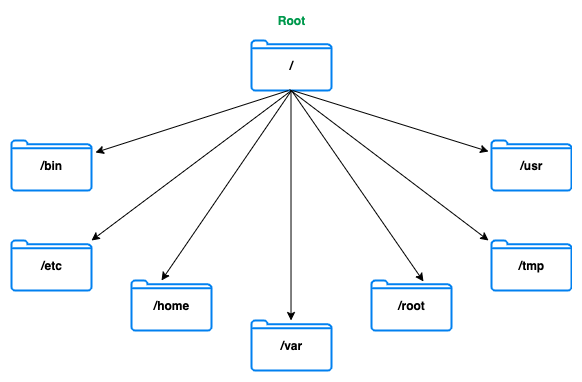
**Basics of Linux Directory Structure**

Linux based operating systems are very different from Windows operating systems. We will start with the difference between the directory structure of these two operating systems. In Windows operating systems, all the files are stored in different data drives like **C: D: E:**.

But in Linux, files are ordered in a tree structure starting with the **root** directory represented by a forward slash (/). The **root** directory **contains all other directories and files** on the system. It is called root because it is at the very top of the directory tree.



Some top level directories that are part of the root directory by default are /bin, /etc, /tmp, /usr, /home, /root and /var. You can read more about these directories [here](https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard). We will briefly discuss some common top level directories that you need to be aware of :

**/etc :** Contains the system configuration files for the host computer. You should be careful when making changes to the files contained in this directory.

**/home :** Usually contains users personal data such as your files, notes and user-specific configuration files.

**/var :** The directory name stands for variable and it is used to store variable files. Files whose content is expected to continually change during normal operation of the system - such as logs and lock files.

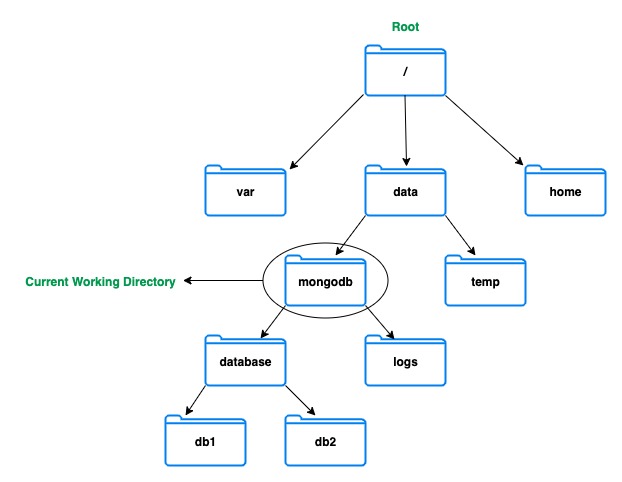
**Important Definitions**

**Current Working Directory :** It is the directory in which the user is currently working in. By default, when you log into your linux system, your current working directory is set to your home directory. In this course, it will be **/root/workspace**. You can check your current working directory by running the **pwd** (Print Working Directory) command. pwd will print out the absolute path to this directory. What is an absolute path?

**Path :** A unique location to a file or a folder in a file system of an OS.

**Absolute Path :** The complete path of a file or directory in the filesystem from the root (/) directory.

**Relative Path :** The path of a file or directory relative to the current working directory.



In the diagram shown above, mongodb is our Current Working Directory.

Let's say that we want to find the path to the db1 directory.

The Absolute Path of the db1 directory - /data/mongodb/database/db1

The Relative Path to the db1 directory from the Current Working Directory of mongodb - database/db1

**Concept of users in linux**

Linux is a multi-user operating system, which means that more than one user can use the Linux machine at the same time. There are three types of users in Linux - **root, regular and service**. In this tutorial we will discuss only two types of user root and regular.

**root user:** This is the main user account in the Linux system. This user is automatically created during the installation and has the highest privileges in the system. The user can perform any administrative task, and is therefore, also known as **superuser** or **administrator**. In this course, you will run commands as the **root** user.

**regular user:** This is the normal user account with moderate privileges. This user does not have all the privileges of root. For instance, the root user can install new softwares, change the ownership of files, and manage other users' accounts.

**What is `sudo` and when to use it ?**

[sudo](https://kb.iu.edu/d/amyi) stands for "superuser do" and is a program that allows you to run programs with the security privileges of another user (by default, as the superuser). It allows you to execute a command with elevated privileges (usually as root). With "sudo", you continue to use your user account, but with root privileges. Sudo creates a thorough audit trail, so everything users do gets logged. You should use it when you are required to perform some administrative task that you cannot perform as a normal user.

**Permissions**

What makes one user different from another user on a Linux machine are the permissions. Permissions are the **"rights"** to act on a file or directory. The basic rights are **read, write and execute**.

* Read permission on a file allows the user to view the content of a file.
* Write permission on a file allows the user to modify the content of that file.
* Execute permission on a file allows the user to run the file and execute a program or script.

**Permissions** can be changed using the [chmod](https://en.wikipedia.org/wiki/Chmod" \t "_blank) command.

**chmod**

chmod command is used to change the permission of a file or directory. Permissions could be **Read, Write or Execute**. In this course you will be using the file permission of 400 for granting read-only access to the user.

COPY

chmod 400 <path-to-the-file>

Using flags such as 400 is an easy and short form to set permissions. If you want to know more what the numbers in chmod mean, please visit [this page](https://www.december.com/unix/ref/chmod.html).

For example, this command will change the permission of *m103-keyfile* file to read only.

COPY

chmod 400 /var/mongodb/pki/m103-keyfile

**Ownership**

Every file is owned by a specific user and a specific group in Linux. In this course the files will be mostly owned by the root user.

Ownership can be changed using the [chown](https://linuxize.com/post/linux-chown-command/" \t "_blank) command.

**chown**

Chown command is used for changing the user/group ownership of a file/directory.

COPY

chown [new owner]:[group] <file name>

If you need to change ownership of a file or directory, you may need to use **sudo** if the directory is owned by root. For example, this command will change the ownership of the /var/mongodb\_directory to **myuser**.

COPY

sudo chown myuser:myuser /var/mongodb\_directory

**Commands You Should Know**

**Creating a directory**

**mkdir**

The [mkdir](https://www.computerhope.com/unix/umkdir.htm" \t "_blank) command creates new directories in the file system. The general syntax is :

COPY

mkdir [modifiers] <name of the directory>

The **-p** modifier creates the parent directory as necessary. For example :

COPY

mkdir -p /db/mongodb\_data/

If **db** directory already exists then this command will create only **mongodb\_data** directory. Otherwise it will create the missing directory as well.

Sometimes, you will create directories inside a parent directory which is owned by **root** then you can use the sudo command. For example :

COPY

sudo mkdir -p /var/data

**Deleting a directory**

**rm**

The [rm](https://linoxide.com/linux-command/linux-rm-command/) command is used for removing or deleting files and directories in the Linux operating system. The general syntax is :

COPY

rm [options] [directory|file]

If you want to delete a directory and all the sub-directories and file recursively without prompting the user then you can use this command.

COPY

rm -rf <directory name>

**Warning :** Be extremely careful when running this command, because once the directories and files have been deleted it cannot be recovered. By default, it will not prompt before any removal.

**File operations in linux**

**ls**

The ls command is used to list information about files and directories within the file system. The general syntax is :

COPY

ls

To get a long listing of every file in a directory, including the hidden files, use :

COPY

ls -la

Long listing format prints some additional information such as : The file permissions, File owner, File group etc.

**cd**

The cd command is used to change the current working directory. The general syntax is :

COPY

cd <directory>

**cat**

The cat command to view the content of any file in the terminal. The general syntax is :

COPY

cat <name of the file>



**Note:** If the file is not in the same directory then we can pass the absolute or relative path of the file to the current directory.

**rm**

The rm command is used to remove a file from the current directory.

COPY

rm <file name>

**Warning:** Be extremely careful when running this command, because once the directories and files have been deleted it cannot be recovered. By default, it will not prompt before any removal.

**Linux process management**

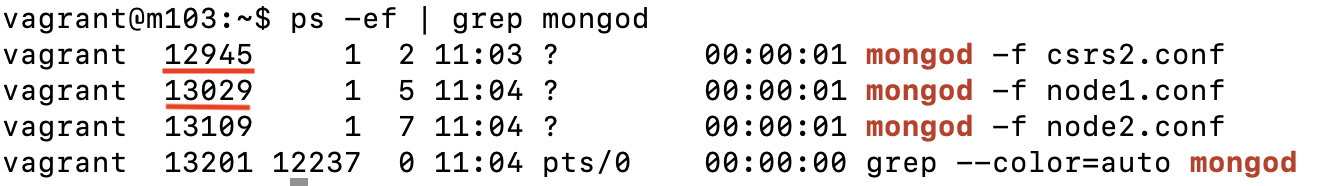
Command to check which mongo, mongod and mongos instances are currently running on your system

COPY

ps -ef | grep mongo

ps -ef | grep mongod

ps -ef | grep mongos



Command to kill any specific process using the process id i.e. **pid**

COPY

kill <pid>

For instance, let's say we want to kill the process associated with **node1.conf** in the above example. We can do so by running this command.

COPY

kill 13029

Command to kill all the currently running mongod instances.

COPY

killall mongod

**Note:** Be careful when running this command as it will kill all the currently running mongod instances in one go.