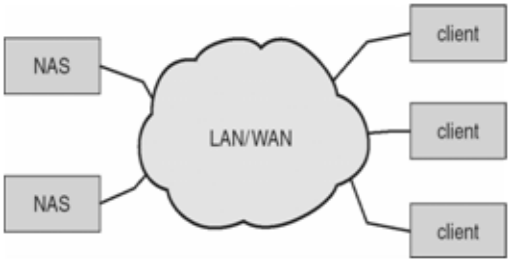
FC is a high-speed serial architecture that can operate over optical fiber or over a four-conductor copper cable. It has two variants. One is a large switched fabric having a 24-bit address space. This variant is expected to dominate in the future and is the basis of storage-area networks (SANs). Because of the large address space and the switched nature of the communication, multiple hosts and storage devices can attach to the fabric, allowing great flexibility in I/O communication. The other PC variant is an arbitrated loop (FC-AL) that can address 126 devices (drives and controllers).

A wide variety of storage devices are suitable for use as host-attached storage. Among these are hard disk drives, RAID arrays, and CD, DVD, and tape drives. The I/O commands that initiate data transfers to a host-attached storage device are reads and writes of logical data blocks directed to specifically identified storage units.

Network-Attached Storage: - A network-attached storage (NAS) device is a special-purpose storage system that is accessed remotely over a data network. Clients access network-attached storage via a remote-procedure-call interface such as NFSfor UNIX systems or CIFS for Windows machines. The remote procedure calls(RPCs) are carried via TCP or UDP over an IP network, usually the same local-area network (LAN) that carries all data traffic to the clients. The network attached storage unit is usually implemented as a RAID array with software that implements the RPC interface. It is easiest to think of NAS as simply another storage-access protocol. For example, rather than using a SCSI device driver and SCSI protocols to access storage, a system using NAS would use RPC overTCP/IP.

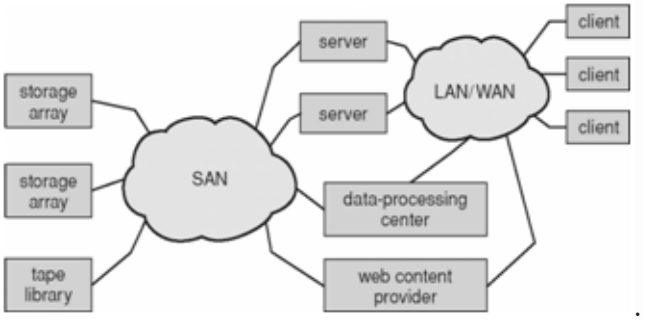
Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage. However, it tends to be less efficient and have lower performance than some direct-attached storage options.

ISCSI is the latest network-attached storage protocol. In essence, it uses the IP network protocol to carry the SCSI protocol. Thus, networks rather than SCSI cables can be used as the interconnects between hosts and their storage. As a result, hosts can treat their storage as if it were directly attached, but the storage can be distant from the host.



Storage-Area Network: -One drawback of network-attached storage systems is that the storage I/O operations consume bandwidth on the data network, thereby increasing the latency of network communication. This problem can be particularly acute in large client-server installations—the communication between servers and clients competes for bandwidth with the communication among servers and storage devices.

A storage-area network (SAN) is a private network (using storage protocols rather than networking protocols) connecting servers and storage units, as shown in Figure. The power of a SAN lies in its flexibility. Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts. A SAN switch allows or prohibits access between the hosts and the storage. As one example, if a host is running low on disk space, the SAN can be configured to allocate more storage to that host



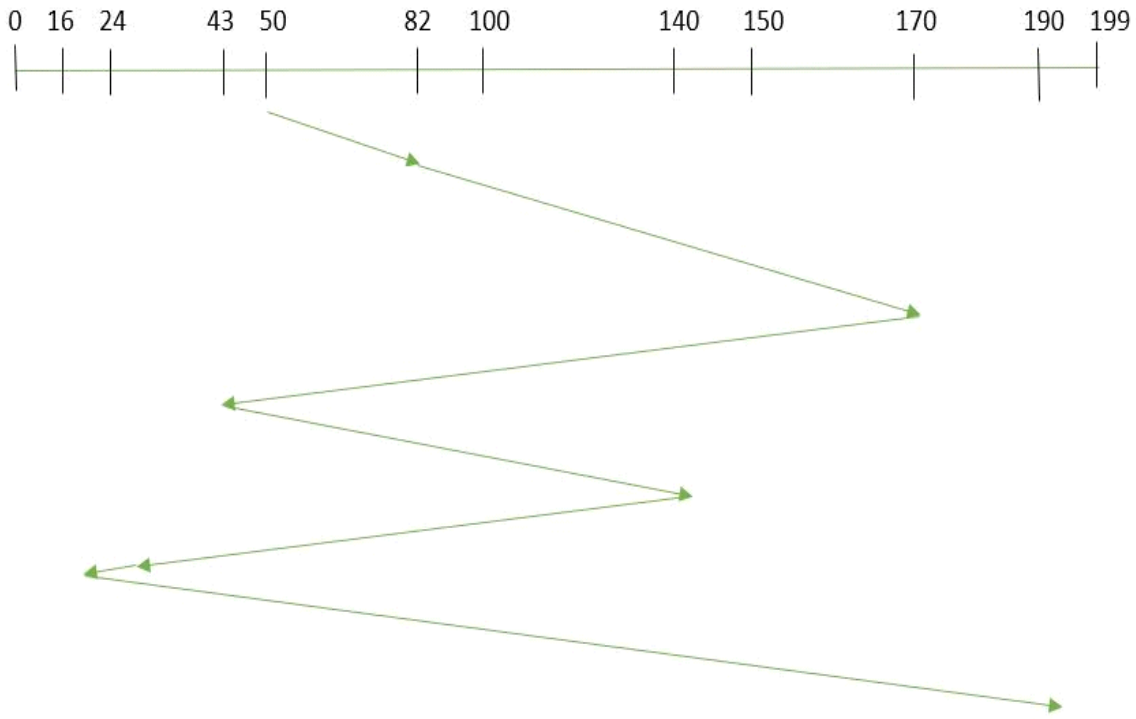
Disk Scheduling Algorithms:

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.Let us understand this with the help of an example.

Example:

Suppose the order of request is- (82,170,43,140,24,16,190)

And current position of Read/Write head is : 50



So, total seek time:

=(82-50)+(170-82)+(170-43)+(140-43)+(140-24)+(24-16)+(190-16)

=642

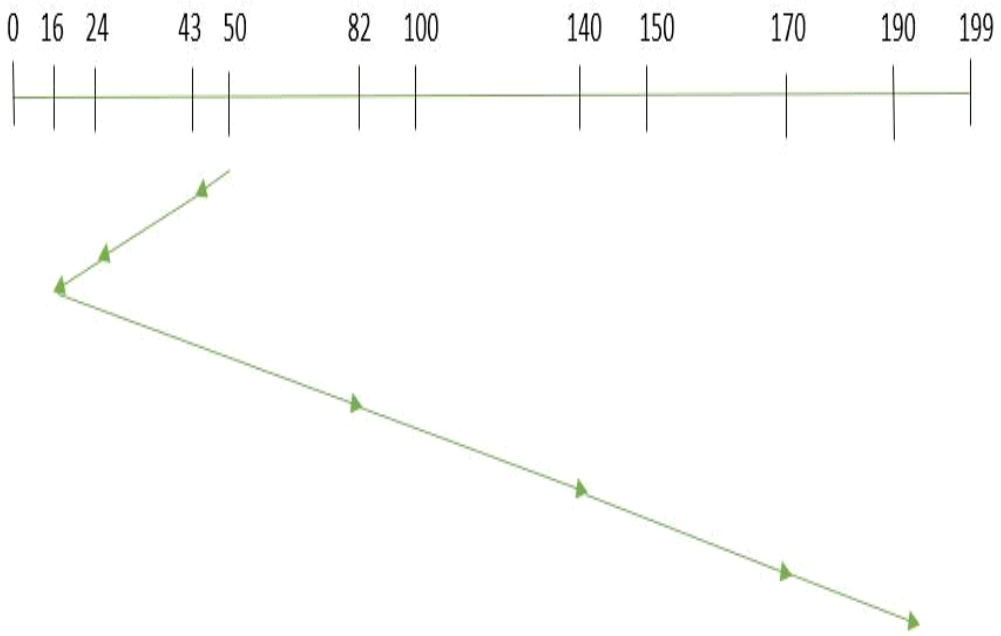
SSTF: In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk

arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system.Let us understand this with the help of an example.

Example:

Suppose the order of request is- (82,170,43,140,24,16,190)

And current position of Read/Write head is : 50



So, total seek time:

=(50-43)+(43-24)+(24-16)+(82-16)+(140-82)+(170-40)+(190-170)

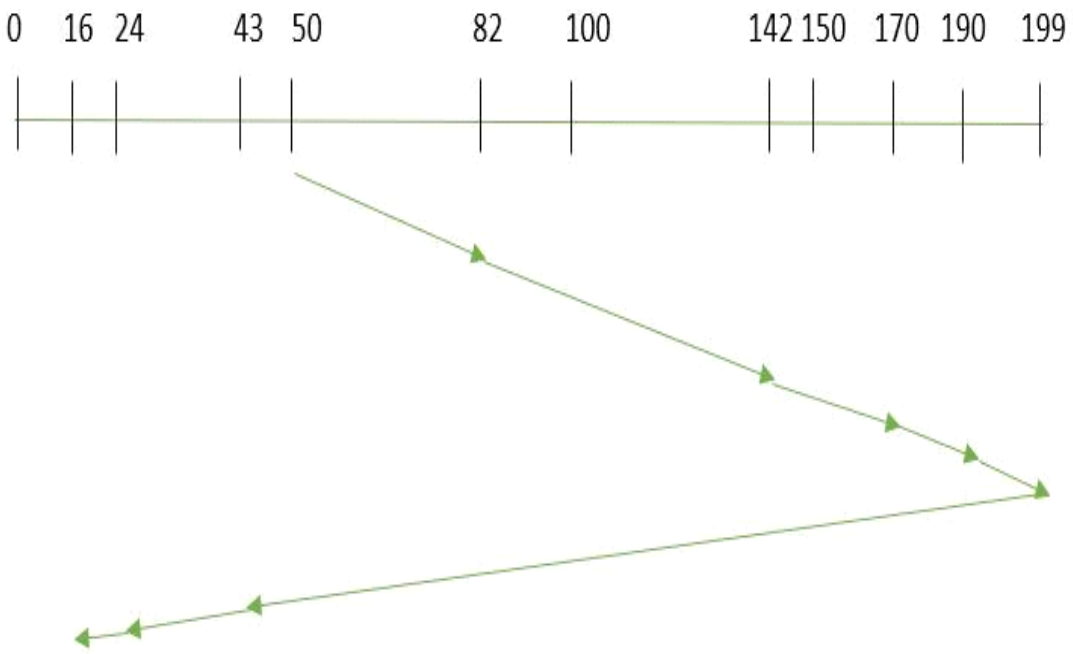
=208

SCAN: In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works as an elevator and hence also known as **elevator algorithm.** As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

Example:

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm

is at 50, and it is also given that the disk arm should move **"towards the larger value".**



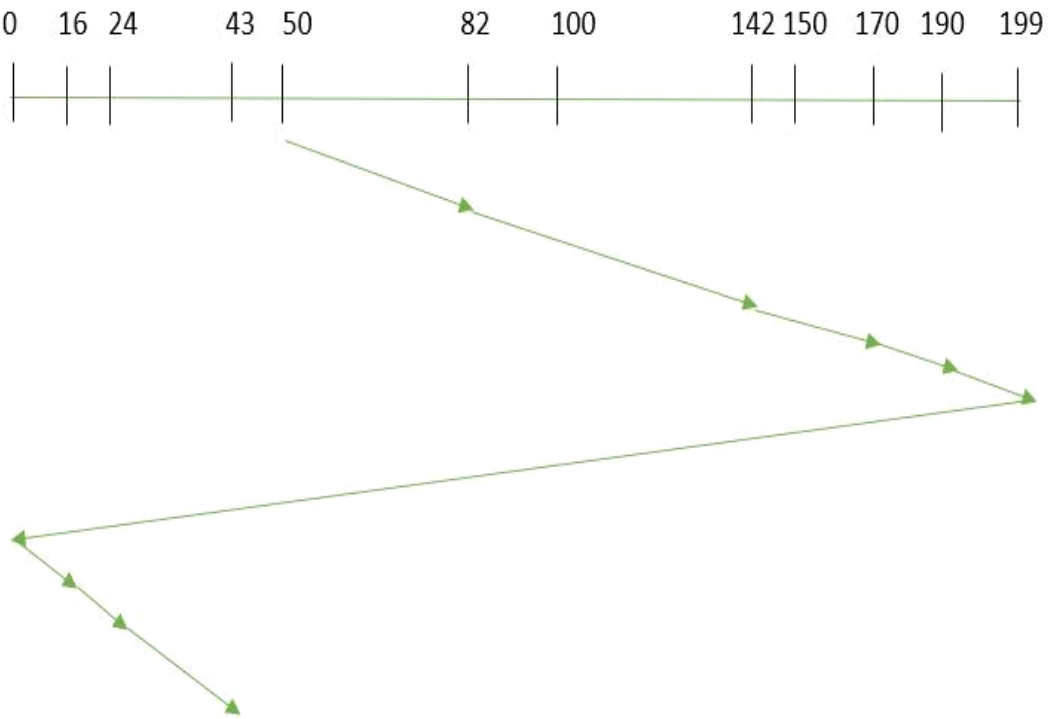
Therefore, the seek time is calculated as:
=(82-50)+(140-82)+(170-140)+(190-170)+(190-43)+(43-24)+(24-16)

=314

CSCAN: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

These situations are avoided in *CSCAN* algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Example:Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move "towards the larger value".



Seek time is calculated as:

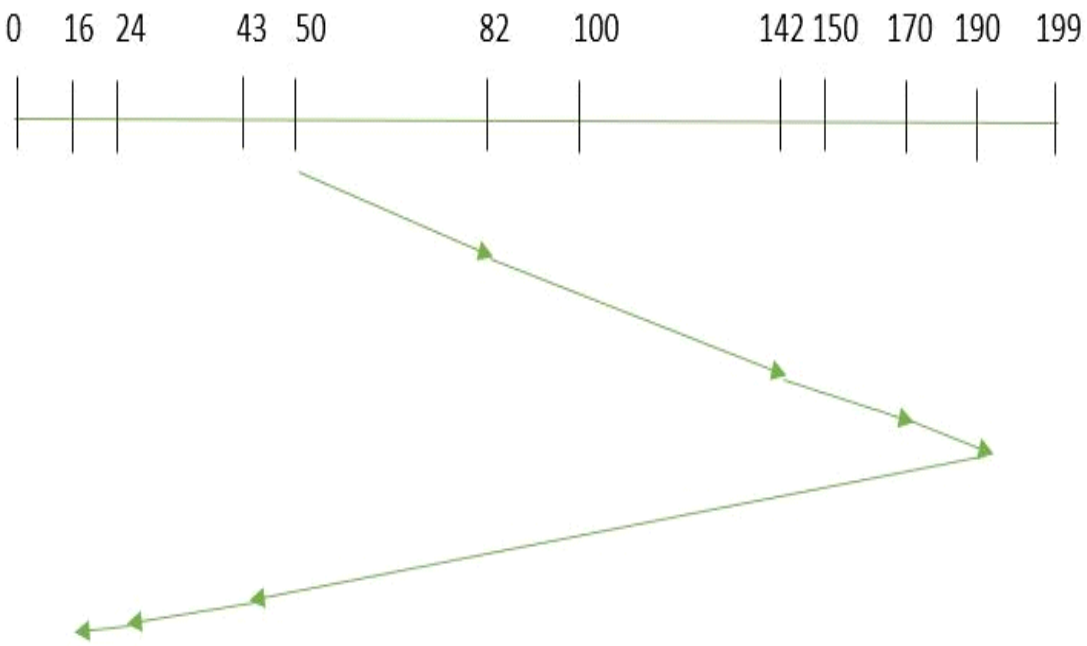
=(82-50)+(140-82)+(170-140)+(190+170)+(199-190)+(16-0)+(24-16)+(43-24)

=192

LOOK: It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Example:
Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm

is at 50, and it is also given that the disk arm should move **“towards the larger value”.**



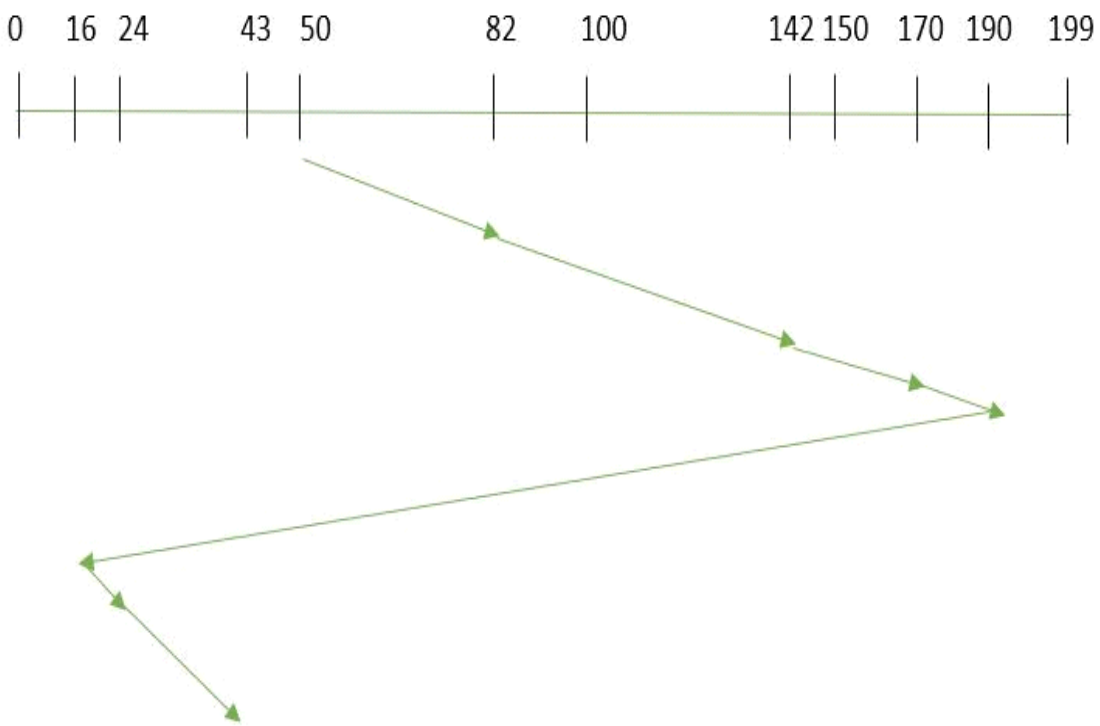
So, the seek time is calculated as:

=(82-50)+(140-82)+(170-140)+(190-170)+(190-43)+(43-24)+(24-16)
=314

=**CLOOK:** As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end’s last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Example:

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **“towards the larger value”**



So, the seek time is calculated as above

=(82-50)+(140-82)+(170-140)+(190-170)+(190-16)+(24-16)+(43-24)

=341

Swap-Space Management

Swapping is a memory management technique used in multi-programming to increase the number of processes sharing the CPU. It is a technique of removing a process from main memory and storing it into secondary memory, and then bringing it back into main memory for continued execution. This action of moving a process out from main memory to secondary memory is called **Swap Out** and the action of moving a process out from secondary memory to main memory is called **Swap In**.

Swap-Space : The area on the disk where the swapped out processes are stored is called swap space.

Swap-Space Management : Swap-Swap management is another low-level task of the operating system. Disk space is used as an extension of main memory by virtual memory. As we know the fact that disk access is much slower than memory access, In the swap-space management we are using disk space, so it will significantly decrease system performance.

Basically, in all our systems we require the best throughput, so the goal of this swap-space implementation is to provide the virtual memory the best throughput
Swap-Space Use : Swap-space is used by the different operating-systems in various ways. The systems which are implementing swapping may use swap space to hold the entire process which may include image, code and data segments. Paging systems may simply store pages that have been pushed out of the main memory. The need for swap space on a system can vary from a megabytes to gigabytes but it also depends on the amount of physical memory, the virtual memory it is backing and the way in which it is using the virtual memory.

It is safer to overestimate than to underestimate the amount of swap space required, because if a system runs out of swap space it may be forced to abort the processes or may crash entirely. Overestimation wastes disk space that could otherwise be used for files, but it does not harm other.Following table shows different system using amount of swap space:

System	Swap-Space
1. Solaris	Equal amount of physical memory
2. Linux	Double the amount of physical memory

Figure – Different systems using amount of swap-space

Explanation of above table :

Solaris, setting swap space equal to the amount by which virtual memory exceeds page-able physical memory. In the past Linux has suggested setting swap space to double the amount of physical memory. Today, this limitation is gone, and most Linux systems use considerably less swap space.

Including Linux, some operating systems; allow the use of multiple swap spaces, including both files and dedicated swap partitions. The swap spaces are placed on the disk so the load which is on the I/O by the paging and swapping will spread over the system's bandwidth.

Swap-Space Location :

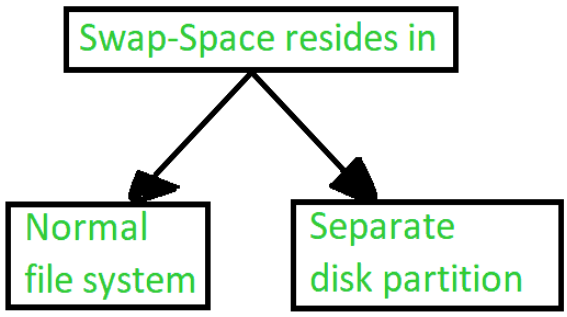


Figure – Location of swap-space

A swap space can reside in one of the two places –

- Normal file system
- Separate disk partition

Stable-Storage Implementation in Operating system

By definition, information residing in the **Stable-Storage** is never lost. Even if the disk and CPU have some errors, it will never lose any data.

Stable-Storage Implementation

To achieve such storage, we need to replicate the required information on multiple storage devices with independent failure modes. The writing of an update should be coordinated in

such a way that it would not delete all the copies of the state and that, when we are recovering from a failure, we can force all the copies to a consistent and correct value, even if another failure occurs during the recovery. In these, we discuss how to meet these needs.

The disk write operation results to one of the following outcome:

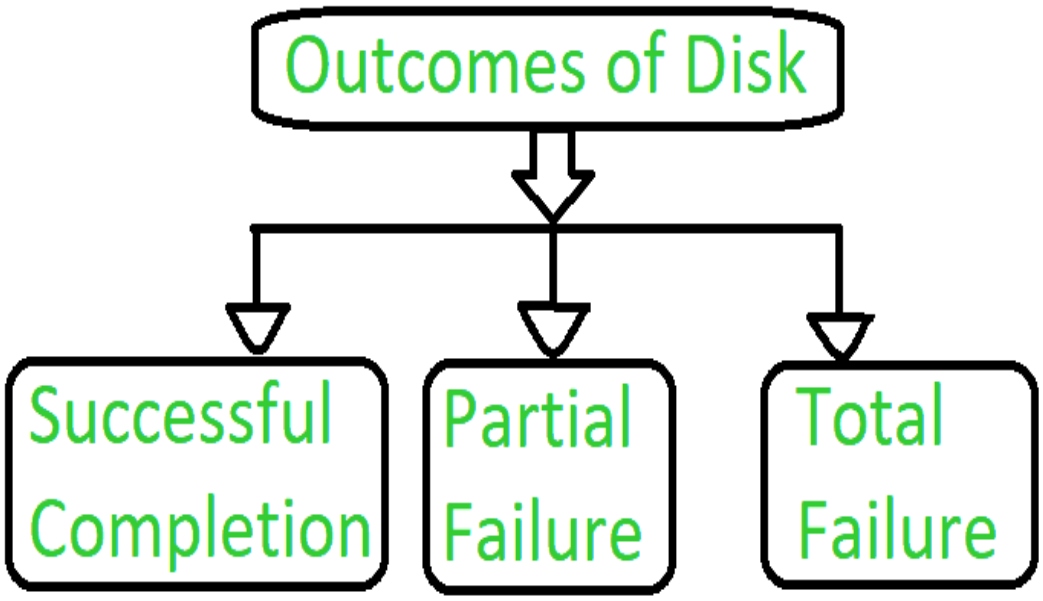


Figure – Outcomes of Disk

Successful completion –

The data will be written correctly on disk.

Partial Failure –

In this case, failure occurred in the middle of the data transfer, so that only some sectors were written with the new data, and the sector which was written during the failure may have been corrupted.

Total Failure –

The failure occurred before the disk write started, so the previous data values on the disk remain intact.While writing a block somehow if failure occurs, the system's first work is to

detect the failure and then invoke a recovery process to restore the consistent state. To do that, the system must contain two physical blocks for each logical block.

An output operation is executed as follows:

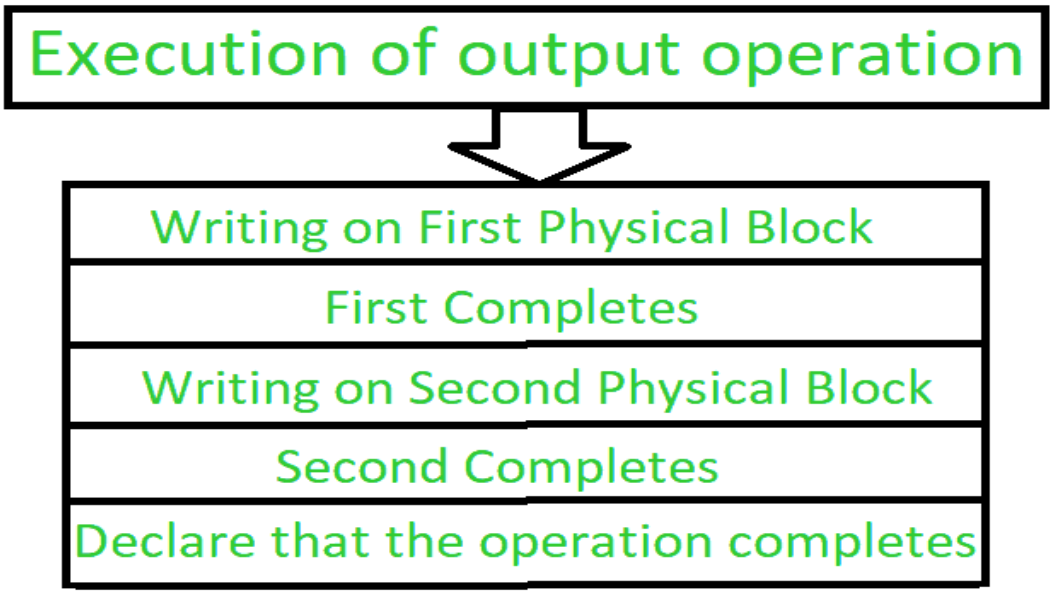


Figure – Process of execution of output operation

- Write the information onto the first physical block.
- When the first write completes successfully, write the same operation onto the second physical block.
- When both the operation declares successfully, declare the operation as complete.

During the recovery from a failure each of the physical blocks is examined. If both are the same and no detectable error exists, then no further action is necessary. If one block contains detectable errors then we replace its content with the value of the other block. If neither block contains the detectable error, but the block differs in content, then we replace the content of first block with the content of the second block.This procedure of the recovery gives us an

conclusion that either the write to stable content succeeds successfully or it results in no change.

This procedure will be extended if we want an arbitrarily large number of copies of each block of the stable storage. With the usage of a large number of copies, the chances of failure reduces. Generally, it is usually reasonable to simulate stable storage with only two copies.. The data that is present in the stable storage is guaranteed to be safe unless a failure destroys all the copies.

Because waiting for disk writes to complete is time consuming, many storage arrays add NVRAM as a cache. Since the memory is no-volatile it can be trusted to store the data en route to the disks. In this way it is considered as a part of the stable storage. Writing to the stable storage is much faster than to disk, so performance is greatly improved.

Tertiary-Storage Structure

- Primary storage refers to computer memory chips; Secondary storage refers to fixed-disk storage systems (hard drives); And ***Tertiary Storage*** refers to ***removable media,*** such as tape drives, CDs, DVDs, and to a lesser extent floppies, thumb drives, and other detachable devices.
- Tertiary storage is typically characterized by large capacity, low cost per MB, and slow access times, although there are exceptions in any of these categories.
- Tertiary storage is typically used for backups and for long-term archival storage of completed work. Another common use for tertiary storage is to swap large little-used files (or groups of files) off of the hard drive, and then swap them back in as needed in a fashion similar to secondary storage providing swap space for primary storage.

Tertiary-Storage Devices

Removable Disks

Removable magnetic disks (e.g. floppies) can be nearly as fast as hard drives, but are at greater risk for damage due to scratches. Variations of removable magnetic disks up to a GB or more in capacity have been developed. (Hot-swappable hard drives?)

A ***magneto-optical*** disk uses a magnetic disk covered in a clear plastic coating that protects the surface.

The heads sit a considerable distance away from the magnetic surface, and as a result do not have enough magnetic strength to switch bits ***at normal room temperature.***

For writing, a laser is used to heat up a specific spot on the disk, to a temperature at which the weak magnetic field of the write head is able to flip the bits.

For reading, a laser is shined at the disk, and the ***Kerr effect*** causes the polarization of the light to become rotated either clockwise or counter-clockwise depending on the orientation of the magnetic field.

Optical disks do not use magnetism at all, but instead use special materials that can be altered (by lasers) to have relatively light or dark spots.

The most common examples of these disks are ***re-writable*** CD-RWs and DVD-RWs.

An alternative to the disks described above are ***Write-Once Read-Many, WORM*** drives.

The original version of WORM drives involved a thin layer of aluminum sandwiched between two protective layers of glass or plastic.

Holes were burned in the aluminum to write bits.

Because the holes could not be filled back in, there was no way to re-write to the disk. (Although data could be erased by burning more holes.)

WORM drives have important legal ramifications for data that must be stored for a very long time and must be provable in court as unaltered since it was originally written. (Such as long-term storage of medical records.)

Modern CD-R and DVD-R disks are examples of WORM drives that use organic polymer inks instead of an aluminum layer.

Read-only disks are similar to WORM disks, except the bits are pressed onto the disk at the factory, rather than being burned on one by one

Protection Mechanisms

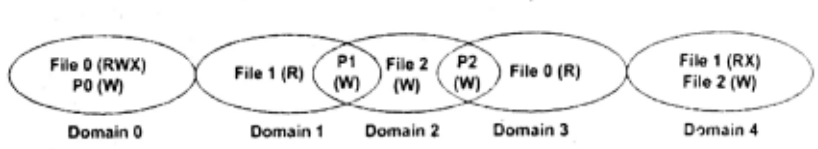
Protection plays a very crucial role in a multiuser environment, where several users will be making concurrent use of the computer resources such as CPU, memory etc. It is the duty of the operating system to provide a mechanism that protects each process from others.

All the items that require protection in a multiuser environment are listed as objects and those that want to access these objects are known as subjects. The operating system grants different 'access rights' to different subjects.

These rights may include read, write, execute, append, delete etc.

1. Domain

A domain is a combination of different objects and a set of different 'access rights' that can be granted to different subjects to operate on each of these objects. An operating system maintains several such domains with different combinations of access rights. The user processes can execute in one of those domains and can access the objects in that domain according to the access rights given to those objects.



Protection domain

A user process executing in domain 0 has access to read from, write into and execute the file 0 and can write to printer P0. Similarly, the process executing in domain 1 has access to read from file 1. The printer P1 is common to both domain 1 and domain 2. The processes executing in domain 1 and domain 2 both can have access to printer P1

In matrix form, the above image can be represented as shown in the below image.

During the execution of a process, it may become necessary for it to access an object, which is in another domain. If it has a right to access that object it switches to the new domain and accesses that file. This process is known as domain switching.

Implementation of Access Matrix

The access matrix can be implemented by using either access control lists or capability lists.

Objects → Domain ↓	File 0	File 1	File 2	Printer 0	Printer 1	Printer 2
0	RWX			W		
1		R			W	
2			W		W	W
3	R					W
4		RX	W			

Protection matrix

In ACL, the data is stored by column by the operating system. The information about the users and their access rights for each file is maintained by the operating system. The empty entries are discarded.

In capability lists, the access control matrix is sliced horizontally by a row. This implies that the operating system will have to maintain for each user a list of all the objects that the user can access and the ways in which he can access them. A combination of ACL and capability list techniques may also be used to design protection mechanisms.

Encryption

It is one of the most powerful and important tools of protection. The process of encryption involves two steps: encryption of the original data into some other form about which nothing is known to the third person and decryption of the data into the original form from the encrypted form.

The most commonly used methods to achieve encryption are: transposition ciphers and substitution ciphers.

In transposition ciphers, the letters in the original message are not changed; only the order in which they are contained in the original message gets changed. For example, consider that the message 'it is raining' needs to be encrypted. It will become 'gniniar si ti' in the encrypted form using a particular form of transposition ciphers algorithm.

The set of characters in the encrypted form will be different from the original ones if we use substitution ciphers. Every letter may be replaced by its previous alphabet, for instance. Now the message 'it is raining' would become, after encryption, 'hs hr qzhmhmf'.

It is very easy to implement these ciphers for characters. The varied forms of these algorithms can be used to encrypt bit streams. For instance, a predetermined bit stream may be added to

the bits in the original stream at a particular position to obtain the encrypted message. The same bit of steam is subtracted at the destination so that the original stream is obtained. This addition and subtraction may be accomplished with the help of simple adder and subtractor circuits.

The key idea behind the encryption schemes is that the encryption process must be restorable. Means, once we encrypt the original message to a different form, there should be a way to restore it to the original form

OS security issues:

Intruders

In the computer security world, people who want to cause some trouble for their fun or for their own commercial profit are called intruders.

Basically, intruders are of the following two types:

- Active intruders
- Passive intruders

Now, let's describe briefly about the above two different types of intruders.

Active Intruders

Active intruders are malicious.

Active intruders always want to make some unauthorized access to the other's system to change, modify, or sometimes delete the data.

Passive Intruders

Passive intruders are less malicious than active one.

Passive intruders want to read the files they aren't authorized to read.

Some of the most common types of ***violations*** include:

> ***Breach of Confidentiality -*** Theft of private or confidential information, such as credit-card numbers, trade secrets, patents, secret formulas, manufacturing procedures, medical information, financial information, etc.

Breach of Integrity - Unauthorized ***modification*** of data, which may have serious indirect consequences. For example a popular game or other program's source code could be modified to open up security holes on users systems before being released to the public.

Breach of Availability - Unauthorized ***destruction*** of data, often just for the "fun" of causing havoc and for bragging rites. Vandalism of web sites is a common form of this violation.

Theft of Service - Unauthorized use of resources, such as theft of CPU cycles, installation of daemons running an unauthorized file server, or tapping into the target's telephone or networking services.

Denial of Service, DOS - Preventing legitimate users from using the system, often by overloading and overwhelming the system with an excess of requests for service.

One common attack is ***masquerading,*** in which the attacker pretends to be a trusted third party. A variation of this is the ***man-in-the-middle,*** in which the attacker masquerades as both ends of the conversation to two targets.

A ***replay attack*** involves repeating a valid transmission. Sometimes this can be the entire attack, (such as repeating a request for a money transfer), or other times the content of the original message is replaced with malicious content.

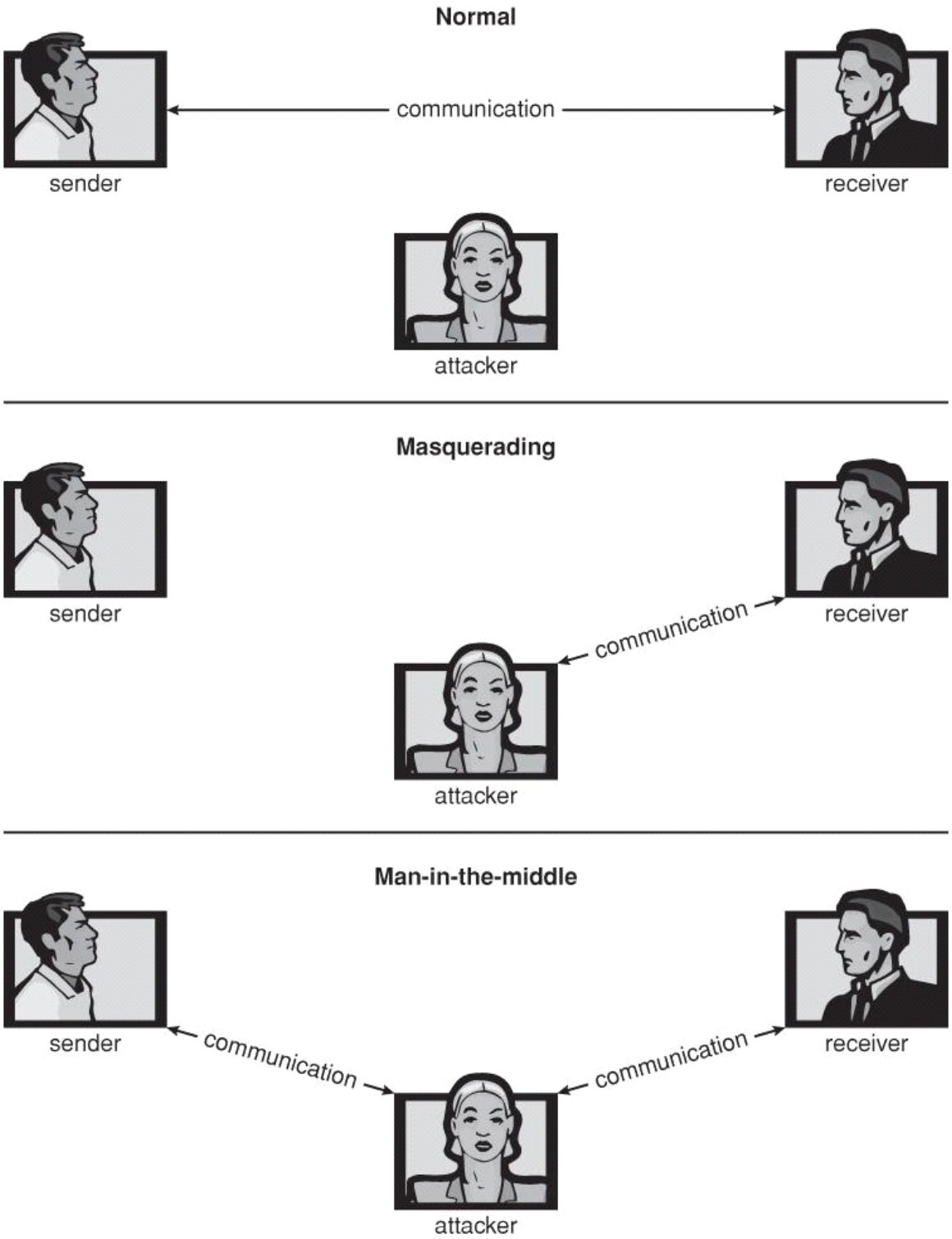


Figure 15.1 - Standard security attacks.

There are four levels at which a system must be protected:

Physical - The easiest way to steal data is to pocket the backup tapes. Also, access to the root console will often give the user special privileges, such as rebooting the system as root from removable media. Even general access to terminals in a computer room offers some opportunities for an attacker, although today's modern high-speed networking environment provides more and more opportunities for remote attacks.

Human - There is some concern that the humans who are allowed access to a system are trustworthy, and that they cannot be coerced into breaching security. However more and more attacks today are made via ***social engineering,*** which basically means fooling trustworthy people into accidentally breaching security.

Phishing involves sending an innocent-looking e-mail or web site designed to fool people into revealing confidential information. E.g. spam e-mails pretending to be from e-Bay, PayPal, or any of a number of banks or credit-card companies.

Dumpster Diving involves searching the trash or other locations for passwords that are written down. (Note: Passwords that are too hard to remember, or which must be changed frequently are more likely to be written down somewhere close to the user's station.)

Password Cracking involves divining users' passwords, either by watching them type in their passwords, knowing something about them like their pet's names, or simply trying all words in common dictionaries. (Note: "Good" passwords should involve a minimum number of characters, include non-alphabetical characters, and not appear in any dictionary (in any language), and should be changed frequently. Note also that it is proper etiquette to look away from the keyboard while someone else is entering their password.)

Operating System - The OS must protect itself from security breaches, such as runaway processes (denial of service), memory-access violations, stack overflow violations, the launching of programs with excessive privileges, and many others.

Network - As network communications become ever more important and pervasive in modern computing environments, it becomes ever more important to protect this area of the system. (Both protecting the network itself from attack, and protecting the local system from

attacks coming in through the network.) This is a growing area of concern as wireless communications and portable devices become more and more prevalent.

Program Threats

There are many common threats to modern systems. Only a few are discussed here.

Trojan Horse

A ***Trojan Horse*** is a program that secretly performs some maliciousness in addition to its visible actions.

Some Trojan horses are deliberately written as such, and others are the result of legitimate programs that have become infected with ***viruses,***

One dangerous opening for Trojan horses is long search paths, and in particular paths which include the current directory (".") as part of the path. If a dangerous program having the same name as a legitimate program (or a common mis-spelling, such as "sl" instead of "ls") is placed anywhere on the path, then an unsuspecting user may be fooled into running the wrong program by mistake.

Another classic Trojan Horse is a login emulator, which records a users account name and password, issues a "password incorrect" message, and then logs off the system. The user then tries again (with a proper login prompt), logs in successfully, and doesn't realize that their information has been stolen.

Spyware is a version of a Trojan Horse that is often included in "free" software downloaded off the Internet. Spyware programs generate pop-up browser windows, and may also accumulate information about the user and deliver it to some central site. (This is an example of ***covert channels,*** in which surreptitious communications occur.) Another common task of spyware is to send out spam e-mail messages, which then purportedly come from the infected user.

Trap Door

A ***Trap Door*** is when a designer or a programmer (or hacker) deliberately inserts a security hole that they can use later to access the system.

Because of the possibility of trap doors, once a system has been in an untrustworthy state, that system can never be trusted again. Even the backup tapes may contain a copy of some cleverly hidden back door.

A clever trap door could be inserted into a compiler, so that any programs compiled with that compiler would contain a security hole. This is especially dangerous, because inspection of the code being compiled would not reveal any problems.

Logic Bomb

A ***Logic Bomb*** is code that is not designed to cause havoc all the time, but only when a certain set of circumstances occurs, such as when a particular date or time is reached or some other noticeable event.

A classic example is the ***Dead-Man Switch***, which is designed to check whether a certain person (e.g. the author) is logging in every day, and if they don't log in for a long time (presumably because they've been fired), then the logic bomb goes off and either opens up security holes or causes other problems.

Stack and Buffer Overflow

This is a classic method of attack, which exploits bugs in system code that allows buffers to overflow.

Viruses

- A virus is a fragment of code embedded in an otherwise legitimate program, designed to replicate itself (by infecting other programs), and (eventually) wreaking havoc.
- Viruses are more likely to infect PCs than UNIX or other multi-user systems, because programs in the latter systems have limited authority to

modify other programs or to access critical system structures (such as the boot block.)

- Viruses are delivered to systems in a ***virus dropper,*** usually some form of a Trojan Horse, and usually via e-mail or unsafe downloads.
- Viruses take many forms (see below.) Figure 15.5 shows typical operation of a boot sector virus:

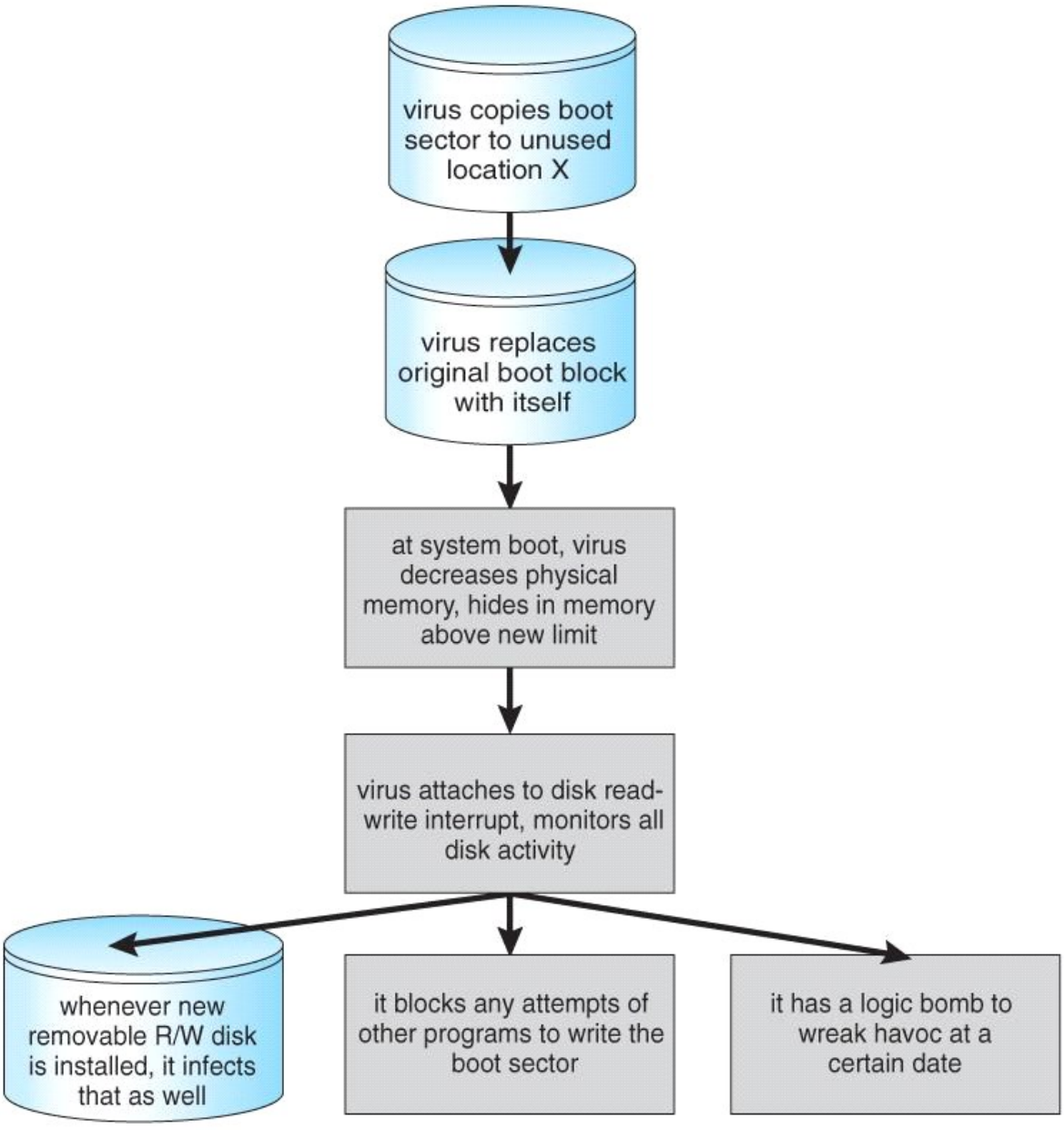


Figure 15.5 - A boot-sector computer virus.

- Some of the forms of viruses include:
    - **File -** A file virus attaches itself to an executable file, causing it to run the virus code first and then jump to the start of the original program. These viruses are termed ***parasitic,*** because they do not leave any new files on the system, and the original program is still fully functional.
    - **Boot -** A boot virus occupies the boot sector, and runs before the OS is loaded. These are also known as ***memory viruses***, because in operation they reside in memory, and do not appear in the file system.
    - **Macro -** These viruses exist as a macro (script) that are run automatically by certain macro-capable programs such as MS Word or Excel. These viruses can exist in word processing documents or spreadsheet files.
    - **Source code** viruses look for source code and infect it in order to spread.
    - **Polymorphic** viruses change every time they spread - Not their underlying functionality, but just their ***signature,*** by which virus checkers recognize them.
    - **Encrypted** viruses travel in encrypted form to escape detection. In practice they are self-decrypting, which then allows them to infect other files.
    - **Stealth** viruses try to avoid detection by modifying parts of the system that could be used to detect it. For example the read( ) system call could be modified so that if an infected file is read the infected part gets skipped and the reader would see the original unadulterated file.
    - **Tunneling** viruses attempt to avoid detection by inserting themselves into the interrupt handler chain, or into device drivers.
    - **Multipartite** viruses attack multiple parts of the system, such as

files, boot sector, and memory.

    - **Armored** viruses are coded to make them hard for anti-virus researchers to decode and understand. In addition many files associated with viruses are hidden, protected, or given innocuous looking names such as "...".
- In 2004 a virus exploited three bugs in Microsoft products to infect hundreds of Windows servers (including many trusted sites) running Microsoft Internet Information Server, which in turn infected any Microsoft Internet Explorer web browser that visited any of the infected server sites. One of the back-door programs it installed was a ***keystroke logger,*** which records users keystrokes, including passwords and other sensitive information.
- There is some debate in the computing community as to whether a ***monoculture,*** in which nearly all systems run the same hardware, operating system, and applications, increases the threat of viruses and the potential for harm caused by them.

System and Network Threats

Most of the threats described above are termed ***program threats***, because they attack specific programs or are carried and distributed in programs. The threats in this section attack the operating system or the network itself, or leverage those systems to launch their attacks.

Worms

A ***worm*** is a process that uses the fork / spawn process to make copies of itself in order to wreak havoc on a system. Worms consume system resources, often blocking out other, legitimate processes.