COURSE NAME : OPERATING SYSTEMS

(COURSE CODE :22516)

COURSE TEACHER : MRS.T.H.GAVHANE

UNIT NO 6 (10Marks)

FILE MANAGEMNET

Course Outcome: Apply File management Techniques

UNIT OUTCOME

6a	**Explain the structure of given file system with example**
6b	**Describe mechanism of the given file access method**
6c	**Explain procedure to create and access directories and assign the given file access permissions**
6d	**Explain the features of the given Raid level structure of hard disk**

6.1 File – Concepts, Attributes Operations, types and file system structure

6.2 Access methods : Sequential , Direct Swapping, File allocation Methods- contiguous, Linked ,Indexed

6.3 Directory Structure :Single Level, two level, three-structured directory, Disk Organization and Disk structure-Physical structure and logical structure, Raid structure of dick, raid level 0 to 6

6.1 File – Concepts, Attributes Operations, types and file system structure

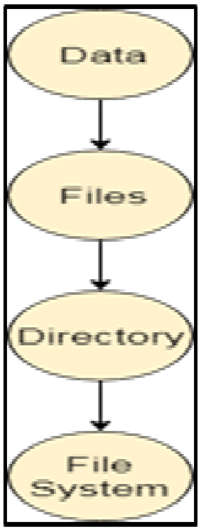
❖ What is a File ?

A file can be defined as a data structure which stores the sequence of records. Files are stored in a file system, which may exist on a disk or in the main memory. Files can be simple (plain text) or complex (specially-formatted).

OR

A file is a collection of correlated information which is recorded on secondary or non-volatile storage like magnetic disks, optical disks, and tapes. It is a method of data collection that is used as a medium for giving input and receiving output from that program.

In general, a file is a sequence of bits, bytes, or records whose meaning is defined by the file creator and user. Every File has a logical location where they are located for storage and retrieval.



The collection of files is known as Directory. The collection of directories at the different levels, is known as File System.

❖ Attributes of the File

1.Name

Every file carries a name by which the file is recognized in the file system. One directory cannot have two files with the same name.

2.Identifier

Along with the name, Each File has its own extension which identifies the type of the file. For example, a text file has the extension **.txt,** A video file can have the extension **.mp4.**

3.Type

In a File System, the Files are classified in different types such as video files, audio files, text files, executable files, etc.

4.Location

In the File System, there are several locations on which, the files can be stored. Each file carries its location as its attribute.

5.Size

The Size of the File is one of its most important attribute. By size of the file, we mean the number of bytes acquired by the file in the memory.

6.Protection

The Admin of the computer may want the different protections for the different files. Therefore each file carries its own set of permissions to the different group of Users.

7.Time and Date

Every file carries a time stamp which contains the time and date on which the file is last modified.

❖ Operations on the File

A file is a collection of logically related data that is recorded on the secondary storage in the form of sequence of operations. The content of the files are defined by its creator who is creating the file. The various operations which can be implemented on a file such as read, write, open and close etc. are called file operations. These operations are performed by the user by using the commands provided by the operating system. Some common operations are as follows:

1.Create operation:

This operation is used to create a file in the file system. It is the most widely used operation performed on the file system. To create a new file of a particular type the associated application program calls the file system. This file system allocates space to the file. As the file system knows the format of directory structure, so entry of this new file is made into the appropriate directory.

2. Open operation:

This operation is the common operation performed on the file. Once the file is created, it must be opened before performing the file processing operations. When the user wants to open a file, it provides a file name to open the particular file in the file system. It tells the operating system to invoke the open system call and passes the file name to the file system.

3. Write operation:

This operation is used to write the information into a file. A system call write is issued that specifies the name of the file and the length of the data has to be written to the file. Whenever the file length is increased by specified value and the file pointer is repositioned after the last byte written.

4. Read operation:

This operation reads the contents from a file. A Read pointer is maintained by the OS, pointing to the position up to which the data has been read.

5. Re-position or Seek operation:

The seek system call re-positions the file pointers from the current position to a specific place in the file i.e. forward or backward depending upon the user's requirement. This operation is generally performed with those file management systems that support direct access files.

6. Delete operation:

Deleting the file will not only delete all the data stored inside the file it is also used so that disk space occupied by it is freed. In order to delete the specified file the directory is searched. When the directory entry is located, all the associated file space and the directory entry is released.

7. Truncate operation:

Truncating is simply deleting the file except deleting attributes. The file is not completely deleted although the information stored inside the file gets replaced.

8. Close operation:

When the processing of the file is complete, it should be closed so that all the changes made permanent and all the resources occupied should be released. On closing it deallocates all the internal descriptors that were created when the file was opened.

9. Append operation:

This operation adds data to the end of the file.

10. Rename operation:

This operation is used to rename the existing file.

❖ Types of File :

There are numerous file types that an operating system uses internally and are not generally used or required by the system user. These files could be application software files, kernel files, configuration files, metadata files, etc. Windows supports the following file types:

1. Regular Files

Regular files consist of information related to the user. The files are usually either ASCII or binary. ASCII files contain lines of text. The major benefit of an ASCII file is that it can be displayed or printed as it is, and it can be edited using a text editor.

Binary files on printing may give some random junk content. Usually, a binary file would have some sort of internal structure that is only known to the program that uses it. A binary file is a sequence of bytes, which if is in the proper format, can be executed by the operating system. Regular files are supported by both Windows as well as UNIX-based operating systems.

2. Directories

A directory in the file system is a structure that contains references to other files and possibly other directories. Files could be arranged by storing related files in the same directory. Directories are supported by both Windows as well as UNIX-based operating systems.

3. Character Special Files

A character special file provides access to an I/O device. Examples of character special files include a terminal file, a system console file, a NULL file, a file descriptor file, etc.

Each character special file has a device major number and a device minor number. The device major number associated with a character special file identifies the device type. The device minor number associated with a character special file identifies a specific device of a given device type. Character special files are supported by UNIX-based operating systems.

4. Block Special Files

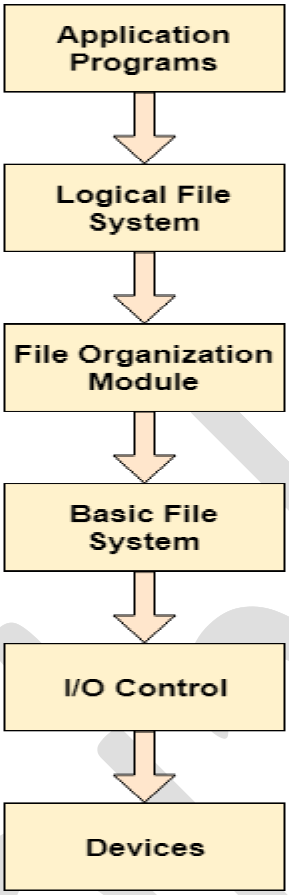
Block special files enable buffered access to hardware devices They also provide some abstraction from their specifics. Unlike character special files, block special files always allow the programmer to read and write a block of any size or alignment. Block special files are supported by UNIX-based operating systems.

❖ File System Structure :

File System provide efficient access to the disk by allowing data to be stored, located and retrieved in a convenient way. A file System must be able to store the file, locate the file and retrieve the file.

Most of the Operating Systems use layering approach for every task including file systems. Every layer of the file system is responsible for some activities.

The image shown below, elaborates how the file system is divided in different layers, and also the functionality of each layer.



- When an application program asks for a file, the first request is directed to the logical file system. The logical file system contains the Meta data of the file and directory structure. If the application program doesn't have the required permissions of the file then this layer will throw an error. Logical file systems also verify the path to the file.
- Generally, files are divided into various logical blocks. Files are to be stored in the hard disk and to be retrieved from the hard disk. Hard disk is divided into various tracks and sectors. Therefore, in order to store and retrieve the files, the logical blocks need to be mapped to physical blocks. This mapping is done by File organization module. It is also responsible for free space management.
- Once File organization module decided which physical block the application program needs, it passes this information to basic file system. The basic file system is responsible for issuing the commands to I/O control in order to fetch those blocks.
- I/O controls contain the codes by using which it can access hard disk. These codes are known as device drivers. I/O controls are also responsible for handling interrupts.

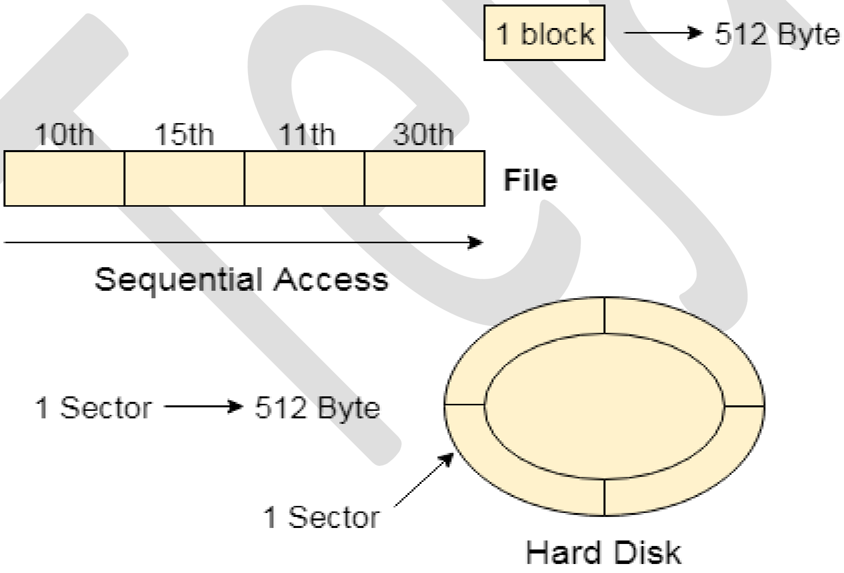
6.2 Access methods : Sequential , Direct Swapping, File allocation Methods- contiguous, Linked ,Indexed

File access is a process that determines the way that files are accessed and read into memory. Generally, a single access method is always supported by operating systems. Though there are some operating system which also supports multiple access methods.

Three file access methods are:

- Sequential access
- Direct random access
- Index sequential access

Sequential Access



Most of the operating systems access the file sequentially. In other words, we can say that most of the files need to be accessed sequentially by the operating system.

In sequential access, the OS read the file word by word. A pointer is maintained which initially points to the base address of the file. If the user wants to read first word of the file then the pointer provides that word to the user and increases its value by 1 word. This process continues till the end of the file.

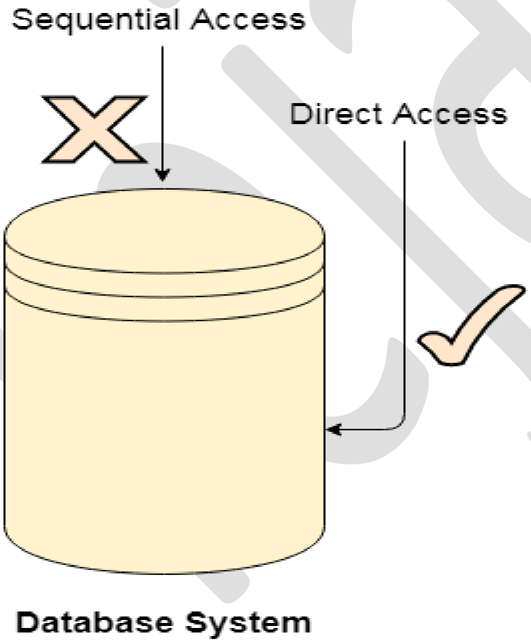
Modern word systems do provide the concept of direct access and indexed access but the most used method is sequential access due to the fact that most of the files such as text files, audio files, video files, etc need to be sequentially accessed.

Direct Access

The Direct Access is mostly required in the case of database systems. In most of the cases, we need filtered information from the database. The sequential access can be very slow and inefficient in such cases.

Suppose every block of the storage stores 4 records and we know that the record we needed is stored in 10th block. In that case, the sequential access will not be implemented because it will traverse all the blocks in order to access the needed record.

Direct access will give the required result despite of the fact that the operating system has to perform some complex tasks such as determining the desired block number. However, that is generally implemented in database applications.



Indexed Access

If a file can be sorted on any of the filed then an index can be assigned to a group of certain records. However, A particular record can be accessed by its index. The index is nothing but the address of a record in the file.

In index accessing, searching in a large database became very quick and easy but we need to have some extra space in the memory to store the index value.

❖ File allocation Methods :

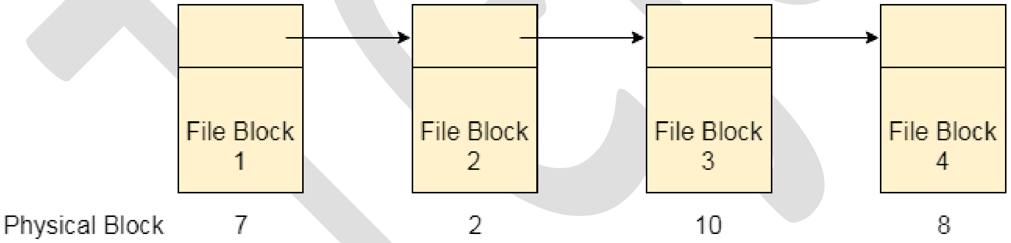
There are various methods which can be used to allocate disk space to the files. Selection of an appropriate allocation method will significantly affect the performance and efficiency of the system. Allocation method provides a way in which the disk will be utilized and the files will be accessed.

There are following methods which can be used for allocation.

- Linked Allocation
- Indexed Allocation
- Contiguous Allocation

a. Linked List Allocation

Linked List allocation solves all problems of contiguous allocation. In linked list allocation, each file is considered as the linked list of disk blocks. However, the disks blocks allocated to a particular file need not to be contiguous on the disk. Each disk block allocated to a file contains a pointer which points to the next disk block allocated to the same file.



Linked List Allocation

Advantages

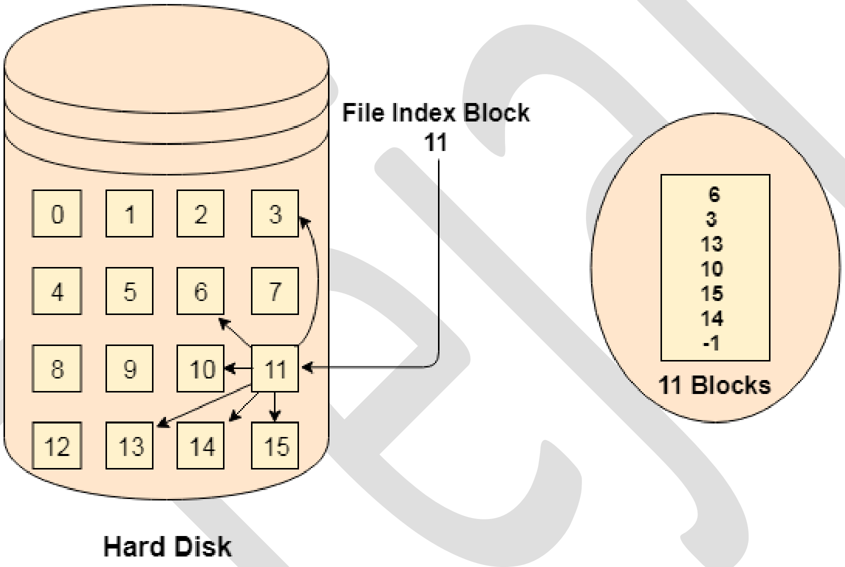
1. There is no external fragmentation with linked allocation.
2. Any free block can be utilized in order to satisfy the file block requests.
3. File can continue to grow as long as the free blocks are available.
4. Directory entry will only contain the starting block address.

Disadvantages

1. Random Access is not provided.
2. Pointers require some space in the disk blocks.
3. Any of the pointers in the linked list must not be broken otherwise the file will get corrupted.
4. Need to traverse each block.

b. Indexed Allocation

Instead of maintaining a file allocation table of all the disk pointers, Indexed allocation scheme stores all the disk pointers in one of the blocks called as indexed block. Indexed block doesn't hold the file data, but it holds the pointers to all the disk blocks allocated to that particular file. Directory entry will only contain the index block address.



Advantages

1. Supports direct access
2. A bad data block causes the lost of only that block.

Disadvantages

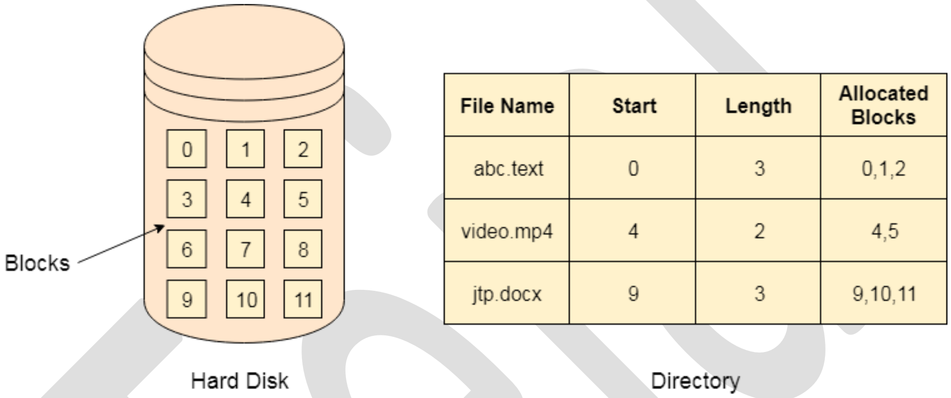
1. A bad index block could cause the lost of entire file.
2. Size of a file depends upon the number of pointers, a index block can hold.
3. Having an index block for a small file is totally wastage.

4. More pointer overhead

c. Contiguous Allocation

If the blocks are allocated to the file in such a way that all the logical blocks of the file get the contiguous physical block in the hard disk then such allocation scheme is known as contiguous allocation.

In the image shown below, there are three files in the directory. The starting block and the length of each file are mentioned in the table. We can check in the table that the contiguous blocks are assigned to each file as per its need.



File Name	Start	Length	Allocated Blocks
abc.text	0	3	0,1,2
video.mp4	4	2	4,5
jtp.docx	9	3	9,10,11

Contiguous Allocation

Advantages

1. It is simple to implement.
2. We will get Excellent read performance.
3. Supports Random Access into files.

Disadvantages

1. The disk will become fragmented.
2. It may be difficult to have a file grow.

6.3 Directory Structure :Single Level, two level, three-structured directory, Disk Organization and Disk structure-Physical structure and logical structure, Raid structure of disk, raid level 0 to 6

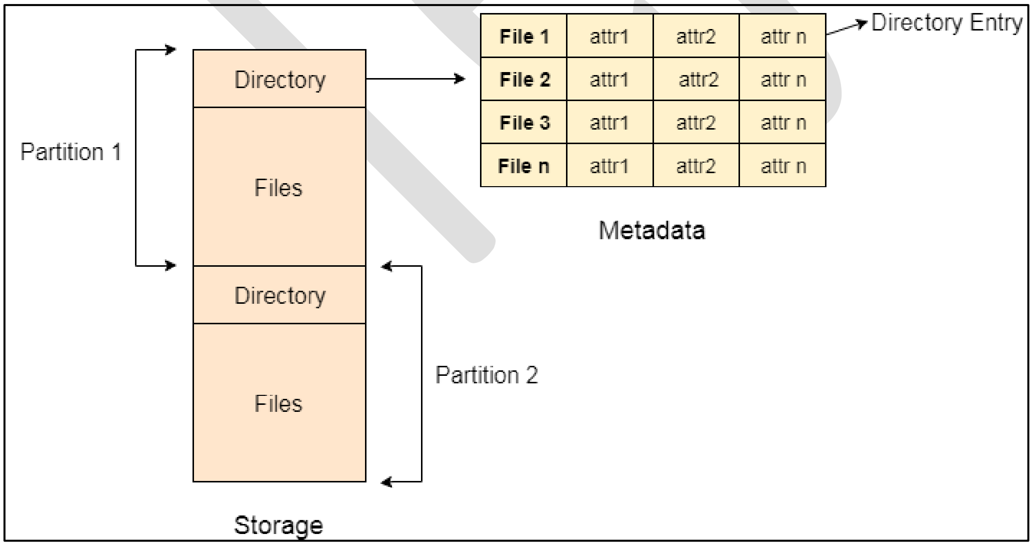
Directory Structure

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.



File 1	attr1	attr2	attr n
File 2	attr1	attr2	attr n
File 3	attr1	attr2	attr n
File n	attr1	attr2	attr n

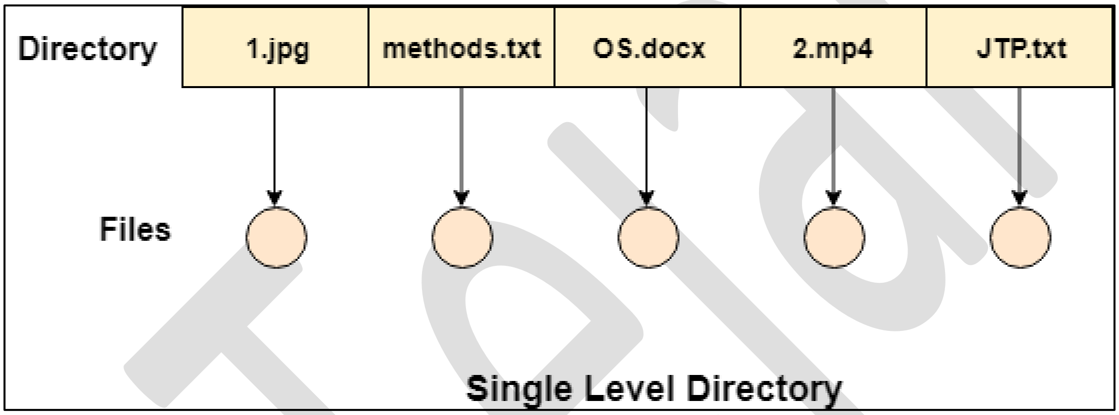
A directory can be viewed as a file which contains the Meta data of the bunch of files.

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

Single Level Directory

The simplest method is to have one big list of all the files on the disk. The entire system will contain only one directory which is supposed to mention all the files present in the file system. The directory contains one entry per each file present on the file system.



This type of directories can be used for a simple system.

Advantages

1. Implementation is very simple.
2. If the sizes of the files are very small then the searching becomes faster.
3. File creation, searching, deletion is very simple since we have only one directory.

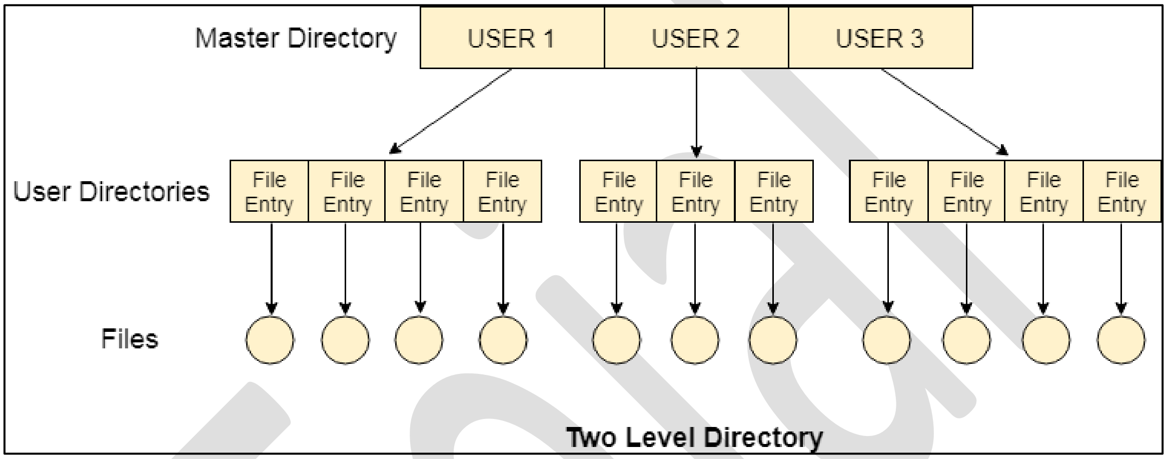
Disadvantages

1. We cannot have two files with the same name.
2. The directory may be very big therefore searching for a file may take so much time.
3. Protection cannot be implemented for multiple users.
4. There are no ways to group same kind of files.

5. Choosing the unique name for every file is a bit complex and limits the number of files in the system because most of the Operating System limits the number of characters used to construct the file name.

Two Level Directory

In two level directory systems, we can create a separate directory for each user. There is one master directory which contains separate directories dedicated to each user. For each user, there is a different directory present at the second level, containing group of user's file. The system doesn't let a user to enter in the other user's directory without permission.



Characteristics of two level directory system

1. Each files has a path name as ***/User-name/directory-name/***
2. Different users can have the same file name.
3. Searching becomes more efficient as only one user's list needs to be traversed.
4. The same kind of files cannot be grouped into a single directory for a particular user.

Every Operating System maintains a variable as **PWD** which contains the present directory name (present user name) so that the searching can be done appropriately.

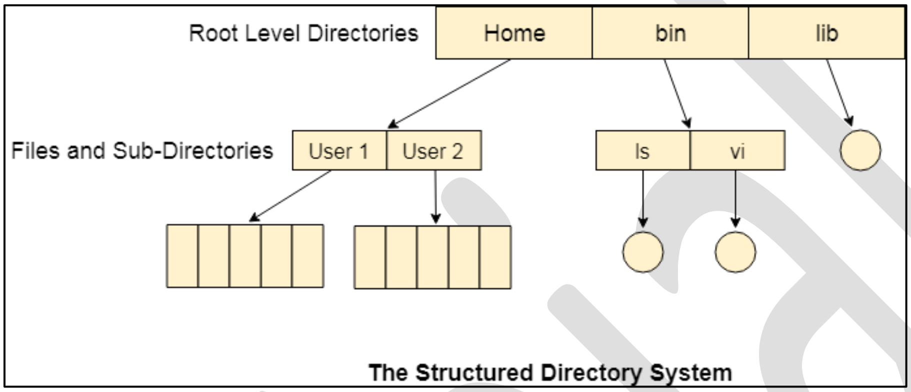
Tree Structured Directory

In Tree structured directory system, any directory entry can either be a file or sub directory. Tree structured directory system overcomes the drawbacks of two level directory system. The similar kind of files can now be grouped in one directory.

Each user has its own directory and it cannot enter in the other user's directory. However, the user has the permission to read the root's data but he cannot write or modify this. Only administrator of the system has the complete access of root directory.

Searching is more efficient in this directory structure. The concept of current working directory is used. A file can be accessed by two types of path, either relative or absolute.

Absolute path is the path of the file with respect to the root directory of the system while relative path is the path with respect to the current working directory of the system. In tree structured directory systems, the user is given the privilege to create the files as well as directories.



Permissions on the file and directory

A tree structured directory system may consist of various levels therefore there is a set of permissions assigned to each file and directory.

The permissions are **R W X** which are regarding reading, writing and the execution of the files or directory. The permissions are assigned to three types of users: owner, group and others.

There is a identification bit which differentiate between directory and file. For a directory, it is **d** and for a file, it is dot **(.)**

The following snapshot shows the permissions assigned to a file in a Linux based system. Initial bit **d** represents that it is a directory.

❖ Disk Organization/ Disk Structure :

• Physical Structure

Hard disks are secondary storage devices we can use to store data. Most modern computers use hard disks to store large amounts of data.

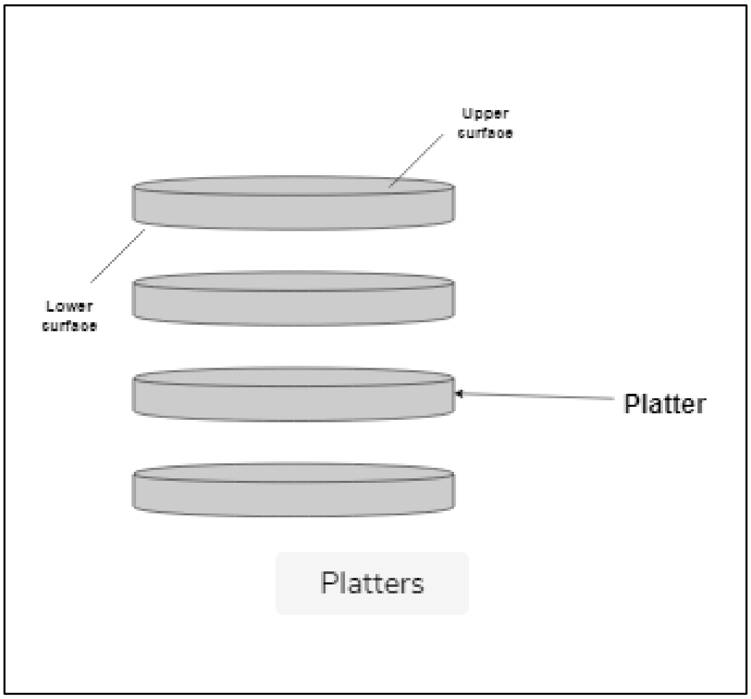
The architecture of a hard disk consists of several physical components that include:

- Platters
- Spindle
- Read/write heads
- Tracks
- Sectors

Platters

Hard disks are organized as a concentric stack of disks. An individual disk is referred to as a **platter**.

Each platter consists of two surfaces: a lower and an upper surface.

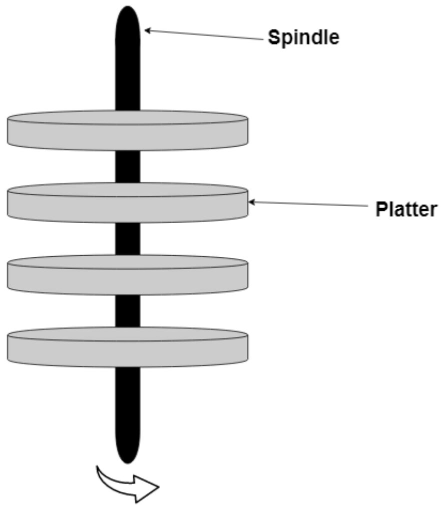


Spindle

The platters within the hard disk are connected by a **spindle** that runs through the middle of the platters.

The spindle moves in a unidirectional manner along its axis (either clockwise or counterclockwise).

The movement of the spindle causes the platters to rotate as well.

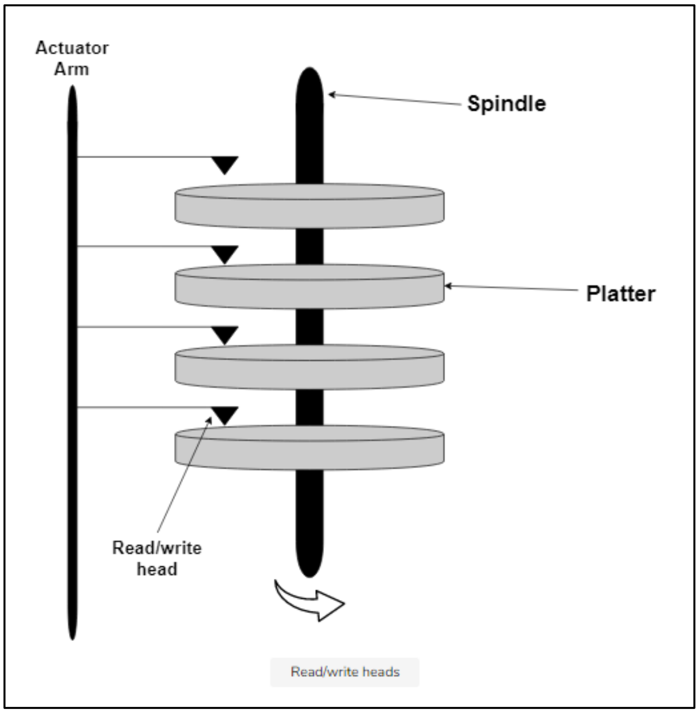


Spindle

Read/write head

Each surface on a platter contains a **read/write head** that is used to read or write data onto the disk.

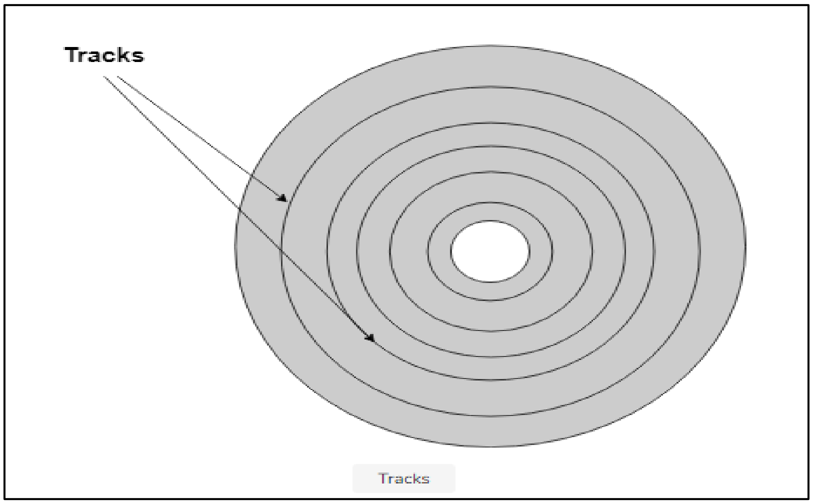
The read/write heads can move back and forth along the surface of a platter. Read/write heads are in turn connected to a single *actuator arm*.



Tracks

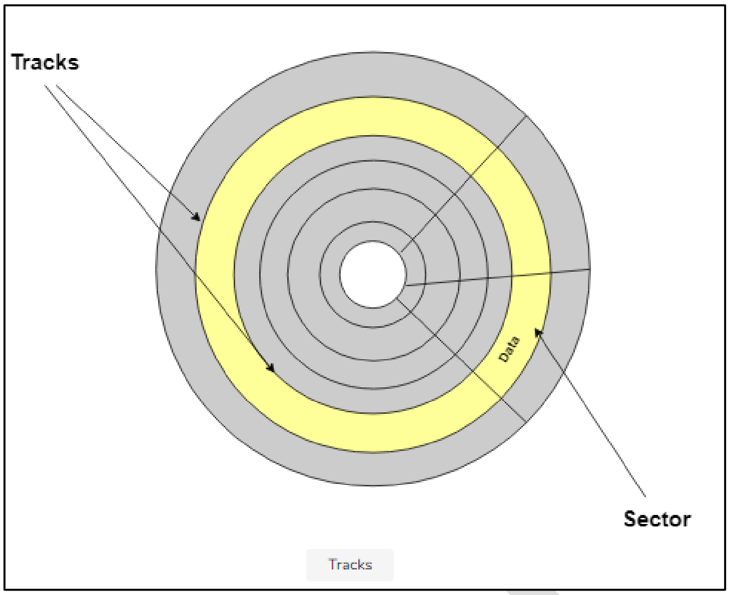
Each surface of a platter consists of a fixed number of **tracks**. These are circular areas on the surface of a platter that decrease in circumference as we move towards the center of the platter.

Data is first written to the outermost track.



Sectors

Each track is divided into a fixed number of **sectors**. Sectors divide track sections and store data.



Logical Structure :

Logical division of hard disk is generally divided into 5 basic terms which are

- MBR (Master Boot Record)
- DBR (DOS Boot Record)
- FAT (File Allocation Tables)
- Root Directory
- Data Area

Master Boot Record:

• It contains a small program to load and start the active partition from the hard disk.

• It is located at absolute sector 0 or we can say at cylinder 0,head 0 and sector 1(The MBR).

• If we have more than one partition, then there are Extended Master Boot Records, located at the beginning of each extended partition volume.

Dos Boot Record:

- After the partition table , the DOS Boot Record(DBR) or sometimes called DOS Boot Sector is the second most important information on your hard disk.

- It contains some important information about disk geometry like:
   Bytes Per Sector
   Sectors per cluster
   Reserved Sectors
- The job of DBR is to load the operating system from the hard disk drive into the main memory of computer and give the systems control to the loaded program

File Allocation Table:

- FAT keeps a map of the complete surface of the disk drive such that, which area is free, which area is taken up by which file etc.
- The FAT manages the disk area in a group of sectors called “CLUSTER”. Root Directory:
- The Root Directory is like a table of contents for the information stored on the hard disk drive.
- The directory area keeps the information about the file name, date and time of the file creation, file attribute, file size and starting cluster of the particular file.

Data Area OR Files Area:

- The data area contains the actual data stored on the disk surfaces.

❖ Raid structure of disk, raid level 0 to 6

RAID (redundant array of independent disks) is a setup consisting of multiple disks for data storage. They are linked together to prevent data loss and/or speed up performance. Having multiple disks allows the employment of various techniques like **disk striping**, **disk mirroring**, and **parity**.

RAID Levels and Types

RAID levels are grouped into the following categories:

- Standard RAID levels
- Non-standard RAID levels
- Nested/hybrid RAID levels

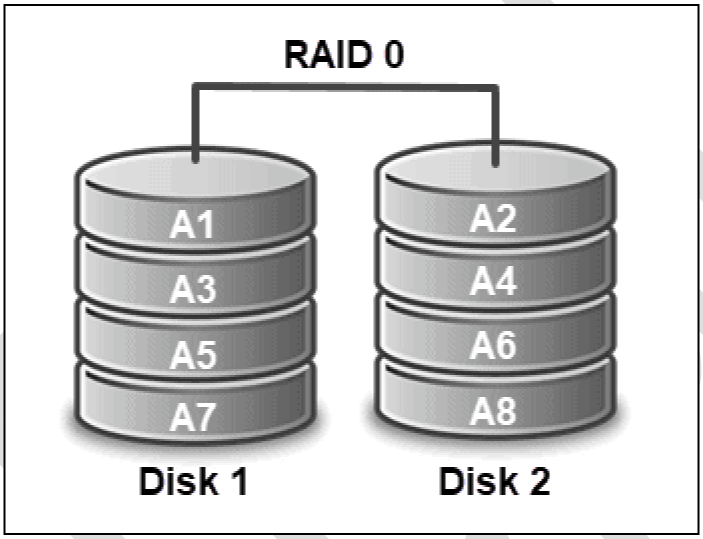
Additionally, you can choose how to implement RAID on your system. Therefore you can choose between hardware RAID, software RAID, and firmware RAID.

The following list explains the standard RAID levels (0, 1, 2, 3, 4, 5, 6) and popular non-standard and hybrid options (RAID 10).

RAID 0: Striping

RAID 0, also known as a striped set or a striped volume, requires a minimum of two disks. The disks are merged into a single large volume where data is stored evenly across the number of disks in the array.

This process is called disk striping and involves splitting data into blocks and writing it simultaneously/sequentially on multiple disks. Configuring the striped disks as a single partition increases performance since multiple disks do reading and writing operations simultaneously. Therefore, RAID 0 is generally implemented to improve speed and efficiency.



It is important to note that if an array consists of disks of different sizes, each will be limited to the smallest disk size in the setup. This means that an array composed of two disks, where one is 320 GB, and the other is 120 GB, actually has the capacity of 2 x 120 GB (or 240 GB in total).

Certain implementations allow you to utilize the remaining 200 GB for different use. Additionally, developers can implement multiple controllers (or even one per disk) to improve performance.

RAID 0 is the most affordable type of redundant disk configuration and is relatively easy to set up. Still, it does not include any redundancy, fault tolerance, or party in its composition. Hence, problems on any of the disks in the array can result in complete data loss. This is why it should only be used for non-critical storage, such as temporary files backed up somewhere else.

Advantages of RAID 0

- Cost-efficient and straightforward to implement.
- Increased read and write performance.
- No overhead (total capacity use).

Disadvantages of RAID 0

- Doesn't provide fault tolerance or redundancy.

When Raid 0 Should Be Used

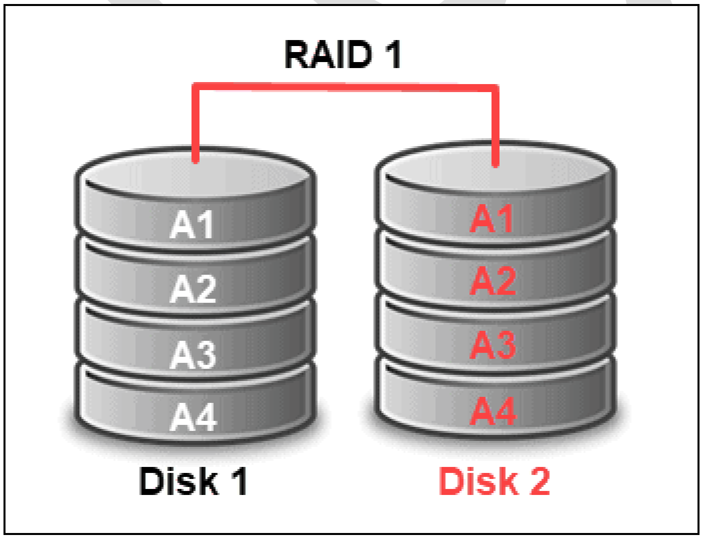
RAID 0 is used when performance is a priority and reliability is not. If you want to utilize your drives to the fullest and don't mind losing data, opt for RAID 0.

On the other hand, such a configuration does not necessarily have to be unreliable. You can set up disk striping on your system along with another RAID array that ensures data protection and redundancy.

RAID 1: Mirroring

RAID 1 is an array consisting of at least two disks where the same data is stored on each to ensure redundancy. The most common use of RAID 1 is setting up a mirrored pair consisting of two disks in which the contents of the first disk is mirrored in the second. This is why such a configuration is also called mirroring.

Unlike with RAID 0, where the focus is solely on speed and performance, the primary goal of RAID 1 is to provide redundancy. It eliminates the possibility of data loss and downtime by replacing a failed drive with its replica.



In such a setup, the array volume is as big as the smallest disk and operates as long as one drive is operational. Apart from reliability, mirroring enhances read performance as a request can be handled by any of the drives in the array. On the other hand, the write performance remains the same as with one disk and is equal to the slowest disk in the configuration.

Advantages of RAID 1

- Increased read performance.
- Provides redundancy and fault tolerance.
- Simple to configure and easy to use.

Disadvantages of RAID 1

- Uses only half of the storage capacity.
- More expensive (needs twice as many drivers).
- Requires powering down your computer to replace failed drive.

When Raid 1 Should Be Used

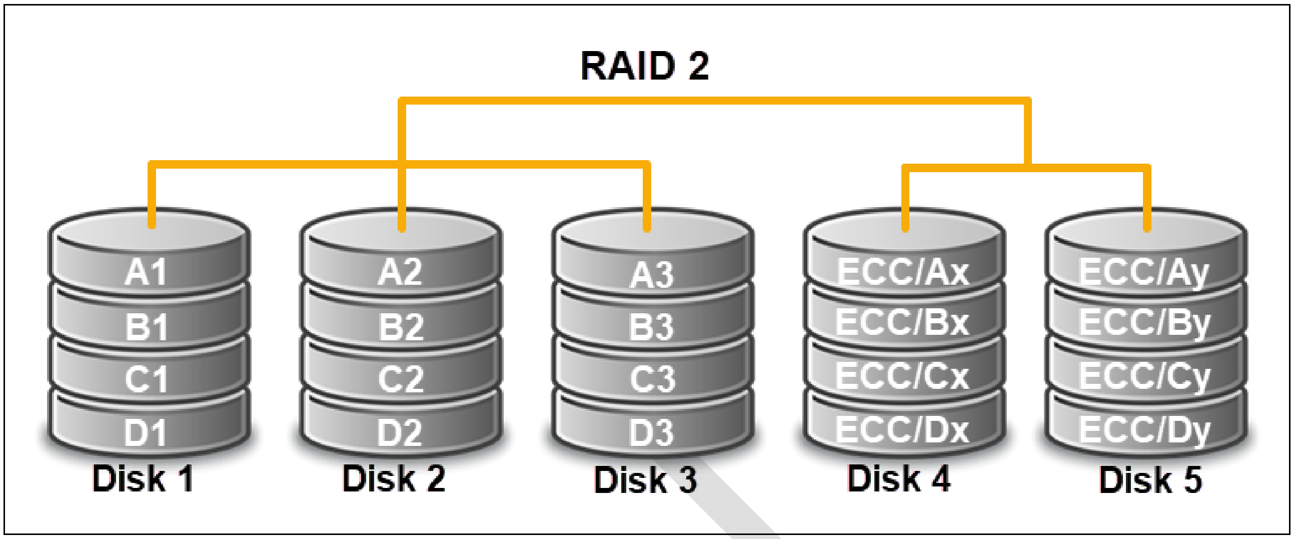
RAID 1 is used for mission-critical storage that requires a minimal risk of data loss. Accounting systems often opt for RAID 1 as they deal with critical data and require high reliability.

It is also suitable for smaller servers with only two disks, as well as if you are searching for a simple configuration you can easily set up (even at home).

Raid 2: Bit-Level Striping with Dedicated Hamming-Code Parity

RAID 2 is rarely used in practice today. It combines bit-level striping with error checking and information correction. This RAID implementation requires two groups of disks – one for writing the data and another for writing error correction codes. RAID 2 also requires a special controller for the synchronized spinning of all disks.

Instead of data blocks, RAID 2 stripes data at the bit level across multiple disks. Additionally, it uses the Humming error ode correction (ECC) and stores this information on the redundancy disk.



The array calculates the error code correction on the fly. While writing the data, it strips it to the data disk and writes the code to the redundancy disk. On the other hand, while reading data from the disk, it also reads from the redundancy disk to verify the data and make corrections if needed.

Advantages of RAID 2

- Reliability.
- The ability to correct stored information.

Disadvantages of RAID 2

- Expensive.
- Difficult to implement.
- Require entire disks for ECC.

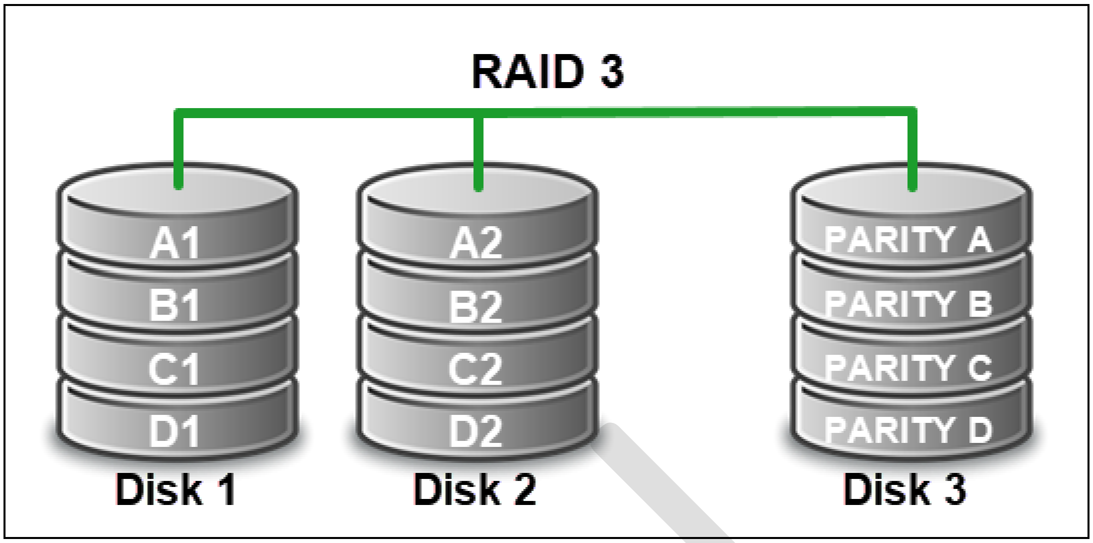
When Raid 2 Should Be Used

RAID 2 is not a common practice today as most of its features are now available on modern hard disks. Due to its cost and implementation requirements, this RAID level never became popular among developers.

Raid 3: Bit-Level Striping with Dedicated Parity

Like RAID 2, RAID 3 is rarely used in practice. This RAID implementation utilizes bit-level striping and a dedicated parity disk. Because of this, it requires at least three drives, where two are used for storing data strips, and one is used for parity.

To allow synchronized spinning, RAID 3 also needs a special controller. Due to its configuration and synchronized disk spinning, it achieves better performance rates with sequential operations than random read/write operations.



Advantages of RAID 3

- Good throughput when transferring large amounts of data.
- High efficiency with sequential operations.
- Disk failure resiliency.

Disadvantages of RAID 3

- Not suitable for transferring small files.
- Complex to implement.
- Difficult to set up as software RAID.

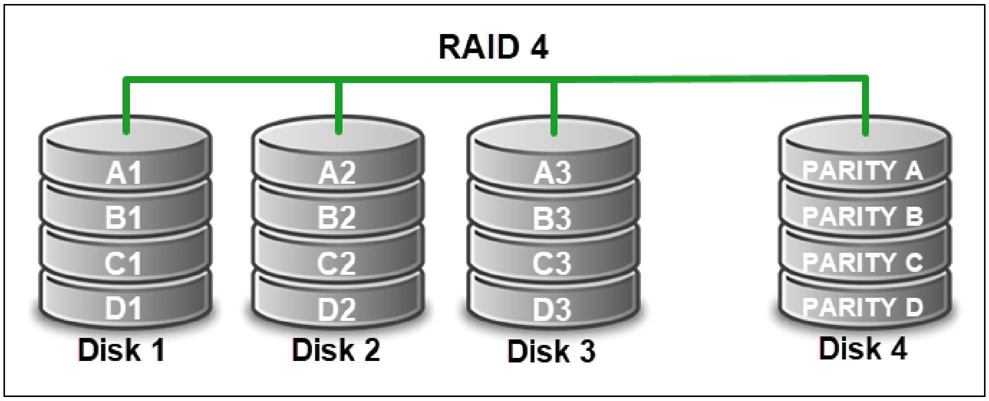
When Raid 3 Should Be Used

RAID 3 is not commonly used today. Its features are beneficial to a limited number of use cases requiring high transfer rates for long sequential reads and writes (such as video editing and production).

Raid 4: Block-Level Striping with Dedicated Parity

RAID 4 is another unpopular standard RAID level. It consists of block-level data striping across two or more independent diss and a dedicated parity disk.

The implementation requires at least three disks – two for storing data strips and one dedicated for storing parity and providing redundancy. As each disk is independent and there is no synchronized spinning, there is no need for a controller.



RAID 4 configuration is prone to bottlenecks when storing parity bits for each data block on a single drive. Such system bottlenecks have a large impact on system performance.

Advantages of RAID 4

- Fast read operations.
- Low storage overhead.
- Simultaneous I/O requests.

Disadvantages of RAID 4

- Bottlenecks that have big effect on overall performance.
- Slow write operations.
- Redundancy is lost if the parity disk fails.

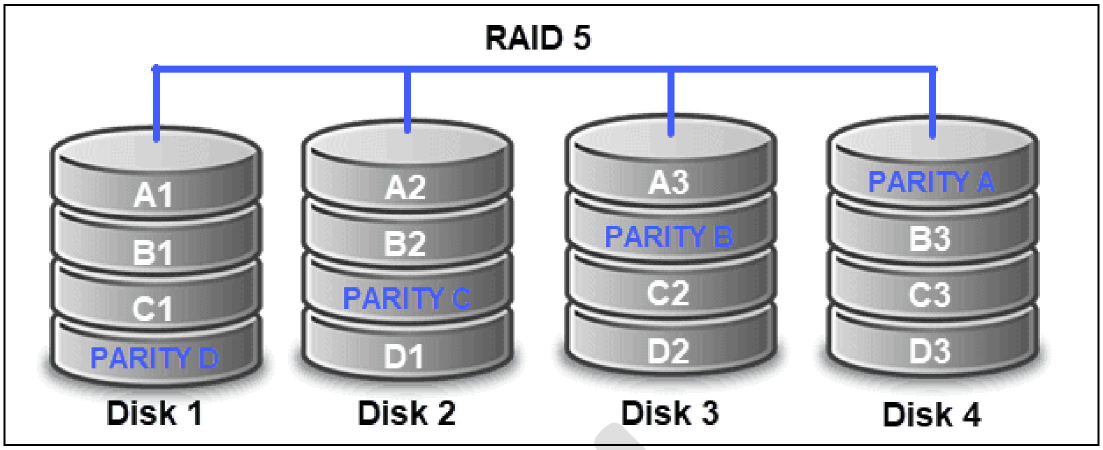
When Raid 4 Should Be Used

Considering its configuration, RAID 4 works best with use cases requiring sequential reading and writing data processes of huge files. Still, just like with RAID 3, in most solutions, RAID 4 has been replaced with RAID 5.

Raid 5: Striping with Parity

RAID 5 is considered the most secure and most common RAID implementation. It combines striping and parity to provide a fast and reliable setup. Such a configuration gives the user storage usability as with RAID 1 and the performance efficiency of RAID 0.

This RAID level consists of at least three hard drives (and at most, 16). Data is divided into data strips and distributed across different disks in the array. This allows for high performance rates due to fast read data transactions which can be done simultaneously by different drives in the array.



Parity bits are distributed evenly on all disks after each sequence of data has been saved. This feature ensures that you still have access to the data from parity bits in case of a failed drive. Therefore, RAID 5 provides redundancy through parity bits instead of mirroring.

Advantages of RAID 5

- High performance and capacity.
- Fast and reliable read speed.
- Tolerates single drive failure.

Disadvantages of RAID 5

- Longer rebuild time.
- Uses half of the storage capacity (due to parity).
- If more than one disk fails, data is lost.
- More complex to implement.

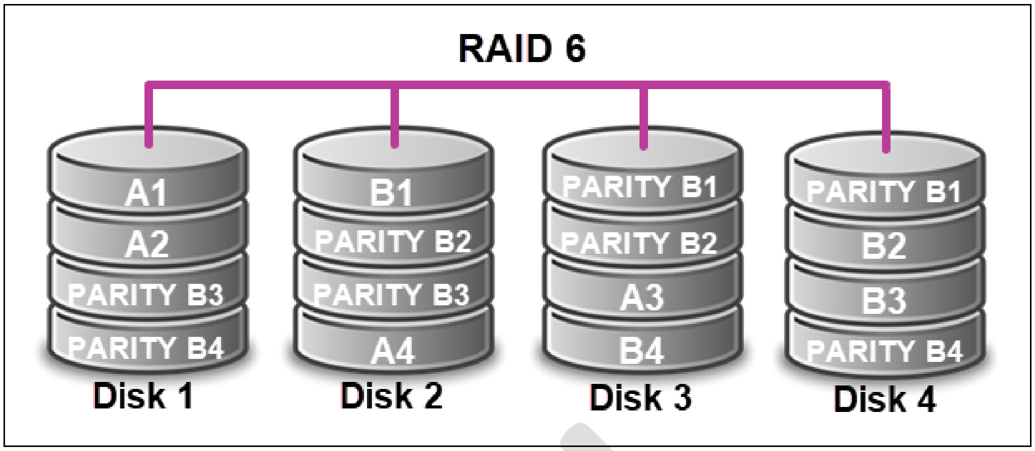
When Raid 5 Should Be Used

RAID 5 is often used for file and application servers because of its high efficiency and optimized storage. Additionally, it is the best, cost-effective solution if continuous data access is a priority and/or you require installing an operating system on the array.

Raid 6: Striping with Double Parity

RAID 6 is an array similar to RAID 5 with an addition of its double parity feature. For this reason, it is also referred to as the double-parity RAID.

This setup requires a minimum of four drives. The setup resembles RAID 5 but includes two additional parity blocks distributed across the disk. Therefore, it uses block-level striping to distribute the data across the array and stores two parity blocks for each data block.



Block-level striping with two parity blocks allows two disk failures before any data is lost. This means that in an event where two disks fail, RAID can still reconstruct the required data.

Its performance depends on how the array is implemented, as well as the total number of drives. Write operations are slower compared to other configurations due to its double parity feature.

Advantages of RAID 6

- High fault and drive-failure tolerance.
- Storage efficiency (when more than four drives are used).
- Fast read operations.

Disadvantages of RAID 6

- Rebuild time can take up to 24 hours.
- Slow write performance.
- Complex to implement.
- More expensive.

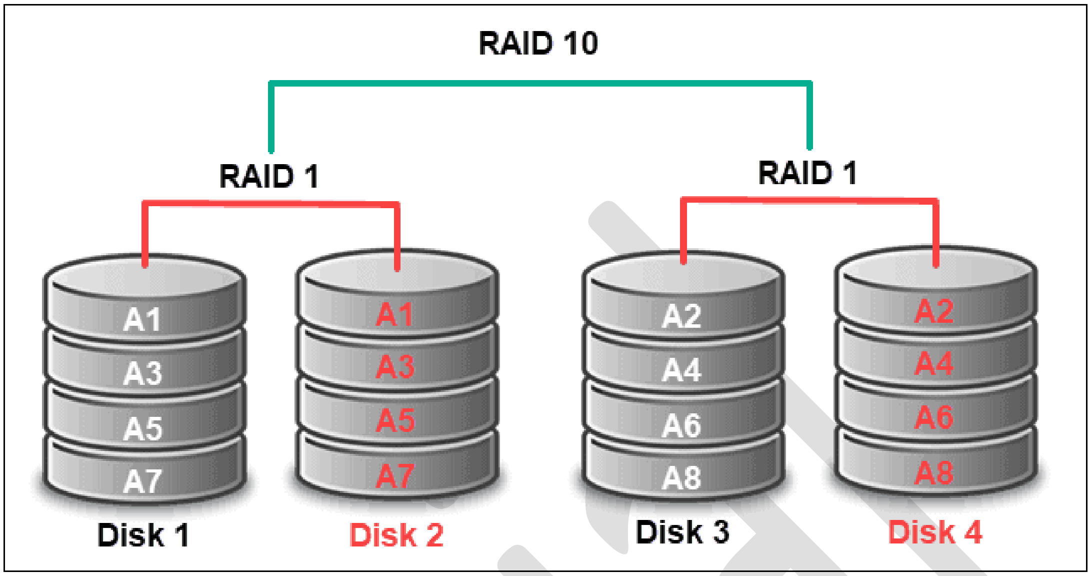
When Raid 6 Should Be Used

RAID 6 is a good solution for mission-critical applications where data loss cannot be tolerated. Therefore, it is often used for data management in defense sectors, healthcare, and banking.

Raid 10: Mirroring with Striping

RAID 10 is part of a group called nested or hybrid RAID, which means it is a combination of two different RAID levels. In the case of RAID 10, the array combines level 1 mirroring and level 0 striping. This RAID array is also known as RAID 1+0.

RAID 10 uses logical mirroring to write the same data on two or more drives to provide redundancy. If one disk fails, there is a mirrored image of the data stored on another disk. Additionally, the array uses block-level striping to distribute chunks of data across different drives. This improves performance and read and write speed as the data is simultaneously accessed from multiple disks.



To implement such a configuration, the array requires at least four drives, as well as a disk controller.

Advantages of RAID 10

- High performance.
- High fault-tolerance.
- Fast read and write operations.
- Fast rebuild time.

Disadvantages of RAID 10

- Limited scalability.
- Costly (compared to other RAID levels).
- Uses half of the disk space capacity.
- More complicated to set up.

When Raid 10 Should Be Used

RAID 10 is often used in use cases that require storing high volumes of data, fast read and write times, and high fault tolerance. Accordingly, this RAID level is often implemented for email servers, web hosting servers, and databases.