Report on Filetype identification

By

Dhwaj verma (ihc2016003)

Brijesh kr. Yadav (icl2016004)

Introduction

A computer file is a computer resource for recording data discretely in a computer storage device. Just as words can be written to paper, so can information be written to a computer file? There are different types of computer files, designed for different purposes. A file may be designed to store a picture, a written message, a video, a computer program, or a wide variety of other kinds of data. Some types of files can store several types of information at once.

Identifying and organizing the files

In modern computer systems, files are typically accessed using names (filenames). In some operating systems, the name is associated with the file itself. In others, the file is anonymous, and is pointed to by links that have names. In the latter case, a user can identify the name of the link with the file itself, but this is a false analogue, especially where there exists more than one link to the same file.

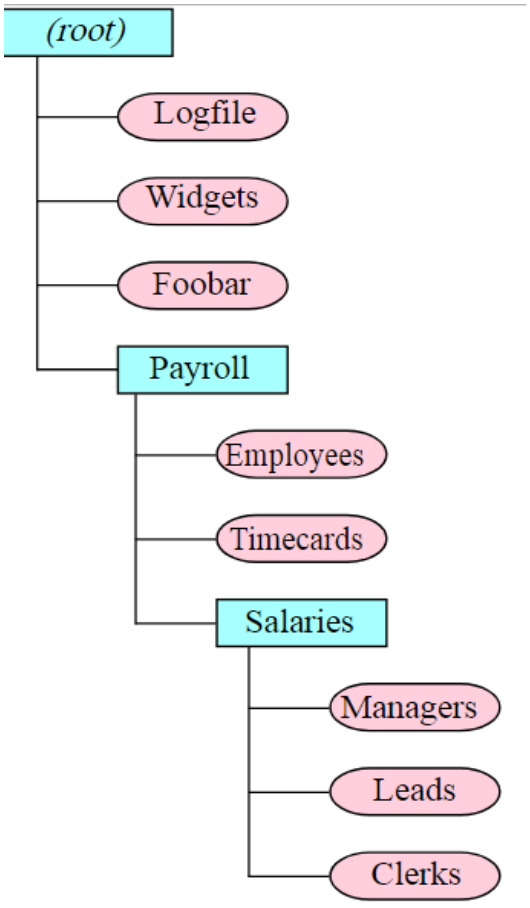
Files (or links to files) can be located in directories. However, more generally, a directory can contain either a list of files or a list of links to files. Within this definition, it is of paramount importance that the term "file" includes directories. This permits the existence of directory hierarchies, i.e., directories containing sub-directories. A name that refers to a file within a directory must be typically unique. In other words, there must be no identical names within a directory. However, in some operating systems, a name may include a specification of type that means a directory can contain an identical name for more than one type of object such as a directory and a file.

In environments in which a file is named, a file's name and the path to the file's directory must uniquely identify it among all other files in the computer system—no two files can have the same name and path. Where a file is anonymous, named references to it will exist within a namespace. In most cases, any name within the namespace will refer to exactly zero or one file. However, any file may be represented within any namespace by zero, one or more names.

Any string of characters may or may not be a well-formed name for a file or a link depending upon the context of application. Whether or not a name is well-formed depends on the type of computer system being used. Early computers permitted only a few letters or digits in the name of a file, but modern computers allow long names (some up to 255 characters) containing almost any combination of unicode letters or unicode digits, making it easier to understand the purpose of a file at a glance. Some computer systems allow file names to contain spaces; others do not. Case-sensitivity of file names is determined by the file system. Unix file systems are usually case sensitive and allow user-level applications to create files whose names differ only in the case of characters. Microsoft Windows supports multiple file systems, each with different policies[which?] regarding case-sensitivity. The common FAT file system can have multiple

files whose names differ only in case if the user uses a disk editor to edit the file names in the directory entries. User applications, however, will usually not allow the user to create multiple files with the same name but differing in case.

Most computers organize files into hierarchies using folders, directories, or catalogs. The concept is the same irrespective of the terminology used. Each folder can contain an arbitrary number of files, and it can also contain other folders. These other folders are referred to as subfolders. Subfolders can contain still more files and folders and so on, thus building a tree-like structure in which one "master folder" (or "root folder" — the name varies from one operating system to another) can contain any number of levels of other folders and files. Folders can be named just as files can (except for the root folder, which often does not have a name). The use of folders makes it easier to organize files in a logical way.



When a computer allows the use of folders, each file and folder has not only a name of its own, but also a path, which identifies the folder or folders in which a file or folder resides. In the path,

some sort of special character—such as a slash—is used to separate the file and folder names. For example, in the illustration shown in this article, the path /Payroll/Salaries/Managers uniquely identifies a file called Managers in a folder called Salaries, which in turn is contained in a folder called Payroll. The folder and file names are separated by slashes in this example; the topmost or root folder has no name, and so the path begins with a slash (if the root folder had a name, it would precede this first slash).

Many (but not all) computer systems use extensions in file names to help identify what they contain, also known as the file type. On Windows computers, extensions consist of a dot (period) at the end of a file name, followed by a few letters to identify the type of file. An extension of .txt identifies a text file; a .doc extension identifies any type of document or documentation, commonly in the Microsoft Word file format; and so on. Even when extensions are used in a computer system, the degree to which the computer system recognizes and heeds them can vary; in some systems, they are required, while in other systems, they are completely ignored if they are presented.

File systems and file managers

The way a computer organizes, names, stores and manipulates files is globally referred to as its file system. Most computers have at least one file system. Some computers allow the use of several different file systems. For instance, on newer MS Windows computers, the older FAT-type file systems of MS-DOS and old versions of Windows are supported, in addition to the NTFS file system that is the normal file system for recent versions of Windows. Each system has its own advantages and disadvantages. Standard FAT allows only eight-character file names (plus a three-character extension) with no spaces, for example, whereas NTFS allows much longer names that can contain spaces. You can call a file "Payroll records" in NTFS, but in FAT you would be restricted to something like payroll.dat (unless you were using VFAT, a FAT extension allowing long file names).

File manager programs are utility programs that allow users to manipulate files directly. They allow you to move, create, delete and rename files and folders, although they do not actually allow you to read the contents of a file or store information in it. Every computer system provides at least one file-manager program for its native file system. For example, File Explorer (formerly Windows Explorer) is commonly used in Microsoft Windows operating systems, and Nautilus is common under several distributions of Linux.

List of different data source where are different file type store

1. **/Home** A user's home directory is intended to contain that user's files; including text documents, music, pictures or videos, etc. It may also include their configuration files of preferred settings for any software they have used there and might have tailored to their liking: web browser bookmarks, favorite desktop wallpaper and themes, passwords to any external services accessed via a given software, etc. The user can install executable software in this directory, but it will only be available to users with permission to this directory. The home directory can be organized further with the use of sub-directories.

   The content of a user's home directory is protected by file system permissions, and by default is accessible to all authenticated users and administrators. Any other user that has been granted administrator privileges has authority to access any protected location on the filesystem including other users home directories.

2. **/Bin** /bin is a standard subdirectory of the root directory in Unix-like operating systems that contains the executable (i.e., ready to run) programs that must be available in order to attain minimal functionality for the purposes of booting (i.e., starting) and repairing a system.
3. /**Opt** This directory is reserved for all the software and add-on packages that are not part of the default installation. For example, StarOffice, Kylix, Netscape Communicator and WordPerfect packages are normally found here. To comply with the FSSTND, all third party applications should be installed in this directory. Any package to be installed here must locate its static files (ie. extra fonts, clipart, database files) must locate its static files in a separate /opt/'package' or /opt/'provider' directory tree (similar to the way in which Windows will install new software to its own directory tree C:\Windows\Progam Files\"Program Name"), where 'package' is a name that describes the software package and 'provider' is the provider's LANANA registered name.
4. **/proc** /proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo-file system. It doesn't contain 'real' files but runtime system information (e.g. system memory, devices mounted, hardware configuration, etc). For this reason it can be regarded as a control and information centre for the kernel. In fact, quite a lot of system utilities are simply calls to files in this directory. For example, 'lsmod' is the same as 'cat /proc/modules' while 'lspci' is a synonym for 'cat /proc/pci'. By altering files located in this directory you can even read/change kernel parameters (sysctl) while the system is running.
5. **/sbin** Linux discriminates between 'normal' executables and those used for system maintenance and/or administrative tasks. The latter reside either here or - the less

important ones - in /usr/sbin. Locally installed system administration programs should be placed into /usr/local/sbin.

6. **/usr** /usr usually contains by far the largest share of data on a system. Hence, this is one of the most important directories in the system as it contains all the user binaries, their documentation, libraries, header files, etc.... X and its supporting libraries can be found here. User programs like telnet, ftp, etc.... are also placed here. In the original Unix implementations, /usr was where the home directories of the users were placed (that is to say, /usr/someone was then the directory now known as /home/someone). In current Unices, /usr is where user-land programs and data (as opposed to 'system land' programs and data) are. The name hasn't changed, but it's meaning has narrowed and lengthened from "everything user related" to "user usable programs and data". As such, some people may now refer to this directory as meaning 'User System Resources' and not 'user' as was originally intended.
7. **/var** Contains variable data like system logging files, mail and printer spool directories, and transient and temporary files. Some portions of /var are not shareable between different systems. For instance, /var/log, /var/lock, and /var/run. Other portions may be shared, notably /var/mail, /var/cache/man, /var/cache/fonts, and /var/spool/news. Why not put it into /usr? Because there might be circumstances when you may want to mount /usr as read-only, e.g. if it is on a CD or on another computer. '/var' contains variable data, i.e. files and directories the system must be able to write to during operation, whereas /usr should only contain static data. Some directories can be put onto separate partitions or systems, e.g. for easier backups, due to network topology or security concerns. Other directories have to be on the root partition, because they are vital for the boot process. 'Mountable' directories are: '/home', '/mnt', '/tmp', '/usr' and '/var'. Essential for booting are: '/bin', '/boot', '/dev', '/etc', '/lib', '/proc' and '/sbin'.
8. **Download** Every file that download form internet by default store in download folder. we can change the path of download files using Browsers setting.
9. **Desktop** at desktop we store icon of some important application and files that frequently access.