File System

The file system is important and highly visible.

It is more than just the way of storing data and programs, persistently.

It also provides organization for the files through a directory structure and maintains metadata related to files.

But what is a file? The snarky UNIX answer is, “Everything is a file!”

As far as the computer is concerned, any data is just 1s and 0s (bytes).

The file is just a logical unit to organize these.

So an area of disk is designated as belonging to a file.

It makes some sense to consider a file to be a structure.

A file has some data (fields, metadata).

There are defined functions, operations to allow users to work with and on files.

Repositioning is also sometimes called a seek operation.

In C this is done with the `fseek()` call.

This adjusts the pointer for reading or writing.

This should be done with caution though, because you can go to an arbitrary location, even the middle of a two (or more) byte character.

And we can't seek when a file is opened for append.

These six operations can be combined for most of the other things we may want to do.

To copy a file, for example, create a new file, read from the old file, and write it into the new file.

We may also have operations to allow a user to access or set various attributes such as the owner, security descriptors, size on disk...

Restrictions on the Operations

Aside from creation and deletion, all operations are restricted to open files.

When a file is opened, a program gets a reference to it, and the operating system keeps track of which files are currently open in which process.

It is good behaviour for a process to close a file when it is no longer using it.

When the process terminates, that will automatically close any open files (hopefully).