Linux Tutorial

70 Key Linux Commands

You Should Know

Overview of the commands:

Command	Description
ls	Lists directory contents, including files and folders.
cd	Changes the current directory to another one.
pwd	Displays the current working directory's full path.
cat	Concatenates and displays file content.
vi	Opens the vi editor for text file creation or editing.
cp	Copies files and directories to a new location.
mv	Moves or renames files and directories.
mkdir	Creates new directories.
touch	Creates a new empty file or updates existing file's timestamp.
less	Views content of large files in a paginated way.

Command	Description
tail	Displays the end of a file's content.
more	Paginates file content for easy reading.
chmod	Changes file access permissions.
chown	Changes file or directory ownership.
diff	Compares files and shows differences.
dd	Converts and copies file data.
ln	Creates links to files and directories.
rm	Removes files and directories.
zip	Compresses files into zip format.
tar	Archives multiple files together.
wc	Counts lines, words, and characters in files.
stat	Displays detailed file or file system status.
uptime	Shows current uptime of the system.
uname	Displays system information.
free	Shows memory usage statistics.
shutdown	Shuts down or reboots the system.
apt	Manages packages on Debian-based systems.
sudo	Executes commands with superuser privileges.
du	Estimates file space usage.
df	Reports file system disk space usage.
lsblk	Lists block devices and their information.
blkid	Shows block device attributes.
date	Displays or sets the system date and time.
hostname	Shows or sets the system's network name.
fdisk	Manages disk partitions.
mount	Attaches filesystems to a directory tree.
systemctl	Manages systemd services and system state.
vmstat	Reports virtual memory statistics.
echo	Displays input as output.
history	Lists previously used commands.
passwd	Changes user passwords.
usermod	Modifies user account properties.

Command	Description
useradd	Adds a new user to the system.
userdel	Deletes a user account from the system.
groupadd	Creates a new user group.
w	Displays logged-in users and their activities.
ssh	Connects to remote servers securely.
top	Monitors real-time process activity.
ps	Displays current processes.
kill	Terminates processes.
fuser	Identifies processes using files or sockets.
lsof	Lists open files and their processes.
ip	Manages network interfaces and routes.
ping	Tests network connectivity.
traceroute	Traces network path to a destination host.
wget	Downloads files from the internet.
scp	Securely transfers files between hosts.
rsync	Synchronizes files between locations.
ss	Displays socket statistics.
nmap	Scans network for devices and open ports.
find	Searches for files and directories.
grep	Searches text using patterns.
tr	Translates or deletes characters.
cut	Extracts sections from each line of files.
uniq	Filters out repeated lines in a file.
tee	Reads from stdin and writes to stdout and files.
awk	Processes and analyzes text data.
sed	Edits text in a stream.
sort	Sorts lines of text files.
xargs	Builds and executes command lines.

Linux Commands that are very commonly used for **NAVIGATION**.

1. ls

The ls command is used to list the contents of directories.

Simply typing ls in the command line and pressing Enter will list the contents of the current directory.

Command:

```
ls
```

You can specify a path to list the contents of a different directory, like so: ls /path/to/directory.

Some of its most useful options:

- `ls -l` : Provides a detailed list of files including the file's type and its permissions.
- `ls -a` : Lists all files, including hidden files.
- `ls -ltr` : Shows a detailed list, sorted by modification time in reverse order (oldest first).
- `ls -d` : List directory information instead of the contents of the directory.

2. cd

The cd command is used to change the current working directory in the command-line interface.

For example to change the working directory to /var, type:

```
cd /var
```

```
ubuntu@Linuxopsys:~$ cd /var
ubuntu@Linuxopsys:/var$ ls
backups  crash  local  log    metrics  run   spool  www
cache    lib    lock   mail   opt      snap  tmp
ubuntu@Linuxopsys:/var$
```

Here we have used absolute path ie start with the root directory `/`. You may also use relative path that is based on the current working directory.

Its usage is straightforward, but there are a few options and shortcuts that can be handy:

- `cd ..` : Moves one directory up in the hierarchy (to the parent directory).
- `cd -` : Changes the directory to the previous working directory. This is useful for toggling between two directories.
- `cd ~` : Takes you to the home directory of the current user.
- `cd` : Running the command with no arguments will change the working directory to the current user's home directory.

3. pwd

The pwd is a command-line utility that displays the full pathname of the current working directory.

Command:

```
pwd
```

```
ubuntu@Linuxopsys:~$ pwd
/home/ubuntu
ubuntu@Linuxopsys:~$
```

This command will return the full path to your current directory, such as "/home/username/."

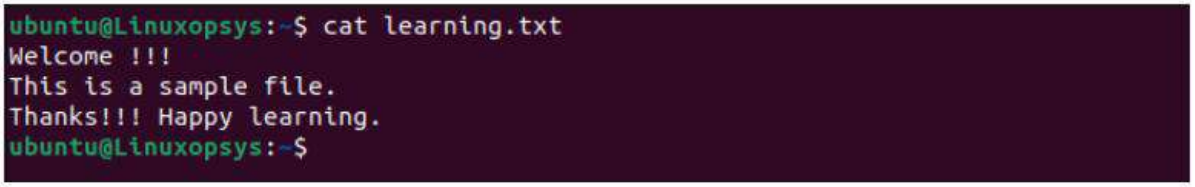
Commands used for **FILE** management:

4. cat

The cat command-line utility displays the content of files to the standard output (usually, the screen). Its primary use is to read and concatenate files, displaying their contents, but it's also used for creating new files and appending content to existing ones.

Example:

```
cat learning.txt
```



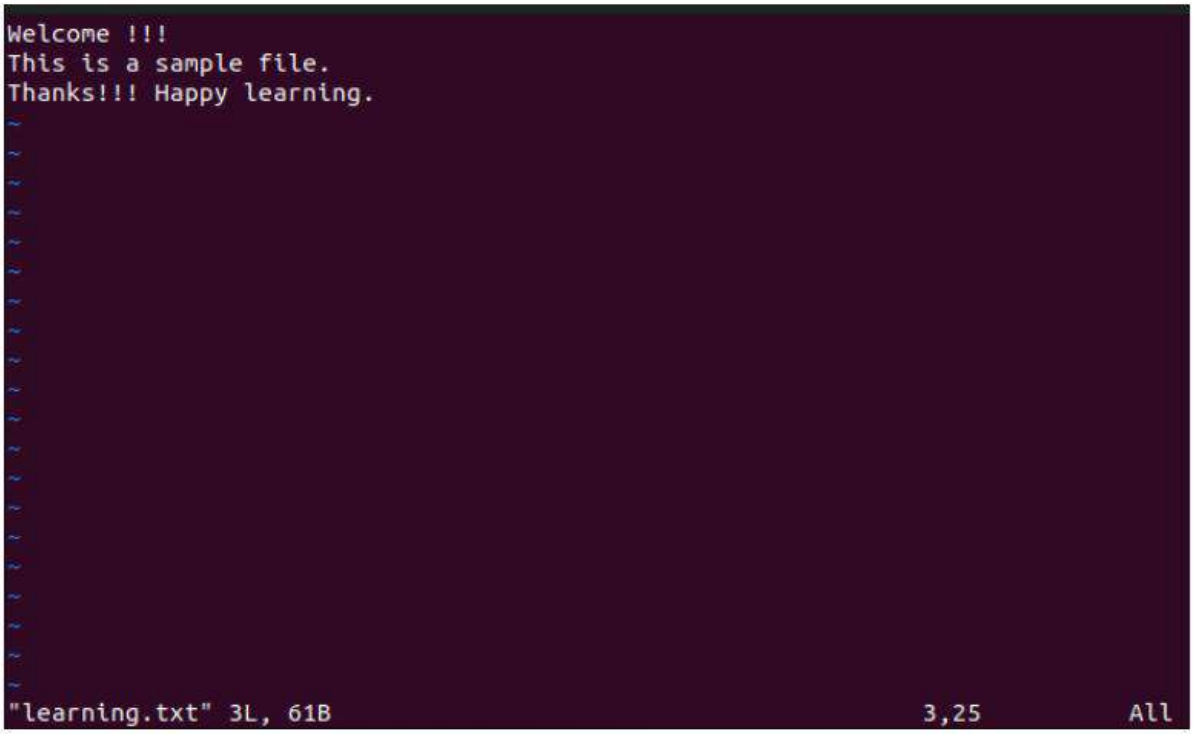
This command will print the content of "learning.txt" to your screen.

5. vi

The vi is a highly popular text editor in Unix-based systems, known for its powerful features in editing plain text, like configuration files and program code. Its primary use is to create, view, or edit text files directly from the command line or within a scripting environment.

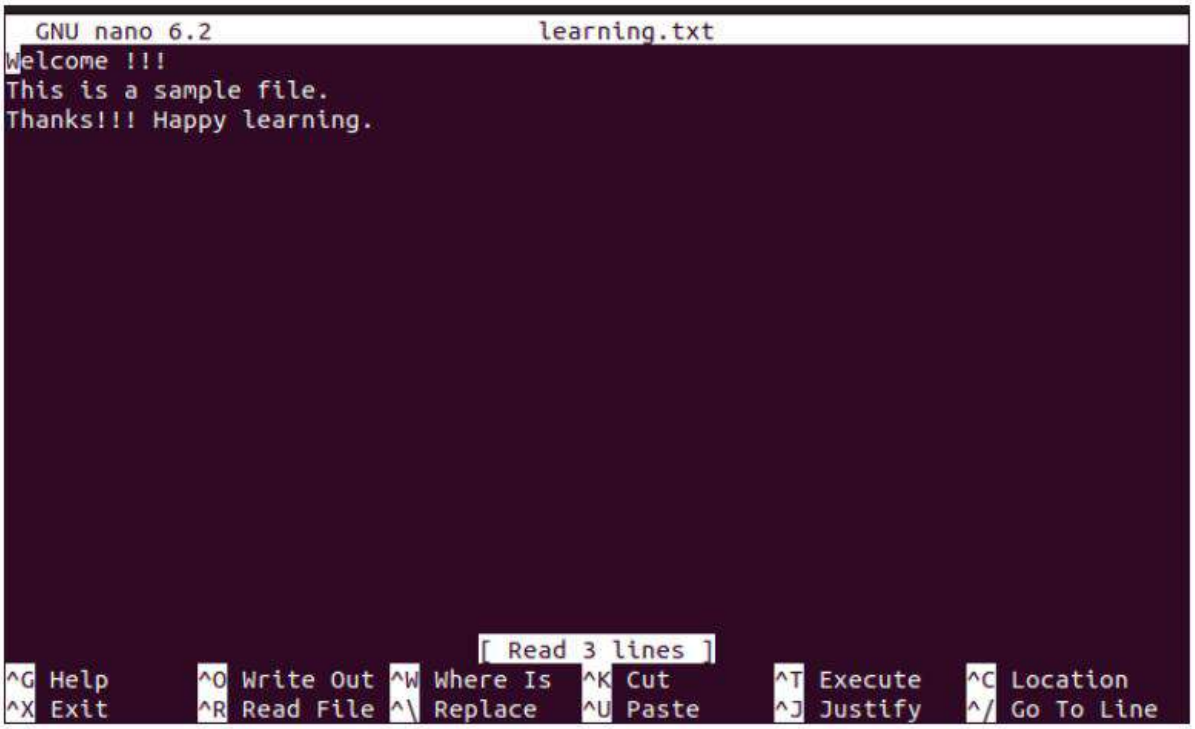
Example:

```
vi learning.txt
```



This command opens "learning.txt" in vi, and if the file doesn't exist, vi will create it upon saving and exiting (:wq).

Some are fans of nano which is a user-friendly text editor that operates in a command line interface which is simpler but has fewer features.



6. cp

The cp command is used to copy files and directories from one location to another within the file system.

Example:

```
cp file1.txt dir1/
```

```
ubuntu@Linuxopsys:~$ cp file1.txt dir1/
ubuntu@Linuxopsys:~$ ls dir1/
file1.txt
ubuntu@Linuxopsys:~$
```

This copies a file named "file1.txt" from the current directory to another directory called "backup".

7. mv

The mv command is used to move or rename files and directories within the filesystem. Its primary use is relocating files or directories and renaming them, without creating a copy.

Example:

```
mv oldname.txt newname.txt
```

```
ubuntu@Linuxopsys:~$ mv oldname.txt newname.txt
ubuntu@Linuxopsys:~$ ls oldname.txt
ls: cannot access 'oldname.txt': No such file or directory
ubuntu@Linuxopsys:~$ ls newname.txt
newname.txt
ubuntu@Linuxopsys:~$
```

This command rename a file from "oldname.txt" to "newname.txt" in the same directory.

8. mkdir

The mkdir command is used to create new directories or folders within the file system.

Command:

```
mkdir dir1
```

9. touch

The touch command is used to create new, empty files and to change the timestamps (access, modify, and change times) of existing files.

Example:

```
touch file1.txt
```

This command creates an empty file named "file1.txt" if it doesn't exist or updates its modification and access times if it does.

10. less

The less command is used for viewing large text files, providing forward and backward navigation, without needing to load the entire file into memory.

Example: To view a file named "/var/log/dmesg" you would use:

```
less /var/log/dmesg
```

```
[    0.000000] kernel: Linux version 6.2.0-34-generic (buildd@bos03-amd
64-059) (x86_64-linux-gnu-gcc-11 (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0,
 GNU ld (GNU Binutils for Ubuntu) 2.38) #34~22.04.1-Ubuntu SMP PREEMPT_
DYNAMIC Thu Sep  7 13:12:03 UTC 2 (Ubuntu 6.2.0-34.34~22.04.1-generic 6
.2.16)
[    0.000000] kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-6.2.0-34-
generic root=UUID=418c9675-3b58-4ec5-a83c-f2dbead4a371 ro quiet splash
[    0.000000] kernel: KERNEL supported cpus:
[    0.000000] kernel:   Intel GenuineIntel
[    0.000000] kernel:   AMD AuthenticAMD
[    0.000000] kernel:   Hygon HygonGenuine
[    0.000000] kernel:   Centaur CentaurHauls
[    0.000000] kernel:   zhaoxin   Shanghai
[    0.000000] kernel: BIOS-provided physical RAM map:
[    0.000000] kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000
9fbff] usable
[    0.000000] kernel: BIOS-e820: [mem 0x000000000009fc00-0x00000000000
9ffff] reserved
/var/log/dmesg
```

This command opens "/var/log/dmesg" in the less program, allowing you to navigate the file with arrow keys, page up/down, or search for specific text with "/" followed by your query.

11. tail

The tail command outputs the last part of files. By default, it shows the last ten lines, but its behavior can be modified with various options.

Example: To display the last 12 lines of a file named "/var/log/dmesg" you would use:

```
tail -n 12 /var/log/dmesg
```

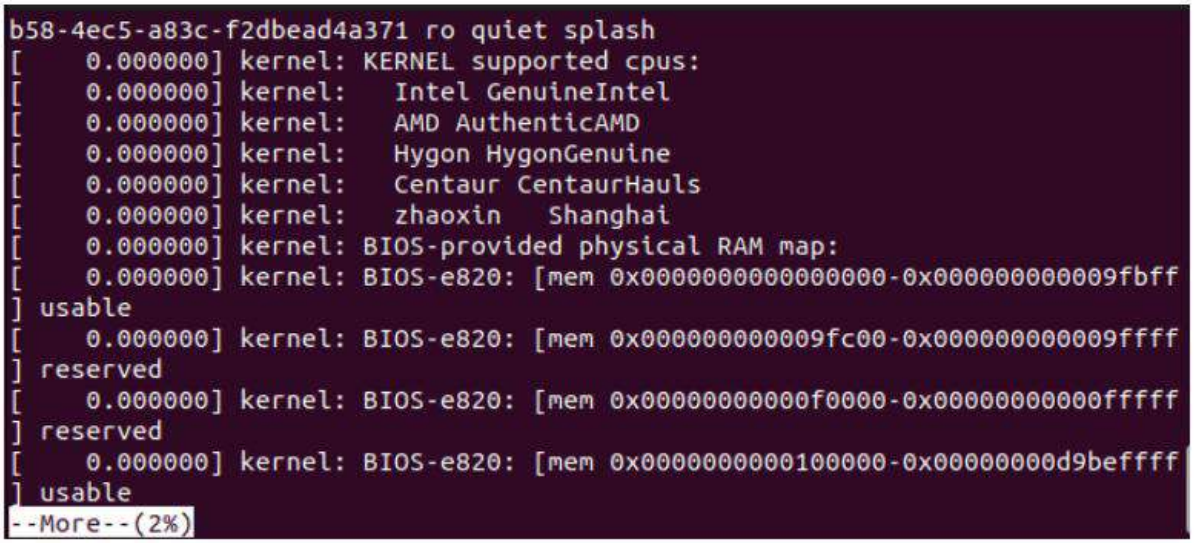
```
[    5.425993] kernel: audit: type=1400 audit(1697490993.516:35): appar
ile_load" profile="unconfined" name="snap-update-ns.snapd-desktop-integ
or_parser"
[    5.443228] kernel: audit: type=1400 audit(1697490993.532:36): appar
ile_load" profile="unconfined" name="snap.firefox.firefox" pid=658 comm
[    5.459109] kernel: audit: type=1400 audit(1697490993.548:37): appar
ile_load" profile="unconfined" name="snap.firefox.geckodriver" pid=668
[    5.465972] kernel: audit: type=1400 audit(1697490993.556:38): appar
ile_load" profile="unconfined" name="snap.firefox.hook.configure" pid=6
[    7.003903] kernel: e1000: enp0s3 NIC Link is Up 1000 Mbps Full Dupl
[    7.009090] kernel: IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becc
[    7.039489] kernel: e1000: enp0s8 NIC Link is Up 1000 Mbps Full Dupl
[    7.039773] kernel: IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becc
[    7.071459] kernel: e1000: enp0s9 NIC Link is Up 1000 Mbps Full Dupl
[    7.071771] kernel: IPv6: ADDRCONF(NETDEV_CHANGE): enp0s9: link becc
[    8.700543] kernel: loop19: detected capacity change from 0 to 8
ubuntu@Linuxopsys:~$
```

12. more

The more command is used to display the contents of a text file one screen at a time.

Example: To view a file named "/var/log/dmesg," you would use:

```
more /var/log/dmesg
```



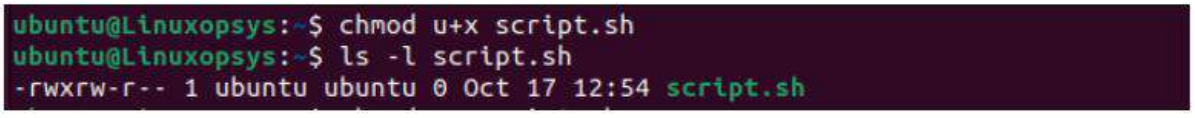
This command opens "/var/log/dmesg" and displays its content one screen at a time, moving to the next screen with the spacebar or the enter key.

13. chmod

The chmod command is used to change the access permissions of file system objects (files and directories). Its primary use is to define who can read, write, or execute a file.

Example: To give the user execute permission on a script named "script.sh," you would use:

```
chmod u+x script.sh
```



While `chmod 755 filename` set octal values to set read, write, and execute permissions for the owner, and read and execute permissions for the group and others.

The -R option apply permissions recursively.

14. chown

The chown command is used to change the ownership of files and directories in the file system.

Example:

```
chown -R ubuntu:developers dir1
```

```
ubuntu@Linuxopsys:~$ sudo chown -R ubuntu:developers dir1
ubuntu@Linuxopsys:~$ ls -ld dir1
drwxrwxr-x 2 ubuntu devs 4096 Oct 17 10:01 dir1
ubuntu@Linuxopsys:~$
```

This command recursively (-R) changes the owner to "ubuntu" and the group to "developers" for "dir1" and all files and directories within it.

15. diff

The diff command is used for comparing different files or directories, displaying the differences between them. It comes particularly useful in tasks like code reviews or checking configuration changes.

Example:

```
diff file1.txt file2.txt
```

This command will output the lines where "file1.txt" and "file2.txt" differ.

The `-r` option allows recursive comparison and -q report only when files differ.

16. dd

The dd command is used for converting and copying files at the byte level. Its primary use encompasses tasks like backing up and restoring an entire disk or partition, copying regions of raw device files, converting data formats, and creating bootable drives.

Example: To create a bootable USB drive from an ISO image "linux-distro.iso," you would use:

```
dd if=linux-distro.iso of=/dev/sdx bs=4M status=progress
```

Must exercise caution with dd as it can overwrite data irreversibly.

17. ln

The ln command creates symbolic links (symlink for short or soft links) and hard links in Linux.

Example of creating a soft link:

```
sudo ln -s /opt/cleanup.sh  /bin/cleanup
```

Useful ln commands and options

- ln example.txt ~/example.txt - creates a hard link to a file example.txt
- ln -sf /opt/cleanup.sh /bin/cleanup - remove the existing destination file if it already exists and create a new link.

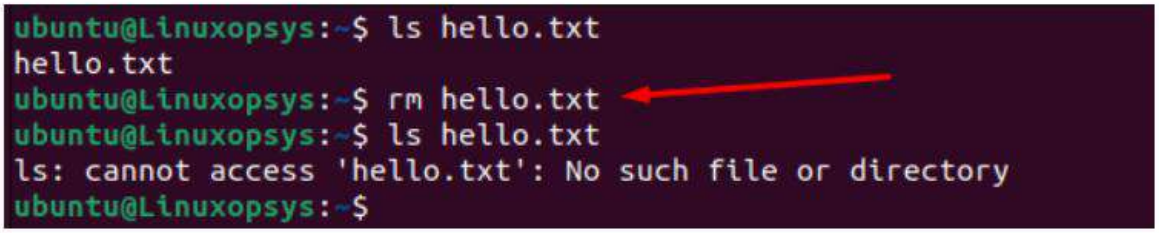
18. rm

The rm command is a Linux built-in used to delete files and directories.

Simply typing the rm command followed by a filename will permanently delete the file from the system thereby freeing up some disk space.

Example

```
rm hello.txt
```



One of the most commonly used options is `-r` or `-R`, which enables recursive removal, this is essential when you need to delete directories and their contents. Coupled with this, the `-f` option, stands for "force", which can be used to avoid prompts for confirmation.

19. zip

The zip command is used to compress files and directories into a single file with the .zip file extension.

Example:

```
zip compressed_output.zip filename1 filename2 filename3
```

The key options for the command:

- `zip -r`: Recursively compress a directory in its file into a single zip file.
- `zip -e`: Password protect a zip file.
- `zip -d`: Delete a specific file from a zip file.
- `zip -u`: Add files to a zip file.
- `zip -v`: Verbose

The zip command does not come pre-installed on most Linux distros. Refer to your distro documentation on how to install it.

20. tar

The tar command allows users to combine multiple files into a single archive file (commonly known as a tarball), which is easier to manage, transfer, and optionally compress.

Example:

```
tar -cvf archive.tar /path/to/directory
```

This creates an archive named archive.tar from the specified directory, with -v enabling verbose output and creating an archive with `-c`. The `-f` option specifies the filename of the archive.

Later you can use `-x` to extract files from the archive.

Compression is often coupled with archiving, utilizing options like -z for gzip compression, etc.

21. wc

The wc command in Linux is used to count the number of lines, words, and characters in text files. The name wc stands for word count.

Example:

```
wc data.txt
```

Various options can tailor its output:

- `wc -l`: Prints the number of lines in a given line.
- `wc -w`: Prints the number of words in a given line.
- `wc -c`: Prints the number of characters/bytes in a file.

22. stat

stat (short for status) is a Linux command-line utility for displaying detailed information about specific files or file systems. It is commonly used to obtain file timestamps related to its last access (`atime`), modification (`mtime`), and change (`ctime`).

Example:

```
stat data.txt
```

```
ubuntu@Linuxopsys:~$ stat data.txt
  File: data.txt
  Size: 33              Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d      Inode: 1179966     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1001/  ubuntu)   Gid: ( 1001/  ubuntu)
Access: 2023-10-26 09:57:32.756268797 +1100
Modify: 2023-10-26 09:57:17.224554138 +1100
Change: 2023-10-26 09:57:17.224554138 +1100
 Birth: 2023-10-26 09:57:17.224554138 +1100
ubuntu@Linuxopsys:~$
```

Useful commands for Linux **SYSTEM MANAGEMENT**:

23. uptime

The uptime command is a Linux utility used to check how long a system has been running since its last reboot. It provides information about the current time, the system's uptime, the number of users currently logged in, and system load averages for the last 1, 5, and 15 minutes.

Example:

```
uptime
```

```
ubuntu@Linuxopsys:~$ uptime
 10:03:01 up 9 min,  1 user,  load average: 0.02, 0.21, 0.17
ubuntu@Linuxopsys:~$
```

24. uname

The uname command (short for Unix name) is a Linux utility used to display information about the operating system and the system hardware.

Example:

```
uname -a
```

```
ubuntu@Linuxopsys:~$ uname -a
Linux Linuxopsys 6.2.0-35-generic #35~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Oct
  6 10:23:26 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
ubuntu@Linuxopsys:~$
```

The `-a` option provides the kernel name, hostname, kernel release, kernel version, and the machine's hardware name.

25. free

The free command provides information about the system's memory usage, including physical memory, swap, and buffer/cache usage.

```
free -h
```

```
ubuntu@Linuxopsys:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           3.3Gi       928Mi       959Mi        42Mi       1.4Gi       2.2Gi
Swap:          923Mi          0B       923Mi
ubuntu@Linuxopsys:~$
```

The `-h` (human-readable) option intelligently chooses the units (KB, MB, GB) for each value to enhance readability.

26. shutdown

The shutdown command in Linux is used to gracefully shut down or reboot the system.

Here is an example

```
shutdown
```

This will schedule the system to be shut down in 1 minute.

You can pass `-r` instructs the system to reboot post-shutdown. To abort a planned shutdown, the `-c` option proves handy.

27. apt

The apt command widely used package management utility in Debian-based Linux distributions like Ubuntu.

The most commonly used command is `apt-get update`, which synchronizes the package index files from their sources, followed by `apt-get upgrade` to upgrade all installed packages to their latest versions.

To install a new package use `apt-get install package_name`, and `apt-get remove package_name` removes it.

As mentioned apt is for Debian-based distro. For Fedora and other RPM-based distributions use dnf, the successor to YUM. Arch Linux uses pacman.

28. sudo

The sudo command allows permitted users to execute commands with the security privileges of another user, typically the superuser or root.

Example:

```
sudo apt update
```

Here apt package management system requires administrative privileges to modify system-wide settings and files so we used sudo.

29. du

The du (short for disk usage) command in Linux is used to estimate and display the disk space used by files and directories (including subdirectories) in the file system.

Running `du` without any options displays the disk usage of the current directory and its subdirectories in bytes.

The `-h` option is commonly used that display disk usage in human-readable format (eg: KB, MB, GB...). For a summarized output of just the total size of a directory, the `-s` option helps.

```
du -sh /path/to/directory
```

```
ubuntu@Linuxopsys:~$ du -sh /home/ubuntu/
495M    /home/ubuntu/
ubuntu@Linuxopsys:~$
```

30. df

The df (short for disk free) command in Linux is used to display the amount of free and used disk space of mounted filesystems.

The `-h` option is particularly popular as it presents the information in a "human-readable" format.

```
df -h
```

```
ubuntu@Linuxopsys:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           337M  3.1M  334M   1% /run
/dev/sda3        20G   15G  4.1G  78% /
tmpfs           1.7G     0  1.7G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  6.1M  506M   2% /boot/efi
tmpfs           337M  112K  337M   1% /run/user/1001
ubuntu@Linuxopsys:~$
```

Another important option is `-T`, which outputs the type of each filesystem like ext4, xfs, or nfs.

This displays the disk space usage in human-readable format for all mounted file systems.

31. lsblk

The lsblk (short for list block) command in Linux is used to display all available block devices in Linux (mounted or not), which typically includes physical storage devices like hard drives, solid-state drives, and partitions, as well as virtual devices like loop devices and LVM logical volumes.

Example:

```
lsblk | grep -v '^loop'
```

```
ubuntu@Linuxopsys:~$ lsblk | grep -v '^loop'
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda           8:0    0    20G  0 disk
├─sda1        8:1    0     1M  0 part
├─sda2        8:2    0   513M  0 part /boot/efi
└─sda3        8:3    0  19.5G  0 part /var/snap/firefox/common/host-hunspell
                                      /
sdb           8:16   0     1G  0 disk
└─vg02-data 253:0    0   400M  0 lvm
sdc           8:32   0   1.1G  0 disk
└─sdc1        8:33   0   1.1G  0 part
sr0          11:0    1  1024M  0 rom
ubuntu@Linuxopsys:~$
```

The output looks tree-like structure of all block devices, showcasing their names, sizes, mount points, and types, etc. Here I have piped grep -v '^loop' to exclude loop* snap devices from the output to look neat.

32. blkid

The blkid (short for block identifier) command in Linux is used to display information about block devices. It retrieves filesystems, partitions, and block devices information such as UUIDs, filesystem types, labels, and other metadata. This comes especially helpful for identifying storage devices and their associated filesystems or labels.

Command:

```
blkid
```

33. date

The date command in Linux is used to display or set system date and time. It allows users to display time in various formats and calculates the past and future dates.

Command:

```
date
```

This displays the current date (including timezone) and time in a predefined format.

Using + specifier followed by format strings allows to change format output and `-d` option lets users display a date other than the current one.

34. hostname

The hostname command is Linux command-line utility used to to view or set the system's hostname and domain name.

Command:

```
hostname
```

This command displays the current hostname set on the system. To modify the hostname, users typically input the desired name as an argument. Note that such changes might be temporary, reverting after a reboot, unless persisted in system-specific configuration files.

You can use hostnamectl command on systemd machine which is more descriptive.

35. fdisk

The fdisk command is a Linux menu-based command-line utility used for disk partitioning. It allows users to create, delete, modify, and manage disk partitions on the system.

The `-l` stands out, listing all partitions on all disk drives, and providing a quick overview of the storage layout.

```
sudo fdisk -l
```

You can pipe `sed -e '/Disk \/dev\/loop/,+5d'` to filter out disk information related to loop devices from the fdisk -l output.

Specifying a device, like `fdisk /dev/sda`, which then presents an interactive menu-driven interface to perform various operations on that device.

Approach it with caution, as improper modifications can lead to data loss.

36. mount

The mount command in Linux is an essential tool used to attach filesystem or partitions to a specific directory (known as the mount point) in the file system tree.

Here is an example of mounting a specified partition to the /mnt directory:

```
sudo mount /dev/sdc1 /mnt/
```

A couple of options worth noting:

- `-a`: Mount all filesystems mentioned in fstab file (/etc/fstab).
- `-o`: Allows for the specification of various mount options like `ro` for read-only access.

37. systemctl

The systemctl command is used for examining and controlling the systemd system and service manager. It allows users to perform actions such as starting, stopping, restarting, enabling, and disabling services.

To start or stop service, the commands looks:

```
systemctl start service_name
systemctl stop service_name
```

On top of use enable and disable (say `systemctl disable service_name`) commands for managing whether services start automatically at boot.

The status command, as in `systemctl status service_name`, can be used to inspect the current status of a service, providing details like whether it's running. While `systemctl list-unit-files` shows all unit files along with their enabled/disabled state.

38. vmstat

vmstat (Virtual Memory Statistics) is a command-line utility in Linux that provides reports about various system resources such as processes, memory, paging, block IO, traps, and CPU activity.

Command:

```
vmstat
```

By specifying an interval (in seconds), users can obtain periodic updates.

The `-s` option delivers a summarized report of the memory statistics and `-d` for detailed disk statistics.

39. echo

The echo command is a built-in Linux command-line utility used to print arguments it receives to the standard output (by default your terminal display). It can also be used to display the values of user-defined variables and environment variables.

Here is an example of printing the text “Hello Linux folks!” on the terminal screen:

echo “Hello Linux folks!”

The echo typically appends to its output, you can use `-n` option suppresses the newline character. The `-e` option enables the interpretation of escaped characters, like `\t` for a tab etc.

40. history

The history command is a Linux builtin command used to view and manage a list of previously executed commands in a terminal session. It provides a history of commands that have been priviously executed or entered, making it easier to recall and reuse them without typing them again.

Example:

```
history
```

This simply lists the recent commands, each prefixed by a unique identification number.

The `history 10`, which shows the last 10 commands. The `-c` option clears the entire command history.

Linux Commands for **USER MANAGEMENT:**

41. passwd

The passwd command is used in Linux to change the user account password. If the command is used by the root user, it can change the password for other users.

Simply typing passwd without any arguments would change the currently logged-in user's password.

Here is an example:

```
passwd
```

System administrators can specify a username, like `passwd bob`, to reset the password for user "bob".

Options you should be aware of:

- d: Delete a user's password (make it empty). This is the same as disabling a password for an account.
- -e: Immediately expire an account's password.

- -l: Temporarily locks a user account, `-u` unlocks it.

42. usermod

The usermod is used to modify user account attributes and settings. It has options to change various properties of an existing user account, such as their username, home directory, group membership, shell, etc.

The following is an example of using the usermod command to add the user “bob” into the sudoers file.

```
sudo -aG sudo bob
```

Here are some of the options you can use with the usermod command:

- `-m`: Move the user's home directory.
- `-d`: Specify new home directory location. Commonly used with -m option to move a user's home direction to a new location.
- `-L`: Lock user account -U to unlock.
- `-e`: Set user account expiration date.

43. useradd

The useradd command in Linux is used to create or add new accounts on the system. When creating users, the it gives you more control via many command line options. If you are someone who wants something simple, and interactive, with a guided process then use adduser command.

Here is an example of creating a new user without a login shell:

```
sudo useradd -d /path/to/home/directory -s /bin/false tom
```

The -d option tells useradd to specify the user's home directory. You can combine both -m and -d option to create custom home directory and ensure it is created if it doesn't exist.

```
useradd -m -d /data/home/toms toms
```

The `-g` and `-G` options assign the primary group and supplementary groups to the user, respectively.

44. userdel

The userdel is used to delete a user account in Linux.

Here is an example of deleting the user we created earlier named tom:

```
sudo userdel tom
```

This doesn't remove the user's home directory. To erase the associated home directory and its contents, use `-r` option.

45. groupadd

In Linux, the groupadd command is used to create a new group on the system. This group is known as the secondary group. You can add users to it using the usermod command.

For example to create a group named developers, type:

```
sudo groupadd developers
```

Once created, you can add users to it using the usermod command.

46. w

The w command displays information about the currently logged-in users on a system and their ongoing activities.

Here is an example:

```
w
```

The output provides a list of users who are logged into the system, along with additional details such as their terminal, login time, idle time, and

the IP address they are connected from.

47. ssh

The ssh command in Linux is used to establish secure remote connections between servers or remote computers over the internet. SSH or Secure Shell in full, provides a secure and encrypted method for logging into and managing remote systems.

For example, to connect to a remote server with the IP address 192.168.1.77 using the username "tom" you would use the following command:

```
ssh tom@192.168.1.77
```

You will be prompted to provide the password for the user tom for authentication.

Helpful Linux commands for managing **PROCESSES**:

48. top

The top is a command-line utility that displays real-time information about the processes that are running on the system. They also display the system summary, which includes memory usage. The top command not only displays the information, but also allows the user to interactively control its behavior, sort the display, and send signals to processes.

Simply typing top without any options into your terminal will start it:

```
top
```

By default, top sorts processes based on their CPU usage, making it easy to identify resource-intensive tasks.

49. ps

The ps command, short for "process status," is used in UNIX and UNIX-like operating systems to view information about processes running in a system. It provides you insights into the system's process activity.

Running the ps command without any options will display a basic list of processes that are associated with the terminal from which the command is executed.

```
ps
```

combination is `ps aux`, which provides a comprehensive view of all processes system-wide, complete with resource utilization metrics.

The following options of ps comes useful:

- `-A`, `-e`: Display information about all processes.
- `-f`: Provide a full-format listing.
- `-l`: Display long format.
- `-u`: user: Display processes owned by the user.
- `-x`: Include processes not attached to a terminal.
- `-p`: PID: Display information for a specific process ID (or list of PIDs).
- `-T`: Display information about processes and threads.

The combination `ps aux`, which provides a comprehensive view of all processes system-wide, complete with resource utilization metrics.

50. kill

The kill command is a powerful tool for managing processes in Linux. It is often used in conjunction with other commands like ps, pidof or pgrep to identify the process by name or other attributes and then send signals to it using kill.

Example:

```
kill 1234
```

This sends the `SIGTERM` (signal 15) to gracefully request 1234 process to terminate. If a process doesn't respond to `SIGTERM`, a more forceful `SIGKILL` (signal 9) can be used.

```
kill -9 $(pgrep firefox)
```

51. fuser

The fuser command in Linux is used to identify processes that are using specific files, sockets, or file systems. It is commonly employed to identify processes that are accessing or locking a specific file, especially when there's a need to edit, save, or delete that file without interference

Here is an example of identifying all processes that are accessing the file /usr/bin/python3:

```
fuser /usr/bin/python3
```

52. lsof

The lsof command, short for "list open files", provides detailed information about files opened by processes.

Default lsof lists all open files and the processes that opened them.

Another user case to list processes using TCP port 22 (SSH), type:

```
lsof -i TCP:22
```

Additionally, use the -p option followed by the PID to list all open files associated with that process.

When comes to **NETWORKING** following are the useful commands:

53. ip

The ip command is a versatile tool in Linux for managing network configuration. Its part of the iproute2 package. This package replaces old

tools such ifconfig, route, etc. The ip command is commonly used to manage IP addresses, network interfaces, routes, ARP cache, and tunnels.

It comes with lots of objects and options. Here are some useful object examples:

- `ip addr show`: Display IP addresses for all available network interfaces.
- `ip addr add/del`: For adding/deleting IP addresses.
- `ip route show`: Display routes.
- `ip route add`: Create new routes.
- `ip link show`: Display the state of all network interfaces.

54. ping

The ping command uses ICMP (Internet Control Message Protocol) packets for diagnosing network connectivity and latency issues. Packets are sent to a specified target, be it an IP address or a domain name, and then listen for the echo replies.

To continuously send packets to "example.com" until interrupted.

```
ping example.com
```

You can press `Ctrl+C` to disconnect.

`ping -c 5 example.com`, which sends precisely five packets.

55. traceroute

The traceroute command network diagnostic tool used to trace the path that an IP packet takes to reach a destination.

Generally used without any options as

```
traceroute example.com
```

This displays the route taken by packets to reach 'example.com'.

56. wget

The wget command in Linux is a non-interactive command-line utility that allows you to download files from the internet. wget supports various network protocols, such as HTTP, HTTPS, FTP, and etc.

To download file with wget, simply provide the url to the file as an argument. For example:

The following options are useful:

- `-O`: Downloads a file using a different file name.
- `-b` or `--background`: Downloads a file in the background and frees up the terminal.
- `-i`, `--input-file`: Allows to specify a list of URLs in a file to download.
- `-c`: Resumes download of a partially downloaded file
- `-P`: Specifies the directory that a file will be downloaded
- `-r` or `--recursive`: Follow links and download recursively.

57. scp

The scp short for secure copy protocol is a file transfer network protocol that makes it simple and safe to copy files between two remote computers or between a remote computer and a local computer. SCP makes use of the SSH (Secure Shell) protocol for authentication and encryption, ensuring that the information transmitted is secure from malicious actors.

In this example, we'll use the scp command to transfer the file log.txt to a specified remote user's home directory. In this case, the remote user is 'linux':

```
scp log.txt linux@remote-hostname-or-ip:/home/linux
```

Among options, the `-r` flag is particularly notable, allowing for the recursive transfer of entire directories.

58. rsync

The rsync command in Linux is a powerful utility for synchronizing files and directories between two locations. It's commonly used to perform backups, transfer files between systems, and keep data synchronized.

Basic example:

```
rsync -a source_directory/ destination_directory/
```

Here the content of `source_directory` is mirrored at `destination_directory`, with the `-a` option preserving file permissions, timestamps, and other metadata.

Among its many options, the `--delete` flag stands out for its ability to remove files in the destination that are no longer present in the source.

59. ss

The ss (Socket Statistics) command in Linux displays network socket information. A modern replacement for the traditional netstat tool.

The default ss command provides a list of open sockets.

```
ss
```

The `-t` and `-u`, which display TCP and UDP sockets respectively. The `-l` option is handy for listing only the listening sockets.

60. nmap

The nmap (Network Mapper) is a network scanning and security auditing tool in Linux that is used to discover hosts and services on a computer network.

Basic Usage:

```
$ nmap 192.168.1.1
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-14 08:31 UTC
Nmap scan report for 192.168.1.1.ip.linuxsystem01.com (192.168.1.1)
Host is up (0.00052s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
9090/tcp open  zeus-admin

Nmap done: 1 IP address (1 host up) scanned in 5.10 seconds
```

This scans the specified IP address for open ports and service details.

Some of the useful options are:

The `-sS` conducts a SYN stealth scan to identify open ports without establishing a full TCP connection, reducing the likelihood of detection. The `-p` option specifies the ports to scan, and can range from individual ports (`-p 80`) to ranges (`-p 1-65535`).

Useful Linux commands for **SEARCH** Operations:

61. find

The find command is a powerful tool used in Linux to search for files and directories based on a variety of attributes such as name, size, type, and more.

For instance

```
find /path/to/search -name 'filename.txt'
```

This searches for a file named 'filename.txt' starting from the specified path.

The `-type` option allows users to filter results based on file type, with arguments like `f` for regular files or `d` for directories. The `-size` option

proves invaluable in finding files based on their size. Use -mtime to search for files based on modification time.

62. grep

The grep (Global Regular Expression Print in full) is one of the most used and powerful commands in Linux for searching and matching text patterns within one or more files.

It operates by matching lines against a specified pattern.

To search for a pattern “Linux” in a single file called data.txt, run:

```
grep “Linux” data.txt
```

This would output all lines containing 'Linux' from the specified file.

Among its many options, `-i` stands out for enabling case-insensitive matching, ensuring that the search is not constrained by letter case. The `-r` or `-R` option enables recursive searching through directories. Furthermore, the `-E` option empowers `grep` to interpret the pattern as an extended regular expression, allowing for more complex search criteria.

Powerful commands for **FILTER** operations in Linux:

63. tr

The tr command, which stands for translate, is a Linux tool for translating, deleting, or squeezing repeated characters from standard input and writing the result to standard output.

A typical use case involves translating characters from one set to another. Example:

```
echo "hello" | tr 'a-z' 'A-Z'
```

This converts lowercase letters to uppercase, resulting in "HELLO".

The `-d` option is crucial when there's a need to delete characters from a set

```
echo "hello 123" | tr -d '0-9'   ###This remove all digits
```

The `-s` option, on the other hand, squeezes or replaces repeated characters into a single character, which is useful for normalizing spaces or other delimiters in text.

64. cut

The cut is used to extract and display specific columns or fields from each line of a file or standard input. It is commonly utilized to parse and manipulate structured text data, such as CSV or TSV files.

The most frequently used options include `-d`, which specifies a delimiter character, and `-f`, which indicates the fields or columns to be extracted.

```
cut -d ',' -f 1 filename.csv
```

The above command would extract and display the first column of a comma-delimited file.

The `-c` option comes useful to select specific character positions from each line, as in `cut -c 1-5 filename.txt`, which would display the first five characters of each line.

65. uniq

The uniq command is a Linux command line utility that identifies adjacent lines that are duplicated in an input file and prints unique lines to the standard output or write to an output file.

A use case example:

```
sort filename.txt | uniq
```

This case file is sorted before duplicate lines are removed.

uniq command line options you should be aware of:

- `-u`, `- -unique` : The unique lines will only be printed from the input content.
- -d, `- -repeated` : The duplicate lines will only be printed from the input content where it displays one line per each repeated line.
- -c, `- -count` : Displays the duplicate count of each repeated line as a number before the line.
- -D, `- -all-repeated` : Only outputs all the duplicated lines and ignores unique lines.
- -z, `- -zero-terminated` : A line will end with a NULL or 0 bytes. By default, each line ended with a new line.

66. tee

The Linux tee command is a powerful and useful command used to read from standard input and write the output to both standard output and one or more files. It's named after the T-splitter used in plumbing, which divides water into two streams, one flowing to the tap and the other to the drain.

To display the output of a specific command at the same time while writing it to a file you would run:

```
df -h | tee file1.txt
```

You can use `-a` option for appending the output to the specified files, rather than overwriting them.

67. awk

awk is a powerful command-line utility as well as a scripting language that is used to manipulate data and format output reports. This command supports several variables, functions, and logical operators to get the desired output.

Let's look at a simple example of displaying a specific column in a file:

```
awk ‘{print $2}’ records.txt
```

This example will print all the records in second column.

Here are a few options that comes with awk:

- `-F` : Specifies the field separator (delimiter) for input data.
- `-v var=value` : Assigns a value to an awk variable.
- `-f scriptfile` : Reads awk commands from a script file.

68. sed

The Linux sed command is a stream editor that is used to process text file content like searching for patterns, finding and replacing, insertion, and deletion.

To simply replace a string in a file, for example, you would run:

```
sed 's/World/Folks!/' example.txt
```

This will replace the first occurrence of the word "World" with "Folks!" on each line in the file example.txt.

Whereas to replace all instances:

```
sed 's/World/Folks!/' example.txt
```

69. sort

The sort command is a Linux utility that allows you to sort lines of text files alphabetically or numerically. It is often used with other commands, such as grep and uniq, to analyze and manipulate text data.

For example, to sort a file you would run:

```
sort example.txt
```

By default, sort arranges lines in lexicographic order but is equipped with options to customize the sorting behavior.

To sort based on specific fields or columns, the `-k` option is used, such as in `sort -k2,2 filename.txt` which sorts the data based on the second field

70. xargs

xargs is a powerful and versatile command line utility in Linux to construct and execute command lines by gathering arguments from standard input. It is frequently used to enhance the utility of commands like cp, mv, rm, and many others.

One common usage

```
find . -type f -name "*.txt" | xargs rm
```

This finds all text files in the current directory and its subdirectories and deletes them.

Another use case

```
ls *.txt | xargs -I {} mv {} {}.bak
```

This lists all `.txt` files in the current directory and then, using `xargs`, renames each file by appending a `.bak` extension to it.
