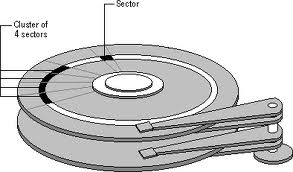
A sector is a physical spot on a formatted disk that holds information. When a disk is formatted, tracks are defined (concentric rings from inside to the outside of the disk platter. Each track is divided into a slice, which is a sector. On hard drives and floppies, each sector can hold 512 bytes of data.  
A block, on the other hand, is a group of sectors that the operating system can address (point to). A block might be one sector, or it might be several sectors (2,4,8, or even 16). The bigger the drive, the more sectors that a block will hold.



So why are there blocks. Why doesn't the operating system just point straight to the sectors? Because there are limits to the number of blocks, or drive addresses, that an operating system can address. By defining a block as several sectors, an OS can work with bigger hard drives without increasing the number of block addresses. For example, PC DOS (earlier versions at least) could only address 65,536 blocks (64K), and each block could could only be a single sector. Thus, the largest size a disk volume could be was 32mb (64K \* 512K). (Earlier versions of the Mac OS had a 16mb volume limit for similar reasons). If you increase the size of a block to, say, 4K, that same version of DOS can now work with volumes as large as 256MB (64K addresses \* 4K blocks).

With current versions of the OS's, the formatting software will look at the size of the drive, and figure out the smallest number of sectors that need to be in a block in order to be able to use the entire drive. So, when you format a floppy disk, the block size will be one sector. When you format a 230MB drive, for example, the block size is 8 sectors (4K).  Why does this matter?

**What is File System?**

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.

## Objective of File management System

Here are the main objectives of the file management system:

* It provides I/O support for a variety of storage device types.
* Minimizes the chances of lost or destroyed data
* Helps OS to standardized I/O interface routines for user processes.
* It provides I/O support for multiple users in a multiuser systems environment.

## Properties of a File System

Here, are important properties of a file system:

* Files are stored on disk or other storage and do not disappear when a user logs off.
* Files have names and are associated with access permission that permits controlled sharing.
* Files could be arranged or more complex structures to reflect the relationship between them.

## File structure

A File Structure needs to be predefined format in such a way that an operating system understands . It has an exclusively defined structure, which is based on its type.

Three types of files structure in OS:

* A text file: It is a series of characters that is organized in lines.
* An object file: It is a series of bytes that is organized into blocks.
* A source file: It is a series of functions and processes.

## File Attributes

A file has a name and data. Moreover, it also stores meta information like file creation date and time, current size, last modified date, etc. All this information is called the attributes of a file system.

Here, are some important File attributes used in OS:

* **Name:** It is the only information stored in a human-readable form.
* **Identifier**: Every file is identified by a unique tag number within a file system known as an identifier.
* **Location:** Points to file location on device.
* **Type:** This attribute is required for systems that support various types of files.
* **Size**. Attribute used to display the current file size.
* **Protection**. This attribute assigns and controls the access rights of reading, writing, and executing the file.
* **Time, date and security:** It is used for protection, security, and also used for monitoring

## File Type

It refers to the ability of the operating system to differentiate various types of files like text files, binary, and source files. However, Operating systems like MS\_DOS and UNIX has the following type of files:

### Character Special File

It is a hardware file that reads or writes data character by character, like mouse, printer, and more.

### Ordinary files

* These types of files stores user information.
* It may be text, executable programs, and databases.
* It allows the user to perform operations like add, delete, and modify.

### Directory Files

* Directory contains files and other related information about those files. Its basically a folder to hold and organize multiple files.

### Special Files

* These files are also called device files. It represents physical devices like printers, disks, networks, flash drive, etc.

## Functions of File

* Create file, find space on disk, and make an entry in the directory.
* Write to file, requires positioning within the file
* Read from file involves positioning within the file
* Delete directory entry, regain disk space.
* Reposition: move read/write position.

## File Access Methods

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

In this type of file access method, records are accessed in a certain pre-defined sequence. In the sequential access method, information stored in the file is also processed one by one. Most compilers access files using this access method.

### Random Access

The random access method is also called direct random access. This method allow accessing the record directly. Each record has its own address on which can be directly accessed for reading and writing.

### Sequential Access

This type of accessing method is based on simple sequential access. In this access method, an index is built for every file, with a direct pointer to different memory blocks. In this method, the Index is searched sequentially, and its pointer can access the file directly. Multiple levels of indexing can be used to offer greater efficiency in access. It also reduces the time needed to access a single record.

## Space Allocation

In the Operating system, files are always allocated disk spaces.

Three types of space allocation methods are:

* Linked Allocation
* Indexed Allocation
* Contiguous Allocation

### Contiguous Allocation

In this method,

* Every file users a contiguous address space on memory.
* Here, the OS assigns disk address is in linear order.
* In the contiguous allocation method, external fragmentation is the biggest issue.

### Linked Allocation

In this method,

* Every file includes a list of links.
* The directory contains a link or pointer in the first block of a file.
* With this method, there is no external fragmentation
* This File allocation method is used for sequential access files.
* This method is not ideal for a direct access file.

### Indexed Allocation

In this method,

* Directory comprises the addresses of index blocks of the specific files.
* An index block is created, having all the pointers for specific files.
* All files should have individual index blocks to store the addresses for disk space.

## File types- name, extension

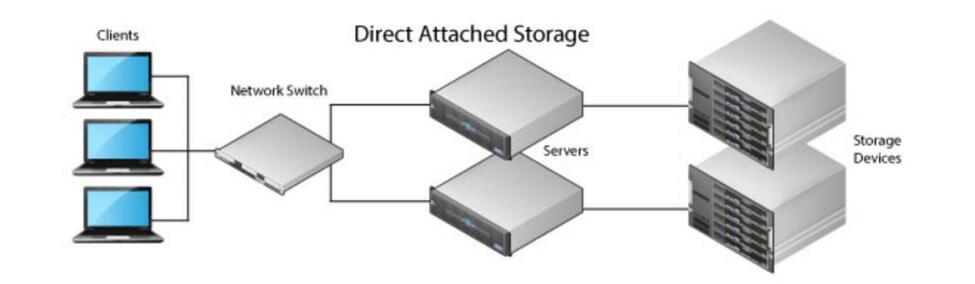
|  |  |  |
| --- | --- | --- |
| **File Type** | **Usual extension** | **Function** |
| Executable | exe, com, bin or none | ready-to-run machine- language program |
| Object | obj, o | complied, machine language, not linked |
| Source code | c. p, pas, 177, asm, a | source code in various languages |
| Batch | bat, sh | Series of commands to be executed |
| Text | txt, doc | textual data documents |
| Word processor | doc,docs, tex, rrf, etc. | various word-processor formats |
| Library | lib, h | libraries of routines |
| Archive | arc, zip, tar | related files grouped into one file, sometimes compressed. |

**DAS**

DAS is the storage in the personal [computer](https://www.router-switch.com/computers-office-price.html?utm_source=blog&utm_medium=blog&utm_campaign=nas_san_das), and it is a part of the [server](https://www.router-switch.com/servers-price.html?utm_source=blog&utm_medium=blog&utm_campaign=nas_san_das). The external storage device is directly connected to the application server through the SCSI or FC interface.

In fact, this DAS storage model is generally widely used in small and medium-sized enterprises. It relies more on the host’s operating system to implement data IO reading and writing, data management, and data backup.

But this storage mode also has certain shortcomings. For example, problems such as poor manageability, weak flexibility in expansion, and difficulty in keeping up with IT development trends. For example, a server/computer will only be equipped with fixed-capacity DAS storage. If the capacity is not enough and the storage space is too small, it is difficult to implement elastic expansion from the inside (external expansion capacity also has drawbacks), if the storage space Too big can also lead to waste of resources, which is especially obvious in the [server](https://www.router-switch.com/servers-price.html?utm_source=blog&utm_medium=blog&utm_campaign=nas_san_das) field.



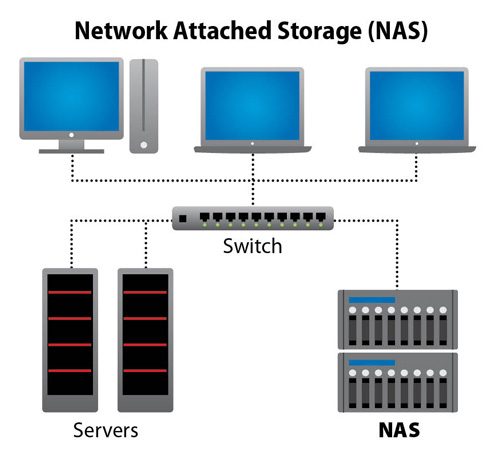
**NAS**

NAS is the kind invitation of many people. It has its own operating system and its own storage space. It contains many “master” secrets.

NAS can be simply understood as a convenient local area network storage device, a device that achieves storage purposes through the network.

NAS (Network Attached Storage) is a standard file-level storage method. It uses network technologies (TCP/IP, ATM, FDDI) to connect the storage system and the server host through a network switch to establish a storage private network. Its main feature is to integrate storage devices, network interfaces and Ethernet technology, and access data directly through the Ethernet network. It can quickly achieve department-level storage capacity requirements and file transfer requirements.

Compared with the above two, NAS network storage has more independence and good compatibility. It not only has its own operating system, but also can be used in a mixed Unix/Windows NT LAN without modification. It is compatible with various operating systems and has good flexibility.



**SAN**

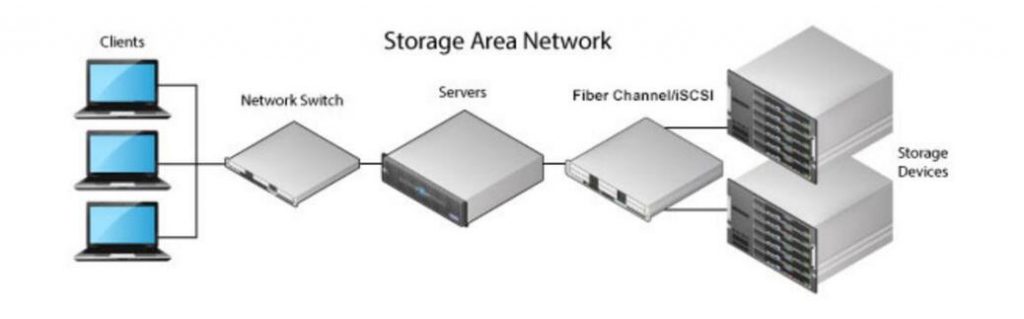
SAN (Storage Area Network) is a mesh channel (FC for short) technology that connects storage arrays and server hosts through FC [switches](https://www.router-switch.com/switches-price.html?utm_source=blog&utm_medium=blog&utm_campaign=nas_san_das) to establish an area network dedicated t data storage.

We can see from the definition that this is a storage method dedicated to enterprise-level applications. It can be simply understood as a high-level network that provides data transmission between the computer and the storage system.

In theory, SAN supports hundreds of disks, provides massive storage space, and solves the problem of large-capacity storage; at the logical level, this massive space can be divided into LUNs of different sizes as needed, and then allocated to servers. Solve the problem that only small capacity storage is needed.

It can be said that the emergence of SAN has adapted to the general trend of information development. It separates computing from storage and enhances the flexibility of storage expansion. After all, nowadays there are more and more networked devices, the amount of data is getting bigger and bigger, and our storage needs are getting stronger and stronger. The SAN’s “collaboration” feature makes storage more flexible and more convenient for capacity expansion.

At the same time, due to the unique storage structure, SAN needs to connect storage arrays and servers through fiber optic [switches](https://www.router-switch.com/switches-price.html?utm_source=blog&utm_medium=blog&utm_campaign=nas_san_das) to establish a dedicated data storage network.



Split brain syndrome, in a clustering context, is a state in which a [cluster](https://whatis.techtarget.com/definition/cluster) of [nodes](https://techtarget.com/searchnetworking/definition/node) gets divided (or partitioned) into smaller clusters of equal numbers of nodes, each of which believes it is the only active cluster.

Believing the other clusters are dead, each cluster may simultaneously access the same application data or disks, which can lead to [data corruption](https://searchsqlserver.techtarget.com/definition/data-corruption). A split brain situation is created during cluster reformation. When one or more node fails in a cluster, the cluster reforms itself with the available nodes. During this reformation, instead of forming a single cluster, multiple fragments of  the cluster with an equal number of nodes may be formed. Each cluster fragment assumes that it is the only active cluster -- and that other clusters are dead -- and starts accessing the data or disk. Since more than one cluster is accessing the disk, the data gets corrupted.

Here's how it works in more detail:

* Let's say there are 5 nodes A,B,C,D and E which form a cluster, X.
* Now a node (say E) fails.
* Cluster reformation takes place. Actually, the remaining nodes A,B,C and D should form cluster X.
* But split brain situation may occur which leads to formation of two clusters X1 (containing A and B) and X2 (containing C and D).
* Both X1 and X2 clusters think that they are the only active cluster. Both clusters start accessing the data or disk, leading to data corruption.