UNIT 5:
File Management and Security

File Systems

File system is the part of the operating system which is responsible for file management. It provides a mechanism to store the data and access to the file contents including data and programs. Some Operating systems treats everything as a file for example Ubuntu.

The File system takes care of the following issues

- **File Structure**

  We have seen various data structures in which the file can be stored. The task of the file system is to maintain an optimal file structure.

- **Recovering Free space**

  Whenever a file gets deleted from the hard disk, there is a free space created in the disk. There can be many such spaces which need to be recovered in order to reallocate them to other files.

- **disk space assignment to the files**

  The major concern about the file is deciding where to store the files on the hard disk. There are various disks scheduling algorithm which will be covered later in this tutorial.

- **tracking data location**

  A File may or may not be stored within only one block. It can be stored in the non contiguous blocks on the disk. We need to keep track of all the blocks on which the part of the files reside.

File Access Methods

What is file access method

File access method is a way of accessing and manipulating data stored in a file. It determines how data is read and written in computer storage devices. There are several file access methods, each with its own characteristics, advantages, and disadvantages.

The three main types of file access methods are :

- sequential access
- random access
- direct access
- indexed access method

Sequential access reads and writes data in a linear order, while random access allows direct access to specific data within the file. Direct access involves accessing data directly by its physical location in the file. The choice of file access method depends on the specific needs of the application or device using the file. Understanding the differences between each method is important for efficient and effective data management.

Importance of file access method in operating systems

The file access method is a critical component of an operating system because it determines how files are stored, organized, and accessed by applications and users.

The importance of the file access method in operating systems can be seen in the following ways −

- **Efficiency −** The file access method can significantly impact the efficiency of a computer system, as it determines how quickly files can be accessed and how efficiently data can be written or read.
- **Data integrity −** The file access method ensures that data is stored and accessed correctly, protecting the integrity of the data stored in files.
- **Security −** The file access method can help ensure the security of files, by controlling access to them, limiting who can view, modify, or delete files.
- **Resource management −** The file access method plays an important role in resource management, helping the operating system manage disk space and allocate resources efficiently.

Definition of sequential access

Sequential access is a file access method in which data is accessed in a linear or sequential order. This means that data can only be accessed in the order in which it is stored in the file. Sequential access reads or writes data one after the other, starting from the beginning of the file and ending at the end of the file.

How data is read/written in sequential access

In sequential access, data is accessed in a particular order. For example, to access the 10th record in a file, a program must first read the first nine records sequentially, starting from the beginning of the file, until it reaches the 10th record. The same is true for writing data in a sequential file. The data must be written in the order that it is to be stored in the file.

Advantages and disadvantages of sequential access

Advantages of sequential access include that it is simple and easy to implement, it requires less memory, and it is suitable for storing large amounts of data. However, sequential access is not efficient for accessing specific data or making changes to the data. It is slow when it comes to reading or writing data in the middle of the file since the program must read or write all the data before the required data.

Examples of devices that use sequential access − Sequential access is commonly used in devices such as tape drives, which require reading or writing data in a linear or sequential order. Sequential access is also used in some types of disk storage systems, but random access is more commonly used for disk storage.

Definition of random access

Random access is a file access method in which data can be accessed from any location within the file. This means that data can be read or written to any location in the file without having to read through all the data that comes before it. Random access provides the ability to directly access any record or data element in the file.

How data is read/written in random access

In random access, data can be read or written at any location in the file without the need to read all the preceding data. This is possible because random access uses an index or address to locate the specific data required, making it faster and more efficient than sequential access.

Advantages and disadvantages of random access

Random access provides fast and efficient access to specific data within the file. It is also efficient for editing and updating data in the file. However, random access requires more memory to store index or address information, which can make the file size larger than with sequential access. Additionally, if the index or address information becomes corrupted, data can become inaccessible.

Examples of devices that use random access − Random access is commonly used in devices such as hard drives, solid-state drives, and USB drives. These devices require fast and efficient access to specific data, which makes random access an ideal file access method. Random access is also commonly used in database systems, where fast access to specific records is required.

Definition of direct access

Direct access is a file access method that allows data to be accessed directly by using the data's physical location within the file. In other words, data can be read or written to any location in the file, much like with random access. However, direct access does not use an index or address like random access, and instead relies on the physical location of the data within the file.

How data is read/written in direct access

In direct access, data is read or written directly to the physical location in the file. The data can be accessed by using the record number, byte position, or block number. This allows for fast and efficient access to specific data within the file.

Advantages and disadvantages of direct access

Direct access provides fast and efficient access to specific data within the file, similar to random access. It also does not require the additional memory needed for index or address information, making the file size smaller than with random access. However, direct access requires knowledge of the physical layout of the data within the file, and may require special hardware or software to access the data directly. Additionally, if data is deleted or moved, gaps can be left in the file which can impact performance.

Examples of devices that use direct access − Direct access is commonly used in devices such as magnetic disk drives, optical disk drives, and flash memory. These devices require fast and efficient access to specific data, which makes direct access an ideal file access method. Direct access is also commonly used in database systems, where fast access to specific records is required.

Indexed Access Method

The indexed access method involves accessing files through an index or directory that contains a list of file names and their corresponding locations on the disk. This method is suitable for applications that need to access files by their names or attributes, such as file managers or search engines. The indexed access method provides a fast and efficient way to locate and access files.

The indexed access method uses a file index or directory to keep track of the locations of files on the disk. The file index is stored in a separate file or in a specific location on the disk. When a file is created, its name and location are added to the file index. To access a file, an application searches the file index for the file name and then uses the direct access method to read the file from its location on the disk.

Comparison of Access Methods

Comparison of advantages and disadvantages

Sequential Access

- **Advantages** − Simple and easy to implement, suitable for storing large amounts of data, requires less memory.
- **Disadvantages** − Not efficient for accessing specific data or making changes to the data, slow for reading or writing data in the middle of the file.

Random Access

- **Advantages** − Provides fast and efficient access to specific data within the file, efficient for editing and updating data, suitable for devices that require fast access to specific data.
- **Disadvantages** − Requires more memory to store index or address information, file size can be larger than with sequential access, data can become inaccessible if index or address information becomes corrupted.

Direct Access

- **Advantages** − Provides fast and efficient access to specific data within the file, suitable for devices that require fast access to specific data, file size is smaller than with random access.
- **Disadvantages** − Requires knowledge of the physical layout of the data within the file, may require special hardware or software to access the data directly, gaps can be left in the file which can impact performance.

Indexed access

- **Advantages** − Provides fast and efficient access to files by name or attributes, making it suitable for applications that require searching and retrieving specific files quickly.
- **Disadvantages** − The index must be maintained, which can require additional disk space and processing time.

Which method is best for certain situations?

The best file access method for a particular situation depends on the requirements of the application or device using the file.

- Sequential access is best suited for applications that require reading or writing data in a linear order, such as logging data, audio/video streaming, or processing large datasets in batches.
- Random access is best suited for applications that require fast access to specific data or records, such as database systems, search engines, or file systems used in operating systems.
- Direct access is best suited for applications that require fast access to specific data and use low-level disk operations, such as device drivers, file systems used in operating systems, or media streaming applications.

In summary, the choice of file access method depends on the specific needs of the application or device using the file. Sequential access is best for linear data processing, random access is best for fast access to specific data, and direct access is best for low-level disk operations.

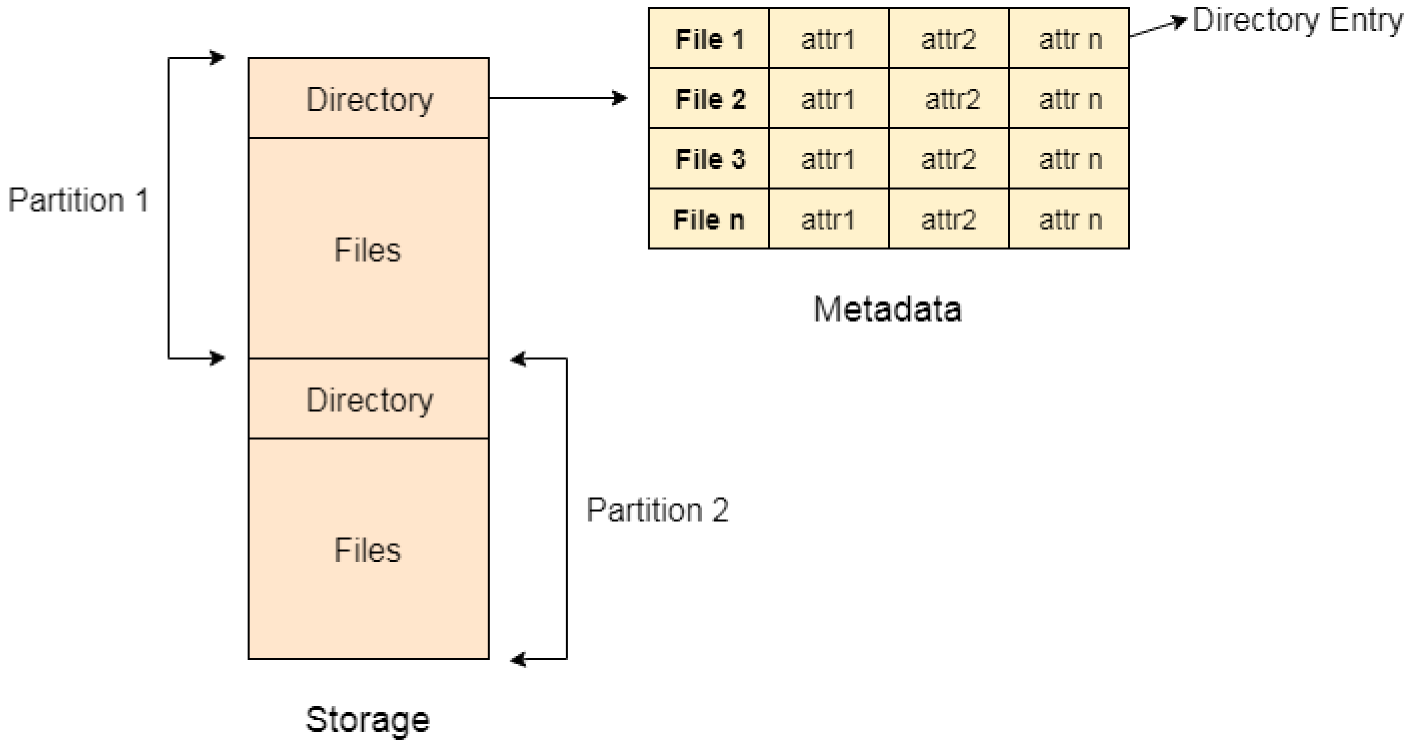
Directory Structure in OS (Operating System)

What is a directory?

Directory can be defined as the listing of the related files on the disk. The directory may store some or the entire file attributes.

To get the benefit of different file systems on the different operating systems, A hard disk can be divided into the number of partitions of different sizes. The partitions are also called volumes or mini disks.

Each partition must have at least one directory in which, all the files of the partition can be listed. A directory entry is maintained for each file in the directory which stores all the information related to that file.


Directory Entry
File 1 attr1 attr2 attr n
File 2 attr1 attr2 attr n
File 3 attr1 attr2 attr n
File n attr1 attr2 attr n
Metadata
Partition 1
Directory
Files
Directory
Files
Partition 2
Storage

A directory can be viewed as a file which contains the Meta data of the bunch of files.

Advertisement

Every Directory supports a number of common operations on the file:

1. File Creation
2. Search for the file
3. File deletion
4. Renaming the file
5. Traversing Files
6. Listing of files

File Protection

File protection in an operating system is the process of securing files from unauthorized access, alteration, or deletion. It is critical for data security and ensures that sensitive information remains confidential and secure. Operating systems provide various mechanisms and techniques such as file permissions, encryption, access control lists, auditing, and physical file security to protect files. Proper file protection involves user authentication, authorization, access control, encryption, and auditing. Ongoing updates and patches are also necessary to prevent security breaches. File protection in an operating system is essential to maintain data security and minimize the risk of data breaches and other security incidents.

What is File protection?

File protection in an operating system refers to the various mechanisms and techniques used to secure files from unauthorized access, alteration, or deletion. It involves controlling access to files, ensuring their security and confidentiality, and preventing data breaches and other security incidents.

Operating systems provide several file protection features, including file permissions, encryption, access control lists, auditing, and physical file security. These measures allow administrators to manage access to files, determine who can access them, what actions can be performed on them, and how they are stored and backed up. Proper file protection requires ongoing updates and patches to fix vulnerabilities and prevent security breaches. It is crucial for data security in the digital age where cyber threats are prevalent. By implementing file protection measures, organizations can safeguard their files, maintain data confidentiality, and minimize the risk of data breaches and other security incidents.

Type of File protection

File protection is an essential component of modern operating systems, ensuring that files are secured from unauthorized access, alteration, or deletion. In this context, there are several types of file protection mechanisms used in operating systems to provide robust data security.

- **File Permissions** − File permissions are a basic form of file protection that controls access to files by setting permissions for users and groups. File permissions allow the system administrator to assign specific access rights to users and groups, which can include read, write, and execute privileges. These access rights can be assigned at the file or directory level, allowing users and groups to access specific files or directories as needed. File permissions can be modified by the system administrator at any time to adjust access privileges, which helps to prevent unauthorized access.
- **Encryption** − Encryption is the process of converting plain text into ciphertext to protect files from unauthorized access. Encrypted files can only be accessed by authorized users who have the correct encryption key to decrypt them. Encryption is widely used to secure sensitive data such as financial information, personal data, and other confidential information. In an operating system, encryption can be

applied to individual files or entire directories, providing an extra layer of protection against unauthorized access.

- **Access Control Lists (ACLs)** − Access control lists (ACLs) are lists of permissions attached to files and directories that define which users or groups have access to them and what actions they can perform on them. ACLs can be more granular than file permissions, allowing the system administrator to specify exactly which users or groups can access specific files or directories. ACLs can also be used to grant or deny specific permissions, such as read, write, or execute privileges, to individual users or groups.
- **Auditing and Logging** − Auditing and logging are mechanisms used to track and monitor file access, changes, and deletions. It involves creating a record of all file access and changes, including who accessed the file, what actions were performed, and when they were performed. Auditing and logging can help to detect and prevent unauthorized access and can also provide an audit trail for compliance purposes.
- **Physical File Security** − Physical file security involves protecting files from physical damage or theft. It includes measures such as file storage and access control, backup and recovery, and physical security best practices. Physical file security is essential for ensuring the integrity and availability of critical data, as well as compliance with regulatory requirements.

Overall, these types of file protection mechanisms are essential for ensuring data security and minimizing the risk of data breaches and other security incidents in an operating system. The choice of file protection mechanisms will depend on the specific requirements of the organization, as well as the sensitivity and volume of the data being protected. However, a combination of these file protection mechanisms can provide comprehensive protection against various types of threats and vulnerabilities.

Advantages of File protection

File protection is an important aspect of modern operating systems that ensures data security and integrity by preventing unauthorized access, alteration, or deletion of files. There are several advantages of file protection mechanisms in an operating system, including −

- **Data Security** − File protection mechanisms such as encryption, access control lists, and file permissions provide robust data security by preventing unauthorized access to files. These mechanisms ensure that only authorized users can access files, which helps to prevent data breaches and other security incidents. Data security is critical for organizations that handle sensitive data such as personal data, financial information, and intellectual property.
- **Compliance** − File protection mechanisms are essential for compliance with regulatory requirements such as GDPR, HIPAA, and PCI-DSS. These regulations require organizations to implement appropriate security measures to protect sensitive data from unauthorized access, alteration, or deletion. Failure to comply with these regulations can result in significant financial penalties and reputational damage.
- **Business Continuity** − File protection mechanisms are essential for ensuring business continuity by preventing data loss due to accidental or malicious deletion,

corruption, or other types of damage. File protection mechanisms such as backup and recovery, auditing, and logging can help to recover data quickly in the event of a data loss incident, ensuring that business operations can resume as quickly as possible.

- **Increased Productivity** − File protection mechanisms can help to increase productivity by ensuring that files are available to authorized users when they need them. By preventing unauthorized access, alteration, or deletion of files, file protection mechanisms help to minimize the risk of downtime and data loss incidents that can impact productivity.
- **Enhanced Collaboration** − File protection mechanisms can help to enhance collaboration by allowing authorized users to access and share files securely. Access control lists, file permissions, and encryption can help to ensure that files are only accessed by authorized users, which helps to prevent conflicts and misunderstandings that can arise when multiple users access the same file.
- **Reputation** − File protection mechanisms can enhance an organizations reputation by demonstrating a commitment to data security and compliance. By implementing robust file protection mechanisms, organizations can build trust with their customers, partners, and stakeholders, which can have a positive impact on their reputation and bottom line.

Overall, these advantages of file protection mechanisms highlight the importance of data security and the need for organizations to implement appropriate measures to protect their sensitive data. File protection mechanisms can help to prevent data breaches and other security incidents, ensure compliance with regulatory requirements, and ensure business continuity in the event of a data loss incident. By implementing a comprehensive file protection strategy, organizations can enhance productivity, collaboration, and reputation, while minimizing the risk of data loss and other security incidents.

Disadvantages of File protection

There are also some potential disadvantages of file protection in an operating system, including −

- **Overhead** − Some file protection mechanisms such as encryption, access control lists, and auditing can add overhead to system performance. This can impact system resources and slow down file access and processing times.
- **Complexity** − File protection mechanisms can be complex and require specialized knowledge to implement and manage. This can lead to errors and misconfigurations that compromise data security.
- **Compatibility Issues** − Some file protection mechanisms may not be compatible with all types of files or applications, leading to compatibility issues and limitations in file usage.
- **Cost** − Implementing robust file protection mechanisms can be expensive, especially for small organizations with limited budgets. This can make it difficult to achieve full data protection.
- **User Frustration** − Stringent file protection mechanisms such as complex passwords, frequent authentication requirements, and restricted access can frustrate users and impact productivity.

File Allocation Methods

The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

The main idea behind these methods is to provide:

- Efficient disk space utilization.
- Fast access to the file blocks.

All the three methods have their own advantages and disadvantages as discussed below:

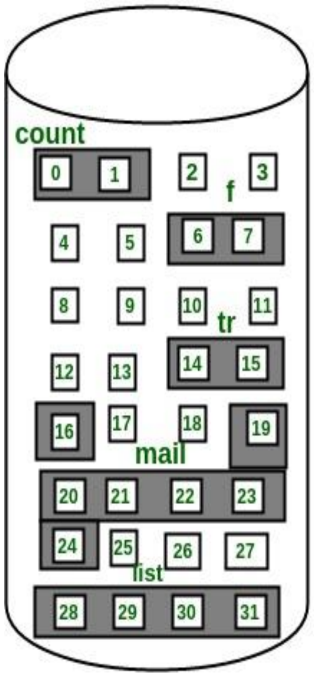
1. Contiguous Allocation

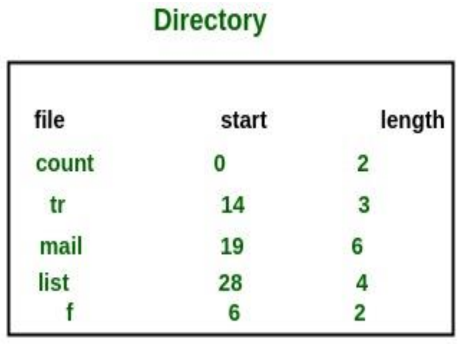
In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,……b+n-1.* This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.

The directory entry for a file with contiguous allocation contains

- Address of starting block
- Length of the allocated portion.

The *file 'mail'* in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.





Directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Advantages:

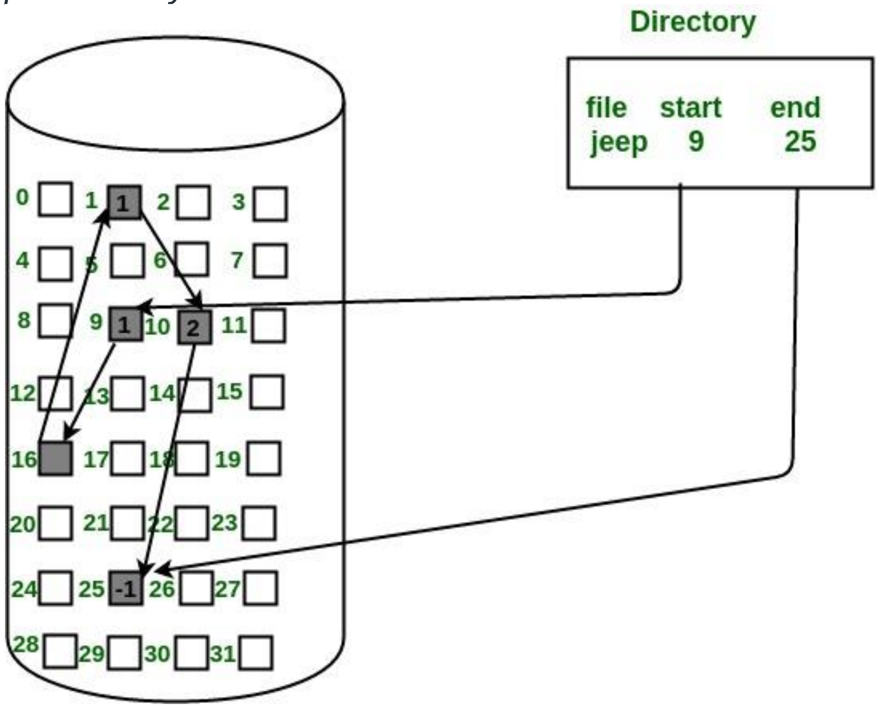
- Both the Sequential and Direct Accesses are supported by this. For direct access, the address of the kth block of the file which starts at block b can easily be obtained as (b+k).
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

Disadvantages:

- This method suffers from both internal and external fragmentation. This makes it inefficient in terms of memory utilization.
- Increasing file size is difficult because it depends on the availability of contiguous memory at a particular instance.

2. Linked List Allocation

In this scheme, each file is a linked list of disk blocks which **need not be** contiguous. The disk blocks can be scattered anywhere on the disk.
The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file.
The file 'jeep' in following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.



Advantages:

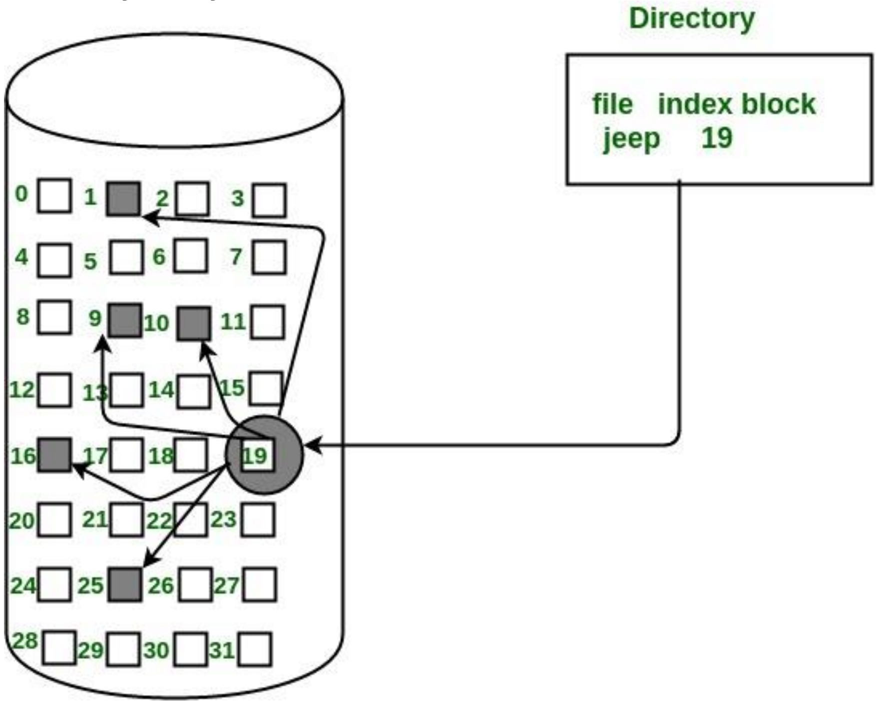
- This is very flexible in terms of file size. File size can be increased easily since the system does not have to look for a contiguous chunk of memory.
- This method does not suffer from external fragmentation. This makes it relatively better in terms of memory utilization.

Disadvantages:

- Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually. This makes linked allocation slower.
- It does not support random or direct access. We can not directly access the blocks of a file. A block k of a file can be accessed by traversing k blocks sequentially (sequential access) from the starting block of the file via block pointers.
- Pointers required in the linked allocation incur some extra overhead.

3. Indexed Allocation

In this scheme, a special block known as the **Index block** contains the pointers to all the blocks occupied by a file. Each file has its own index block. The ith entry in the index block contains the disk address of the ith file block. The directory entry contains the address of the index block as shown in the image:



Advantages:

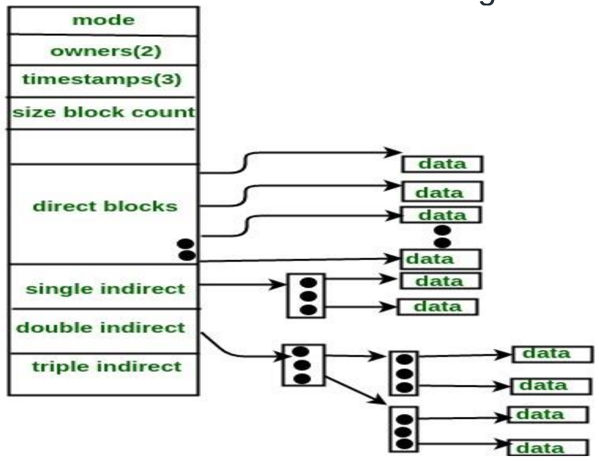
- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

Disadvantages:

- The pointer overhead for indexed allocation is greater than linked allocation.
- For very small files, say files that expand only 2-3 blocks, the indexed allocation would keep one entire block (index block) for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block.

For files that are very large, single index block may not be able to hold all the pointers.
Following mechanisms can be used to resolve this:

1. **Linked scheme:** This scheme links two or more index blocks together for holding the pointers. Every index block would then contain a pointer or the address to the next index block.
2. **Multilevel index:** In this policy, a first level index block is used to point to the second level index blocks which inturn points to the disk blocks occupied by the file. This can be extended to 3 or more levels depending on the maximum file size.
3. **Combined Scheme:** In this scheme, a special block called the **Inode (information Node)** contains all the information about the file such as the name, size, authority, etc and the remaining space of Inode is used to store the Disk Block addresses which contain the actual file *as shown in the image below.* The first few of these pointers in Inode point to the **direct blocks** i.e the pointers contain the addresses of the disk blocks that contain data of the file. The next few pointers point to indirect blocks. Indirect blocks may be single indirect, double indirect or triple indirect. **Single Indirect block** is the disk block that does not contain the file data but the disk address of the blocks that contain the file data. Similarly, **double indirect blocks** do not contain the file data but the disk address of the blocks that contain the address of the blocks containing the file data.



Operating System Security

Every computer system and software design must handle all security risks and implement the necessary measures to enforce security policies. At the same time, it's critical to strike a balance because strong security measures might increase costs while also limiting the system's usability, utility, and smooth operation. As a result, system designers must assure efficient performance without compromising security.

In this article, you will learn about operating system security with its issues and other features.

What is Operating System Security?

The process of ensuring OS availability, confidentiality, integrity is known as operating system security. OS security refers to the processes or measures taken to protect the operating system from dangers, including viruses, worms, malware, and remote hacker intrusions. Operating system security comprises all preventive-control procedures that protect any system assets that could be stolen, modified, or deleted if OS security is breached.

Security refers to providing safety for computer system resources like software, CPU, memory, disks, etc. It can protect against all threats, including viruses and unauthorized access. It can be enforced by assuring the operating system's **integrity, confidentiality**, and **availability**. If an illegal user runs a computer application, the computer or data stored may be seriously damaged.

System security may be threatened through two violations, and these are as follows:

1. Threat

A program that has the potential to harm the system seriously.

2. Attack

A breach of security that allows unauthorized access to a resource.

There are two types of security breaches that can harm the system: malicious and accidental. Malicious threats are a type of destructive computer code or web script that is designed to cause system vulnerabilities that lead to back doors and security breaches. On the other hand, Accidental Threats are comparatively easier to protect against.

Security may be compromised through the breaches. Some of the breaches are as follows:

1. Breach of integrity

This violation has unauthorized data modification.

2. Theft of service

It involves the unauthorized use of resources.

3. Breach of confidentiality

It involves the unauthorized reading of data.

4. Breach of availability

It involves the unauthorized destruction of data.

5. Denial of service

It includes preventing legitimate use of the system. Some attacks may be accidental.

The goal of Security System

There are several goals of system security. Some of them are as follows:

1. Integrity

Unauthorized users must not be allowed to access the system's objects, and users with insufficient rights should not modify the system's critical files and resources.

2. Secrecy

The system's objects must only be available to a small number of authorized users. The system files should not be accessible to everyone.

3. Availability

All system resources must be accessible to all authorized users, i.e., no single user/process should be able to consume all system resources. If such a situation arises, service denial may occur. In this case, malware may restrict system resources and preventing legitimate processes from accessing them.

Types of Threats

There are mainly two types of threats that occur. These are as follows:

Program threats

The operating system's processes and kernel carry out the specified task as directed. Program Threats occur when a user program causes these processes to do malicious operations. The common example of a program threat is that when a program is installed on a computer, it could store and transfer user credentials to a hacker. There are various program threats. Some of them are as follows:

1.Virus

A virus may replicate itself on the system. Viruses are extremely dangerous and can modify/delete user files as well as crash computers. A virus is a little piece of code that is implemented on the system program. As the user interacts with the program, the virus becomes embedded in other files and programs, potentially rendering the system inoperable.

2. Trojan Horse

This type of application captures user login credentials. It stores them to transfer them to a malicious user who can then log in to the computer and access system resources.

3. Logic Bomb

A logic bomb is a situation in which software only misbehaves when particular criteria are met; otherwise, it functions normally.

4. Trap Door

A trap door is when a program that is supposed to work as expected has a security weakness in its code that allows it to do illegal actions without the user's knowledge.

System Threats

System threats are described as the misuse of system services and network connections to cause user problems. These threats may be used to trigger the program threats over an entire network, known as program attacks. System threats make an environment in which OS resources and user files may be misused. There are various system threats. Some of them are as follows:

1. Port Scanning

It is a method by which the cracker determines the system's vulnerabilities for an attack. It is a fully automated process that includes connecting to a specific port via TCP/IP. To protect the attacker's identity, port scanning attacks are launched through Zombie Systems, which previously independent systems now serve their owners while being utilized for such terrible purposes.

2. Worm

The worm is a process that can choke a system's performance by exhausting all system resources. A Worm process makes several clones, each consuming system resources and preventing all other processes from getting essential resources. Worm processes can even bring a network to a halt.

3. Denial of Service

Denial of service attacks usually prevents users from legitimately using the system. For example, if a denial-of-service attack is executed against the browser's content settings, a user may be unable to access the internet.

Threats to Operating System

There are various threats to the operating system. Some of them are as follows:

Malware

It contains viruses, worms, trojan horses, and other dangerous software. These are generally short code snippets that may corrupt files, delete the data, replicate to propagate further, and even crash a system. The malware frequently goes unnoticed by the victim user while criminals silently extract important data.

Network Intrusion

Network intruders are classified as masqueraders, misfeasors, and unauthorized users. A masquerader is an unauthorized person who gains access to a system and uses an authorized person's account. A misfeasor is a legitimate user who gains unauthorized access to and misuses programs, data, or resources. A rogue user takes supervisory authority and tries to evade access constraints and audit collection.

Buffer Overflow

It is also known as buffer overrun. It is the most common and dangerous security issue of the operating system. It is defined as a condition at an interface under which more input may be placed into a buffer and a data holding area than the allotted capacity, and it may overwrite other information. Attackers use such a situation to crash a system or insert specially created malware that allows them to take control of the system.

How to ensure Operating System Security?

There are various ways to ensure operating system security. These are as follows:

Authentication

The process of identifying every system user and associating the programs executing with those users is known as authentication. The operating system is responsible for implementing a security system that ensures the authenticity of a user who is executing a specific program. In general, operating systems identify and authenticate users in three ways.

1. Username/Password

Every user contains a unique username and password that should be input correctly before accessing a system.

2. User Attribution

These techniques usually include biometric verification, such as fingerprints, retina scans, etc. This authentication is based on user uniqueness and is compared to database samples already in the system. Users can only allow access if there is a match.

3. User card and Key

To login into the system, the user must punch a card into a card slot or enter a key produced by a key generator into an option provided by the operating system.

One Time passwords

Along with standard authentication, one-time passwords give an extra layer of security. Every time a user attempts to log into the One-Time Password system, a unique password is needed. Once a one-time password has been used, it cannot be reused. One-time passwords may be implemented in several ways.

1. Secret Key

The user is given a hardware device that can generate a secret id that is linked to the user's id. The system prompts for such a secret id, which must be generated each time you log in.

2. Random numbers

Users are given cards that have alphabets and numbers printed on them. The system requests numbers that correspond to a few alphabets chosen at random.

3. Network password

Some commercial applications issue one-time passwords to registered mobile/email addresses, which must be input before logging in.

Firewalls

Firewalls are essential for monitoring all incoming and outgoing traffic. It imposes local security, defining the traffic that may travel through it. Firewalls are an efficient way of protecting network systems or local systems from any network-based security threat.

Physical Security

The most important method of maintaining operating system security is physical security. An attacker with physical access to a system may edit, remove, or steal important files since operating system code and configuration files are stored on the hard drive.

Operating System Security Policies and Procedures

Various operating system security policies may be implemented based on the organization that you are working in. In general, an OS security policy is a document that specifies the procedures for ensuring that the operating system maintains a specific level of integrity, confidentiality, and availability.

OS Security protects systems and data from worms, malware, threats, ransomware, backdoor intrusions, viruses, etc. Security policies handle all preventative activities and procedures to ensure an operating system's protection, including steal, edited, and deleted data.

As OS security policies and procedures cover a large area, there are various techniques to addressing them. Some of them are as follows:

1. Installing and updating anti-virus software
2. Ensure the systems are patched or updated regularly
3. Implementing user management policies to protect user accounts and privileges.
4. Installing a firewall and ensuring that it is properly set to monitor all incoming and outgoing traffic.

OS security policies and procedures are developed and implemented to ensure that you must first determine which assets, systems, hardware, and date are the most vital to your organization. Once that is completed, a policy can be developed to secure and safeguard them properly.