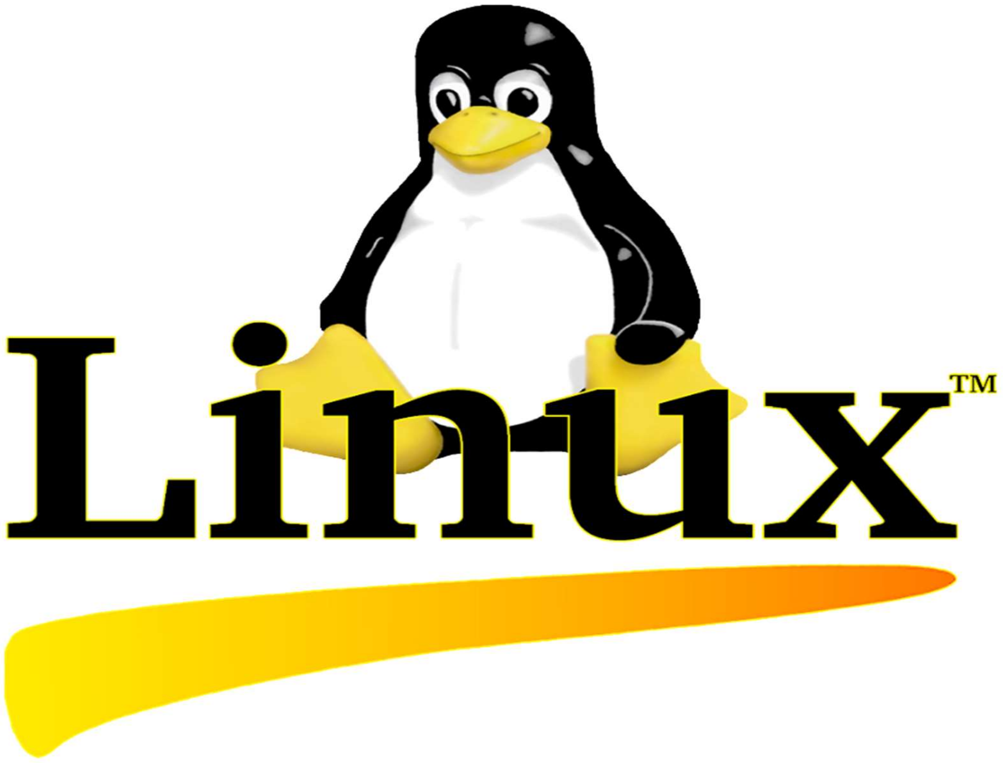
Linux Commands: Explanation and Practical Examples



Davoud Azari

Contents

1. File and Directory Management

- **ls** – List directory contents:

  ```
  ls
  ls -l          # detailed list
  ls -a          # includes hidden files
  ls -h          # human-readable file sizes
  ls -R          # recursive listing
  ls -S          # sort by file size
  ls -t          # sort by modification time
  ```

- **cd** – Change directory:

  ```
  cd /path/to/directory
  cd ..          # move up one directory
  cd ~           # go to home directory
  cd -           # go to previous directory
  ```

- **pwd** – Print working directory:

  ```
  pwd
  ```

- **cp** – files and directories:

  ```
  cp file1.txt /path/to/destination
  cp -r folder1 /path/to/destination  #  directory recursively
  cp -i file1.txt /path/to/destination  # prompt before overwriting
  cp -u file1.txt /path/to/destination  #  only if source is newer
  cp -v file1.txt /path/to/destination  # verbose output
  ```

- **mv** – Move or rename files and directories:

  ```
  mv file1.txt /path/to/destination  # move file
  mv oldname.txt newname.txt         # rename file
  mv -i file1.txt /path/to/destination  # prompt before overwriting
  mv -u file1.txt /path/to/destination  # move only if source is newer
  ```

- **rm** – Remove files or directories:

  ```
  rm file1.txt
  rm -r directory_name  # remove directory recursively
  rm -f file1.txt       # force remove, skip confirmation
  rm -i file1.txt       # prompt before removing
  rm -v file1.txt       # verbose output
  ```

- **mkdir** – Make directories:

  ```
  mkdir new_directory
  mkdir -p /path/to/directory  # create parent directories if not exist
  mkdir -v new_directory  # verbose output
  ```

- **rmdir** – Remove empty directories:

```
rmdir directory_name
rmdir -v directory_name  # verbose output
```

- **touch** – Change file timestamps or create empty files:

```
touch newfile.txt  # create empty file if not exists
touch -c file1.txt  # do not create file if it does not exist
touch -t 202502191210.30 file1.txt  # set specific timestamp
```

- **find** – Search for files in a directory hierarchy:

```
find /path/to/directory -name "file*.txt"
find /path/to/directory -type f  # find files only
find /path/to/directory -type d  # find directories only
find /path/to/directory -size +100M  # find files larger than 100MB
find /path/to/directory -mtime -7  # files modified in the last 7 days
find /path/to/directory -exec rm {} \;  # find and delete files
```

- **locate** – Find files by name:

```
locate filename
locate -i filename  # case insensitive search
```

- **tree** – Display directories in a tree-like format:

```
tree
tree -L 2  # limit depth to 2 levels
tree -d    # list directories only
tree -f    # show full path of files
```

- **chmod** – Change file permissions:

```
chmod 755 file1.sh  # owner can read/write/execute, others can
read/execute
chmod +x script.sh  # add execute permission
chmod -R 755 directory  # apply recursively
chmod u+x file1.sh  # add execute permission to the user
chmod g-w file1.txt  # remove write permission from the group
```

- **chown** – Change file owner and group:

```
chown user:group file1.txt
chown -R user:group directory  # recursively change owner and group
```

- **chgrp** – Change group ownership:

```
chgrp group_name file1.txt
chgrp -R group_name directory  # recursively change group ownership
```

- **stat** – Display file or file system status:

```
stat file1.txt
```

2. File Viewing and Editing

- **cat** – Concatenate and display file content:

```
cat file.txt
cat -n file.txt  # number the lines
cat -b file.txt  # number non-blank lines only
cat -A file.txt  # show all control characters
```

- **tac** – Concatenate and display file content in reverse:

```
tac file.txt
```

- **more** – View file content interactively (page by page):

```
more file.txt
more +10 file.txt  # start viewing from line 10
```

- **less** – View file content interactively (scrollable):

```
less file.txt
less +G file.txt  # go to end of file
less +/pattern file.txt  # search for pattern
```

- **head** – Output the first part of a file:

```
head file.txt
head -n 20 file.txt  # first 20 lines
head -c 50 file.txt  # first 50 bytes
```

- **tail** – Output the last part of a file:

```
tail file.txt
tail -f file.txt  # follow file as it's being updated
tail -n 20 file.txt  # last 20 lines
tail -c 100 file.txt  # last 100 bytes
```

- **nano** – Text editor (terminal-based):

```
nano file.txt
nano -w file.txt  # disable line wrapping
nano -v file.txt  # start in view-only mode
```

- **vim / vi** – Advanced text editors:

```
vim file.txt
```

```
vi file.txt
vim -y file.txt  # read-only mode
vim -R file.txt  # open in read-only mode
```

- **emacs** – Text editor:

```
emacs file.txt
emacs -nw file.txt  # start in terminal mode (no GUI)
```

- **grep** – Search text using patterns:

```
grep "pattern" file.txt
grep -i "pattern" file.txt  # case insensitive
grep -r "pattern" /path/to/dir  # search recursively
grep -v "pattern" file.txt  # exclude matching lines
grep -l "pattern" *.txt  # show filenames containing pattern
```

- **sed** – Stream editor for filtering and transforming text:

```
sed 's/old/new/g' file.txt  # replace 'old' with 'new' in the file
sed -i 's/old/new/g' file.txt  # in-place edit
sed '1,5d' file.txt  # delete lines 1 to 5
```

- **awk** – Pattern scanning and processing language:

```
awk '{print $1}' file.txt  # print the first column of each line
awk -F, '{print $1}' file.txt  # specify a delimiter (comma)
```

- **cut** – Remove sections from each line of files:

```
cut -d',' -f1 file.txt  # extract first field of CSV file
cut -f1-3 file.txt  # extract fields 1 to 3
```

- **sort** – Sort lines of text files:

```
sort file.txt
sort -r file.txt  # reverse order
sort -n file.txt  # numerical sort
```

- **uniq** – Report or omit repeated lines:

```
uniq file.txt
uniq -c file.txt  # count occurrences of each line
uniq -d file.txt  # show only duplicate lines
```

3. Process Management

- **ps** – Report a snapshot of current processes:

```
ps
ps -e             # list all processes
ps -ef            # full-format listing
ps -aux           # user-oriented format
ps -u user        # processes by specific user
```

- **top** – Display Linux tasks:

```
top
top -u user       # show processes for a specific user
top -d 5          # update every 5 seconds
top -p pid        # monitor specific process ID
```

- **htop** – Interactive process viewer (advanced top):

```
htop
htop -u user      # show processes of a specific user
htop -d 5         # set refresh delay
```

- **kill** – Send a signal to a process, typically to terminate:

```
kill pid                # terminate process by PID
kill -9 pid             # force kill (SIGKILL)
kill -SIGTERM pid       # send specific signal
```

- **killall** – Terminate processes by name:

```
killall process_name    # terminate all processes by name
killall -9 process_name # force kill all processes
```

- **bg** – Resume a suspended job in the background:

```
bg job_number          # resume a specific job
bg %1                  # resume the first job
```

- **fg** – Bring a job to the foreground:

```
fg job_number          # bring a specific job to the foreground
fg %1                  # bring the first job to the foreground
```

- **jobs** – List active jobs:

```
jobs
jobs -l                # list jobs with PID
```

- **nice** – Run a program with modified scheduling priority:

```
nice -n 10 command     # run a command with lower priority
nice -n -10 command    # run a command with higher priority
```

- **renice** – Alter priority of running processes:

```
renice -n 10 -p pid    # change priority of a process by PID
renice -n -10 -p pid   # change priority to higher for a process
```

- **uptime** – Show how long the system has been running:

```
uptime
```

- **time** – Measure program running time:

```
time command
```

4. Disk Management

- **df** – Report file system disk space usage:

```
df
df -h                   # human-readable format
df -T                   # show file system type
df /path/to/directory  # disk space of a specific directory
```

- **du** – Estimate file space usage:

```
du file.txt             # space used by a file
du -sh /path            # human-readable summary
du -a /path             # show space used by all files
du -h /path             # human-readable format
```

- **fdisk** – Partition table manipulator for Linux:

```
fdisk -l                # list all partitions
fdisk /dev/sda          # manipulate partitions on /dev/sda
```

- **lsblk** – List information about block devices:

```
lsblk
lsblk -f                # show filesystem type
lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT  # custom output
```

- **mount** – Mount a file system:

```
mount /dev/sda1 /mnt   # mount device to /mnt
mount -t ext4 /dev/sda1 /mnt  # specify filesystem type
```

- **umount** – Unmount a file system:

```
umount /mnt
umount /dev/sda1
```

- **parted** – A partition manipulation program:

```
parted /dev/sda
parted -l            # list partitions
parted /dev/sda mkpart primary ext4 1GB 10GB  # create a partition
```

- **mkfs** – Create a file system:

```
mkfs.ext4 /dev/sda1   # create ext4 file system
mkfs.xfs /dev/sda1    # create XFS file system
```

- **fsck** – File system consistency check and repair:

```
fsck /dev/sda1        # check and repair the file system
fsck -f /dev/sda1     # force check even if the system thinks it's
clean
```

- **blkid** – Locate/print block device attributes:

```
blkid
blkid /dev/sda1       # get attributes for specific device
```

5. Networking

- **ifconfig** – Configure network interfaces:

```
ifconfig
ifconfig eth0 up         # bring interface eth0 up
ifconfig eth0 down       # bring interface eth0 down
ifconfig eth0 192.168.1.10 netmask 255.255.255.0  # set IP and netmask
ifconfig -a              # show all interfaces, even inactive ones
```

- **ip** – Show/manipulate routing, devices, and tunnels:

```
ip addr                 # show IP addresses
ip link                 # show network interfaces
ip route                # show routing table
ip link set eth0 up     # bring interface eth0 up
ip addr add 192.168.1.10/24 dev eth0  # assign IP address
```

- **ping** – Send ICMP Echo requests to network hosts:

```
ping 192.168.1.1        # ping IP address
ping -c 4 google.com    # send 4 ICMP packets to google.com
ping -i 0.5 192.168.1.1 # set interval between pings to 0.5 seconds
```

- **netstat** – Network statistics:

```
netstat                 # show network connections
netstat -tuln           # show listening ports and associated processes
netstat -i              # show network interface statistics
netstat -r              # show routing table
```

- **ss** – Socket statistics (faster than netstat):

```
ss                        # display all sockets
ss -tuln                  # show listening TCP/UDP ports
ss -p                     # show process using the socket
```

- **traceroute** – Trace the route packets take to a network host:

```
traceroute google.com   # trace route to google.com
traceroute -m 20 google.com  # set max hops to 20
```

- **nslookup** – Query Internet name servers interactively:

```
nslookup google.com       # look up the IP address for google.com
nslookup 8.8.8.8          # look up the domain for IP address 8.8.8.8
```

- **dig** – DNS lookup utility:

```
dig google.com            # query DNS records for google.com
dig @8.8.8.8 google.com # query DNS using a specific DNS server
dig +short google.com   # show only the IP address
```

- **wget** – Non-interactive network downloader:

```
wget http://example.com/file.tar.gz  # download a file
wget -c http://example.com/file.tar.gz  # continue a previously
interrupted download
wget -r http://example.com/  # download a website recursively
```

- **curl** – Transfer data with URLs:

```
curl http://example.com    # fetch data from a URL
curl -O http://example.com/file.tar.gz  # download file
curl -I http://example.com  # show HTTP headers
curl -u user:password http://example.com  # use basic authentication
```

- **scp** – Secure copy files between hosts:

```
scp file.txt user@remote_host:/path/to/destination
scp -r folder user@remote_host:/path/to/destination  # copy directory
recursively
```

- **ssh** – Secure shell for remote login:

```
ssh user@remote_host    # login to a remote machine
ssh -p 2222 user@remote_host  # specify a different port
ssh -i /path/to/private_key user@remote_host  # use an SSH key for
authentication
```

- **ftp** – File Transfer Protocol client:

```
ftp example.com          # connect to FTP server
ftp> ls                  # list files in the FTP server directory
ftp> get file.txt        # download a file from the server
ftp> put file.txt        # upload a file to the server
```

6. User and Group Management

- **useradd** – Add a user to the system:

```
useradd newuser          # create a new user
useradd -m -s /bin/bash newuser  # create a user with home directory
and bash shell
```

- **usermod** – Modify a user account:

```
usermod -aG groupname username  # add user to a group
usermod -s /bin/zsh username    # change user shell
```

- **userdel** – Delete a user account:

```
userdel username         # delete user account
userdel -r username      # delete user and their home directory
```

- **groupadd** – Add a group to the system:

```
groupadd newgroup        # create a new group
```

- **groupdel** – Delete a group:

```
groupdel groupname       # delete group
```

- **passwd** – Change user password:

```
passwd username          # change password for a user
passwd                   # change the current user's password
```

- **chage** – Change user password expiry information:

```
chage -l username        # display password expiration info
chage -E 2025-12-31 username  # set password expiration date
```

- **whoami** – Print the current logged-in user:

```
whoami
```

- **who** – Show who is logged in:

```
who
```

- **w** – Show who is logged in and what they're doing:

```
w
```

- **id** – Display user and group information:

```
id username             # display user and group IDs
```

- **groups** – Show user's groups:

```
groups username         # show the groups a user belongs to
```

7. System Information and Monitoring

- **uname** – Print system information:

```
uname                  # basic system info (kernel name)
uname -a               # all system information
uname -r               # kernel version
uname -m               # machine hardware name
```

- **hostname** – Show or set the system's hostname:

```
hostname               # show the current hostname
hostname new-hostname  # set the system hostname
```

- **uptime** – How long the system has been running:

```
uptime
```

- **dmesg** – Boot and system messages:

```
dmesg                  # display boot and system messages
dmesg | grep error     # filter dmesg output for errors
```

- **free** – Display memory usage:

```
free                   # display memory usage
free -h                # human-readable format
free -m                # display in MB
free -g                # display in GB
```

- **top** – Display Linux tasks:

```
top                    # display tasks
top -u user            # show processes of a specific user
top -p pid             # monitor a specific process by PID
top -d 5               # update every 5 seconds
```

- **vmstat** – Report virtual memory statistics:

```
vmstat                 # display virtual memory stats
vmstat 5               # update every 5 seconds
```

- **lscpu** – Display information about the CPU architecture:

```
lscpu                  # display CPU architecture details
```

- **lsusb** – List USB devices:

```
lsusb                  # list USB devices
lsusb -v               # detailed information about USB devices
```

- **lspci** – List PCI devices:

```
lspci                  # list PCI devices
lspci -v               # detailed information about PCI devices
```

- **lshw** – List hardware configuration:

```
lshw                   # list hardware configuration
lshw -short            # show a summary of hardware
lshw -C network        # show network devices only
```

8. Archiving and Compression

- **tar** – Archive files:

```
tar -cf archive.tar /path/to/directory  # create a tarball
tar -xf archive.tar  # extract tarball
```

  - **Compress files using gzip**:

```
tar -czf archive.tar.gz /path/to/directory  # create a gzipped
tarball
tar -xzf archive.tar.gz  # extract gzipped tarball
```

- **zip** – Package and compress files into a ZIP archive:

```
zip archive.zip file1 file2 file3  # create a zip archive
zip -r archive.zip /path/to/directory  # zip a directory recursively
```

- **unzip** – Extract files from a ZIP archive:

```
unzip archive.zip   # extract the zip archive
unzip archive.zip -d /path/to/extract  # extract to a specific
directory
```

- **gzip** – Compress files using the gzip algorithm:

```
gzip file.txt       # compress a file
gzip -d file.txt.gz # decompress a file
```

- **gunzip** – Decompress files compressed with gzip:

  ```
  gunzip file.txt.gz  # decompress a file
  ```

- **bzip2** – Compress files using the bzip2 algorithm:

  ```
  bzip2 file.txt      # compress a file
  bzip2 -d file.txt.bz2  # decompress a file
  ```

- **bunzip2** – Decompress files compressed with bzip2:

  ```
  bunzip2 file.txt.bz2  # decompress a file
  ```

- **xz** – Compress files using the xz algorithm:

  ```
  xz file.txt          # compress a file
  xz -d file.txt.xz    # decompress a file
  ```

- **unxz** – Decompress files compressed with xz:

  ```
  unxz file.txt.xz     # decompress a file
  ```

9. Package Management (Depends on Distribution)

Debian-based (e.g., Ubuntu)

- **apt-get** – APT package handling utility:
  - **apt-get install <package>** – Install a package:

    ```
    sudo apt-get install package_name
    ```

  - **apt-get update** – Update package list:

    ```
    sudo apt-get update
    ```

  - **apt-get upgrade** – Upgrade installed packages:

    ```
    sudo apt-get upgrade
    ```

  - **apt-get remove <package>** – Remove a package:

    ```
    sudo apt-get remove package_name
    ```

- **apt-cache** – Query APT cache:
  - **apt-cache search <package>** – Search for a package:

```
apt-cache search package_name
```

- **apt-cache show <package>** – Show package details:

  ```
  apt-cache show package_name
  ```

Red Hat-based (e.g., CentOS, Fedora)

- **yum** – Package manager for RPM-based systems:
  - **yum install <package>** – Install a package:

    ```
    sudo yum install package_name
    ```

  - **yum update** – Update installed packages:

    ```
    sudo yum update
    ```

  - **yum remove <package>** – Remove a package:

    ```
    sudo yum remove package_name
    ```

- **dnf** – Next-generation package manager (Fedora, CentOS 8+):
  - **dnf install <package>** – Install a package:

    ```
    sudo dnf install package_name
    ```

  - **dnf update** – Update installed packages:

    ```
    sudo dnf update
    ```

  - **dnf remove <package>** – Remove a package:

    ```
    sudo dnf remove package_name
    ```

General Commands

- **rpm** – RPM package manager:
  - **rpm -i <package.rpm>** – Install an RPM package:

    ```
    sudo rpm -i package_name.rpm
    ```

  - **rpm -e <package>** – Remove an RPM package:

    ```
    sudo rpm -e package_name
    ```

- **dpkg** – Debian package manager:
  - **dpkg -i <package.deb>** – Install a Debian package:

```
sudo dpkg -i package_name.deb
```

- **dpkg -r <package>** – Remove a Debian package:

```
sudo dpkg -r package_name
```

10. System Services and Daemon Management

- **systemctl** – Control the systemd system and service manager:
  - **systemctl start <service>** – Start a service:

    ```
    sudo systemctl start service_name
    ```

  - **systemctl stop <service>** – Stop a service:

    ```
    sudo systemctl stop service_name
    ```

  - **systemctl restart <service>** – Restart a service:

    ```
    sudo systemctl restart service_name
    ```

  - **systemctl enable <service>** – Enable a service to start on boot:

    ```
    sudo systemctl enable service_name
    ```

  - **systemctl disable <service>** – Disable a service from starting on boot:

    ```
    sudo systemctl disable service_name
    ```

  - **systemctl status <service>** – Check service status:

    ```
    systemctl status service_name
    ```

- **service** – Older service management command (used in non-systemd systems):
  - **service <service> start** – Start a service:

    ```
    sudo service service_name start
    ```

  - **service <service> stop** – Stop a service:

    ```
    sudo service service_name stop
    ```

  - **service <service> restart** – Restart a service:

    ```
    sudo service service_name restart
    ```

  - **service <service> status** – Check service status:

```
sudo service service_name status
```

11. Scheduling Tasks

- **cron** – Daemon for running scheduled commands:
  - **crontab -e** – Edit cron jobs for the current user:

    ```
    crontab -e
    ```

  - **crontab -l** – List the current user's cron jobs:

    ```
    crontab -l
    ```

  - **crontab -r** – Remove the current user's cron jobs:

    ```
    crontab -r
    ```

- **at** – Run commands at a specified time:
  - **at 09:00** – Schedule a command to run at 09:00 AM:

    ```
    at 09:00
    # Enter command to execute at 09:00
    ```

- **batch** – Run commands when the system load is low:

  ```
  batch
  # Enter commands to run when system load is low
  ```

- **sleep** – Delay for a specified time:
  - **sleep 5s** – Sleep for 5 seconds:

    ```
    sleep 5s
    ```

---

12. File Permissions and Security

- **chmod** – Change file permissions:

  ```
  chmod 755 file.sh   # owner can read/write/execute, others can
  read/execute
  chmod +x file.sh    # add execute permission
  chmod -x file.sh    # remove execute permission
  ```

- **chown** – Change file owner and group:

  ```
  chown user:group file.txt
  ```

- **chgrp** – Change the group ownership of a file:

```
chgrp group_name file.txt
```

- **umask** – Set default permissions for new files:

```
umask 022         # set default permissions to rw-r--r--
```

- **setfacl** – Set file access control lists (ACL):

```
setfacl -m u:username:rwx file.txt   # give user 'username'
read/write/execute permissions
```

- **getfacl** – Get file access control lists (ACL):

```
getfacl file.txt
```

- **sudo** – Execute a command as another user (usually root):

```
sudo command       # execute command as root
sudo -u user command  # execute command as specific user
```

- **visudo** – Edit the sudoers file safely:

```
sudo visudo         # edit sudoers file with proper syntax checking
```

- **passwd** – Change a user's password:

```
sudo passwd user    # change password for specific user
```

- **sudoers** – Manage sudo access for users:

```
sudo visudo         # add or modify sudo access for users
```

- **gpasswd** – Administer group password:

```
sudo gpasswd -a user group_name  # add user to group
```

- **ss** – Display socket statistics (for secure network connections):

```
ss -tuln            # display listening sockets
```

---

13. System Backup and Restore

- **rsync** – Remote file and directory synchronization:
  - **rsync -avz source/ destination/** – Synchronize files:

    ```
    rsync -avz source/ destination/
    ```

    - **rsync -avz -e ssh source/ user@remote:/destination/** – Sync over SSH:

      ```
      rsync -avz -e ssh source/ user@remote:/destination/
      ```

- **cpio** – Copy files to and from archives:

  ```
  cpio -o < file_list > archive.cpio  # create an archive
  cpio -i < archive.cpio              # extract files from an archive
  ```

- **dd** – Low-level copying and backup of entire filesystems:
    - **dd if=/dev/sda of=/path/to/backup.img** – Backup a disk/partition:

      ```
      dd if=/dev/sda of=/path/to/backup.img
      ```

    - **dd if=/path/to/backup.img of=/dev/sda** – Restore a disk/partition:

      ```
      dd if=/path/to/backup.img of=/dev/sda
      ```

14. System Diagnostics and Troubleshooting

- **dmesg** – Print the kernel ring buffer messages (system boot and hardware-related messages):

  ```
  dmesg
  dmesg | grep error       # filter for error messages
  dmesg -T                # show human-readable timestamps
  ```

- **journalctl** – Query and view logs from systemd's journal:

  ```
  journalctl
  journalctl -xe           # show detailed error messages
  journalctl -u service_name  # show logs for a specific service
  journalctl -f            # follow logs in real-time
  ```

- **strace** – Trace system calls and signals:

  ```
  strace <command>         # trace a specific command's system calls
  strace -p <pid>          # trace a specific process by PID
  strace -f <command>      # follow child processes
  ```

- **lsof** – List open files (useful for debugging):

  ```
  lsof                     # list all open files
  lsof /path/to/file       # show processes using a specific file
  lsof -i :80              # list processes using port 80
  ```

- **vmstat** – Report virtual memory statistics:

  ```
  vmstat                  # basic memory statistics
  vmstat 5                # update every 5 seconds
  ```

- **iostat** – Report CPU and I/O statistics:

```
iostat                   # basic CPU and I/O stats
iostat -x                # extended statistics
```

- **mpstat** – Report CPU usage statistics:

```
mpstat                   # basic CPU usage statistics
mpstat -P ALL            # CPU usage per core
```

- **pidstat** – Report statistics by process:

```
pidstat                  # general process stats
pidstat -u               # show CPU usage per process
pidstat -d               # show disk I/O stats per process
```

- **free** – Display memory usage:

```
free                     # general memory usage stats
free -h                  # human-readable format
free -m                  # in MB
```

- **uptime** – How long the system has been running:

```
uptime                   # show uptime
```

- **watch** – Execute a program periodically, showing output:

```
watch -n 1 free          # watch memory usage every second
watch -d ls              # watch directory contents and highlight
changes
```

- **lshw** – List hardware configuration:

```
lshw                     # list hardware configuration
lshw -short              # concise summary of hardware
lshw -C network          # show network devices only
```

- **htop** – Interactive process viewer (better than top):

```
htop                     # interactive process viewer
htop -u user             # show processes of a specific user
htop -d 2                # set refresh interval to 2 seconds
```

- **netstat** – Network statistics (deprecated in favor of ss):

```
netstat                  # general network statistics
netstat -tuln            # show listening ports and associated
processes
netstat -r               # show routing table
```

- **ss** – Show socket statistics (more efficient than netstat):

  ```
  ss                         # basic socket statistics
  ss -tuln                   # show listening TCP/UDP ports
  ss -p                      # show process using the socket
  ```

---

15. Networking & Remote Management

- **ifconfig** – Configure network interfaces (older command, replaced by ip):

  ```
  ifconfig                   # show network interfaces
  ifconfig eth0 up           # bring eth0 interface up
  ifconfig eth0 down         # bring eth0 interface down
  ifconfig -a                # show all interfaces
  ```

- **ip** – A more modern alternative for managing network interfaces and routing:
  - **ip addr** – Show IP addresses:

    ```
    ip addr                    # show IP addresses
    ip addr show eth0          # show IP for specific interface
    ```

  - **ip link** – Show or manipulate network interfaces:

    ```
    ip link                    # show interfaces
    ip link set eth0 up        # bring interface eth0 up
    ip link set eth0 down      # bring interface eth0 down
    ```

  - **ip route** – Show or manipulate routing tables:

    ```
    ip route                   # show routing table
    ip route add 192.168.1.0/24 via 192.168.0.1  # add a route
    ```

- **ss** – Display socket statistics (useful for diagnosing network issues):

  ```
  ss                         # general socket stats
  ss -tuln                   # show listening TCP/UDP ports
  ```

- **nmap** – Network exploration tool (can be used for security auditing):

  ```
  nmap 192.168.1.1           # scan a specific IP address
  nmap -p 80 192.168.1.1     # scan port 80 of a specific IP
  nmap -sP 192.168.1.0/24    # ping scan to discover live hosts
  ```

- **telnet** – User interface to the TELNET protocol (less common nowadays):

  ```
  telnet example.com         # connect to a host using TELNET
  ```

- **nc (Netcat)** – Network utility for reading and writing from network connections:
  - **nc -l -p 1234** – Listen on port 1234:

```
nc -l -p 1234
```

  - **nc <host> <port>** – Connect to a host and port:

    ```
    nc example.com 1234
    ```

- **iptables** – Administration tool for IPv4 packet filtering and NAT (Network Address Translation):

  ```
  sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT  # allow SSH
  sudo iptables -L               # list current rules
  sudo iptables -F               # flush all rules
  ```

- **firewalld** – Frontend for managing firewall rules (used in some distros like Fedora and CentOS):

  ```
  sudo firewall-cmd --add-port=80/tcp --permanent  # allow HTTP
  sudo firewall-cmd --reload  # apply changes
  ```

- **ufw** – Uncomplicated firewall (front-end for iptables):
  - **ufw enable** – Enable firewall:

    ```
    sudo ufw enable
    ```

  - **ufw allow <port>** – Allow traffic on a specific port:

    ```
    sudo ufw allow 80/tcp  # allow HTTP
    ```

- **tcpdump** – Command-line packet analyzer:

  ```
  sudo tcpdump -i eth0        # capture packets on eth0 interface
  sudo tcpdump -i eth0 port 80 # capture HTTP traffic
  ```

- **curl** – Transfer data from or to a server using various protocols (HTTP, FTP, etc.):

  ```
  curl http://example.com     # fetch data from a URL
  curl -O http://example.com/file.txt  # download a file
  curl -I http://example.com   # fetch headers only
  ```

- **wget** – Download files from the web via HTTP, HTTPS, FTP:

  ```
  wget http://example.com/file.txt  # download a file
  wget -c http://example.com/file.txt  # continue a paused download
  ```

- **scp** – Secure copy over SSH (used to copy files between systems):
  - **scp file.txt user@remote:/path/to/destination/** – Copy file to remote server:

    ```
    scp file.txt user@remote:/path/to/destination/
    ```

- **rsync** – Remote file and directory synchronization (often used for backups):
  - **rsync -avz /local/path/ remote:/remote/path/** – Sync directories:

    ```
    rsync -avz /local/path/ remote:/remote/path/
    ```

16. Text Processing Utilities

- **grep** – Search for patterns within files:
  - **`grep 'pattern' file.txt`** – Search for a pattern in a file:

    ```
    grep 'pattern' file.txt
    ```

  - **`grep -r 'pattern' /dir/`** – Recursively search for a pattern:

    ```
    grep -r 'pattern' /dir/
    ```

  - **`grep -i 'pattern' file.txt`** – Case-insensitive search:

    ```
    grep -i 'pattern' file.txt
    ```

  - **`grep -v 'pattern' file.txt`** – Exclude lines matching the pattern:

    ```
    grep -v 'pattern' file.txt
    ```

  - **`grep -l 'pattern' file.txt`** – Show filenames containing the pattern:

    ```
    grep -l 'pattern' file.txt
    ```

- **sed** – Stream editor for filtering and transforming text:
  - **`sed 's/old/new/g' file.txt`** – Replace old with new globally:

    ```
    sed 's/old/new/g' file.txt
    ```

  - **`sed -i 's/old/new/g' file.txt`** – In-place edit (modify the file directly):

    ```
    sed -i 's/old/new/g' file.txt
    ```

- **awk** – A powerful text processing language:
  - **`awk '{print $1}' file.txt`** – Print the first column of each line in a file:

    ```
    awk '{print $1}' file.txt
    ```

  - **`awk -F':' '{print $1}' file.txt`** – Set a custom delimiter (e.g., colon):

    ```
    awk -F':' '{print $1}' file.txt
    ```

- **cut** – Remove sections from each line of a file:

- `cut -d ':' -f 1 /etc/passwd` – Print the first field of each line, delimited by ":":

  ```
  cut -d ':' -f 1 /etc/passwd
  ```

- `cut -c 1-5 file.txt` – Cut specific character positions from each line:

  ```
  cut -c 1-5 file.txt
  ```

- **sort** – Sort lines of text files:
  - `sort file.txt` – Sort file content in ascending order:

    ```
    sort file.txt
    ```

  - `sort -r file.txt` – Sort in reverse order:

    ```
    sort -r file.txt
    ```

- **uniq** – Report or omit repeated lines in a file:
  - `sort file.txt | uniq` – Sort and remove duplicate lines:

    ```
    sort file.txt | uniq
    ```

  - `uniq -c file.txt` – Count occurrences of each line:

    ```
    uniq -c file.txt
    ```

- **tee** – Read from standard input and write to standard output and files:
  - `echo "text" | tee file.txt` – Write to file and show output on screen:

    ```
    echo "text" | tee file.txt
    ```

- **tr** – Translate or delete characters:
  - `echo "hello" | tr 'a-z' 'A-Z'` – Convert lowercase to uppercase:

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

  - `echo "hello" | tr -d 'a'` – Delete character 'a' from the input:

    ```
    echo "hello" | tr -d 'a'
    ```

- **paste** – Merge lines of files:
  - `paste file1.txt file2.txt` – Combine lines of file1 and file2 side by side:

    ```
    paste file1.txt file2.txt
    ```

- **wc** – Word, line, character, and byte count:
  - `wc -l file.txt` – Count lines in a file:

```
wc -l file.txt
```

  - **`wc -w file.txt`** – Count words in a file:

```
wc -w file.txt
```

---

17. System Shutdown and Reboot

- **shutdown** – Shut down the system:
  - **`shutdown -h now`** – Immediately shut down:

```
sudo shutdown -h now
```

  - **`shutdown -r now`** – Reboot the system:

```
sudo shutdown -r now
```

  - **`shutdown -h +10`** – Shut down after 10 minutes:

```
sudo shutdown -h +10
```

- **reboot** – Reboot the system:

```
sudo reboot
```

- **halt** – Halt the system immediately (equivalent to turning off power):

```
sudo halt
```

- **poweroff** – Power off the system:

```
sudo poweroff
```

- **init** – Change the runlevel (old-style system manager):
  - **`init 0`** – Shutdown:

```
sudo init 0
```

  - **`init 6`** – Reboot:

```
sudo init 6
```

18. File System Mounting and Management

- **mount** – Mount a file system:
  - **`mount /dev/sda1 /mnt`** – Mount partition to a directory:

```
mount /dev/sda1 /mnt
```

  - **`mount -t ext4 /dev/sda1 /mnt`** – Mount a partition with a specific file system type:

```
mount -t ext4 /dev/sda1 /mnt
```

  - **`mount -o loop file.iso /mnt`** – Mount an ISO file:

```
mount -o loop file.iso /mnt
```

- **umount** – Unmount a file system:
  - **`umount /mnt`** – Unmount the file system mounted at /mnt:

```
umount /mnt
```

- **fstab** – File system table (configuration file for mounting file systems):
  - **`/etc/fstab`** – View and configure persistent mount points:

```
cat /etc/fstab
```

- **blkid** – Display block device attributes:

```
blkid
blkid /dev/sda1      # get details about a specific device
```

- **fsck** – Check and repair a file system:
  - **`fsck /dev/sda1`** – Check and repair /dev/sda1:

```
sudo fsck /dev/sda1
```

  - **`fsck -A`** – Check all file systems mentioned in `/etc/fstab`:

```
sudo fsck -A
```

---

19. Filesystem Permissions and Security

- **chmod** – Change file permissions:
  - **`chmod 755 file.txt`** – Give read, write, and execute permissions to owner, and read-execute permissions to others:

```
chmod 755 file.txt
```

- **`chmod u+x file.txt`** – Add execute permission to the owner:

```
chmod u+x file.txt
```

- **`chmod -R 755 directory/`** – Apply permissions recursively to a directory:

```
chmod -R 755 directory/
```

- **chown** – Change file owner and group:
  - **`chown user:group file.txt`** – Change owner and group of a file:

```
chown user:group file.txt
```

  - **`chown -R user:group directory/`** – Change owner and group recursively:

```
chown -R user:group directory/
```

- **chgrp** – Change group ownership of a file:
  - **`chgrp group file.txt`** – Change the group of a file:

```
chgrp group file.txt
```

- **umask** – Set default permissions for new files:
  - **`umask 022`** – Set default permissions for newly created files to 755:

```
umask 022
```

- **setfacl** – Set access control lists (ACL) for file permissions:
  - **`setfacl -m u:username:rwx file.txt`** – Give read, write, and execute permissions to a specific user:

```
setfacl -m u:username:rwx file.txt
```

  - **`setfacl -m g:groupname:rx file.txt`** – Give read and execute permissions to a specific group:

```
setfacl -m g:groupname:rx file.txt
```

- **getfacl** – Get access control lists (ACL) for file permissions:
  - **`getfacl file.txt`** – View ACL for a file:

    ```
    getfacl file.txt
    ```