

# Linux Operating System:

## 1. Introduction to Linux

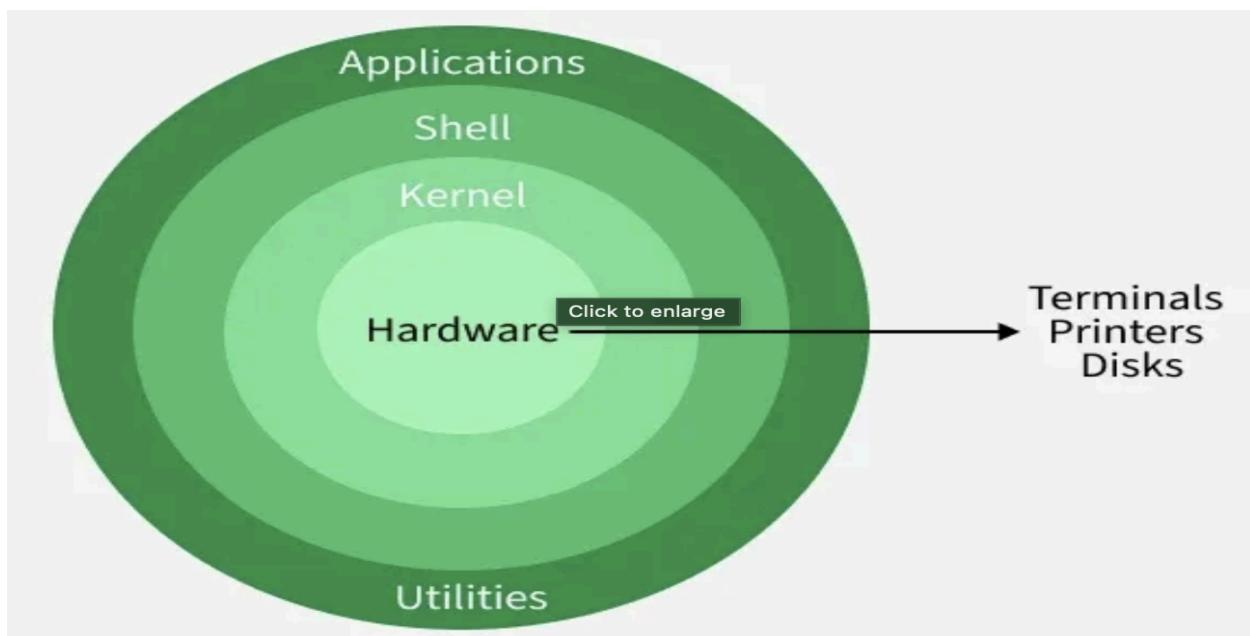
Linux is a free and open-source operating system based on **UNIX**, a powerful multitasking and multi-user system developed in the 1970s at AT&T Bell Labs.

### Key Characteristics

- **Open-Source:** The source code is accessible to everyone, promoting global innovation and collaboration.
- **Performance & Security:** Known for efficient resource management and robust security features.
- **Versatility:** Works across a vast range of devices, from embedded systems to supercomputers.
- **Completeness:** Combines the kernel with file systems, utilities, and applications to create a functional computing environment.

## 2. Linux Architecture

The architecture of Linux is a layered structure that defines how hardware and software components interact.



1. **Hardware Layer:** The physical components (CPU, RAM, Storage).
2. **Kernel:** The core of the OS. It manages hardware resources and acts as the communication bridge between hardware and software.
  - *Types:* Monolithic, Microkernel, Hybrid, Exokernel.
3. **System Libraries:** Pre-written code that allows applications to access kernel features without direct interaction.
4. **Shell:** The Command-Line Interface (CLI) that interprets user commands and sends them to the kernel.
5. **System Utilities & Applications:** Tools for management (installing software, user accounts) and end-user tasks (web browsers, office suites).

### 3. Understanding Distributions (Distros)

A **Linux Distribution** is a complete operating system built around the Linux kernel. With over 600+ distros available, users can choose based on stability, performance, or ease of use.

#### Popular Distributions

- **General Use:** Ubuntu, Linux Mint, Debian, Fedora.
- **Advanced/Custom:** Arch Linux, Gentoo.
- **Specialized:** Kali Linux (Security), MX Linux, openSUSE.

### 4. Applications of Linux

Linux is dominant across various industries due to its stability and open-source nature.

Domain	Key Use Case
<b>Servers &amp; Cloud</b>	Powers the majority of the world's web servers and data centers.
<b>Development</b>	Offers a native environment for coding, debugging, and multiple languages.
<b>Cybersecurity</b>	The gold standard for ethical hacking and penetration testing (e.g., Kali Linux).
<b>Embedded Systems</b>	Used in IoT devices, routers, and smart appliances due to its efficiency.
<b>Supercomputers</b>	Used almost exclusively for massive scientific calculations and simulations.

## 5. Selecting the Right Distribution

Choosing a distro depends entirely on your goals and experience level.

### Selection Guide

- **For Beginners:** **Ubuntu** (high support) or **Linux Mint** (Windows-like feel).
- **For Developers:** **Fedora** (latest tech) or **Debian** (rock-solid stability).
- **For Servers:** **Ubuntu Server** or **CentOS/RHEL** (enterprise stability).
- **For Security:** **Kali Linux** (pre-installed forensics and hacking tools).
- **For Older Hardware:** **Lubuntu** or **Puppy Linux** (extremely lightweight).

## 6. Importance of Linux

- **Cost-Effective:** Zero licensing fees for the core OS.
- **Customizable:** You can modify the system at every level to suit your needs.
- **Community Support:** Backed by a massive global community providing constant updates and troubleshooting.

# Linux Commands:

- **sudo su**: This command switches you to the **SuperUser** (root) account. **sudo** gives you administrative privileges, and **su** stands for "switch user." It is used when you need to perform multiple administrative tasks without typing **sudo** before every command.

```
ubuntu:~$ sudo su
ubuntu:~$ whoami
root
ubuntu:~$ pwd
/root
ubuntu:~$
```

- **whoami**: A simple but vital command that prints the **username** of the current user. It's helpful when you've switched between multiple accounts (like **root** and a standard user) and need to verify your current identity.

```
ubuntu:~$ whoami
root
```

- **touch**: Used to create a new, **empty file**. It is also used to update the access and modification timestamps of existing files.

```
ubuntu:~$ touch file1 file2
ubuntu:~$ ls
file1  file2  filesystem
```

- **ls**: lists the **files and folders** in a directory

```
ubuntu:~$ ls  
file1 file2 filesystem  
ubuntu:~$
```

- **rm:** Stands for **remove**. It deletes files. By default, it does not delete directories unless specified.

```
ubuntu:~$ rm file1  
ubuntu:~$ ls  
file2 filesystem  
ubuntu:~$
```

- **vi:** Open file if present else creates one. We can insert the data by clicking **i** then paste the data the **:wq** to save

```
ubuntu:~$ vi file1  
ubuntu:~$ cat file1  
Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.  
ubuntu:~$
```

- **nano:** A simpler, more user-friendly terminal text editor compared to **vi**. It displays the keyboard shortcuts at the bottom of the screen, making it great for beginners.

```
ubuntu:~$ nano file1
```

```
GNU nano 7.2                                         file1
Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.

[ Read 1 line ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo    M-A Set Mark
^X Exit      ^R Read File   ^Y Replace     ^U Paste      ^J Justify    ^G Go To Line M-E Redo    M-G Copy
```

- **cat**: Stands for **concatenate**. While it can merge files, its most common use is to quickly **display the entire contents** of a file directly in the terminal window.

```
ubuntu:~$ cat file1
Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.
```

- **mkdir**: Stands for **make directory**. It creates a new folder.

```
ubuntu:~$ mkdir dir1
ubuntu:~$ ls
dir1  file1  filesystem
ubuntu:~$
```

- **rmdir**: Stands for **remove directory**. This command only works if the directory is **empty**. If there are files inside, you must use **rm -r**.

```
ubuntu:~/dir1$ rmdir dir1.1/  
rmdir: failed to remove 'dir1.1/': Directory not empty  
ubuntu:~/dir1$ cd ..  
ubuntu:~$ rmdir dir2  
ubuntu:~$ ls  
dir1  dir3  filesystem  
ubuntu:~$
```

- **rm -rf**: The "nuclear" option.
  - **-r** (recursive) deletes a directory and everything inside it.
  - **-f** (force) ignores non-existent files and never prompts for confirmation.
  - **Caution:** In the industry, using this on a production server without checking your path first is a major risk.

```
ubuntu:~$ ls  
dir1  dir3  filesystem  
ubuntu:~$ rm -rf dir1  
ubuntu:~$ ls  
dir3  filesystem  
ubuntu:~$
```

- **mv**: Stands for **move**. It is used to move files or directories from one location to another. It is also the standard way to **rename** a file in Linux.

```
ubuntu:~$ mv file1 file  
ubuntu:~$ ls  
dir3  file  filesystem
```

```
ubuntu:~$ mv file dir3
ubuntu:~$ ls
dir3  filesystem
ubuntu:~$ cd dir3/
ubuntu:~/dir3$ ls
file
ubuntu:~/dir3$
```

- **cp**: Stands for **copy**. It creates a duplicate of a file or directory. To copy a directory, you usually need the **-r** flag.

```
ubuntu:~$ cp filee dir2
ubuntu:~$ ls
dir1  dir2  dir3  filee  filesystem
ubuntu:~$ cd dir2/
ubuntu:~/dir2$ ls
filee
ubuntu:~/dir2$
```

- **cd**: Stands for **change directory**. It is how you navigate through the file system (e.g., **cd /var/log**).

```
ubuntu:~/dir2$ pwd
/root/dir2
ubuntu:~/dir2$ cd ..
ubuntu:~$ cd dir2/
ubuntu:~/dir2$ pwd
/root/dir2
ubuntu:~/dir2$
```

- **pwd**: Stands for **print working directory**. It displays the full absolute path of the directory you are currently in.

```
ubuntu:~/dir2$ pwd
/root/dir2
```

- **wget**: A command-line utility for **downloading files** from the internet via protocols like HTTP, HTTPS, and FTP.
  - *Example:* `wget https://example.com/file.zip`

```
ubuntu:~/dir2$ wget www.google.com
--2026-01-30 06:24:31-- http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.20.36, 2a00:1450:4007:80f::2004
Connecting to www.google.com (www.google.com)|172.217.20.36|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html                                              [ =>                               ] 19.74K  --.-KB/s   in 0.01s

2026-01-30 06:24:31 (1.56 MB/s) - 'index.html' saved [20212]

ubuntu:~/dir2$ ls
filee  index.html
ubuntu:~/dir2$
```

- **ping**: Used to test the **reachability** of a host (like a server or website) on an IP network. It sends small packets of data and measures how long it takes for them to return.

```
ubuntu:~/dir2$ ping www.google.com
PING www.google.com (172.217.20.36) 56(84) bytes of data.
64 bytes from arn11s01-in-f4.1e100.net (172.217.20.36): icmp_seq=1 ttl=115 time=13.2 ms
64 bytes from arn11s01-in-f4.1e100.net (172.217.20.36): icmp_seq=2 ttl=115 time=12.8 ms
64 bytes from arn11s01-in-f4.1e100.net (172.217.20.36): icmp_seq=3 ttl=115 time=15.1 ms
64 bytes from arn11s01-in-f4.1e100.net (172.217.20.36): icmp_seq=4 ttl=115 time=12.7 ms
64 bytes from arn11s01-in-f4.1e100.net (172.217.20.36): icmp_seq=5 ttl=115 time=13.0 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 12.704/13.353/15.060/0.867 ms
ubuntu:~/dir2$
```

## SYSTEM COMMANDS:

- **uname**: Displays the operating system name (e.g., Linux).

```
ubuntu:~$ uname  
Linux
```

- **uname -r**: Prints the **Kernel Release** version. This is critical when checking if a server is compatible with specific software like Docker.

```
ubuntu:~$ uname -r  
6.8.0-90-generic
```

- **uname -a**: The "All" flag. It provides a complete summary: OS name, hostname, kernel version,

```
ubuntu:~$ uname -a  
Linux ubuntu 6.8.0-90-generic #91-Ubuntu SMP PREEMPT_DYNAMIC Tue Nov 18 14:14:30 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

- **clear** (or **Ctrl + L**): Wipes the current terminal screen of previous text to give you a clean workspace.
- **who**: Lists all users currently logged into the system, including where they are logged in from and for how long.

```
ubuntu:~$ who  
ubuntu:~$ whoami  
root
```

- **whoami**: Returns the username of the effective user you are currently operating as.

```
ubuntu:~$ who
ubuntu:~$ whoami
root
```

- **uptime**: Shows how long the system has been running, the number of users, and the **load average** (a vital stat for SREs to check if a server is overloaded).

```
ubuntu:~$ uptime
04:44:08 up 1:09, 0 user, load average: 0.01, 0.00, 0.00
```

- **uptime -p**: Displays the uptime in a "pretty" or simplified format (e.g., "up 2 hours, 30 minutes").

```
ubuntu:~$ uptime -p
up 1 hour, 9 minutes
```

- **date**: Displays the current system date, time, and timezone.

```
ubuntu:~$ date
Sat Jan 31 04:46:18 UTC 2026
```

- **timedatectl**: A modern tool to view and change the system clock, timezone, and network time synchronization (NTP) settings.

```
ubuntu:~$ timedatectl
          Local time: Sat 2026-01-31 04:46:28 UTC
          Universal time: Sat 2026-01-31 04:46:28 UTC
                  RTC time: Sat 2026-01-31 04:46:28
                 Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
          NTP service: active
       RTC in local TZ: no
```

- **timedatectl set-timezone [Zone]**: Permanently changes the system's timezone.
  - *Example:* `timedatectl set-timezone Asia/Kolkata`

```
ubuntu:~$ timedatectl set-timezone Asia/Kolkata
```

```
ubuntu:~$ date
Sat Jan 31 10:30:39 IST 2026
```

- **hostname**: Displays the system's network name.

```
ubuntu:~$ hostname
ubuntu
```

- **hostname -i**: Displays the IP address(es) associated with the hostname.

```
ubuntu:~$ hostname -i
127.0.0.1
```

- **hostnamectl set-hostname [Name]**: A persistent way to change the server's name. This updates the configuration files so the name stays changed even after a reboot.

```
ubuntu:~$ hostnamectl set-hostname "swiggy"
ubuntu:~$ hostname
swiggy
```

- **ip addr**: The modern replacement for **ifconfig**. It shows all network interfaces and their assigned IP addresses.

```
ubuntu:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 2a:11:8e:9f:da:7f brd ff:ff:ff:ff:ff:ff
        inet 172.30.1.2/24 brd 172.30.1.255 scope global dynamic noprefixroute enp1s0
            valid_lft 86309374sec preferred_lft 75520174sec
        inet6 fe80::4511:769f:a2b7:c398/64 scope link
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1454 qdisc noqueue state DOWN group default
    link/ether 56:75:bb:65:9d:29 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
```

- **ip route**: Shows the **routing table**, telling you how traffic leaves the system and what the "Default Gateway" is.

```
ubuntu:~$ ip route
default via 172.30.1.1 dev enp1s0 proto dhcp src 172.30.1.2 metric 1002 mtu 1500
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.30.1.0/24 dev enp1s0 proto dhcp scope link src 172.30.1.2 metric 1002 mtu 1500
```

- **ifconfig**: A legacy tool used to view or configure network interfaces. Note that many modern Linux distros require you to install **net-tools** to use this.

```
ubuntu:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1454
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 56:75:bb:65:9d:29 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.30.1.2 netmask 255.255.255.0 broadcast 172.30.1.255
    inet6 fe80::4511:769f:a2b7:c398 prefixlen 64 scopeid 0x20<link>
        ether 2a:11:8e:9f:da:7f txqueuelen 1000 (Ethernet)
        RX packets 26559 bytes 33369468 (33.3 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 9834 bytes 14842159 (14.8 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 84 bytes 6264 (6.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 84 bytes 6264 (6.2 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- **ps**: Stands for **Process Status**. By default, it shows processes running in the current terminal session.
  - *Industry Tip*: Use `ps -ef` or `ps aux` to see every process running on the entire system.

```
ubuntu:~$ ps
  PID TTY          TIME CMD
1865 pts/0    00:00:00 bash
2196 pts/0    00:00:00 ps
```

- **kill -9 [PID]**: Sends a "SIGKILL" signal to a process ID. The `-9` flag is the "force" option, used to immediately stop a process that is frozen or won't close normally.

```
ubuntu:~$ kill -9 PID
bash: kill: PID: arguments must be process or job IDs
ubuntu:~$ ps
  PID TTY          TIME CMD
1865 pts/0    00:00:00 bash
2196 pts/0    00:00:00 ps
```

Using the `+` sign allows you to extract specific parts of the timestamp, which is incredibly useful for **shell scripting** (like naming a backup file with today's date).

Command	Output Description	Example
<code>date</code> <code>+ "%d"</code>	Day of the month	31
<code>date</code> <code>+ "%m"</code>	Month number	01
<code>date</code> <code>+ "%y"</code>	Last two digits of year	26

<b>date</b> +"%H"	Hour (24-hour format)	10
<b>date</b> +"%M"	Minute	07
<b>date</b> +"%S"	Seconds	46
<b>date</b> +"%D"	Date as MM/DD/YY	01/31/26
<b>date</b> +"%F"	Full date (YYYY-MM-DD)	2026-01- 31
<b>date</b> +"%A"	Full name of the weekday	Saturday
<b>date</b> +"%B"	Full name of the month	January

```
ubuntu:~$ date
Sat Jan 31 10:18:13 IST 2026
ubuntudate +"%d""%d"
"31"
ubuntudate +"%m""%m"
"01"
ubuntudate +"%y""%y"
"26"
ubuntudate +"%H""%H"
"10"
ubuntudate +"%M""%M"
"18"
ubuntudate +"%S""%S"
"04"
ubuntu:~$ date +"%D"
01/31/26
ubuntudate +"%F""%F"
"2026-01-31"
ubuntudate +"%A""%A"
"Saturday"
ubuntudate +"%B""%B"
"January"
ubuntu:~$ date
Sat Jan 31 10:30:39 IST 2026
```

## HARDWARE COMMANDS:

- **lscpu**: This is the most user-friendly way to view CPU details. It gathers information from the system architecture and displays the number of CPUs, cores, threads, and the CPU model name in a clean, structured list.

```
ubuntu:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         39 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                1
On-line CPU(s) list:  0
Vendor ID:             GenuineIntel
BIOS Vendor ID:       Red Hat
Model name:            Intel Xeon E312xx (Sandy Bridge, IBRS update)
BIOS Model name:      RHEL-9.6.0 PC (Q35 + ICH9, 2009) CPU @ 2.0GHz
BIOS CPU family:      1
CPU family:            6
Model:                 42
Thread(s) per core:   1
Core(s) per socket:   1
Socket(s):             1
Stepping:              1
BogoMIPS:              7008.00
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc rep_good noopl xtopology cpuid tsc_known_freq pni pclmulqdq sse3 cx16 pdid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx hypervisor lahf_lm cpuid_fault pti ssbd ibrs ibpb stibp tsc_adjust xsaveopt arat md_clear

Virtualization features:
Hypervisor vendor:    KVM
Virtualization type:  full
Caches (sum of all):
L1d:                  32 KiB (1 instance)
L1i:                  32 KiB (1 instance)
L2:                   4 MiB (1 instance)
L3:                   16 MiB (1 instance)
NUMA:
NUMA node(s):          1
NUMA node0 CPU(s):     0
Vulnerabilities:
Gather data sampling: Not affected
Itlb multihit:         KVM: Mitigation: VMX unsupported
L1tf:                  Mitigation; PTE Inversion
Mds:                   Mitigation; Clear CPU buffers; SMT Host state unknown
```

- **cat /proc/cpuinfo**: This command reads directly from the virtual file system. It provides a highly detailed, technical breakdown for *every single logical core* in the system. If you have 16 cores, it will print 16 long blocks of text.
  - **Industry Use:** Use **lscpu** for a quick summary and **/proc/cpuinfo** if you need to find specific hardware flags or cache sizes.

```
ubuntu:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 42
model name     : Intel Xeon E312xx (Sandy Bridge, IBRS update)
stepping        : 1
microcode      : 0x1
cpu MHz         : 3504.000
cache size      : 16384 KB
physical id    : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid : 0
fpu             : yes
fpu_exception   : yes
cpuid level    : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscal
l nx rdtscp lm constant_tsc rep_good nopl xtopology cpuid tsc_known_freq pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2 x2api
c popcnt tsc_deadline_timer aes xsave avx hypervisor lahf_lm cpuid_fault pti ssbd ibrs ibpb stibp tsc_adjust xsaveopt arat
md_clear
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs itlb_multihit mmio_unknown bhi
bogomips        : 7008.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

ubuntu:~$ lsblk -a
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0   0B  0 loop
loop1  7:1    0   0B  0 loop
loop2  7:2    0   0B  0 loop
loop3  7:3    0   0B  0 loop
loop4  7:4    0   0B  0 loop
loop5  7:5    0   0B  0 loop
```

- **lsblk -a**: Lists all "Block Devices" (hard drives, SSDs, partitions, and even RAM disks). The **-a** flag ensures it shows empty devices as well.
  - **Key Insight:** It shows a "tree" structure, making it easy to see which partition belongs to which physical disk.
  - **Industry Use:** Essential when you are adding a new volume to a cloud instance (like an AWS EBS volume) to verify the system "sees" the new hardware.

```
ubuntu:~$ lsblk -a
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0   0B  0 loop
loop1  7:1    0   0B  0 loop
loop2  7:2    0   0B  0 loop
loop3  7:3    0   0B  0 loop
loop4  7:4    0   0B  0 loop
loop5  7:5    0   0B  0 loop
loop6  7:6    0   0B  0 loop
loop7  7:7    0   0B  0 loop
vda   253:0   0   20G 0 disk
|---vda1 253:1   0   19G 0 part /
|---vda14 253:14 0   4M  0 part
|---vda15 253:15 0   106M 0 part /boot/efi
`---vda16 253:0   0   913M 0 part /boot
```

- **free**: Displays the total amount of free and used physical memory (RAM) and swap memory in the system. By default, it displays values in **Kilobytes (KB)**, which can be hard for humans to read quickly.

```
ubuntu:~$ free
total        used         free      shared  buff/cache   available
Mem:       1948940      483732      501892          1064     1160980      1465208
Swap:      1048572           0      1048572
```

- **cat /proc/meminfo**: Like the CPU version, this reads directly from the kernel's data. it provides a massive list of memory statistics, including "Dirty" memory, "Cached" memory, and "Buffers."

```
ubuntu:~$ cat /proc/meminfo
MemTotal:              1948940 kB
MemFree:               501892  kB
MemAvailable:          1465244  kB
Buffers:                125904  kB
Cached:                 784380  kB
SwapCached:              0      kB
Active:                 456956  kB
Inactive:               604204  kB
Active(anon):           160692  kB
Inactive(anon):            0      kB
Active(file):            296264  kB
Inactive(file):          604204  kB
Unevictable:             27284   kB
Mlocked:                27284   kB
SwapTotal:               1048572 kB
SwapFree:                1048572 kB
Zswap:                  0      kB
Zswapped:                0      kB
Dirty:                   12     kB
Writeback:                0      kB
AnonPages:               178208 kB
Mapped:                  170732 kB
Shmem:                   1064   kB
KReclaimable:            250732 kB
Slab:                     308716 kB
SReclaimable:            250732 kB
SUnreclaim:               57984  kB
KernelStack:                3004  kB
PageTables:                6840  kB
SecPageTables:                0    kB
NFS_Unstable:                0    kB
Bounce:                  0      kB
WritebackTmp:                0    kB
CommitLimit:              2023040 kB
Committed_AS:              877704 kB
VmallocTotal:              34359738367 kB
VmallocUsed:                16228 kB
VmallocChunk:                  0   kB
```

- **free -m**: The same as **free**, but it converts all values into **Megabytes (MB)**.
  - **Industry Use:** In 2026, most engineers prefer **free -h** (human-readable), which automatically switches between MB, GB, and TB so you don't have to do the math in your head.

```
ubuntu:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:       1903         472         490           1        1133        1430
Swap:      1023          0       1023
```

- **df -h**: Stands for "disk free." It reports how much space is used and available on all mounted file systems.
  - **The -h flag:** "Human-readable" format (shows 50G instead of 52428800).
  - **Industry Use:** This is a "must-know" command. If your **MySQL** database stops working, the first thing you check with **df -h** is if the disk is 100% full.

```
ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            191M  980K  190M   1% /run
/dev/vda1        19G   5.2G   14G  29% /
tmpfs            952M   84K  952M   1% /dev/shm
tmpfs             5.0M    0   5.0M   0% /run/lock
/dev/vda16       881M  117M  703M  15% /boot
/dev/vda15       105M   6.2M   99M   6% /boot/efi
ubuntu:~$
```

## GREP (Global Regular Expression Print):

```
ubuntu:~$ ls
filesystem
ubuntu:~$ vi file
ubuntu:~$ cat
^C
ubuntu:~$ cat file
Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.
ubuntu:~$
```

- **grep "word" filename:** The basic search. It scans the file and prints every line that contains a match for "word."

```
ubuntu:~$ grep "Linux" file
1:Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.
```

- **grep -n "word" filename:** Adds the **line number** from the file to each matching line.
- **Industry Use:** Very helpful when you find an error in a long configuration file and need to know exactly where to go to fix it.

```
ubuntu:~$ grep -n "Linux" file
1:Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.
```

- **grep -c "word" filename:** Instead of showing the lines, it gives you a **count** of how many lines contained the match.

```
ubuntu:~$ grep -c "Linux" file
1
```

- **grep -i "word" filename:** Performs a **case-insensitive** search. It will match "Word", "WORD", or "wOrD".

```
ubuntu:~$ grep -i "linux" file
1:Linux commands allow users to control and manage the system using the terminal, which works similarly to the Command Prompt in Windows. These commands are case-sensitive and are commonly used for performing system and administrative tasks efficiently.
```

- **cat /etc/passwd**: Displays the raw system file containing user account information. Each line represents a user and includes their UID, GID, home directory, and default shell.
- **getent passwd**: A more robust command that "gets entries" from the password database. While it often shows the same as **/etc/passwd**, it is preferred in professional environments because it can also pull users from external sources like LDAP or Active Directory.
- **useradd [username]**: Creates a new user account.
  - **The "Home" Note:** By default, it creates a personal directory at **/home/ [username]**. This is where the user stores their personal files and configurations.
  - **The "Group" Note:** Most modern Linux distributions use **User Private Groups (UPG)**. This means for every user added, a group with the exact same name is created, and the user is made the only member of that group.
- **useradd -M [username]**: The **-M** flag stands for "**No Home Directory**."
  - **Industry Use:** This is commonly used for **System Users** (like a **mysql** or **nginx** user). Since these "users" are just used to run background services and won't be logging in to type commands, they don't need a **/home** folder.
- **userdel [username]**: Deletes the user account from the system and removes their name from the **/etc/group** file.
  - **The "Safety" Note:** It leaves the **/home** folder untouched as a safety measure so that important data isn't accidentally deleted.
- **userdel -r [username]**: The **-r** flag stands for "**Recursive**." It deletes the user account AND their home directory and mail spool simultaneously.
- **passwd [username]**: Assigns or changes a password for a user. Without a username, it changes the password for the account you are currently using.

- **su - [username]**: Stands for "Switch User."