Assignments - 25th – May-2022

1) Operating systems architecture?

: An operating system is a program that acts as an interface between a user of a computer and the computer resources. The purpose of an operating system is to provide an environment in which a user may execute programs.

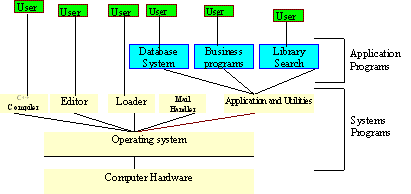
**General Architecture of an Operating System**

Operating systems architecture refers to the overall design of hardware and software components and their operational effectiveness as a whole.

To be effective, however, an operating system must not only be cognizant of the collection of hardware and software modules, but must also be designed in light of the programs and data which the system processes and the people which it serves.

The absence of formal theory on operating systems and the lack of standard terminology have caused much confusion among users.

The problem is particularly apparent when comparing systems where the same terms are applied to a variety of concepts and levels of implementation.



2)What is Linux Architecture?

: The Linux operating system's architecture mainly contains some of the components: the Kernel, System Library, Hardware layer, System, and Shell utility.

**Hardware**

The hardware consists of the memory, CPU, arithmetic-logic unit, various bulk storage devices, I/O, peripheral devices and other physical devices.

**Kernel**

In computing, the kernel is the central component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level.

**Shell**

A shell is a piece of software that provides an interface for users to an operating system which provides access to the services of a kernel.



1.**Kernel**: - The kernel is one of the core sections of an operating system. It is responsible for each of the major actions of the Linux OS.  This operating system contains distinct types of modules and cooperates with underlying hardware directly. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

* Monolithic Kernel
* Micro kernels
* Exo kernels
* Hybrid kernels

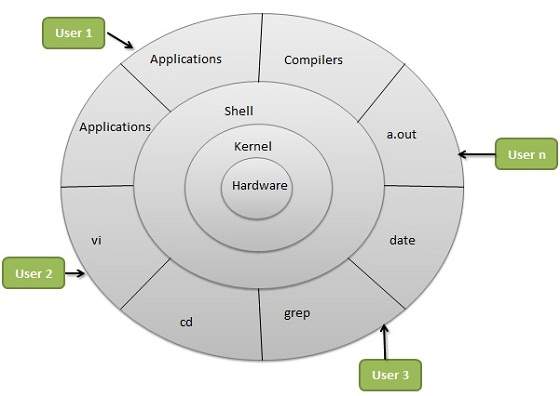
**2. System Libraries: -** These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

**3. System Utility Programs: -** It is responsible for doing specialized level and individual activities.

**4. Hardware layer:-** Linux operating system contains a hardware layer that consists of several peripheral devices like [CPU](https://www.javatpoint.com/central-processing-unit), [HDD](https://www.javatpoint.com/hdd), and [RAM](https://www.javatpoint.com/ram).

**5. Shell: -** It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are the **graphical shells** and **command-line shells**.

The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.



3) What is a computer system BIOS?

: As your PC's most important start up program, BIOS, or Basic Input/Output System, is the built-in core processor software responsible for booting up your system.

Typically embedded into your computer as a motherboard chip, the BIOS functions as a catalyst for PC functionality action.

OR

In computing, BIOS is firmware used to provide runtime services for operating systems and programs and to perform hardware initialization during the booting process.

The BIOS firmware comes pre-installed on an IBM PC or IBM PC compatible's system board and exists in UEFI-based systems too.

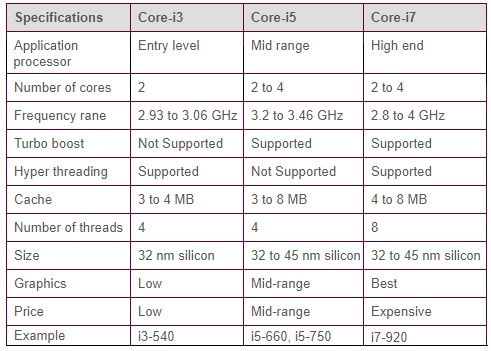
4)Difference between 13-bit o.s and 64-bit o.s?

* As its name suggests, the 32-bit OS can store and handle lesser data than the 64-bit OS.
* More specifically, it addresses a maximum of 4,294,967,296 bytes (4 GB) of RAM.
* The 64-bit OS, on the other hand, can handle more data than the 32-bit OS.
* When it comes to computers, the difference between 32-bit and a 64-bit is all about processing power.
* Computers with 32-bit processors are older, slower, and less secure, while a 64-bit processor is newer, faster, and more secure.

5)What are the difference between i3 i5 and i7?

* Core i3 processors have two cores
* Core i5 CPUs have four and Core
* i7 models also have four.
* Some Core i7 Extreme processors have six or eight cores.

Generally speaking, we find that most applications can't take full advantage of six or eight cores, so the performance boost from extra cores isn't as great.



6)What is an OS interrupt driven?

* interrupt-driven Denoting a process that is restarted by the occurrence of an interrupt.
* Interrupts are important because they give the user better control over the computer.
* Without interrupts, a user may have to wait for a given application to have a higher priority over the CPU to be ran. This ensures that the CPU will deal with the process immediately.

WORKING:

Interrupt driven I/O is an alternative scheme dealing with I/O.

Interrupt I/O is a way of controlling input/output activity whereby a peripheral or terminal that needs to make or receive a data transfer sends a signal.

This will cause a program interrupt to be set.

7)What is Interrupt handling?

* Interrupt handling is a key function in real-time software, and comprises interrupts and their handlers.
* Only those physical interrupts which of high enough priority can be cantered into system interrupt table.
* The software assigns each interrupt to a handler in the interrupt table.

8)What are the function of interrupt?

: Interrupts are commonly used to service hardware timers, transfer data to and from storage (e.g., disk I/O) and communication interfaces (e.g., UART, Ethernet), handle keyboard and mouse events, and to respond to any other time-sensitive events as required by the application system.

OR

Role of Interrupts. Interrupts are signals sent to the CPU by external devices, normally I/O devices

. They tell the CPU to stop its current activities and execute the appropriate part of the operating system.

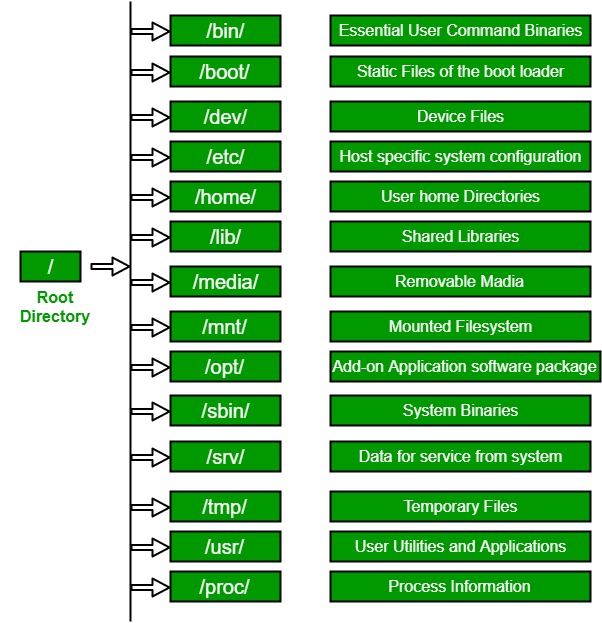
9)Linux commands list?



Assignments - 27th – May-2022

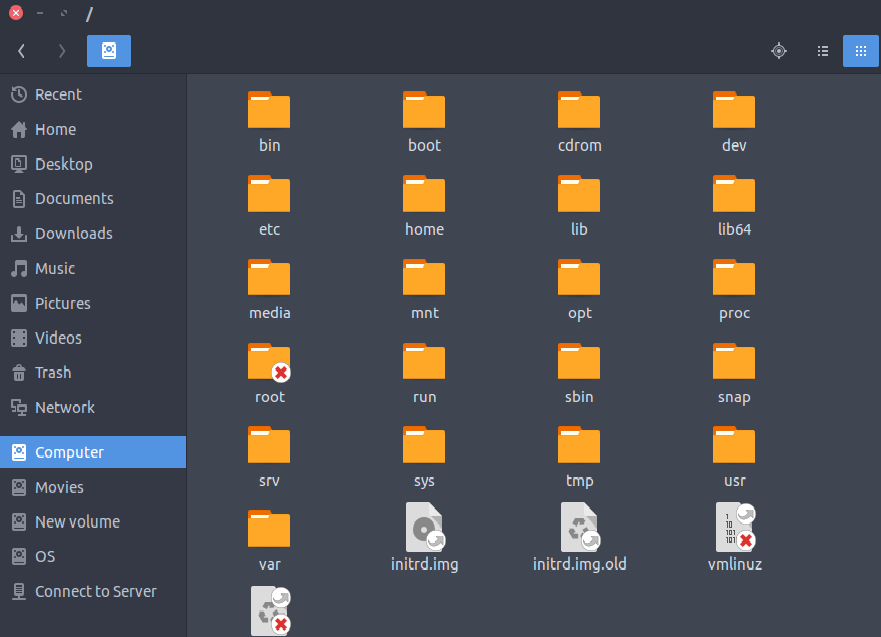
1. Linux directory hierarchy?

:



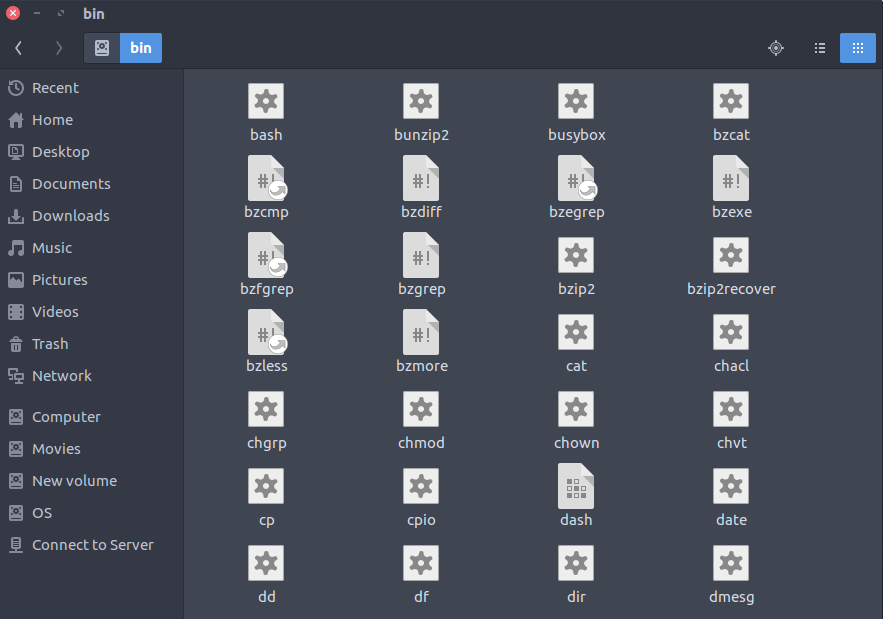
1. / (Root): Primary hierarchy root and root directory of the entire file system hierarchy.

* Every single file and directory starts from the root directory
* The only root user has the right to write under this directory
* /root is the root user’s home directory, which is not the same as /



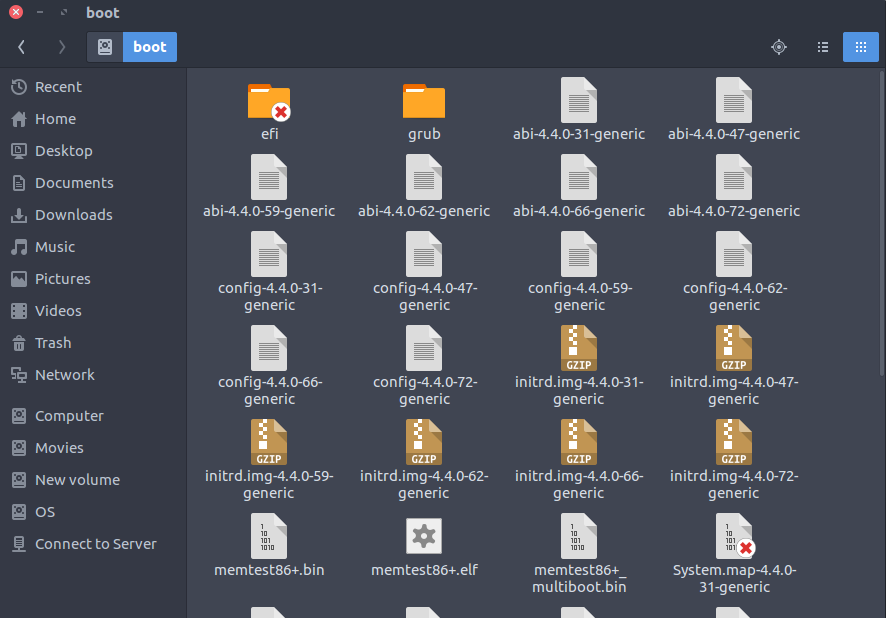
2. /bin : Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.

* Contains binary executables
* Common linux commands you need to use in single-user modes are located under this directory.
* Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp



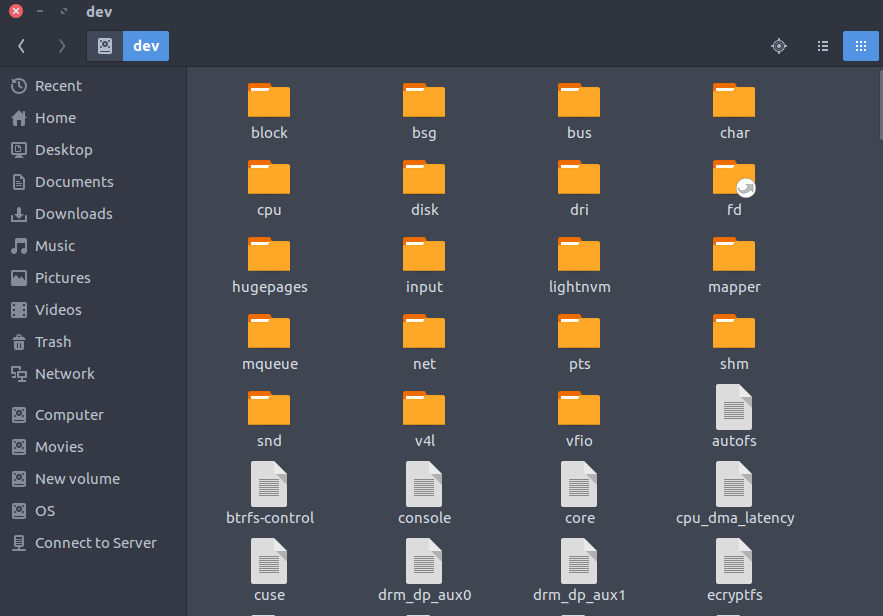
3. /boot : Boot loader files, e.g., kernels, initrd.

* Kernel initrd, vmlinux, grub files are located under /boot
* Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic



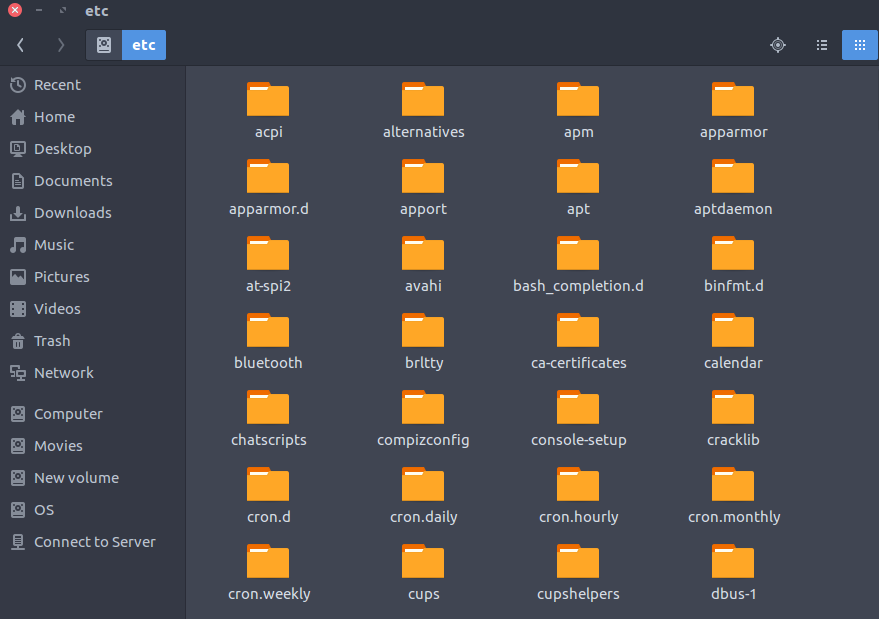
4. /dev : Essential device files, e.g., /dev/null.

* These include terminal devices, usb, or any device attached to the system.
* Example: /dev/tty1, /dev/usbmon0



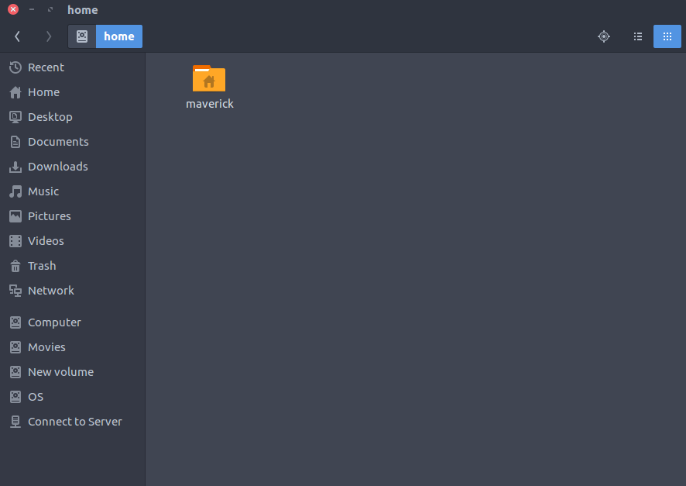
5. /etc : Host-specific system-wide configuration files.

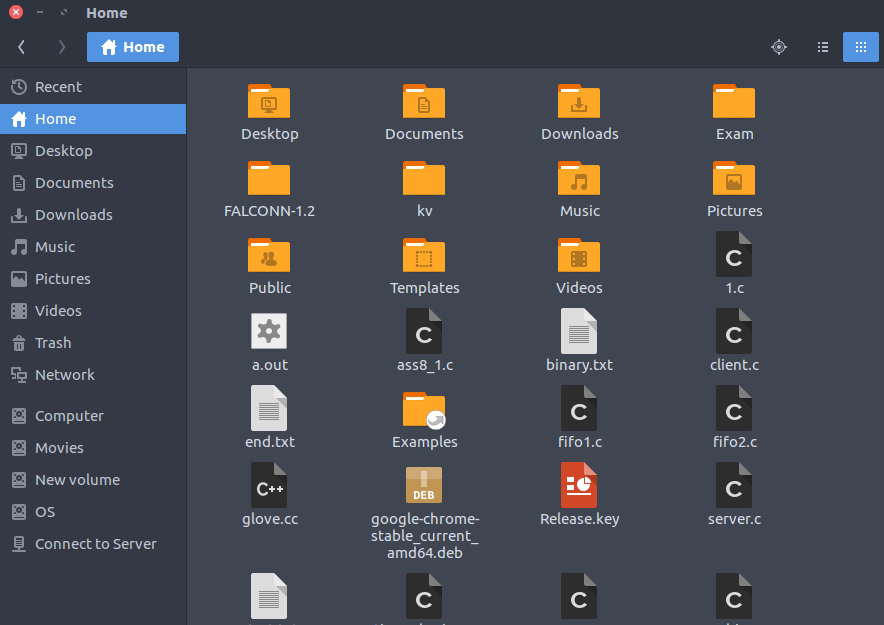
* Contains configuration files required by all programs.
* This also contains startup and shutdown shell scripts used to start/stop individual programs.
* Example: /etc/resolv.conf, /etc/logrotate.conf.



6. /home : Users’ home directories, containing saved files, personal settings, etc.

* Home directories for all users to store their personal files.
* example: /home/kishlay, /home/kv

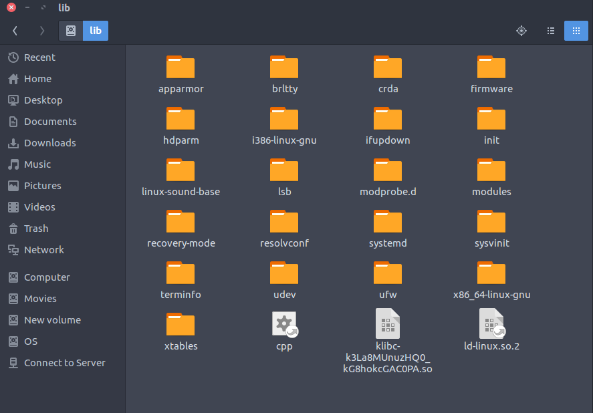




7. /lib : Libraries essential for the binaries in /bin/ and /sbin/.

• Library filenames are either ld\* or lib\*.so.\*

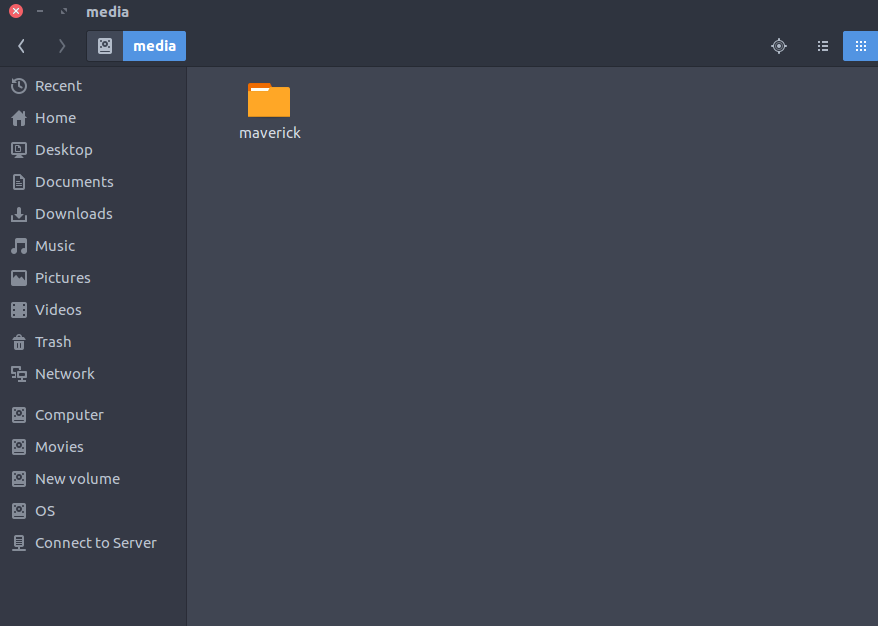
• Example: ld-2.11.1.so, libncurses.so.5.7



8. /media : Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).

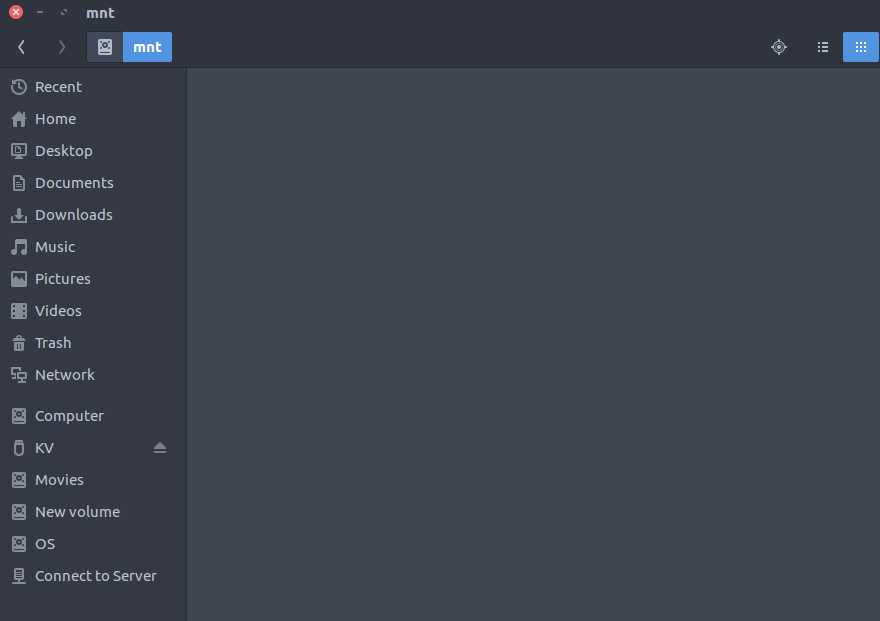
• Temporary mount directory for removable devices.

• Examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer



9. /mnt : Temporarily mounted filesystems.

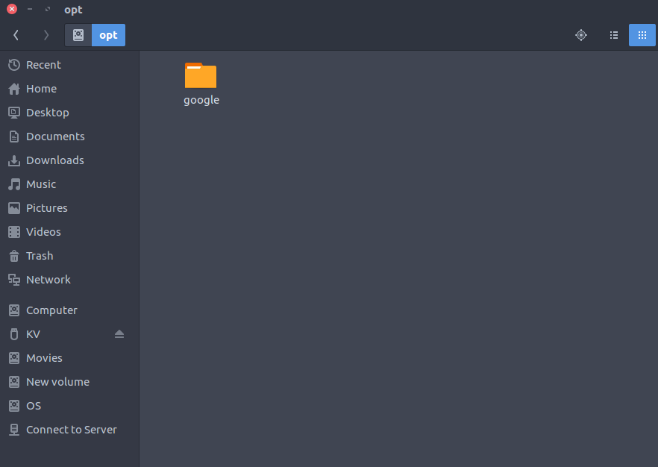
• Temporary mount directory where sysadmins can mount filesystems.



10. /opt : Optional application software packages.

• Contains add-on applications from individual vendors.

• Add-on applications should be installed under either /opt/ or /opt/ sub-directory.

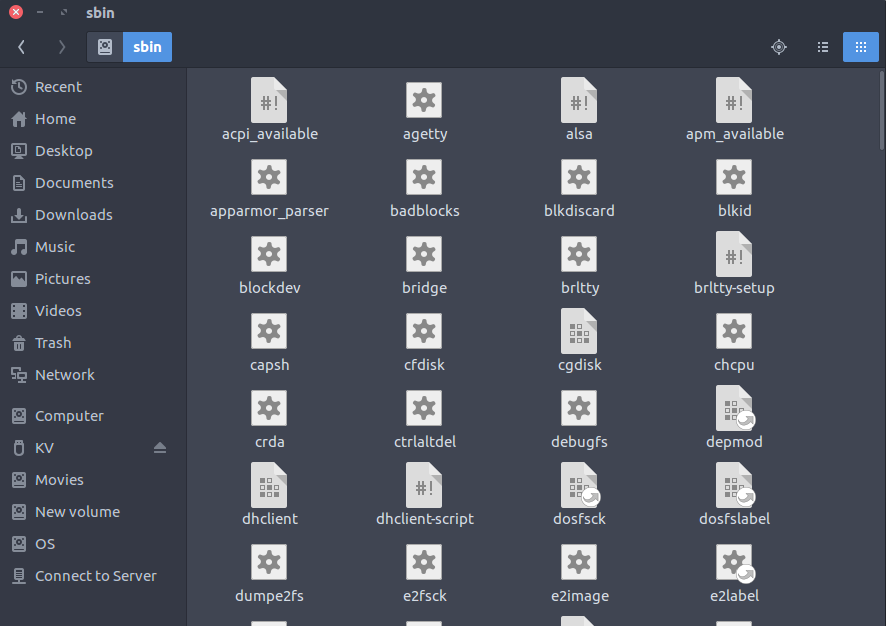


11. /sbin : Essential system binaries, e.g., fsck, init, route.

• Just like /bin, /sbin also contains binary executables.

• The linux commands located under this directory are used typically by system administrator, for system maintenance purpose.

• Example: iptables, reboot, fdisk, ifconfig, swapon

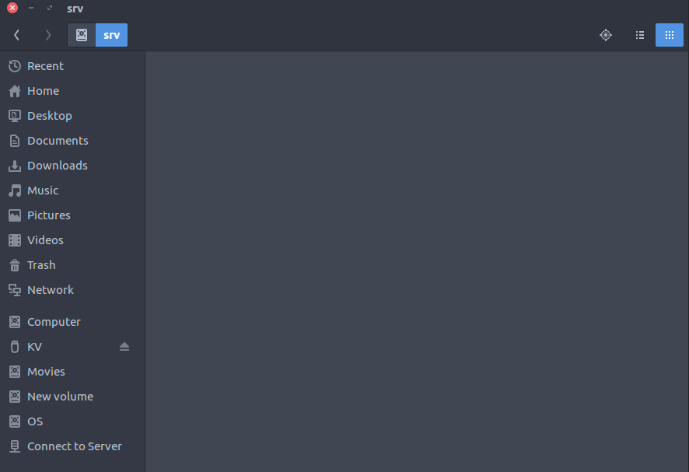


12. /srv : Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems.

• srv stands for service.

• Contains server specific services related data.

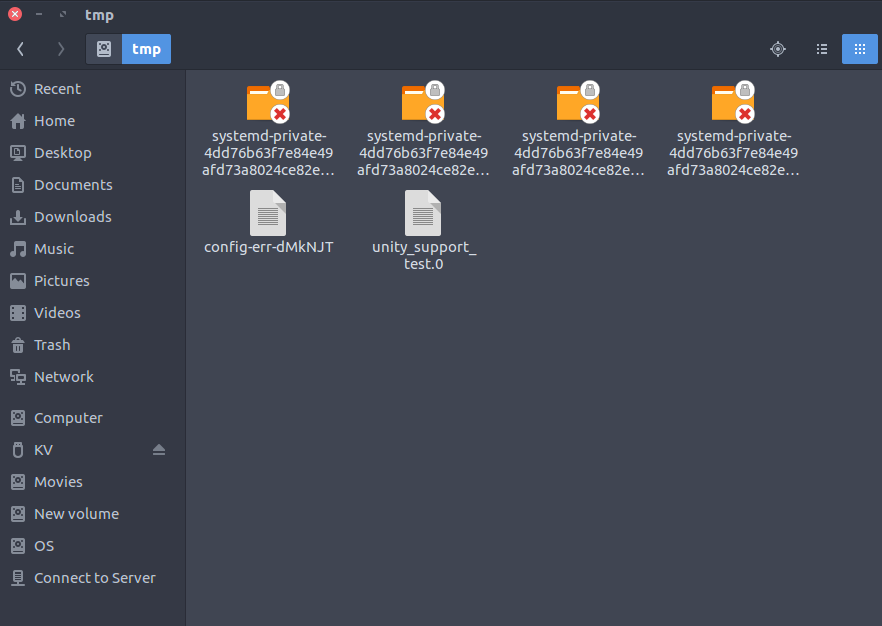
• Example, /srv/cvs contains CVS related data.



13. /tmp : Temporary files. Often not preserved between system reboots, and may be severely size restricted.

• Directory that contains temporary files created by system and users.

• Files under this directory are deleted when system is rebooted.



14. /usr : Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

• Contains binaries, libraries, documentation, and source-code for second level programs.

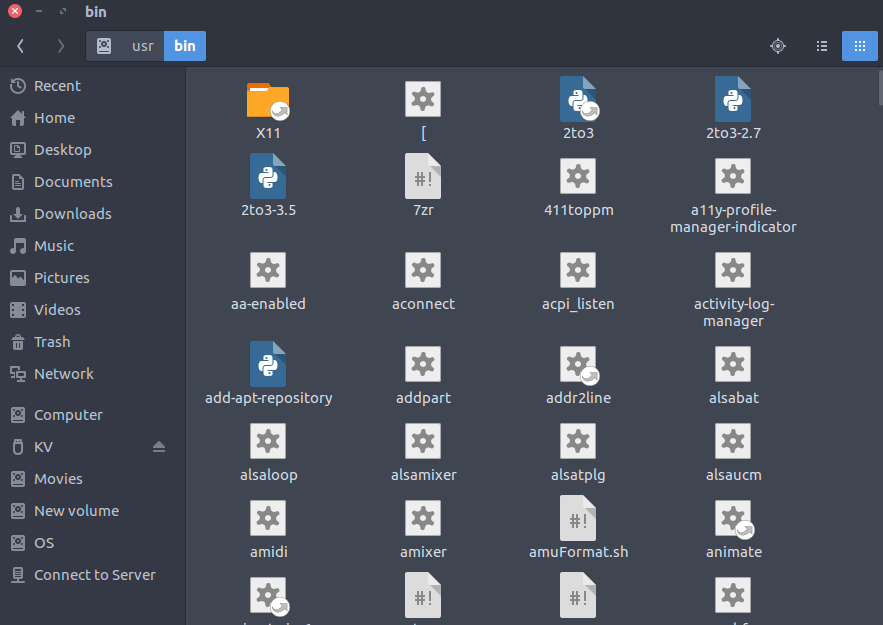
• /usr/bin contains binary files for user programs. If you can’t find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp

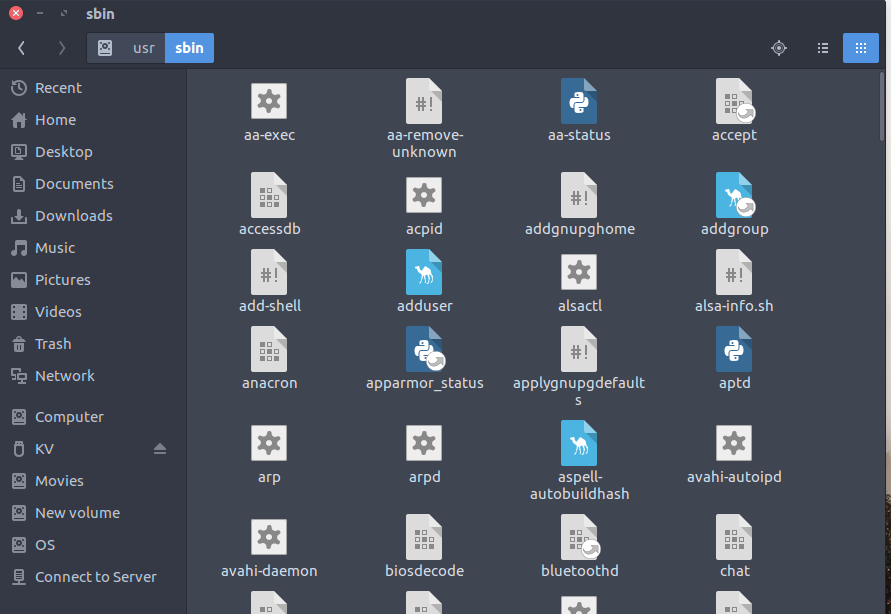
• /usr/sbin contains binary files for system administrators. If you can’t find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel

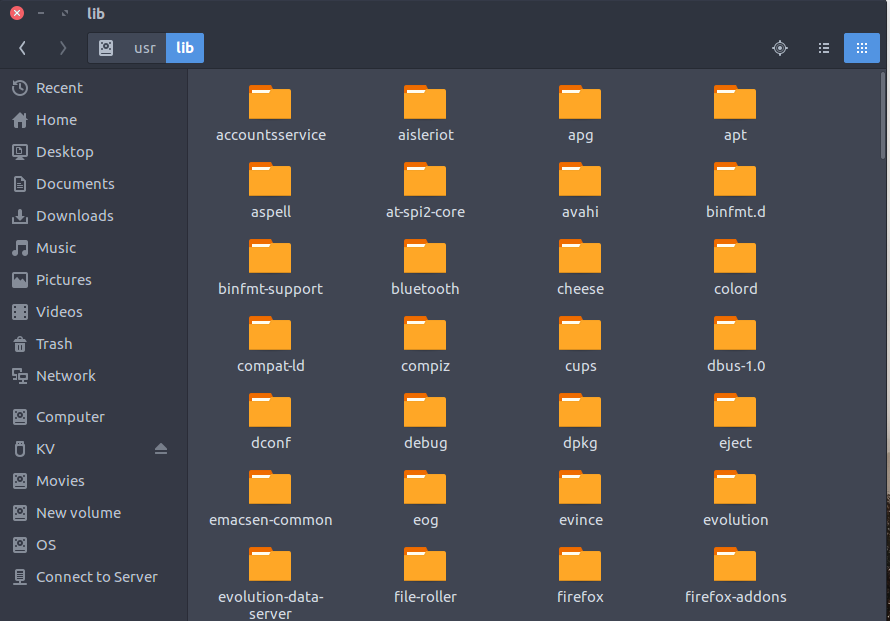
• /usr/lib contains libraries for /usr/bin and /usr/sbin

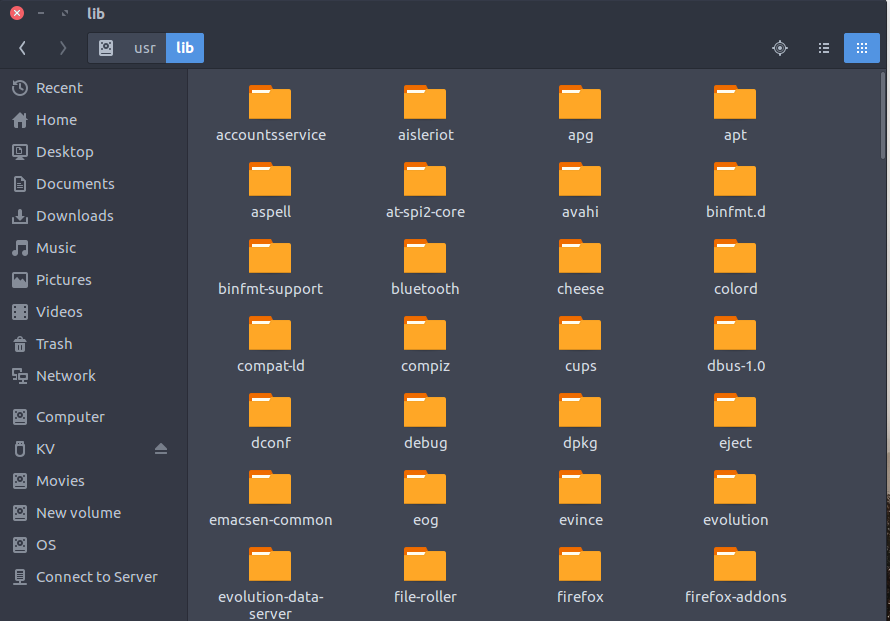
• /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

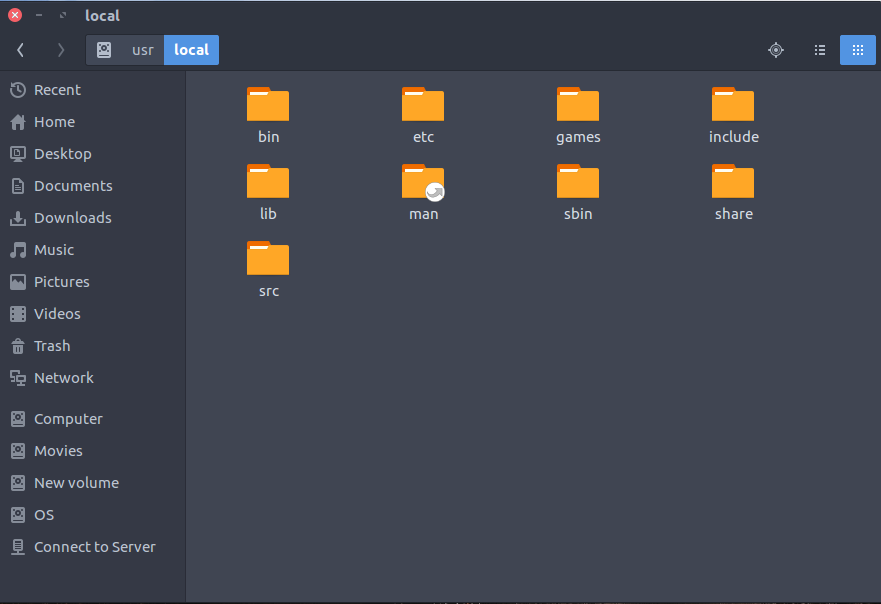
• /usr/src holds the Linux kernel sources, header-files and documentation.

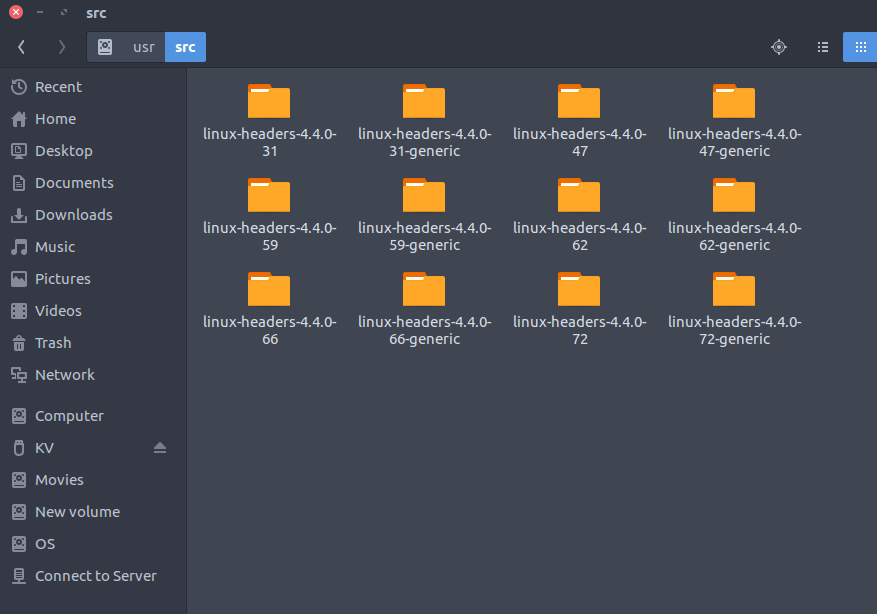










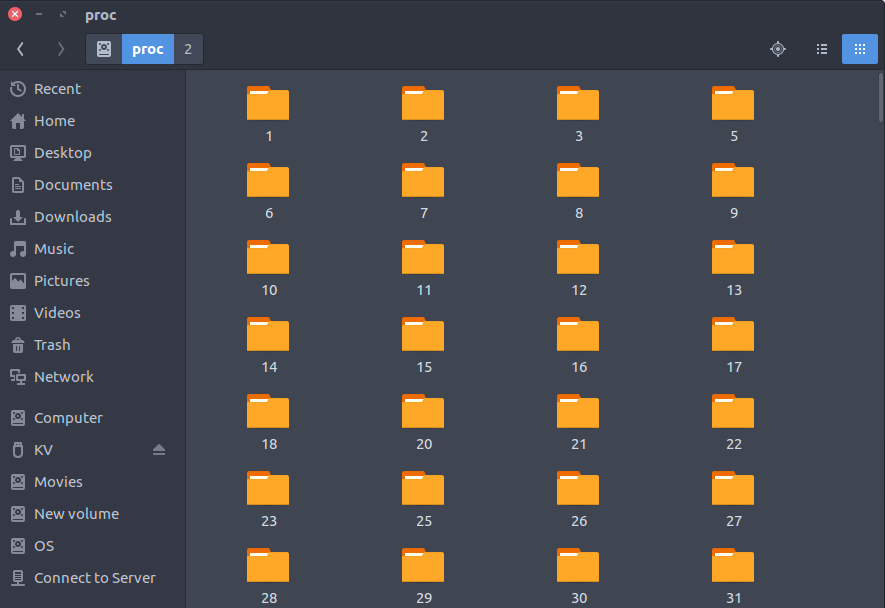


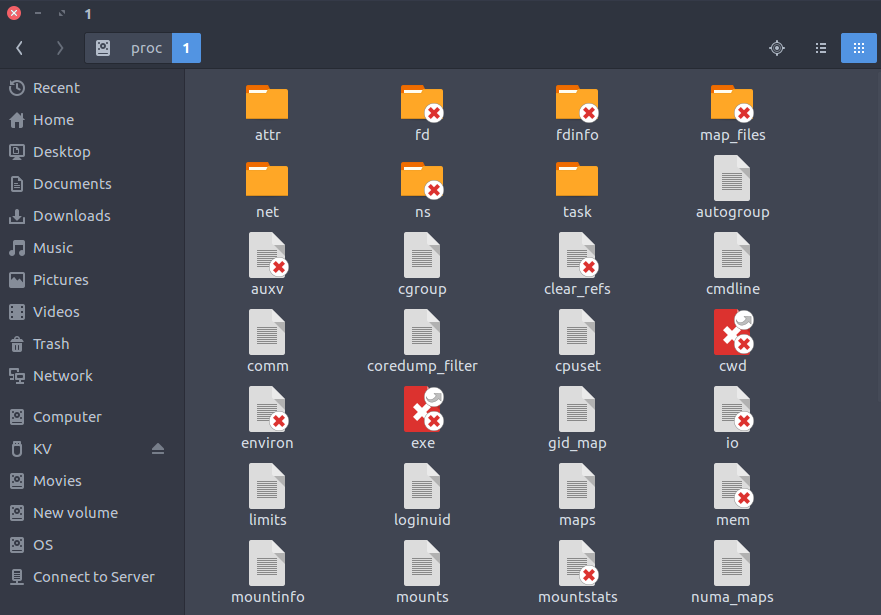
15. /proc : Virtual filesystem providing process and kernel information as files. In Linux, corresponds to a procfs mount. Generally automatically generated and populated by the system, on the fly.

• Contains information about system process.

• This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.

• This is a virtual filesystem with text information about system resources. For example: /proc/uptime

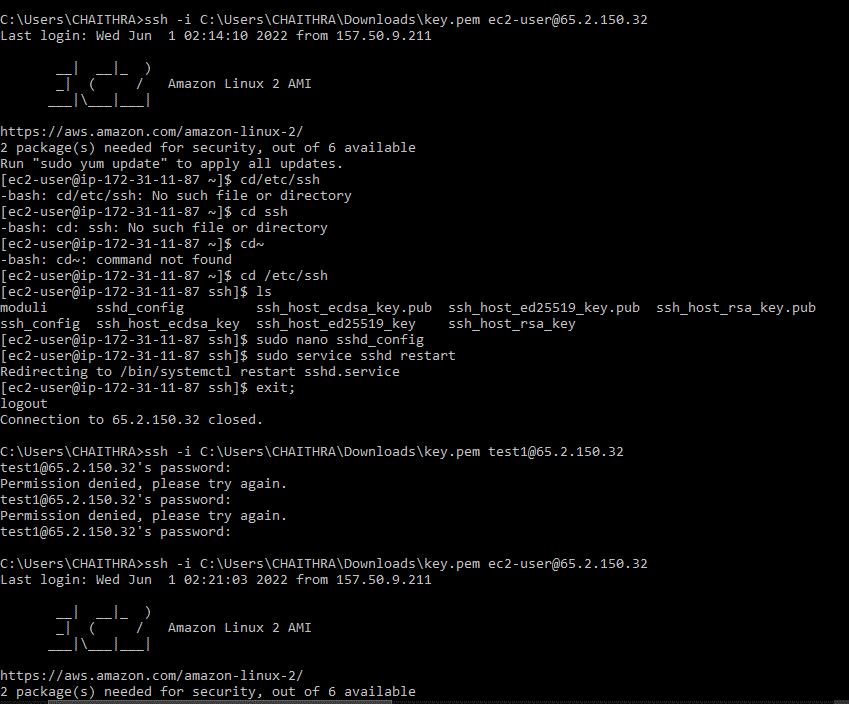


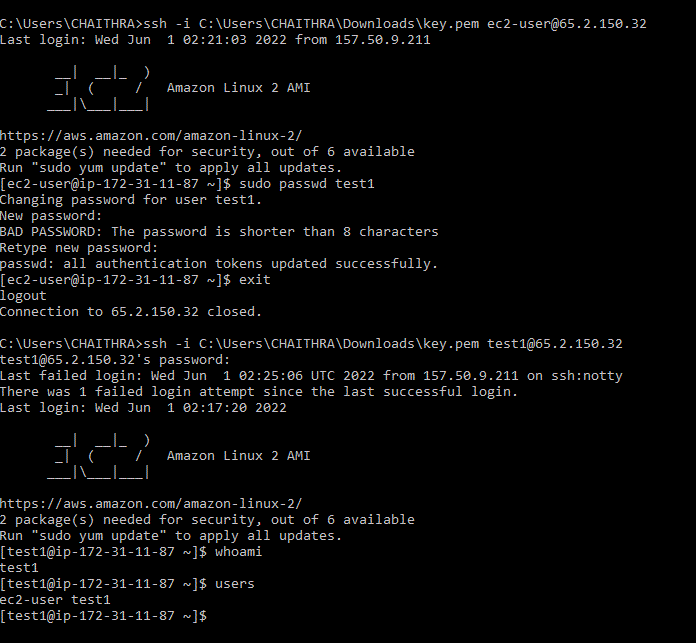


Assignments - 31th – May-2022

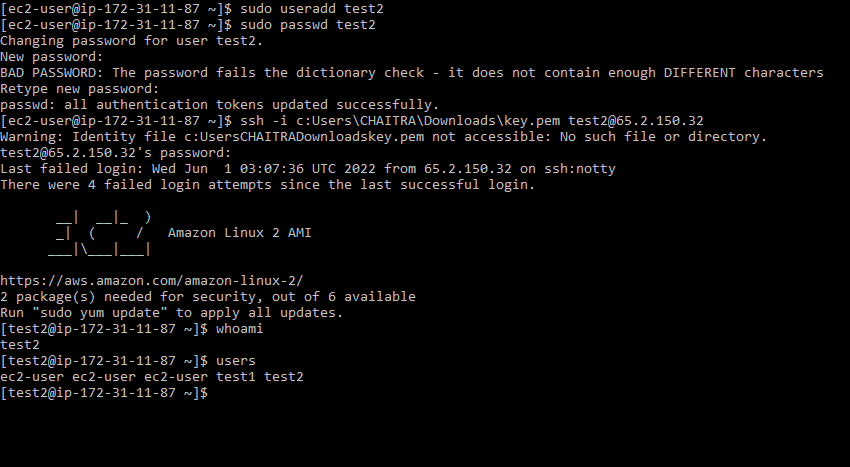
**Task:**

1. Create Users - test1 & test2?

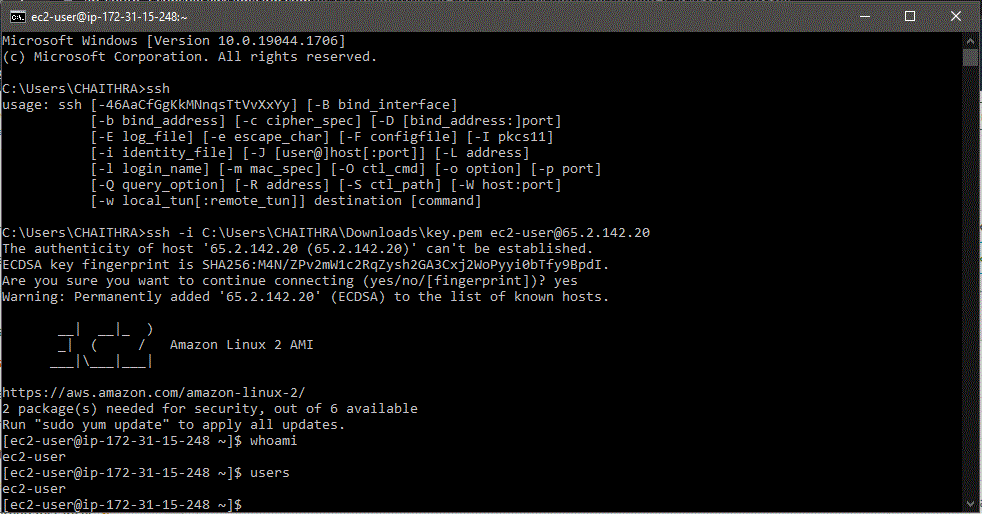


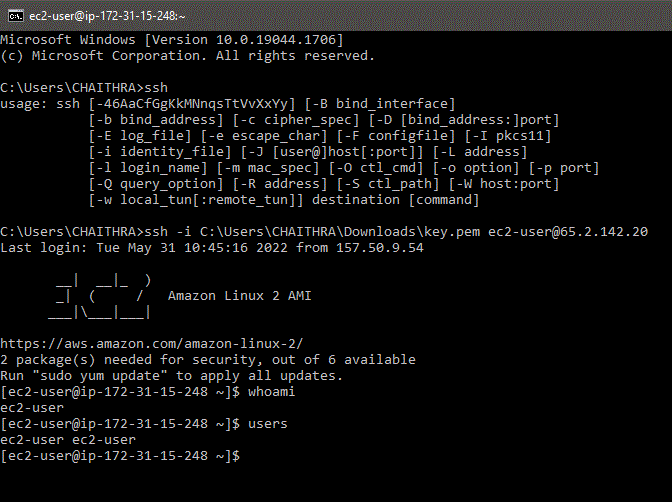


Test2

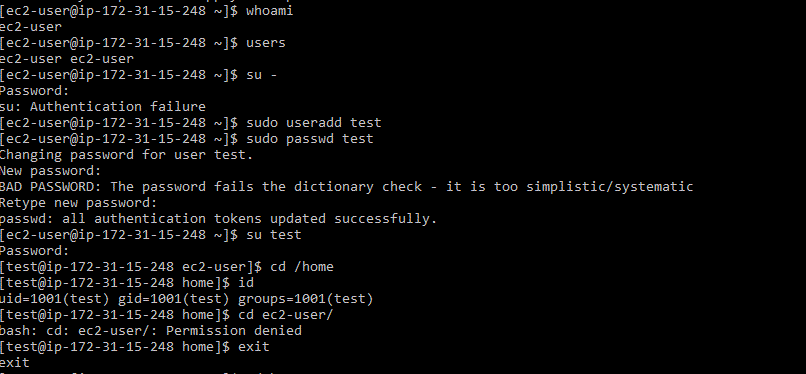


1. Login with these two credentials of the newly created users?

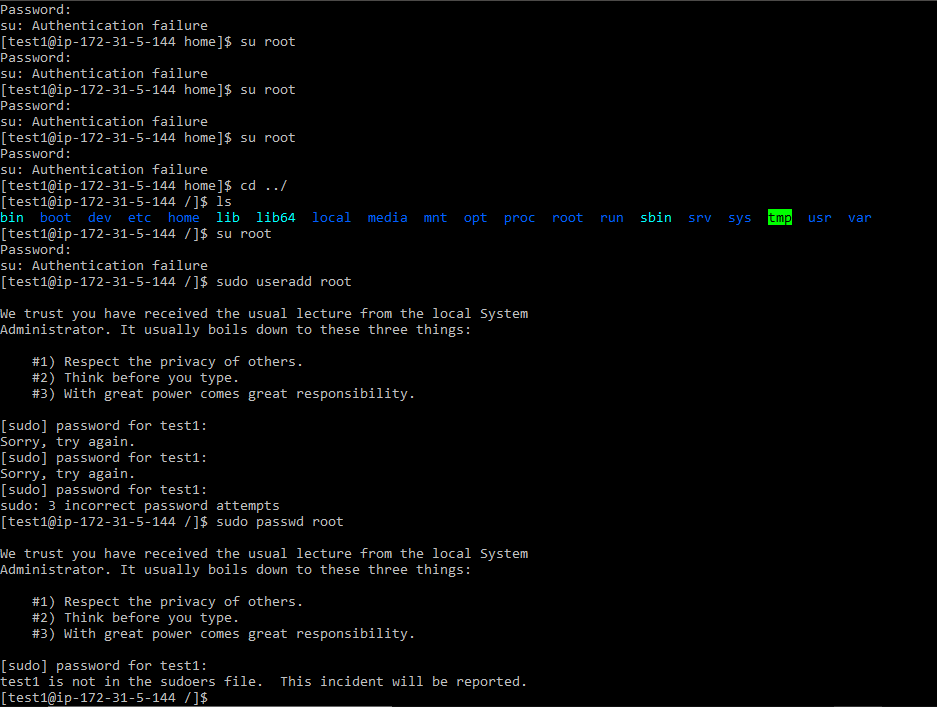


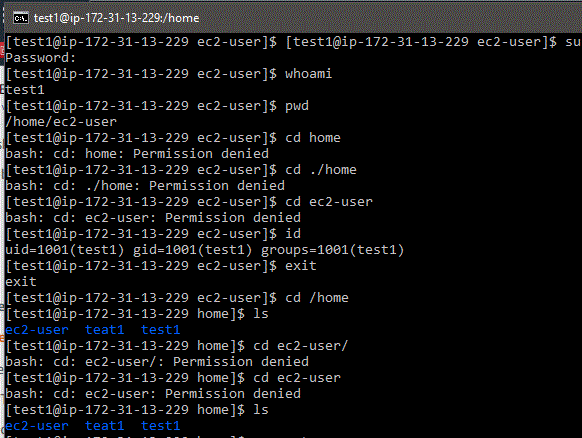


1. execute 'users' command - It will show the list of logged in users?

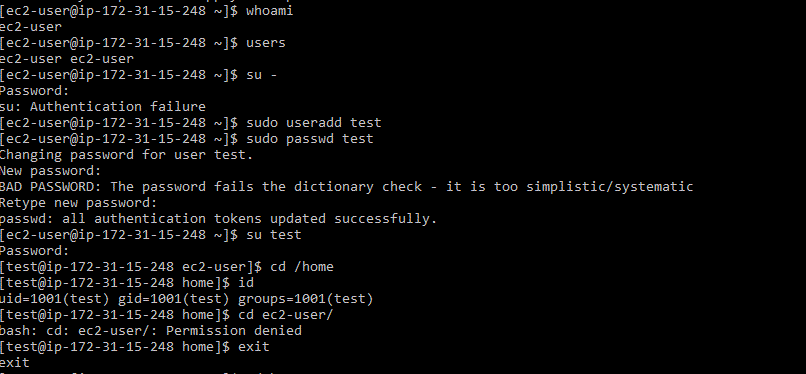


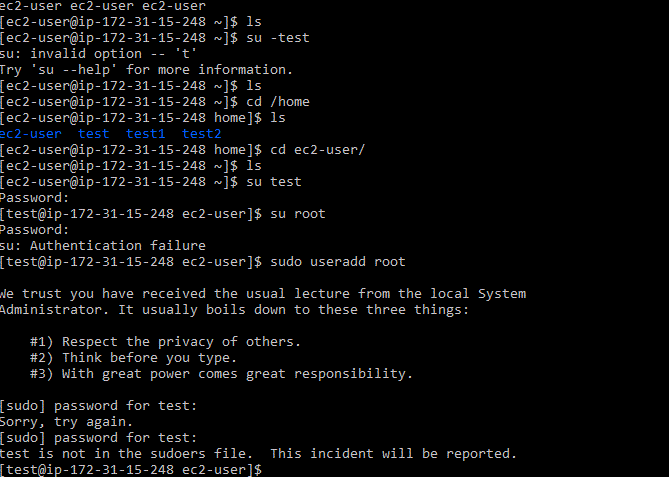
1. find the path for the sudoers users?



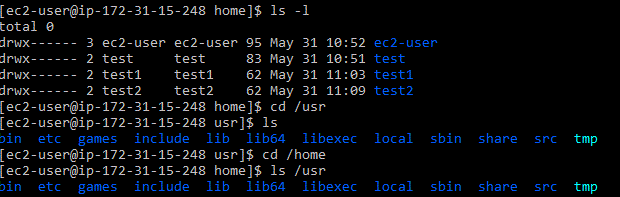


1. switch user to test /add user & password / id commends?

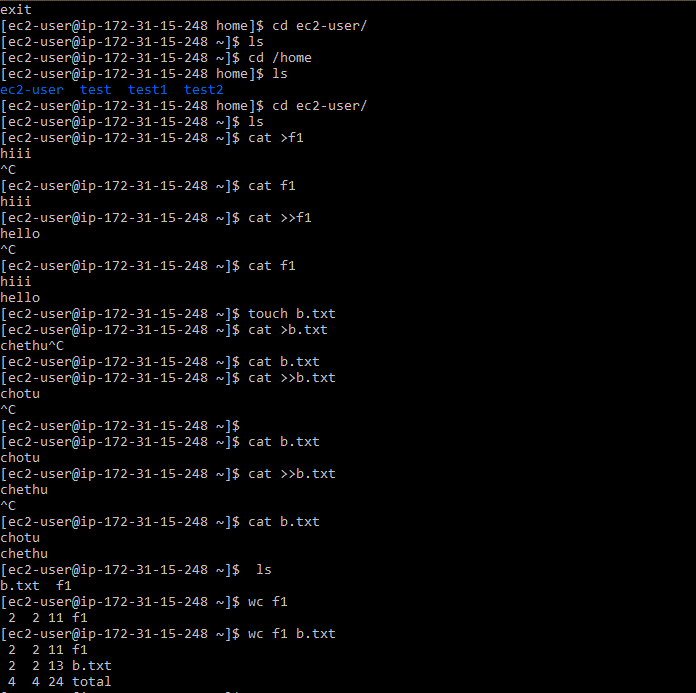


switch user to root /cd home /cd ec2-user?

1. ( ls -l ) / ( cd /usr ) / (ls / usr)?



1. Create file / wc command?



Assignments - 9th - Jun -2022

1) write a shell script that will add two number,which are supplled as command line argument

Requirement:

* the script should accept the input from the command line.
* if u don’t put input, then display an error message to execute the script correctly.

Hint:

* use $# - to check status:
* expressions to test file status:
* -f <file> file is an ordinary file

: nano sum .sh

echo $#

if [$# -eq 2]

then

echo sum : `expr $1 + $2`

else

echo “invalid argument pls provide 2 no”

fi

o/p :-

(1) sh sum.sh

invalid argument pls provide 2 no

(2) sh sum.sh 2 4

sum:6

2)write a script to determine whether given file exist or not?

* file name is supplied as command line argument
* also check for sufficient number of command line arguments

: (a) echo dir1

read dir1

if [ -d dir1 ]

then

echo " directory has been found "

else

echo " directory has not found "

or

if [ -f $ file ]

then

echo $ file is found

else

echo $ file is not found

Assignments - 10th - Jun -2022

1. Shell Script to print the reverse of a number?

: Source Code

read -p "Enter a number: " number

temp=$number

while [ $temp -ne 0 ]

do

reverse=$reverse$((temp%10))

temp=$((temp/10))

done

echo "Reverse of $number is $reverse"

Output

Enter a number: 123

Reverse of 123 is 321

temp%10 is used to obtain the last digit and then we concatenated it with reverse variable.

2)Shell script to find greatest of three numbers?

: Algorithm

1.Get three numbers.

Say num1, num2, num2

2. If (num1 > num2) and (num1 > num3)

echo value of num1

3. elif(num2 > num1) and (num2 > num3)

echo value of num2

4. Otherwise,

echo value of num3

input

echo "Enter Num1”

read num1

echo "Enter Num2"

read num2

echo "Enter Num3"

read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo "greter is" $num1

elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]

then

echo "greter is" $num2

else

echo"greter is" $num3

fi

Output

Enter Num1

1

Enter Num2

34

Enter Num3

2

greter is 34

3)shell script to find the age?

echo enter ur age

read age

if[ $age -ge 18 ]

then

echo "u can vote"

else

echo "u can't vote"

fi

4)shell script for finding the file?

echo enter ur file name

read test

if [-f $test]

then

echo"file found"

else

echo "file not found"

5) shell script to find parameter?

echo 1st per:$1

read $1

echo 2nd per: $2

read $2

echo total per:$#

6)to give user no?

echo $#

echo enter 1st no

read a

echo enter 2nd no

read b

echo sum:`expr $a + $b `

7)To do arithmetic programs?

echo enter 1st no:$2

echo enter 2nd no:$1

echo sum :`expr $1 + $2`

echo sub :`expr $1 - $2`

echo div :`expr $1 / $2`

echo mul :`expr $1 \\* $2`

Assignments - 1th - 7 -2022

1) what is conflict?

: Git can handle most merges on its own with automatic merging features.

A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other

. Conflicts will most likely happen when working in a team environment.

2) How to resolve conflicts?

: There are a few steps that could reduce the steps needed to resolve merge conflicts in Git: -

* The easiest way to resolve a conflicted file is to open it and make any necessary changes
* After editing the file, we can use the git add a command to stage the new merged content
* The final step is to create a new commit with the help of the git commit command
* Git will create a new merge commit to finalize the merge

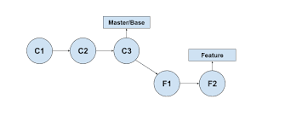
3) What are the 2 way & 3 way of merge?

: (A) 2 way merge - two-way merge An algorithm that merges two ordered files into one single sorted file.

* Two are the ones that are involved in a two-way merge

(B) 3 way merge - is where two changesets to one base file are merged as they are applied, as opposed to applying one, then merging the result with the other

* the third one is the base file or the common ancestor with which these two files will be compared.



4) what is the difference between git fetch and git pull?

**Git Fetch**

* is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn't do any file transferring.
* It's more like just checking to see if there are any changes available).

**Git Pull**

* on the other hand does that AND brings (copy) those changes from the remote repository.

