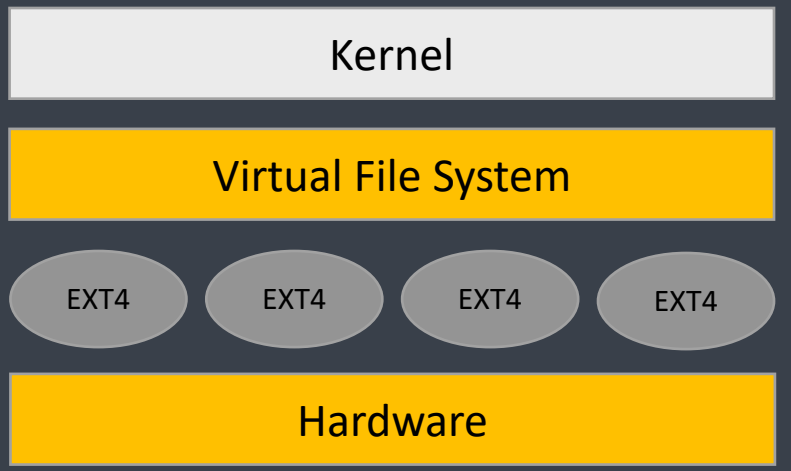
Working With Files

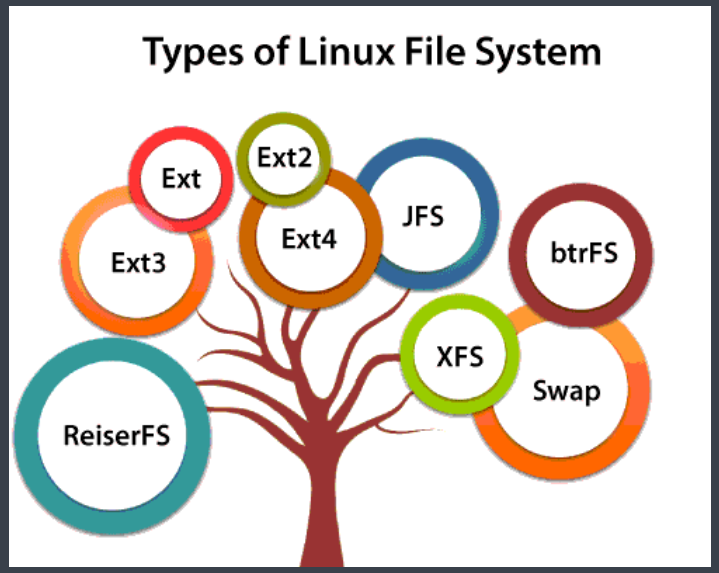
Linux Hierarchical File System

- A Linux file system is a structured collection of files on a disk drive or a partition
- It is built-in layer on Linux Operating System used to handle data management of the storage
- It helps to arrange the file on the disk storage
- Linux file system contains two-part file system software implementation architecture
- Linux Virtual File System
  - It provides a single set of commands for the kernel and developers to access the file system
  - This virtual file system requires the specific system driver to give an interface to the file system



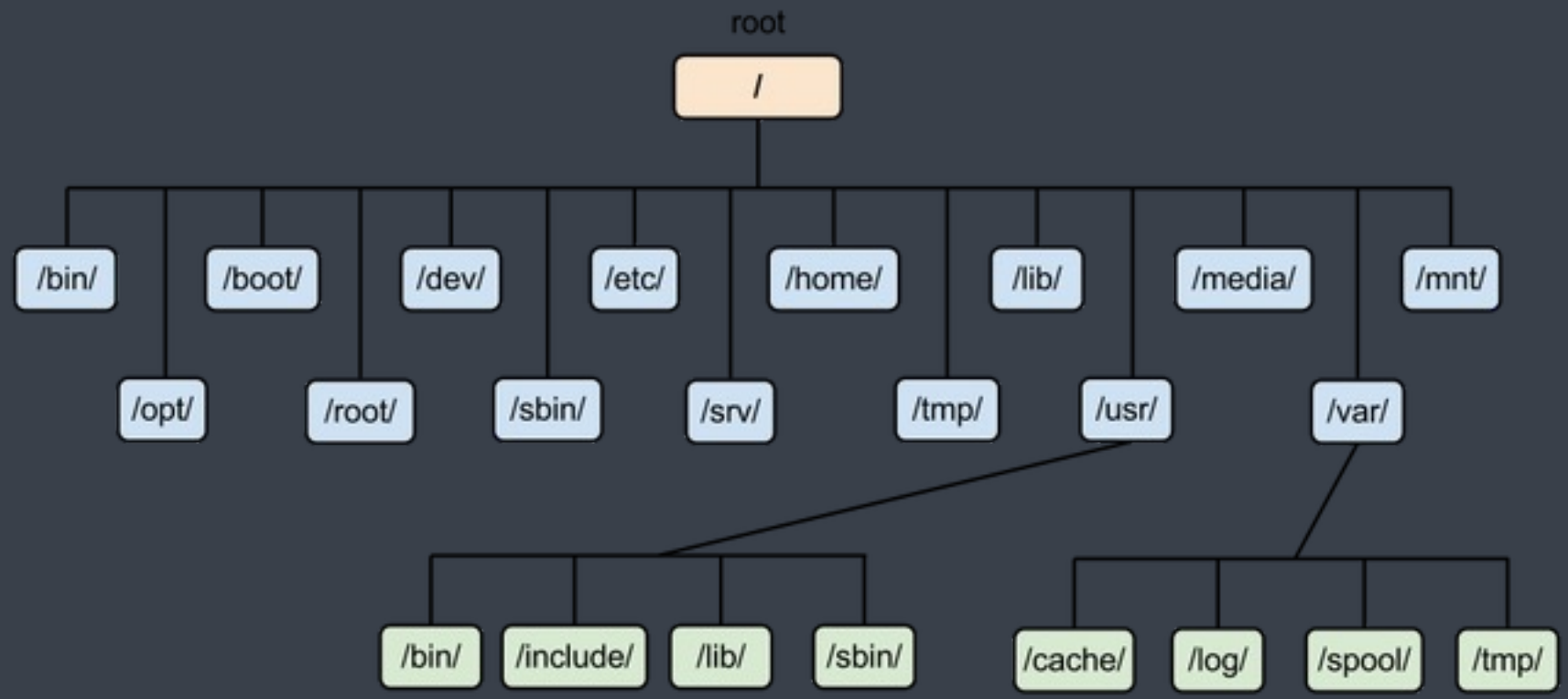
Types of Linux File Systems

- Ext, Ext2, Ext3 and Ext4 file system
  - The file system Ext stands for **Extended File System**
  - It was primarily developed for **MINIX OS**
- JFS File System
  - JFS stands for **Journaled File System**, and it is developed by **IBM for AIX Unix**
  - It can also be used in place of Ext4, where stability is needed with few resources
- ReiserFS File System
  - ReiserFS is an alternative to the Ext3 file system
  - It has improved performance and advanced features
- XFS File System
  - XFS file system was considered as high-speed JFS
  - It is developed for parallel I/O processing
  - NASA still using this file system with its high storage server (300+TB servers)



The Linux File System Hierarchy

- All files on a Linux system are stored on file systems which are organized into a single inverted tree of directories, known as a file system hierarchy
- This tree is inverted because the root of the tree is said to be at the top of the hierarchy, and the branches of directories and subdirectories stretch below the root



File System Hierarchy

- /bin - Stores essential command-line utilities and binaries. For example, the /bin/ls is the binary for the ls command
- /boot - Stores the files necessary to boot the Linux operating system
- /dev - Stores hardware and software device drivers. This directory maintains file system entries that represent the devices connected to the system (for example, the /dev/sda1 partition)
- /etc - Stores basic configuration files. For example, the /etc/ samba/smb.conf file stores Samba configuration data
- /home - Stores users' home directories, including personal files
- /lib - Stores shared program libraries required by the kernel, command-line utilities, and binaries
- /media - Stores mount points for removable media such as CD-ROMs and floppy disks
- /mnt - This is the mount point for temporarily mounting file systems
- /opt - Stores optional files of large software packages. These packages normally create a subdirectory bearing their name under the /opt directory and then place their files in the subdirectory. For example, the /opt/nessus subdirectory contains files for the Nessus vulnerability scanning program
- /proc - This is a virtual file system (VFS) that represents continually updated kernel information to the user in a typical file format (for example, the /proc/mounts file)
- /root - The home directory of the root user

File System Hierarchy

- /sbin - Stores binaries that are used for completing the booting process and also the ones that are used by the root user. For example, the /sbin/ifconfig file is the binary for the ifconfig command that is used to manage network interfaces on the system
- /sys - This is another VFS, and it primarily stores information about devices. For example, /sys/block includes links to devices that are stored in various subdirectories under the /sys/ devices/ location, which presents a hierarchy of devices in the kernel
- /tmp - Stores temporary files that may be lost on system shutdown
- /usr - A read-only directory that stores small programs and files accessible to all users
- The /usr directory contains some important subdirectories
  - /usr/bin - Includes executable programs that can be executed by all users
  - /usr/local - Includes custom build applications that are stored here by default
  - /usr/lib - Includes object libraries and internal binaries that are needed by the executable programs
  - /usr/lib64 - Serves the same purpose as /usr/lib, except it is meant only for 64-bit systems
  - /usr/share - Includes read-only architecture independent files. These files can be shared among different architectures of an operating system
  - /var - Stores variable files, or files that are expected to constantly change as the system runs. Examples include log files, printer spools, and some networking services' configuration files

File Path

- The path of a file or directory specifies its unique file system location
- Following a file path traverses one or more named subdirectories, delimited by a forward slash (/), until the destination is reached
- Standard file behavior definitions apply to directories the same as other file types
- Absolute Path
  - An absolute path is a fully qualified name, specifying the files exact location in the file system hierarchy
  - It begins at the root (/) directory and specifies each subdirectory that must be traversed to reach the file
  - E.g. /etc/passwd
- Relative Path
  - Like an absolute path, a relative path identifies a unique file, specifying only the path necessary to reach the file from the working directory
  - A path name with *anything other than* a forward slash as the first character is a relative path name
  - E.g. A user in /etc directory can refer passwd directly

Key Directories

- /etc
  - The /etc directory contains files and subdirectories that hold configuration information for the system and its services
  - It's reasonable to assume that if you need to adjust firewall rules, manage Secure Shell (SSH) access, or configure software-management settings, you will need to work with files stored in /etc
- /home
  - The home directory contains a user's personal files or files that are otherwise specific to that user
  - The home directory is where you are placed when you log in to the system
  - In Linux, by default, every user except the root user is assigned a subdirectory in /home that corresponds to their user name. A user can create subdirectories and files within this directory
  - The home directory path is set to a variable named $HOME. When the string $HOME is used, it references the path to the current user's home directory
  - This text will often reference the home directory using the $HOME variable
  - In many shells, including KornShell, C shell, and Bash, the tilde character (~) represents your home directory
- /root
  - The home directory of the root user is /root
  - This is not to be confused with the root directory (/), which is the top-most part of the file system hierarchy
- /var/log
  - Linux and installed services typically write to log files stored in the /var/log directory
  - As a sysadmin investigating a system issue or auditing system access, you will likely find the information required in log files stored at /var/log

File System Navigation

- pwd: shows the current working directory user is in
- ls: shows the list of the files and directories of the destination path
  - -l : used to show the files and directories in list format
  - -a : shows all files
  - -h : human readable sizes
  - -r : reverse the order while sorting
  - -R : recursively print all the sub-directories
- cd: used to change the directory
- tree: used to display the filesystem in a more familiar structure, making it easy to understand the directory and its location in the file system hierarchy

File System Globbing

- Globbing is a shell feature that helps matching the filenames
- You should not confuse this with regular expressions (which helps finding text patterns)
- For documentation: see man 7 glob
- E.g.
  - ls host* : find files which start with host followed by any characters
  - ls ?ost : find files which start with any letter followed by ost
  - ls [hm]ost : find files starting with either h or m followed by ost
  - ls [!hm]ost : find files starting with neither h nor m followed by ost
  - ls script[0-9][0-9] : find file starting with script followed by any 2 digit number

File Metadata

- The stat command displays file metadata in a relatively user-friendly structure
- File size, access information, storage data, and more are displayed
- Use the man page to display various options that supplement stat
- File Naming Conventions
  - A file name is a string of characters that identify a file
  - By using the right combination of characters in file names, you can ensure that the files are unique and easy to recognize
  - On an ext4 file system, a file name may be up to 255 bytes long and contain any byte except NULL (\0) and the forward slash (/)
  - File names of user files may not be a single dot . or two dots .. because these are special, reserved file names
  - Various file systems may enforce different requirements for file names
  - Although file names may contain a space, convention on Linux systems dictates that words in a file name are more frequently demarcated by a hyphen or an underscore, as these are easier to manage on the command-line

File System Operations

- touch: creates an empty file
- mkdir: creates a directory
  - mkdir –p: create a path
- cp: copy a file from one location to another
- mv
  - Moves a file from one location to another
  - Can also be used for renaming a file
- rmdir: deletes an empty directory
- rm: remove file

Finding files

- ls command can be used to list the files and not to find the files
- which command looks for binaries in the $PATH
- locate uses a database, built by updatedb to find the files in a database
- find is the most flexible tool that allows you to find files based on many criteria
  - find / -name “host”
  - find / -size +2G
  - find / -type f –size +100M
  - find / -user sunbeam

Links

Understanding Links

- Links are the pointers to files in a different location
- I can be compared to shortcuts in other operating systems
- Links can be useful to make the same file available on multiple locations
- Linux uses hard link and symbolic links

Understanding Hard Links

- Each hard linked file is assigned the same Inode value as the original, therefore they reference the same physical file location
- Hard links more flexible and remain linked even if the original or linked files are moved throughout the file system, although hard links are unable to cross different file systems
- ls -l command shows all the links with the link column shows number of links
- Links have actual file contents
- Removing any link, just reduces the link count, but doesn't affect other links
- We cannot create a hard link for a directory to avoid recursive loops
- If original file is removed then the link will still show the content of the file
- E.g.
  - ln [original filename] [link name]

Understanding Symbolic Link

- A soft link is similar to the file shortcut feature which is used in Windows Operating systems
- Each soft linked file contains a separate Inode value that points to the original file
- As similar to hard links, any changes to the data in either file is reflected in the other
- Soft links can be linked across different file systems, although if the original file is deleted or moved, the soft linked file will not work correctly (called hanging link)
- Soft Link contains the path for original file and not the contents
- Removing soft link doesn't affect anything but removing original file, the link becomes "dangling" link which points to nonexistent file
- A soft link can link to a directory
- E.g.
  - ln -s [original filename] [link name]

Archive and Compression

Working With Compressed Files

- **Compression** is a way of reducing the size of a file on disk using different algorithms and mathematical calculations
- There are two ways of compression
  - Lossy
    - If some information is getting lost while compression, it is known as lossy compression
    - e.g. mp3, JPEG
  - Lossless
    - This method creates a file smaller than the original that can be used to reconstruct the original file
    - This type of compression does not use approximations to compress data, and instead uses certain algorithms to recognize repeated portions of a file
    - It removes these and replaces them with a placeholder
    - It continues on and replaces later occurrences of the pattern with references to the same placeholder

Understanding tar

- Stands for tape archive, which was used to archive the files to a tape drive
- Linux provides tar command to archive and unarchive the files
- Commands
  - tar –cvf <archive file name> <file list>
  - tar –xvf <archive file name>
  - tar –tvf <archive file name>
- Add compression using –z, -j or -J command line options

Compression tools

- Linux comes with different compression tools like
- gzip
  - Considered as classic method of compressing the data on Linux machine
  - It has been around since 1992
  - The gzip tool uses a compression algorithm know as DEFLATE
- bzip2
  - Alternate utility for compression
- xzip
  - Modern compression utility