**Linux terminal commands**

**Log in: ssh <user/acount\_name>@<ip\_addr>**

Log out**: exit**

Root directory (root of FS): **cd /**

Home directory (root/home/user\_name**): cd ~**

Change account to root for WSL: You’d need to know your distributions command to access it via windows cmd. You can view it on Windows store for the installed distribution. For Debian: debian amnd for linux unbuntu: ubuntu.

Now just type this in windows command prompt: <distribution\_access\_command> config –default-user root

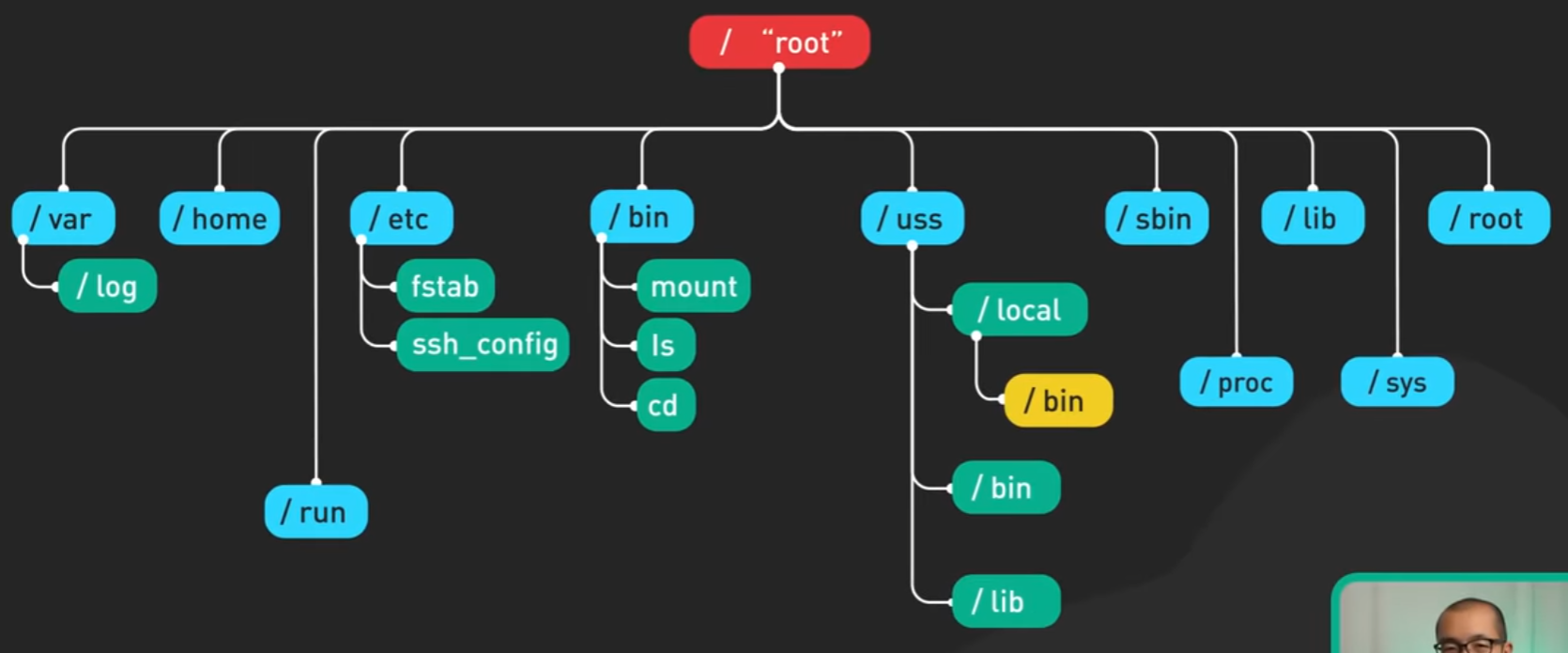
Or to change it back to normal user/account: <ditribution\_access\_command> config –default-user <account>

Confirm OS/Distribution on Linux: see contents of etc/os-release

All user/account details: see contents of etc/passwd

Command binary location:> **which <command>**

**Linux File system:**



**Major directories in root directory:**

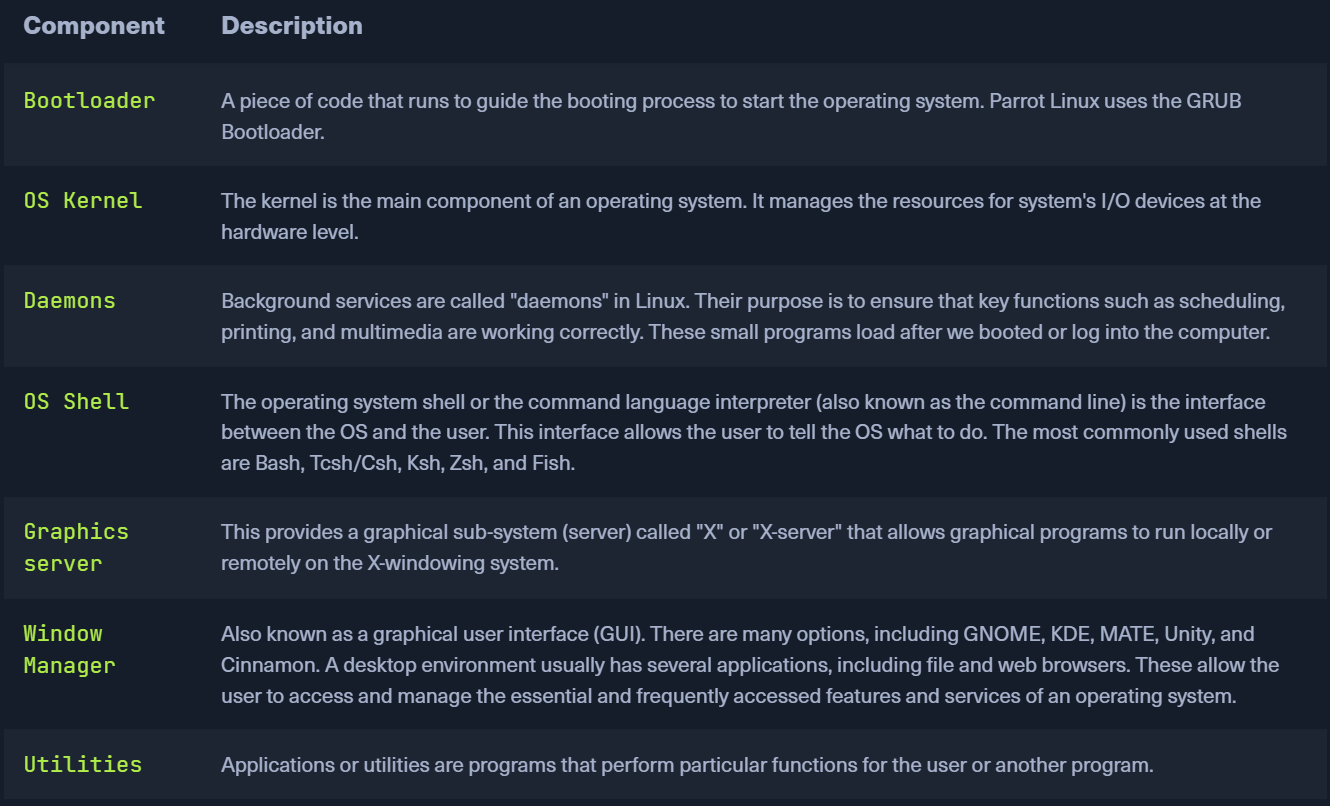
|  |  |
| --- | --- |
| / | The top-level directory is the root filesystem and contains all of the files required to boot the operating system before other filesystems are mounted as well as the files required to boot the other filesystems. After boot, all of the other filesystems are mounted at standard mount points as subdirectories of the root. |
| /bin | Contains essential command binaries. Holds system level binary files (system binaries). |
| /sbin | Contains executable binaries used by the super user (root if sudo group has no other member). |
| /boot | Consists of the static bootloader, kernel executable, and files required to boot the Linux OS. |
| /dev | Contains device files to facilitate access to every hardware device attached to the system. NOTE: /dev/null is directory which deletes automatically whatever is pushed into it. |
| /etc | Home to Local system-wide configuration files. Configuration files for installed applications may be saved here as well. (.conf files) |
| /home | Each user on the system has a subdirectory here for storage. |
| /lib | Shared library files that are required for system boot. |
| /media | Holds files system for External removable media devices such as USB drives are mounted here. |
| /mnt | Temporary mount point for regular filesystems. |
| /opt | Optional files such as third-party tools can be saved here. |
| /root | Like Special /home dir for super user |
| /temp | Directory holding temporary files for the running services |
| /var | Directory holding log files for applications and system srevices and tasks. |

**Note:** /bin contains essential system binaries that are needed for the system to boot and perform basic operations even before user gets mounted at boot time. Where as /usr/bin holds non-essential system binaries for user programs. And usr/local/bin holds installed program’s executable binaries. In summary, /bin contains essential binaries for system operation, /usr/bin holds non-essential user binaries, and /usr/local/bin is for locally installed software that is not part of the system's package management.

**Note: /media holds FS of the devices in /dev, attached to the file tree.**

**NOTE: Everytihng in Linux is a File**

**Components of Linux OS:**



**Package managers for different Linux distributions:**

* Debian: Advance package tools (**apt**)
* RHLE: Yellowdog Updater Modified (**yum**) for versions above 6, Dandified YUM (**dnf**) for versions <=6.
* openSUSE: Zypp (**zypper**)

1. Usermod: modify users/groups:

**:> sudo usermod -aG <group(s) seperated by commas> <username>**

This adds user with <username> to a group/groups, Provided that current user is in sudoers file. (is admin or in sudo group)

**:>sudo groupadd <groupName>**

Create a group and add it in the list of groups

**:> groups <username>**

List all the groups where this user with <username> belongs

**:> usermod -l <new\_username> <old\_username>**

Chnages the username of the user.

**:>usermod -L <username>**

Lockt a user with <username> from root/other user(with root priviledges) account. This will not allow the user with <username> to login to their account as the admin/root has locked him/her. Inorder to unlock the user, root/user (with admin priviledges must) use -U flag.

**:> usermod -U <username>**

In the same way we can set an expiration date for a user account, from root/user with root priviledges. This can be done from root by:

**:> usermod <username> -e YYYY-MM-DD**

And we can check that using “chage” comand:

**:> chage -l <username**>

This will deliver us the account expiry details

1. Sudo: Grant Root privileges to current acount.

**Installation of this utility: sudo apt update**

**sudo apt install sudo**

**Switching to root: su - (switch user/login to another user, root here)**

Logout: exit

Check if <user> is from ‘sudo’ group**:> groups <user>** { list all groups:> groups }

Add the user, if he is not from ‘sudo’ group**:> usermod -aG <group> <user\_name>**

**:> usermod -aG sudo <user>**

NOTE: In order to get the new group membership working, reboot/logout-login to the user acccount.

How to check what sudo-command permissions does current user have:> sudo -l

Add sudo to previously (top on history stack) written non-priviledged command:> sudo !!

How to see list of admin-priviledged users/groups: View the content of the sudoers file (/etc/sudoer)

This file can be editted as wellm under root privileges. Root permissions can look like: root ALL=(ALL:ALL) ALL

Which translates to ‘For All the servers, All the users under All the groups have access to All the commands’

To edit this file (given that we are in priviledged user account):> sudo visudo

New\_user ALL=(ALL:ALL) /usr/bin/apt (**New user can only user sudo utility with apt command)**

How to add a user\_account:> sudo adduser <new\_user\_username>

Switch to (another user), say new\_account:> sudo su - <new\_usename>

Everytime you learn a priviledged command, you would be asked to type passord. To avoid this we can add something like this {say we want new\_user to use sudo only with apt command without ever asking for password}, the modify the sudoer file like this:

New\_user ALL=(ALL:ALL) NOPASSWD: /usr/bin/apt

1. Htop: monitor system/machine resources.

**Installation of this utility: sudo apt update**

**sudo apt install htop**

**:> htop**

In htop interface, you can sort the processes by the CPU usage/memory usage using**: Shift+P or Shift+M shortcuts. You can narow down the list of processes by user by presing ‘u’ and selecting username for which you want to monitor system resources. All the setup shortcuts are present in the htop interface.**

1. **ps** command: analyse about the processes running on the machine/server.

**:> ps**: Displays a snapshot of processes associated with the current terminal.

**:>** **ps aux**: Displays a detailed list of all processes running on the system, including those of other users. This is often used to get a comprehensive view of all processes.

**:>** ps -e: Shows information about every process on the system.

**:>** ps -ef: Similar to ps -e, providing a full listing of processes along with additional information.

The ps command, when used without specific options, typically shows a snapshot of processes associated with the terminal in which it is run.

Different terminal sessions may have different sets of running processes, and the ps command without specific options will display processes associated with the terminal from which it is executed. If you want to see a comprehensive list of all processes on the system, regardless of the terminal session, you can use the ps aux or ps -ef command as mentioned earlier.

1. **Data Streams in Linux**: In linux data streams have 3 by default channels. These 3 channels are fixed to 0, 1 and 2 for standard input stdin, standard output/stdout and standard error/stderr channel. You can use the output/err output from one command to be utilised for the next command using ‘**>’ operator** between to commands. This is called comand chaining. Format is

<command1> <channel\_number> **>** <command2>………..

1. **Nano editor:** :>nano <file\_name>

It is a simple text editor to change and read file contents. Write out option is to save. ‘Where is ’ is search option. ‘Cut text’ is to cut entire line if not seleceted a specific part. ‘paste text’ is to paste/uncut option.

:> nano +<line\_number> <file\_name> //open at exactly given line number