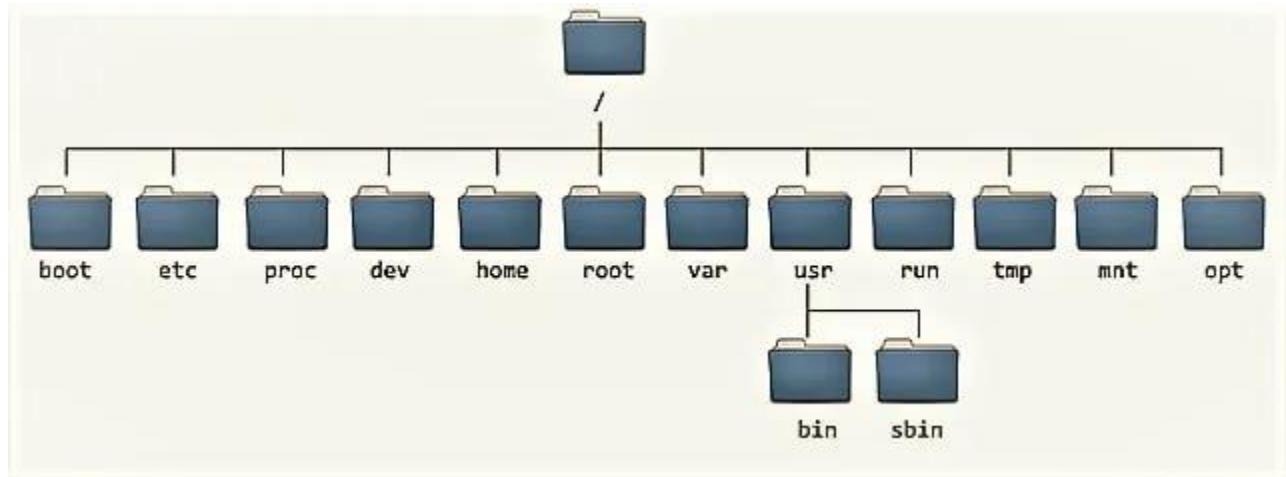


# Operating System

## Assignment – 01

Date-01/09/25

**Q1. Connect to the Linux server and understand the basic directory structure of Linux.**



### Common Linux Directories:

```
root@UbuntuVM:~# ls /
bin          dev   lib64           mnt   run           srv  var
bin.usr-is-merged  etc   lib.usr-is-merged  opt   sbin           sys
boot         home  lost+found       proc  sbin.usr-is-merged  tmp
cdrom        lib    media          root  snap           usr
root@UbuntuVM:~#
```

**(i) / (Root) -**

- The top-level directory of Linux.
- Everything starts from here.
- All other directories are inside /.

```
guest@UbuntuVM:~$ cd /
guest@UbuntuVM:$
```

## **(ii) /home (User Homes) -**

- Personal directories for each user.
- Example: /home/alex.
- Stores user files and settings.

```
guest@UbuntuVM:/$ cd home  
guest@UbuntuVM:/home$
```

## **(iii) /root (Root Home) -**

- Home directory of the root user.
- Similar to /home, but only for admin.
- Has full system control.

```
root@UbuntuVM:/home# cd /root  
root@UbuntuVM:~#
```

## **(iv) /bin (User Binaries) -**

- Contains essential commands for all users.
- Examples: ls, cp, cat.
- Needed for basic system use.

```
root@UbuntuVM:~# cd /bin  
root@UbuntuVM:/bin#
```

## **(v) /sbin (System Binaries) -**

- Holds system administration commands.
- Examples: shutdown, mount.
- Usually for root user.

```
root@UbuntuVM:/bin# cd /sbin  
root@UbuntuVM:/sbin#
```

## **(vi) /var (Variable Data) -**

- Stores frequently changing files.
- Examples: logs, mail, cache.
- Used by running services.

```
root@UbuntuVM:/sbin# cd /var  
root@UbuntuVM:/var#
```

## **(vii) /usr (User Programs) -**

- Contains installed software and libraries.
- Has subfolders like /usr/bin, /usr/lib.
- Similar to “Program Files” in Windows.

```
root@UbuntuVM:/var# cd /usr  
root@UbuntuVM:/usr#
```

## **(viii) /tmp (Temporary Files) -**

- Holds temporary data from apps.
- Usually cleared on reboot.
- World-writable for programs.

```
root@UbuntuVM:/usr# cd /tmp  
root@UbuntuVM:/tmp#
```

## **(ix) /etc (Configuration Files) -**

- Stores system-wide config files.
- Examples: /etc/passwd, /etc/hosts.
- Controls how the system runs.

```
root@UbuntuVM:/lib# cd /  
root@UbuntuVM:/# cd /etc  
root@UbuntuVM:/etc#
```

## (x) /lib (Libraries) -

- Stores shared libraries for binaries.
- Needed by /bin and /sbin programs.
- Like DLL files in Windows.

```
root@UbuntuVM:# cd /lib  
root@UbuntuVM:/lib# cd /
```

## (xi) /dev (Devices) -

- Represents hardware as files.
- Examples: /dev/sda, /dev/tty.
- Used to interact with devices.

```
root@UbuntuVM:# cd /dev  
root@UbuntuVM:/dev# cd ..
```

## (xii) /proc (Process Info) -

- Virtual filesystem for processes.
- Example: /proc/cpuinfo.
- Updates dynamically in real-time.

```
root@UbuntuVM:# cd /proc  
root@UbuntuVM:/proc# cd ..
```

## (xiii) /sys (System Info) -

- Provides info about devices and drivers.
- Works with the kernel.
- Useful for system tuning.

```
root@UbuntuVM:# cd /sys  
root@UbuntuVM:/sys# cd ..
```

## (xiv) /opt (Optional Software) -

- Used for third-party software.
- Keeps add-on apps separate.

- Example: /opt/google.

```
root@UbuntuVM:/# cd /opt
root@UbuntuVM:/opt#
```

## Q2. To understand help commands like man, info, help, what is, apropos.

- **man** – the manual pages (most common)

- Shows detailed documentation for commands, system calls, and files.
- Used to check syntax, options, and usage examples.
- Example: man ls opens the manual for the ls command..

```
MAN(1)                               Manual pager utils                               MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

    The table below shows the section numbers of the manual followed by the types of pages they contain.

    1  Executable programs or shell commands
    2  System calls (functions provided by the kernel)
    3  Library calls (functions within program libraries)
    4  Special files (usually found in /dev)
```

### EXAMPLES:

➤ **man ls**

Display the manual page for the item (program) ls.

➤ **man man. 7**

Display the manual page for macro package man from section 7.  
(This is an alternative spelling of "man 7 man".)

- **info** - GNU's hypertext manuals (often more tutorial-like)
  - Provides book-style, hyperlinked manuals (mainly for GNU tools).
  - Easier to read step-by-step explanations than man.
  - **Example:** info coreutils 'ls invocation' shows the ls section

```

LS(1)                               User Commands                               LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILEs (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
    do not ignore entries starting with .

  -A, --almost-all
    do not list implied . and ..

  --author
    with -l, print the author of each file

  -b, --escape
    print C-style escapes for nongraphic characters

  --block-size=SIZE
    with -l, scale sizes by SIZE when printing them; e.g.,

```

## Navigation keys inside `info`

- **h** - help tutorial · **Space** - next screen · **Backspace** - previous
- **n** - next node · **p** - previous · **u** - up one level · **d** - top directory
- **m** - choose a menu item · **s** - search · **q** - quit

- **help** - documentation for shell *builtins* (Bash/Zsh/etc.)
  - Displays information about shell built-in commands (like cd, echo).
  - Only works inside shells like Bash or Zsh.
  - **Example:** help cd shows how the cd builtin works.

```

ls --help
gu
Usage: date [-o[PTION]... [FORMAT]]
  or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display date and time in the given FORMAT.
With -s, or with [MMDDhhmm[[CC]YY][.ss]], set the date and time.

Mandatory arguments to long options are mandatory for short options too.
-d, --date=STRING          display time described by STRING, not 'now'
--debug                     annotate the parsed date,
                           and warn about questionable usage to stderr
-f, --file=DATEFILE        like --date; once for each line of DATEFILE
-I[FMT], --iso-8601[=FMT]   output date/time in ISO 8601 format.
                           FMT='date' for date only (the default),
                           'hours', 'minutes', 'seconds', or 'ns'
                           for date and time to the indicated precision.
                           Example: 2006-08-14T02:34:56-06:00
--resolution                output the available resolution of timestamps
                           Example: 0.00000001
-R, --rfc-email             output date and time in RFC 5322 format.
                           Example: Mon, 14 Aug 2006 02:34:56 -0600
--rfc-3339=FMT              output date/time in RFC 3339 format.
                           FMT='date', 'seconds', or 'ns'
                           for date and time to the indicated precision.
                           Example: 2006-08-14 02:34:56-06:00
-r, --reference=FILE        display the last modification time of FILE
-s, --set=STRING             set time described by STRING
-u, --utc, --universal      print or set Coordinated Universal Time (UTC)
--help                      display this help and exit
--version                   output version information and exit

```

- **what is** - one-line descriptions (quick glance)
  - Gives a one-line summary of a command or function.
  - Good for quickly checking what a command does.
  - Example: whatis grep → “grep (1) - print lines matching a pattern”.

```

root@UbuntuVM:/home# whatis
whatis what?
root@UbuntuVM:/home# whatis pwd
pwd (1)           : print name of current/working directory

```

- **Syntax** of the `whatis` command in Linux:

**whatis [option] [command\_name]**

- **[option]** = replace it with the desired option of yours.
- **[command\_name]** = replace it with the desired command you want.

- **apropos** - search by topic/keywords (most useful for discovery)
  - Searches manual page descriptions for a keyword.
  - Useful when you don't remember the exact command name.
  - **Example:** apropos network lists all commands related to networking.

```
guest@UbuntuVM:~$ apropos ls
proc_kallsyms (5)    - kernel exported symbols
_llseek (2)           - reposition read/write file offset
_Static_assert (3)   - fail compilation if assertion is false
aconnect (1)          - ALSA sequencer connection manager
add-shell (8)         - add shells to the list of valid login shells
afs_syscall (2)       - unimplemented system calls
alsa-info (8)         - command-line utility to gather information about the ALSA subsystem
alsabat (1)           - command-line sound tester for ALSA sound card driver
alsactl (1)           - advanced controls for ALSA soundcard driver
alsactl_init (7)      - alsactl management - initialization
alsaloop (1)          - command-line PCM loopback
alsamixer (1)         - soundcard mixer for ALSA soundcard driver, with ncurses interface
alsatplg (1)          - ALSA Topology Compiler
alsaucm (1)           - ALSA Use Case Manager
```

## Options:

**-d:** This option is used to emit debugging messages. When this option is used then terminal returns man directories, global path, path directory, warnings, etc. of each command which is related to the search keyword.

```
sc@ubuntu:~$ apropos -d email
From the config file /etc/manpath.config:

Mandatory mandir '/usr/man'.
Mandatory mandir '/usr/share/man'.
Mandatory mandir '/usr/local/share/man'.
Path '/bin' mapped to mandir '/usr/share/man'.
Path '/usr/bin' mapped to mandir '/usr/share/man'.
Path '/sbin' mapped to mandir '/usr/share/man'.
Path '/usr/sbin' mapped to mandir '/usr/share/man'.
Path '/usr/local/bin' mapped to mandir '/usr/local/man'.
Path '/usr/local/bin' mapped to mandir '/usr/local/share/man'.
Path '/usr/local/sbin' mapped to mandir '/usr/local/man'.
Path '/usr/local/sbin' mapped to mandir '/usr/local/share/man'.
Path '/usr/X11R6/bin' mapped to mandir '/usr/X11R6/man'.
Path '/usr/bin/X11' mapped to mandir '/usr/X11R6/man'.
Path '/usr/games' mapped to mandir '/usr/share/man'.
Path '/opt/bin' mapped to mandir '/opt/man'.
```

**-v:** This option is used to print verbose warning messages.

**-e, --exact:** This option is used to search each keyword for exact match. If no option is used, the apropos command returns list of all the command whose description in the man page description matches with the keyword or which are somehow related to the keyword given in the argument. However, when the -e option is used, the apropos only returns the command whose description exactly matches with the keyword.

```
sc@ubuntu:~$ apropos -e compress
bzexe (1)           - compress executable files in place
gunzip (1)          - compress or expand files
gzexe (1)          - compress executable files in place
gzip (1)           - compress or expand files
lzcat (1)          - Compress or decompress .xz and .lzma files
lzma (1)           - Compress or decompress .xz and .lzma files
mscompress (1)      - compress data using LZ77 algorithm
msexpand (1)        - decompress data compressed using mscompress(1) or COMP...
uncompress (1)      - compress or expand files
```

**-w, --wildcard:** This option is used when the keyword(s) contain wildcards. apropos will independently search the page name and the description matching against the keyword(s). All the command related to sudo are listed when sudo is given as a wildcard.

```
sc@ubuntu:~$ apropos -w sudo
sudo (8)           - execute a command as another user
sudo.conf (5)       - configuration for sudo front end
sudo_plugin (8)     - Sudo Plugin API
sudoers (5)         - default sudo security policy plugin
sudoreplay (8)      - replay sudo session logs
sc@ubuntu:~$ █
```

**-a, --and:** This option is used when we want all the keywords to match. It returns nothing if any one of the keywords supplied has no matching in the man page or description. In the below input, two keywords have been given and only two commands are displayed in the result since there is only one command that contains both the keyword.

```
sc@ubuntu:~$ apropos -a email send
nautilus-sendto (1) - convenience application to send a file via email or in...
xdg-email (1)       - command line tool for sending mail using the user's pr...
sc@ubuntu:~$ █
```

**-l, --long:** By default, the output is trimmed to the terminal width. This option becomes handy when we don't want the result to be truncated.

```
sc@ubuntu:~$ apropos -l email
Email::Valid (3pm)  - Check validity of Internet email addresses
Mail::Internet (3pm) - manipulate email messages
mailto.conf (5)      - configuration file for cups email notifier
nautilus-sendto (1) - convenience application to send a file via email or instant messenger
```

### Q3. To understand basic directory navigation commands like cat, cd, mv, cp, rm, mkdir, rmdir, file and pwd commands.

- **cat (concatenate and view files)**

- Used to display contents of a file on the terminal.
- Can also join multiple files together.
- **Example:** cat file.txt

```
root@UbuntuVM:/home# cd /
root@UbuntuVM:# cat > demo
This is a unix file
root@UbuntuVM:# cat demo
This is a unix file
root@UbuntuVM:# cat demo > democpy
root@UbuntuVM:# cat democpy
This is a unix file
root@UbuntuVM:# cat > file1
Contains text data
root@UbuntuVM:# cat demo file1 >> democpy
root@UbuntuVM:# cat democpy
This is a unix file
This is a unix file
Contains text data
```

- **cd (change directory)**

- Copies files or directories.
- Needs -r option to copy folders.
- **Example:** cp file.txt backup.txt

```
root@UbuntuVM:# cd /bin
root@UbuntuVM:/bin# cd ..
root@UbuntuVM:# ls
bin          demo    file1  lib usr-is-merged opt   sbin           sys
bin.usr-is-merged democpy home  lost+found      proc  sbin.usr-is-merged tmp
boot         dev     lib   media        root  snap           usr
cdrom        etc    lib64  mnt        run   srv            var
root@UbuntuVM:# cd /democpy
bash: cd: /democpy: Not a directory
root@UbuntuVM:# cd /home
root@UbuntuVM:/home# cd ..
root@UbuntuVM:# cd /bi
bash: cd: /bi: No such file or directory
root@UbuntuVM:# cd /bin
root@UbuntuVM:/bin# cd ..
root@UbuntuVM:# cd /home
root@UbuntuVM:/home# cd ..
root@UbuntuVM:#
```

## • mv (move/rename files)

- Moves files/directories from one location to another.
- Also used to rename files.
- **Example:** mv old.txt new.txt

```
root@UbuntuVM:/home# cd demodr1
root@UbuntuVM:/home/demodr1# cat > new
abcd
root@UbuntuVM:/home/demodr1# cd ..
root@UbuntuVM:/home# mv demodr1/new demodr2
root@UbuntuVM:/home# cd demodr2
root@UbuntuVM:/home/demodr2# ls
new
root@UbuntuVM:/home/demodr2# cat new
abcd
root@UbuntuVM:/home/demodr2# mv new old
root@UbuntuVM:/home/demodr2# ls
old
root@UbuntuVM:/home/demodr2#
```

## • cp (copy files/directories)

- Copies files or directories.
- Needs -r option to copy folders.
- **Example:** cp file.txt backup.txt

```
root@UbuntuVM:/home# cat > f1
1234
root@UbuntuVM:/home# cat > f2
56789
root@UbuntuVM:/home# cp f1 f2
root@UbuntuVM:/home# cp f1 f01
root@UbuntuVM:/home# ls
Admin demo_dir demodr1 demodr2 f01 f1 f2 guest
root@UbuntuVM:/home# cat f1
1234
root@UbuntuVM:/home# cat f2
1234
root@UbuntuVM:/home#
```

## • rm (remove files/directories)

- Removes an empty directory only.
- Fails if the folder has files inside.
- **Example:** rmdir old\_folder

```
root@UbuntuVM:/home/demodr2# ls
old
root@UbuntuVM:/home/demodr2# cp old temp
root@UbuntuVM:/home/demodr2# rm temp
root@UbuntuVM:/home/demodr2# cp old l2
root@UbuntuVM:/home/demodr2# ls
l2 old
root@UbuntuVM:/home/demodr2# cd ..
root@UbuntuVM:/home# rm -R demodr2
root@UbuntuVM:/home# ls
Admin demo_dir demodr1 f01 f1 f2 guest
root@UbuntuVM:/home#
```

## • **mkdir (make directory)**

- Removes an empty directory only.
- Fails if the folder has files inside.
- **Example:** rmdir old\_folder

```
root@UbuntuVM:/home# mkdir demo_dir
root@UbuntuVM:/home# mkdir demodr1 demodr2
root@UbuntuVM:/home# ls
Admin  demo_dir  demodr1  demodr2  guest
root@UbuntuVM:/home# mkdir demodr1/akhil
```

## • **file (identify file type)**

- Identifies the type of a file.
- Useful to check text, image, binary, etc.
- **Example:** file photo.jpg

⊕ ll - ll is usually an alias for ls -l that shows a long listing of files and direct with detailed information

```
[paul@centos01 perm]$
[paul@centos01 perm]$ ll
total 8
drwxr--r--. 2 root root 22 Aug  4 11:29 ndir
drwxr-xr-x. 2 root root 22 Aug  4 11:29 rdir
-rw-r--r--. 1 root root 23 Aug  4 11:26 rfile.txt
-rwxr--r--. 1 paul paul 31 Aug  4 11:49 script.sh
[paul@centos01 perm]$
```

⊕ ls -l - It lists files and directories in long format, showing details like permissions, owner, group, size, and modification time.

```
[paul@centos01 perm]$
[paul@centos01 perm]$ ll
total 12
drwxr--r--. 2 root root 22 Aug  4 11:29 ndir
-rw-----. 1 root root  6 Aug  4 12:16 nfile.txt
drwxr-xr-x. 2 root root 22 Aug  4 11:29 rdir
-rw-r--r--. 1 root root 23 Aug  4 11:26 rfile.txt
-rw-r--r--. 1 paul paul 31 Aug  4 11:49 script.sh
[paul@centos01 perm]$
[paul@centos01 perm]$ cat nfile.txt
cat: nfile.txt: Permission denied
[paul@centos01 perm]$
[paul@centos01 perm]$ vi rfile.txt
[paul@centos01 perm]$ cd ndir/
bash: cd: ndir/: Permission denied
[paul@centos01 perm]$ █
```

- **chmod - chmod** (change mode) is used to change the permissions (read, write, execute) of a file or directory.

```
[paul@centos01 perm]$ 
[paul@centos01 perm]$ 
[paul@centos01 perm]$ ll
total 8
drwxr--r--. 2 root root 22 Aug  4 11:29 ndir
drwxr-xr-x. 2 root root 22 Aug  4 11:29 rdir
-rw-r--r--. 1 root root 23 Aug  4 11:26 rfile.txt
-rw-r--r--. 1 paul paul 31 Aug  4 11:49 script.sh
[paul@centos01 perm]$ 
[paul@centos01 perm]$ chmod a+rwx rfile.txt
chmod: changing permissions of 'rfile.txt': Operation not permitted
[paul@centos01 perm]$ 
[paul@centos01 perm]$ chmod u+x script.sh
[paul@centos01 perm]$ 

[paul@centos01 perm]$ 
[paul@centos01 perm]$ mkdir testdir
[paul@centos01 perm]$ cd testdir/
[paul@centos01 testdir]$ touch file1 file2 file3
[paul@centos01 testdir]$ ll
total 0
-rw-r--r--. 1 paul paul 0 Aug  4 14:27 file1
-rw-r--r--. 1 paul paul 0 Aug  4 14:27 file2
-rw-r--r--. 1 paul paul 0 Aug  4 14:27 file3
[paul@centos01 testdir]$ cd 

[paul@centos01 perm]$ 
[paul@centos01 perm]$ chmod u=rwx,g=rw,o=r script.sh
[paul@centos01 perm]$ ll
total 8
drwxr--r--. 2 root root 22 Aug  4 11:29 ndir
drwxr-xr-x. 2 root root 22 Aug  4 11:29 rdir
-rw-r--rw-. 1 root root 32 Aug  4 14:24 rfile.txt
-rwxrw-r--. 1 paul paul 31 Aug  4 11:49 script.sh
[paul@centos01 perm]$ 

[paul@centos01 perm]$ 
[paul@centos01 perm]$ chmod -R u=rwx testdir/
[paul@centos01 perm]$ 
[paul@centos01 perm]$ ls testdir/
file1 file2 file3
[paul@centos01 perm]$ ls -l testdir/
total 0
-rwxr--r--. 1 paul paul 0 Aug  4 14:27 file1
-rwxr--r--. 1 paul paul 0 Aug  4 14:27 file2
-rwxr--r--. 1 paul paul 0 Aug  4 14:27 file3
[paul@centos01 perm]$ 
```

## • **rmdir (remove empty directory)**

- Removes an empty directory only.
- Fails if the folder has files inside.
- **Example:** `rmdir old_folder`

```
root@UbuntuVM:/home# ls
Admin demo_dir demodr1 f01 f1 f2 guest
root@UbuntuVM:/home# rmdir demo_dir
root@UbuntuVM:/home# ls
Admin demodr1 f01 f1 f2 guest
```

## • **pwd (print working directory)**

- Prints the current working directory path.
- Helps know where you are in the system.
- **Example:** `pwd`

```
root@UbuntuVM:/home# pwd
/home
root@UbuntuVM:/home# cd ..
root@UbuntuVM:/# pwd
```