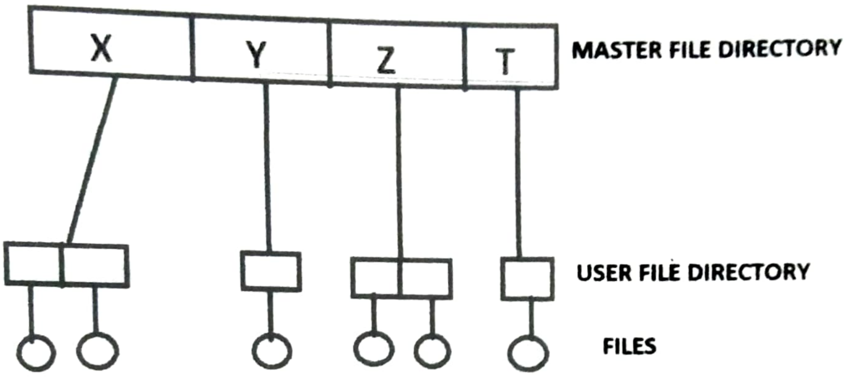
Difference between File System and DBMS

1. File System :

File system is basically a way of arranging the files in a storage medium like hard disk. File system organizes the files and helps in retrieval of files when they are required. File systems consists of different files which are grouped into directories. The directories further contain other folders and files. File system performs basic operations like management, file naming, giving access rules etc.

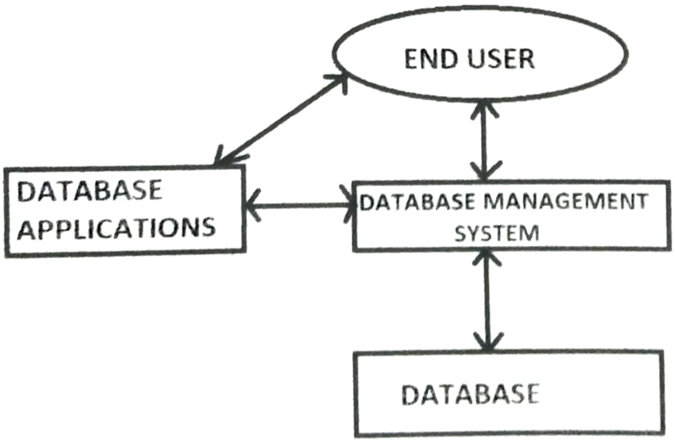
Example:

NTFS(New Technology File System), EXT(Extended File System).



2. DBMS(Database Management System) :

Database Management System is basically a software that manages the collection of related data. It is used for storing data and retrieving the data effectively when it is needed. It also provides proper security measures for protecting the data from unauthorized access. In Database Management System the data can be fetched by SQL queries and relational algebra. It also provides mechanisms for data recovery and data backup.



Example:

Oracle, MySQL, MS SQL server.

S.NO.	FILE SYSTEM	DBMS
1.	File system is a software that manages and organizes the files in a storage medium within a computer.	DBMS is a software for managing the database.
2.	Redundant data can be present in a file system.	In DBMS there is no redundant data.
3.	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
4.	There is no efficient query processing in file system.	Efficient query processing is there in DBMS
5.	There is less data consistency in file system.	There is more data consistency because of the process of normalization.
6.	It is less complex as compared to DBMS.	It has more complexity in handling as compared to file system.
7.	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file system.

DISTRIBUTED FILE SYSTEM

A **Distributed File System (DFS)** as the name suggests, is a file system that is distributed on multiple file servers or multiple locations. It allows programs to access or store isolated files as they do with the local ones, allowing programmers to access files from any network or computer.

The main purpose of the Distributed File System (DFS) is to allows users of physically distributed systems to share their data and resources by using a Common File System. A collection of workstations and mainframes connected by a Local Area Network (LAN) is a configuration on Distributed File System. A DFS is executed as a part of the operating system. In DFS, a namespace is created and this process is transparent for the clients.

DFS has two components:

- **Location Transparency:** Location Transparency achieves through the namespace component.
- **Redundancy:** Redundancy is done through a file replication component.

In the case of failure and heavy load, these components together improve data availability by allowing the sharing of data in different locations to be logically grouped under one folder, which is known as the "DFS root".

It is not necessary to use both the two components of DFS together, it is possible to use the namespace component without using the file replication component and it is perfectly possible to use the file replication component without using the namespace component between servers.

Features of DFS

- **Transparency:**
  - **Structure transparency:** There is no need for the client to know about the number or locations of file servers and the storage devices. Multiple file servers should be provided for performance, adaptability, and dependability.
  - **Access transparency:** Both local and remote files should be accessible in the same manner. The file system should be automatically located on the accessed file and send it to the client's side.
  - **Naming transparency:** There should not be any hint in the name of the file to the location of the file. Once a name is given to the file, it should not be changed during transferring from one node to another.
  - **Replication transparency:** If a file is copied on multiple nodes, both the copies of the file and their locations should be hidden from one node to another.
- **User mobility:** It will automatically bring the user's home directory to the node where the user logs in.
- **Performance:** Performance is based on the average amount of time needed to convince the client requests. This time covers the CPU time + time taken to access secondary storage + network access time. It is advisable that the performance of the Distributed File System be similar to that of a centralized file system.
- **Simplicity and ease of use:** The user interface of a file system should be simple and the number of commands in the file should be small.
- **High availability:** A Distributed File System should be able to continue in case of any partial failures like a link failure, a node failure, or a storage drive crash.
  A high authentic and adaptable distributed file system should have different and independent file servers for controlling different and independent storage devices.

History

The server component of the Distributed File System was initially introduced as an add-on feature. It was added to Windows NT 4.0 Server and was known as "DFS 4.1". Then later on it was included as a standard component for all editions of Windows 2000 Server. Client-side support has been included in Windows NT 4.0 and also in later on version of Windows.

Linux kernels 2.6.14 and versions after it come with an SMB client VFS known as "cifs" which supports DFS. Mac OS X 10.7 (lion) and onwards supports Mac OS X DFS.

Applications

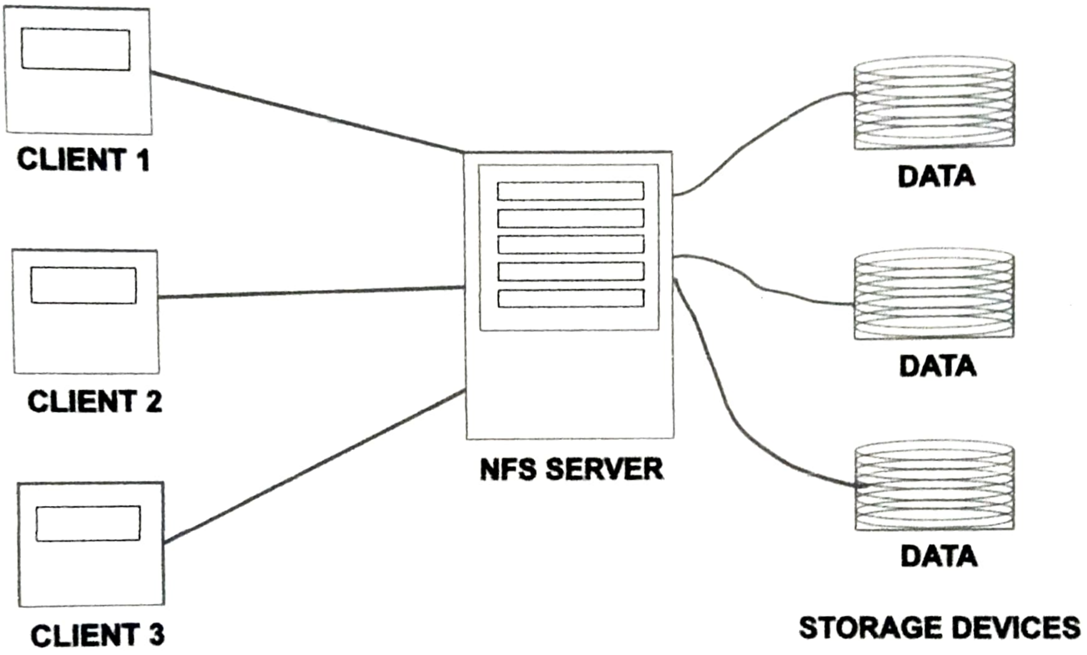
- **NFS:** NFS stands for Network File System. It is a client-server architecture that allows a computer user to view, store, and update files remotely. The protocol of NFS is one of the several distributed file system standards for Network-Attached Storage (NAS).
- **CIFS:** CIFS stands for Common Internet File System. CIFS is an accent of SMB. That is, CIFS is an application of SIMB protocol, designed by Microsoft.
- **SMB:** SMB stands for Server Message Block. It is a protocol for sharing a file and was invented by IMB. The SMB protocol was created to allow computers to perform read and write operations on files to a remote host over a Local Area Network (LAN). The directories present in the remote host can be accessed via SMB and are called as "shares".

- **Hadoop:** Hadoop is a group of open-source software services. It gives a software framework for distributed storage and operating of big data using the MapReduce programming model. The core of Hadoop contains a storage part, known as Hadoop Distributed File System (HDFS), and an operating part which is a MapReduce programming model.
- **NetWare:** NetWare is an abandon computer network operating system developed by Novell, Inc. It primarily used combined multitasking to run different services on a personal computer, using the IPX network protocol.

Working of DFS

There are two ways in which DFS can be implemented:

- **Standalone DFS namespace:** It allows only for those DFS roots that exist on the local computer and are not using Active Directory. A Standalone DFS can only be acquired on those computers on which it is created. It does not provide any fault liberation and cannot be linked to any other DFS. Standalone DFS roots are rarely come across because of their limited advantage.
- **Domain-based DFS namespace:** It stores the configuration of DFS in Active Directory, creating the DFS namespace root accessible at **\\<domainname>\<dfsroot>** or **\\<FQDN>\<dfsroot>**



Advantages

- DFS allows multiple user to access or store the data.
- It allows the data to be share remotely.
- It improved the availability of file, access time and network efficiency.
- Improved the capacity to change the size of the data and also improves the ability to exchange the data.
- Distributed File System provides transparency of data even if server or disk fails.

Disadvantages

- In Distributed File System nodes and connections needs to be secured therefore we can say that security is at stake.
- There is a possibility of lose of messages and data in the network while movement from one node to another.
- Database connection in case of Distributed File System is complicated.
- Also handling of the database is not easy in Distributed File System as compared to a single user system.
- There are chances that overloading will take place if all nodes tries to send data at once.