# WEEK-03

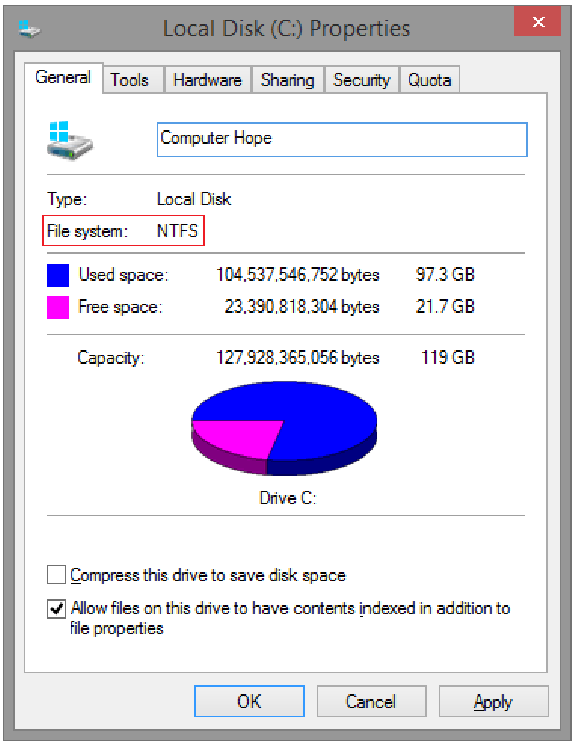
## Contents:

File system:

- Pathnames
- File system structure and its description
- Navigating the file system
- File types, attributes
- Access Control List
- Adding Text to File
- Pipes
- File Comparison
- Filters/Text Processing Commands.

INTRODUCTION

- ➢ A file system is a method of organizing and retrieving files from a storage medium (e.g., hard drive).
- ➢ File systems usually consist of files separated into groups called directories. Directories can contain files or additional directories.
- ➢ The most commonly used file system with Windows is NTFS.
- ➢ Files are most often managed in a hierarchy, which allows to view files in the current directory and then navigate into any subdirectories.



Local Disk (C:) Properties

Computer Hope

Type: Local Disk

File system: NTFS

🟦 Used space: 104,537,546,752 bytes 97.3 GB

🟪 Free space: 23,390,818,304 bytes 21.7 GB

Capacity: 127,928,365,056 bytes 119 GB

Drive C:

□ Compress this drive to save disk space

☑ Allow files on this drive to have contents indexed in addition to file properties

Examples of file systems:

1. exFAT (Extensible File Allocation Table)
2. FAT (e.g., FAT16 and FAT32) (File Allocation Table)
3. GFS (Global File System)
4. HFS (Hierarchical File System)
5. NTFS (New Technology File System)
6. UDF (Universal Disk Format)

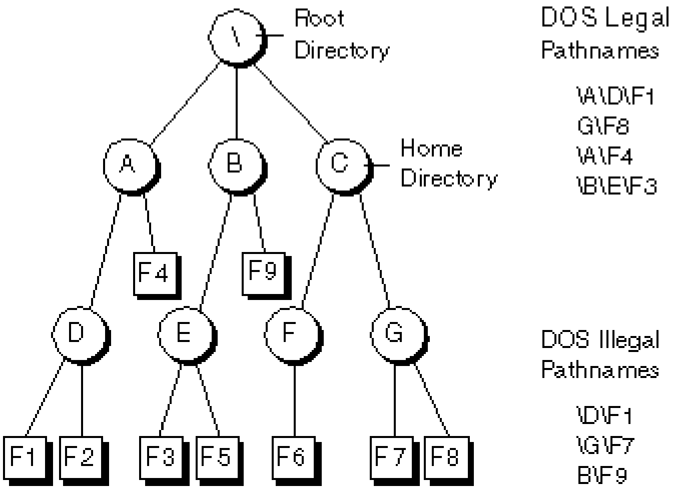
File is a container for storing information. UNIX treats directories and devices as files.

Generally, files are classified into three categories:

1. Ordinary(regular) Files: Contains only data as a stream of characters.
2. Directory Files: It contains names of files and directories and a number associated with each name.
3. Device Files: All devices and peripherals are represented as files.

PATH NAMES:

- A sequence of symbols and names that identifies a file.
- Every file has a name, called a *filename*, so the simplest type of pathname is just a filename.
- If we specify a filename as the pathname, the operating system looks for that file in our current working directory.
- However, if the file resides in a different directory, we must tell the operating system how to find that directory.
- We need do this by specifying a path that the operating system must follow.



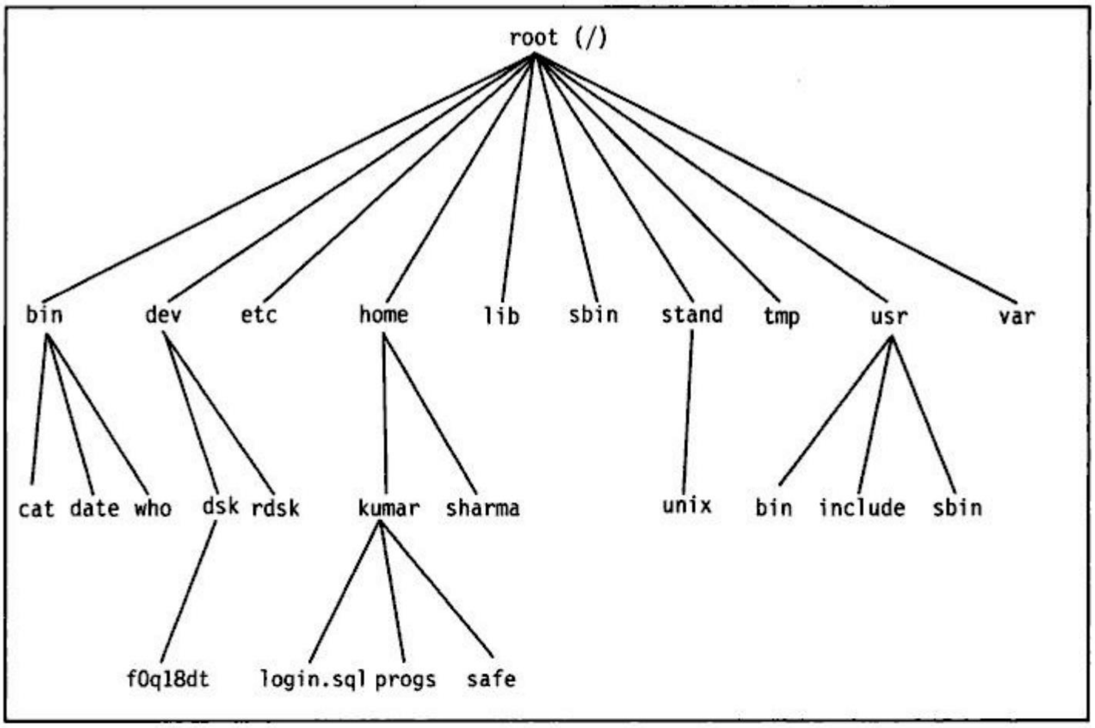
- ✓ Pathnames are classified into Absolute and Relative Pathnames.
- ✓ Absolute Path: Where the first character of the pathname is '/'
  Eg: cat /home/kumar/login.sql

- ✓ Relative Path: Where the first character of the pathname is not '/'
  Eg: cd progs
  cat login.sql
  cd progs/scripts
- ✓ A relative pathname uses following cryptic symbols:
  a. . (a single dot) – This represents the current directory.
  b. .. (two dots) – This represents the parent directory.

  Eg: cp ../sharma/profile . – This copies the file profile to the current directory.

FILE SYSTEM:

- It is a collection of all related files (ordinary, directory and device files) organized in a hierarchical structure.
- This system is also adopted by DOS and Windows.



UNIX File System Tree

- **/:** The topmost directory is called root directory represented as '/', which has many subdirectories or files under it.
- **/bin:** Stands for "binaries" and contains certain fundamental utilities, such as ls or cp, which are generally needed by all users.
- **/boot:** Contains all the files that are required for successful booting process.
- **/dev:** Stands for "devices". Contains file representations of peripheral devices and pseudo-devices.
- **/etc:** Contains system-wide configuration files and system databases.
- **/home:** Contains the home directories for the users.

- **/lib:** Contains system libraries, and some critical files such as kernel modules or device drivers.
- **/media:** Default mount point for removable devices, such as USB sticks, media players, etc.
- **/mnt:** Stands for “mount”. Contains filesystem mount points. These are used, for example, if the system uses multiple hard disks or hard disk partitions. It is also often used for remote (network) filesystems, CD-ROM/DVD drives, etc.
- **/proc:** procfs virtual filesystem showing information about processes as files.
- **/root:** The home directory for the superuser “root” – that is, the system administrator.
- **/tmp:** A place for temporary files. Many systems clear this directory upon startup.
- **/usr:** Originally the directory holding user home directories. It also holds executables, libraries, and shared resources that are not system critical, like the X Window System, KDE, Perl, etc.
- **/usr/bin:** This directory stores all binary programs distributed with the operating system not residing in /bin, /sbin or (rarely) /etc.
- **/usr/include:** Stores the development headers used throughout the system. Header files are mostly used by the #include directive in C/C++ programming language.
- **/usr/lib:** Stores the required libraries and data files for programs stored within /usr or elsewhere.
- **/var:** A short for “variable.” A place for files that may change often – especially in size, for example e-mail sent to users on the system, or process-ID lock files.
- **/var/log:** Contains system log files.
- **/var/mail:** The place where all the incoming mails are stored. Often, this directory is a symbolic link to /var/spool/mail.
- **/var/spool:** Spool directory. Contains print jobs, mail spools and other queued tasks.
- **/var/tmp:** A place for temporary files which should be preserved between system reboots.

NAVIGATION:

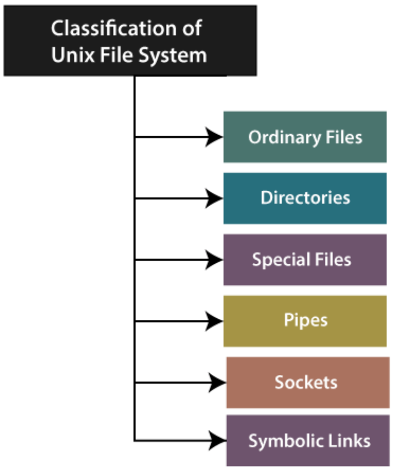
- Moving from one location to another in a file system is called a navigation.
- Navigation in Unix is the same as pointing and clicking in a typical graphical user interface.

  Eg: If you have the folder “ExperimentFolder” on my Desktop, you can point and double-click to open it.

  You can do the same thing by typing cd ~/Desktop/ExperimentFolder in the Terminal and then typing ls to see what’s in the directory.

FILE TYPES:

- ✓ In Unix Operating System, all data is organized into files, each file is organized into directories, and these directories are organized in a tree like structure called a File System.
- ✓ Following are the types of files in UNIX



### 1. Ordinary Files

- ✓ Ordinary files are the files in Unix, which includes program instructions, text, and data.
- ✓ In ordinary files, there are no other files.
- ✓ Ordinary files are always placed under the directory file.
- ✓ In the long-format output of ls -l, the "-" symbol is used to specified such kind of file.

### 2. Directory

- ✓ Directory store both special as well as ordinary files.
- ✓ Unix directories are equivalent to the folders of Mac or Windows.
- ✓ A directory file includes an entry file for each file and a subdirectory which it houses.
- ✓ If there are 5 files in a directory, then there will be 5 entries in the directory. Every entry comprises of 2 components.
  1. The name of the File.
  2. An inode number, which is a unique identification number for the file or directory.
- ✓ Each file is descendant of the root directory (named /) located at the tree's top.
- ✓ In the long-format output of ls -l, we used the "d" symbol in order to specify this kind of file.

### 3. Special Files

- ✓ We use special files to represent the real physical device like terminal, tape drive, and printer. and also used for Input/Output(I/O) operations.

- ✓ Special files or devices are used for Input/Output(I/O) on Linux and Unix systems.
- ✓ They look as similar to the ordinary file or directory in a file system.
- ✓ Mainly two kinds of special files for every device in the Unix system are there such as block special files and character special files.
    1. If we use the block special file for the device input/output (I/O), the data is moved to the higher fixed-size blocks. This kind of access is known as block device access.
    2. In the other kind, one character at a time for the terminal devices.
- ✓ In the long-format output of ls -l, using the "c" symbol character distinct, files are marked.
- ✓ In the long-format output of ls -l, using the "b" symbol block, distinct files are marked.

### 4. Pipes

- ✓ In Unix, with the help of pipes, we can link commands together.
- ✓ Pipes is like a temporary file that only exists to hold the data from one command till it is read by another.
- ✓ A one-way flow of data is provided by the Unix pipe.
- ✓ The first command's output sequence is used like an input to the next command sequence.
- ✓ In order to create a pipe, the vertical bar (|) has to be placed on the command line between 2 commands. Such as ***who | wc -l***
- ✓ In the long-format output of ls -l, known as pipes, are marked using the symbol "p."

### 5. Sockets

- ✓ Unix sockets are a type of file that enables for advanced inter-process communication.
- ✓ It is also known as inter-process communication socket.
- ✓ In the framework of the client-server application, a Unix Socket is used.
- ✓ It is essentially a data stream, same as the network stream and network socket, except each transaction is local to the file system.
- ✓ In the long-format output of ls -l, using the "s" symbol, Unix sockets are marked.

### 6. Symbolic Link

- ✓ The symbolic link is used to reference another file in the file system.
- ✓ The symbolic file will not function if the source file is deleted or moved to the different location.
- ✓ In the case of the long-format output of ls -l, using the "l" symbol, the symbolic link is marked.

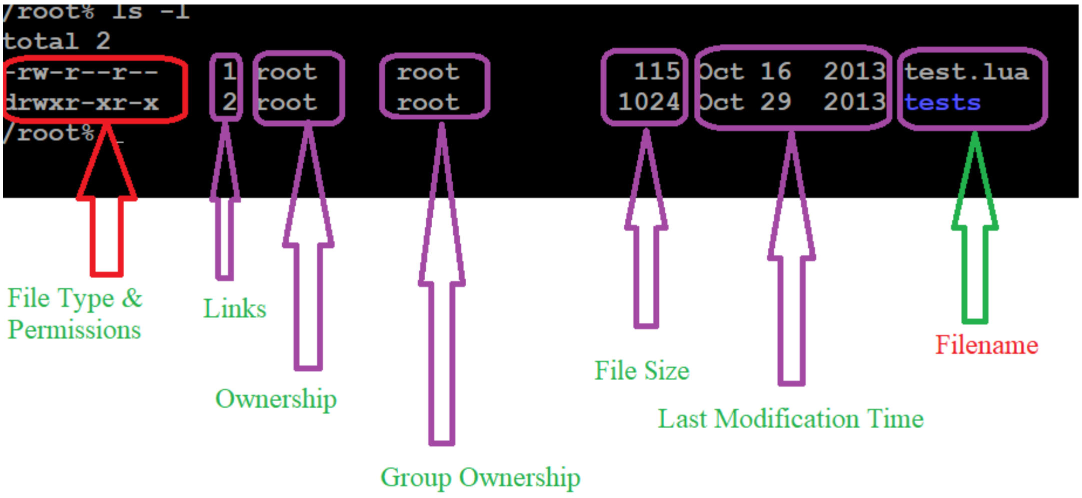
## FILE ATTRIBUTES:

- ✓ In UNIX, using ***ls –l*** command, we can see the attributes of a file and using ***chmod*** command, we can modify its attributes.

- ✓ The ***ls –l*** command lists seven attributes of a file;
  a. File Type and Permissions
  b. Links
  c. File Ownership
  d. Group Ownership
  e. File Size
  f. Last Modification Time and
  g. Filename

- ✓ The Three file permissions are;
  a. read(r)
  b. write(w) and
  c. execute(e)

- ✓ Read – Designated as an "r"; allows a file to be read, but nothing can be written to or changed in the file.
- ✓ Write – Designated as a "w"; allows a file to be written to and changed.
- ✓ Execute - Designated as an "x"; allows a file to be executed by users or the operating system.



```
/root% ls -l
total 2
-rw-r--r--  1 root  root      115 Oct 16  2013 test.lua
drwxr-xr-x  2 root  root     1024 Oct 29  2013 tests
/root% _
```

## ACCESS CONTROL LIST (ACL):

- ✓ Access control list (ACL) provides an additional, more flexible permission mechanism for file systems.
- ✓ It is designed to assist with UNIX file permissions.
- ✓ ACL allows you to give permissions for any user or group to any disc resource.
- ✓ Think of a scenario in which a particular user is not a member of group created by you but still you want to give some read or write access, how you

can do it without making user a member of group, here comes in picture Access Control Lists.

- ✓ **setfacl** and **getfacl** are used for setting up ACL and showing ACL respectively.

  **E.g.:** ***getfacl*** *filename*

  ***Setfacl –m*** *"u:user:permissions" filename*

## ADDING TEXT TO FILE (APPENDING):

- ✓ While working with files in a terminal sometimes we need to append the same data of a command output or file content.
- ✓ Append means simply add the data to the file without erasing existing data.
- ✓ The >> operator redirects output to a file.
- ✓ If the mentioned file does not exist, the file is created and then the text is appended to the file.
- ✓ We can use echo, printf, cat and tee –a commands to append operation.

  **E.g.:** **echo** "Which is the best Linux Distro?" >> file.txt

  **printf** "Which is the best Linux Distro?\n" >> file.txt

  **cat** file1.txt >> file2.txt

  **echo** "Which is the best Linux Distro?" | **tee -a** file.txt

## PIPES:

- ✓ A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing.
- ✓ The Unix/Linux systems allow stdout of a command to be connected to stdin of another command.
- ✓ It can be done by using the pipe character **'|'**.
- ✓ Pipes are unidirectional **i.e data flows from left to right through the pipeline.**

  E.g.: command_1 | command_2 | command_3 | .... | command_N

  ls –l | more

## FILE COMPARISION:

- ✓ The *file comparison command* helps us to *compare* the *files* and find the similarities and *differences* between these *files*.

  **E.g.:** diff, cmp and comm

## FILTERS/ TEXT PROCESSING COMMANDS:

- ✓ Filters are programs that take plain text (either stored in a file or produced by another program) as standard input, transforms it into a meaningful format, and then returns it as standard output. Linux has a number of filters.

  **E.g.:** cat, head, tail, sort, uniq, wc, grep, tac, sed and nl.

***References:***

1. UNIX Concepts and Applications by Sumitabha Das.
2. Various websites.
3. https://www.geeksforgeeks.org/