

**File based structure**

The systems that are used to organize and maintain data files are known as file-based data systems. These file systems are used to handle single or multiple files and are not very efficient.

**Functionalities**

* The functionalities of a file-based data management system are as follows:
* Two files can’t have the same name but can have the same name if the extension is different.
* A file-based system helps in basic data management for any user
* The data stored in the file-based system should remain consistent. Any transactions done in the file-based system should not alter the consistency property.
* The file-based system should not allow any illegal or potentially hazardous operations to occur on the data.
* The file-based system should allow concurrent access by different processes, and this should be carefully coordinated.
* The file-based system should make sure that the data is uniformly structured and stored so it is easier to access it.

Advantages:

* The file-based system is not complicated and is simpler to use. Because of this the system is quite inexpensive.
* Because the file-based system is simple and cheap, it is normally suitable for home users and owners of small businesses.
* Since the file-based system is used by smaller organizations or individual users, it stores comparatively lesser amount of data. Hence, the data can be accessed faster and more easily.

Disadvantages:

* The file-based system is limited to a smaller size and cannot store large amounts of data.
* This system is relatively uncomplicated, but this means it cannot support complicated queries, data recovery etc.

**Directory Based structure**

A directory is a container that is used to contain folders and files. It organizes files and folders in a hierarchal manner.

Single-level directory:

This directory is the simplest directory structure. Inside of it, all files are contained in the same directory which makes it easy to support and understand. It has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all he files are in the same directory, they must have unique names. If two users call their dataset test, then the unique name rule violated.

Pros:

* Since it is a single directory, its implementation is very easy.
* If the files are smaller in size, searching will become faster
* The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

Cons:

* Two files can’t have the same name
* Searching will take time if the directory is large
* This cannot group the same type of files together.

Two-level directory:

In this directory, each user has their own user files directory (UFD). The UFDs have similar structures, but each list only the files of a single user. System's master file directory (MFD) is searches whenever a new user id=s logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.

Pros:

* We can give full path.
* Different users can have the same directory as well as the file name.
* Searching files becomes easier due to pathnames and user-grouping.

Cons:

* A user is not allowed to share files with other users.
* Still, it is not very scalable, two files of the same type cannot be grouped together in the same user.

**Metadata**

Metadata is data that is hidden from the user. It is stored within a document that is not evident by just looking at the file. It is an electronic ‘fingerprint’ that automatically adds identifying characteristics, such as the creator or author of the file, the name of individual who have accessed or edited the file, the location from which the file was accessed, and the amount of time spent editing the file. In addition to data automatically added to a document, there is user-introduced metadata, such as tracked changes, versions, hidden text and embedded objects.

Metadata in a call:

* Both numbers of both people in a call
* Data of who is receiver and the caller
* Who is the service provider
* Data of where the call is connected
* IMA number
* Duration of call
* Geographic location
* Whether you move during the call
* Which signal power is going through
* Device type
* OS
* Did the call connect
* Time of call

Every time you create, open or save a Microsoft Word document, hidden information is created and stored within the document that you may not want others outside of your organization to discover. Hidden information can also reside in other Microsoft application files such as Excel spreadsheets or PowerPoint presentations and includes:

Your name and initials, company name, computer name, location of document on local or network server, attached template, hidden text, comments, track changes, non-visible portions of OLE objects, file properties and summary information, and more...

**Data at rest**

Data at rest is one of the three states of digital data and it refers to any digital information that is stationary and contained within permanent storage devices, such as hard drives and tapes, or information reservoirs such as off-site backups, databases, archives, etc. The other states of digital data are data in motion, and data in use. Once data is transported and settles in its final destination, it is called data at rest during the entire period it remains inactive. If the data needs to be utilized for whatever purpose, and is being processed, it is then classified as data in use.

By leveraging various exploit tactics, attackers can eventually trace their way to classified data. The integrity of data at rest will be threatened if the systems which have direct access to the data at rest are vulnerable, the networks in which the data is stored become contained or if the control over the various. devices containing the data are hijacked.

Data in motion is a term used to label any digital information that is being transferred from one location to another. It is also commonly referred to as data in transit or data in flight. When the data is finally contained in one location, it becomes data at rest.

**Types of software testing**

Unit Testing – Passing each module individually so that each achieves their purpose.

Integration Testing – if they integrate with each other and to expose faults when combining them.

System Testing – Evaluate the system’s compliance with the specified requirement.

Acceptance Testing – Passes any business requirements and if it’s acceptable for delivery.

Manual testing: Manual testing includes testing a software manually, I.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior bug.

Automated Testing: Automated testing is the execution of your test plan by a script instead of a human. It is used to re-run the manually performed test scenarios quickly and repeatedly.

**Disaster Recovery**

**Hot site:**

Hot sites are essentially mirrors of your datacenter infrastructure. The backup site is populated with servers, cooling, power, and office space (if applicable). The most important feature offered by a hot site is that the production environment(s) are running concurrently with your main datacenter. This syncing allows for minimal impact and downtime to business operations. In the event of a significant outage event to your main datacenter, the hot site can take the place of the impacted site immediately. However, this level of redundancy does not come cheap, and businesses will have to weigh the cost-benefit-analysis (CBA) of hot site utilization.

**Cold site:**

A cold site is essentially office or datacenter space without any server-related equipment installed. The cold site provides power, cooling, and/or office space which waits in the event of a significant outage to the main work site or datacenter. The cold site will require extensive support from engineering and IT personnel to get all necessary servers and equipment migrated and functional. Cold sites are the cheapest cost-recovery option for businesses to utilize.

**Warm site:**

A warm site is the middle ground of the two disaster recovery options. Warm sites offer office space/datacenter space and will have some pre-installed server hardware. The difference between a hot site and a warm site is that while the hot site provides a mirror of the production datacenter and its environment(s), a warm site will contain only servers ready for the installation of production environments. Warm sites make sense for aspect of the business, which is not critical, but requires a level of redundancy (ex. administrative roles). A CBA conducted on whether to use a warm site versus a hot site should include the downtime associated with the software-loading/configuration requirements for engineering.

**Why use sites?**

Unplanned outages can severely risk a business’ ability to generate revenue, and service clients. A disaster recovery site can help mitigate the impact of those outages on production systems. Business owners need only add this detail to their disaster recovery plans to ensure collective peace of mind in the event of an emergency.

The hot site will allow the money index to continue trading in a very short space of time, which will protect customer investments against large losses.

Money index can configure the hot site remotely, which allows them to respond immediately to the market trends.