

Linux commands

1. man top

- Description:
The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel.
- Uptime(how long system has been running since the last reboot) and Load Averages(how busy CPU is over different time periods): This portion consists of a single line containing:
 - program or window name, depending on display mode
 - current time and length of time since last boot
 - total number of users
 - system load average over the last 1, 5 and 15 minutes
- Tasks/Threads:
 - This line shows the **total number of tasks (or threads)** currently running on the system, depending on the state of the Threads-mode toggle.
 - The total tasks is further classified into:
 - **Running** → Actively using CPU right now.
 - **Sleeping** → Waiting for an event (like input from a file or network).
 - **Stopped** → Temporarily halted (e.g., by a debugger).
 - **Zombie** → A finished process whose parent hasn't collected its exit status.
- CPU state:

Code	Function	Explanation
us, user	%CPU time, running un-niced user processes.	Includes apps like browsers, text editors. If high your applications are using too much CPU.
sy, system	%CPU time, running kernel processes.	Includes system services like networking, drivers. If high, OS kernel is doing a lot of processing.
ni, nice	Time running niced(low priority) user processes	If a program has been "niced" (given lower priority), its CPU usage appears here.
id, idle	Time spent in kernel idle handler	CPU is not doing anything. High idle means low CPU usage; low idle means CPU is busy.
wa, IO-wait	Time waiting for I/O completion	If high, the system is waiting for disk/network operations to complete, indicating slow storage or overloaded disk operations.
hi	Time spent servicing hardware interrupts	CPU doesn't constantly check if a device (like a keyboard, mouse, or network card) needs attention. Instead, hardware devices send signals called interrupts to the CPU when they need processing.
si	Time spent servicing software interrupts	Related to network processing and system calls.
st	Time stolen from this VM by hypervisor	Shows how much CPU time your VM wanted to use but couldn't because the host system (the real computer running the VM) took it for something else.

- On **multi-core CPUs**, **top** can show separate CPU usage lines for each core.

- **Useful for WiFi debugging:** If one core is overloaded while others are idle, a single process (e.g., a faulty network daemon) might be causing issues.
- CPU usage breakdown in **top**.
 - In the alternate cpu states display modes, beyond the first tasks/threads line, an abbreviated summary is shown consisting of these elements:

```

          a  b   c  d
%Cpu(s): 75.0/25.0 100[ ...

```

Where:

- a) is the combined us and ni percentage; (75% of the CPU is being used by user programs)
- b) is the sy percentage; (25% of the CPU is being used by system tasks)
- c) is the total; (The total CPU usage (75% + 25% = 100% in this case))
- d) is one of two visual graphs of those representations; (visual representation of CPU usage)

- Memory Usage:

- Line 1 reflects physical memory, classified as: total, free, used and buff/cache
- Line 2 reflects mostly virtual memory, classified as: total, free, used and avail
- Avail: estimation of physical memory available for starting new applications, without swapping(no need to swap space).
- Free: Completely unused memory
- In the alternate memory display modes, two abbreviated summary lines are shown consisting of these elements:

```

          a    b      c
GiB Mem : 18.7/15.738 [ ...      (Mem: system's actual physical memory)
GiB Swap: 0.0/7.999  [ ...      (Swap(VM): used when RAM is full.)

```

Where:

- a) is the percentage used;
- b) is the total available;
- c) is one of two visual graphs of those representations.
- This table may help in interpreting the scaled values displayed:
 - KiB = kibibyte = 1024 bytes
 - MiB = mebibyte = 1024 KiB
 - GiB = gibibyte = 1024 MiB
 - TiB = tebibyte = 1024 GiB
 - PiB = pebibyte = 1024 TiB
 - EiB = exbibyte = 1024 PiB

- Fields/Column Usage:

- %CPU: The percentage of CPU time used by a process.
 - If a process has **%CPU = 50**, it means the process is using **50% of one CPU core**.
 - If you have **4 cores**, the total CPU usage can go up to **400%**.
- %MEM: This indicates how much of the **total physical RAM** a process is using.
 - If a process has **%MEM = 10**, it means it is using **10% of total RAM**.
 - Higher memory usage could slow down other applications.
- COMMAND:

2. man lscpu

- Description:
 - The `lscpu` command retrieves CPU architecture details from system files and architecture-specific libraries.
 - It provides information on CPUs, threads, cores, sockets, NUMA nodes, caches, CPU family, model, and other attributes.
 - In virtualized environments, it displays the guest OS configuration, which may differ from the host system.
 - The output can be formatted for readability or parsing, with customizable column selection. Not all architectures support every column.
 - The default terminal output is optimized for readability, while non-terminal output follows a structured "Field: data" format.
- Columns: topology elements (core, socket, etc.) use a sequential unique ID starting from zero, but CPU logical numbers follow the kernel where there is no guarantee of sequential numbering.
 - CPU: Logical CPU number used by the Linux kernel.
 - CORE: Logical core number (a core may contain multiple CPUs).
 - SOCKET: Logical socket number (a socket may contain multiple cores).
 - NODE: Logical NUMA node number (a node may contain multiple drawers).
 - CACHE: Details on CPU cache sharing.
 - ONLINE: Indicates if the CPU is currently in use by the system.
 - MAXMHZ / MINMHZ: Maximum and minimum clock speed of the CPU in MHz.
- Options:
 - `-a, --all` → Show both online and offline CPUs (only with `-e` or `-p`).
 - `-B, --bytes` → Display sizes in bytes instead of human-readable format.
 - `-b, --online` → Show only online CPUs (only with `-e` or `-p`).
 - `-C, --caches[=list]` → Show CPU cache details. Example: `-C=NAME,ONE-SIZE`.
 - `-c, --offline` → Show only offline CPUs (only with `-e` or `-p`).
 - `-e, --extended[=list]` → Display detailed CPU info. Example: `-e=cpu,node`.
 - `-h, --help` → Show help text and exit.
 - `-J, --json` → Output in JSON format.
 - `-p, --parse[=list]` → Optimized output for parsing. Example: `-p=cpu,node`.
 - `-s, --sysroot directory` → Gather CPU data from another Linux instance.
 - `-x, --hex` → Display CPU sets in hexadecimal.
 - `-y, --physical` → Show physical IDs for topology elements (e.g., cores, sockets).
 - `-V, --version` → Show version information.
 - `--output-all` → Show all columns (must be used with `--extended`, `--parse`, or `--caches`).
- Bugs:
 - The basic overview of CPU family, model, etc. is always based on the first CPU.
 - Sometimes in Xen Dom0, the kernel reports wrong data.
 - On virtual hardware, the number of cores per socket, etc. can be wrong.