To goto root directory, and view the file system

sibu@sibu-To-be-filled-by-O-E-M:~/Desktop$ pwd

/home/sibu/Desktop

sibu@sibu-To-be-filled-by-O-E-M:~/Desktop$ cd /../../

sibu@sibu-To-be-filled-by-O-E-M:/$ pwd /

sibu@sibu-To-be-filled-by-O-E-M:/$ ls

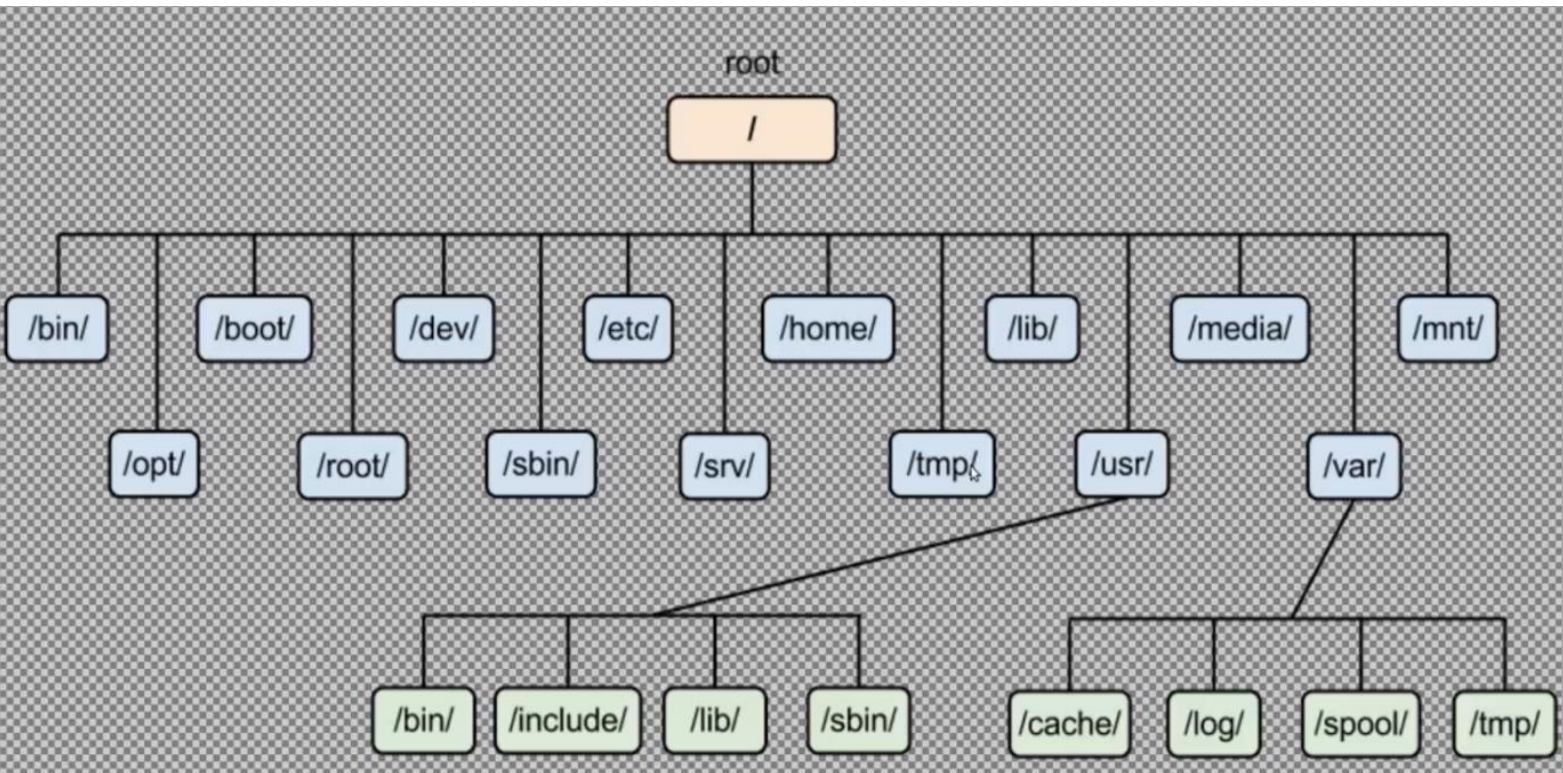bin   dev   lib   libx32   mnt   root   snap   sys

var   boot   etc   lib32   lost+found   opt   run

srv   tmp   cdrom   home   lib64   media

proc   sbin   swapfile   usr

sibu@sibu-To-be-filled-by-O-E-M:/$

Tree Structure

To view the above structure in tree format

/$ tree/
Note: if it is not installed, then install using the command
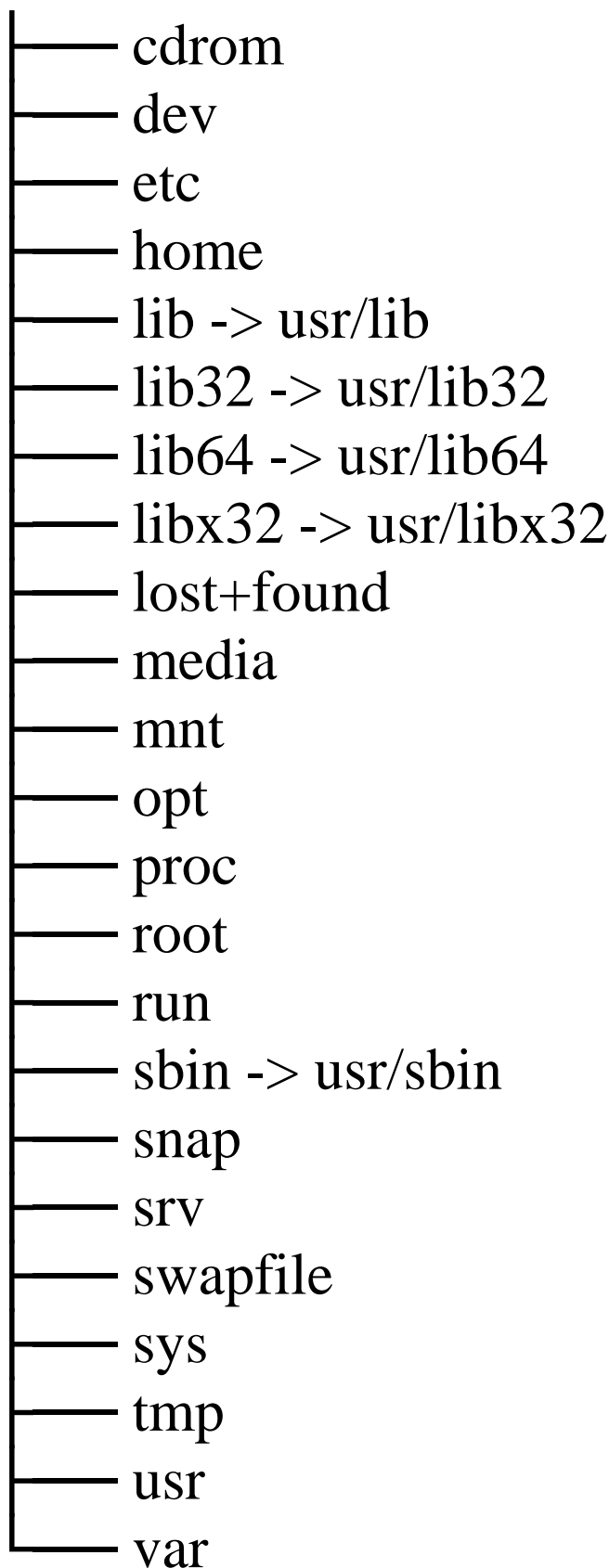/$ sudo apt install tree

after the installation, use
/$ tree/
displays tree structure

sibu@sibu-To-be-filled-by-O-E-M:/$ tree -L 1 /
/
├── bin -> usr/bin
├── boot

```
├────── cdrom
├────── dev
├────── etc
├────── home
├────── lib -> usr/lib
├────── lib32 -> usr/lib32
├────── lib64 -> usr/lib64
├────── libx32 -> usr/libx32
├────── lost+found
├────── media
├────── mnt
├────── opt
├────── proc
├────── root
├────── run
├────── sbin -> usr/sbin
├────── snap
├────── srv
├────── swapfile
├────── sys
├────── tmp
├────── usr
└────── var
```

24 directories, 1 file

shows only the first level tree structure. L indicates the number of levels.

Directories

From top to bottom, the directories you are seeing are as follows.

/bin

/bin is the directory that contains *bin*aries, that is, some of the applications and programs you can run. You will find the *ls* program mentioned above in this directory, as well as other basic tools for making and removing files and directories, moving them around, and so on. There are more *bin* directories in other parts of the file system tree, but we'll be talking about those in a minute.

/boot

The */boot* directory contains files required for starting your system.   If you mess up one of the files in here, you may not be able to run your Linux and it is a pain to repair. On the other hand, don't worry too much about destroying your system by accident: you have to have superuser privileges to do that.

/dev

/dev contains *dev*ice files. Many of these are generated at boot time or even on the fly. For example, if you plug in a new webcam or a USB pendrive into your machine, a new device entry will automatically pop up here.

/etc

/etc is the directory where names start to get confusing. */etc* gets its name from the earliest Unixes and it was literally "et cetera" because it was the dumping ground for system files administrators were not sure where else to put.

Nowadays, it would be more appropriate to say that *etc* stands for "Everything to configure," as it contains most, if not all system-wide configuration files. For example, the files that contain the name of your system, the users and their passwords, the names of machines on your network and when and where the partitions on your hard disks should be mounted are all in here. Again, if you are new to Linux, it may be best if you don't touch too much in here until you have a better understanding of how things work.

/home

/home is where you will find your users' personal directories. In my case, under */home* there are two directories: */home/paul*, which contains all my stuff; and */home/guest*, in case anybody needs to borrow my computer.

/lib

/lib is where *lib*raries live. Libraries are files containing code that your applications can use. They contain snippets of code that applications use to draw windows on your desktop, control peripherals, or send files to your hard disk.

There are more *lib* directories scattered around the file system, but this one, the one hanging directly off of */* is special in that, among other things, it contains the all-important kernel modules. The kernel modules are drivers that make things like your video card, sound card, WiFi, printer, and so on, work.

/media

The */media* directory is where external storage will be automatically mounted when you plug it in and try to access it. As opposed to most of the other items on this list, */media* does not hail back to 1970s, mainly because inserting and detecting storage (pendrives, USB hard disks, SD cards, external SSDs, etc) on the fly, while a computer is running, is a relatively new thing.

/mnt

The */mnt* directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

/opt

The */opt* directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the */opt/bin* directory and libraries in the */opt/lib* directory.

A slight digression: another place where applications and libraries end up in is */usr/local*, When software gets installed here, there will also be */usr/local/bin* and */usr/local/lib* directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

/proc

/proc, like */dev* is a virtual directory. It contains information about your computer, such as information about your CPU and the kernel your Linux system is running. As with */dev*, the files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

/root

/root is the home directory of the superuser (also known as the "Administrator") of the system. It is separate from the rest of the users' home directories BECAUSE YOU ARE NOT MEANT TO TOUCH IT. Keep your own stuff in you own directories, people.

/run

/run is another new directory. System processes use it to store temporary data for their own nefarious reasons.

/sbin

/sbin is similar to */bin*, but it contains applications that only the superuser (hence the initial *s*) will need. You can use these applications with the sudo command that temporarily concedes you superuser powers on many distributions. */sbin* typically contains tools that can install stuff, delete stuff and format stuff. As you can imagine, some of these instructions are lethal if you use them improperly, so handle with care.

/usr

The */usr* directory was where users' home directories were originally kept back in the early days of UNIX. However, now */home* is where users kept their stuff as we saw above. These days, */usr* contains a mish-mash of directories which in turn contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services.

You will also find *bin*, *sbin* and *lib* directories in */usr*. What is the difference with their root-hanging cousins? Not much nowadays. Originally, the */bin* directory (hanging off of root) would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system. */usr/bin* on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps.

But many modern Linux distributions just put everything into */usr/bin* and have */bin* point to */usr/bin* just in case erasing it completely would break something. So, while Debian, Ubuntu and Mint still keep */bin* and */usr/bin* (and */sbin* and */usr/sbin*) separate; others, like Arch and its derivatives just have one "real" directory for binaries, */usr/bin*, and the rest or **bin*s are "fake" directories that point to */usr/bin*.

/srv

The */srv* directory contains data for servers. If you are running a web server from your Linux box, your HTML files for your sites would go into */srv/http* (or */srv/www*). If you were running an FTP server, your files would go into */srv/ftp*.

/sys

/sys is another virtual directory like */proc* and */dev* and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the */sys/devices/pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel_backlight/brightness* file (on

your machine you will probably have a different file). But to do that you have to become superuser. The reason for that is, as with so many other virtual directories, messing with the contents and files in */sys* can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

/tmp

/tmp contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn't need right now, but may need later on.

You can also use */tmp* to store your own temporary files — */tmp* is one of the few directories hanging off / that you can actually interact with without becoming superuser.

/var

/var was originally given its name because its contents was deemed *variable*, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above.

Be that as it may, */var* contains things like logs in the */var/log* subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in */var/log*; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains *spools* for tasks. These "tasks" can be the jobs you send to a shared printer when you have to wait because another user is printing a long document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven't mentioned above. In the screenshot, for example, there is a */snap* directory. That's because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated snap packages as a way of distributing software. The */snap* directory contains all the files and the software installed from snaps.

/var/log/syslog contains lot of system related logfiles.

That is the root directory covered, but many of the subdirectories lead to their own set of files and subdirectories. Figure 2 gives you an overall idea of what the basic file system tree looks like (the image is kindly supplied under a CC By-SA license by Paul Gardner) and Wikipedia has a break down with a summary of what each directory is used for.