System info Host, CPU, memory

START HERE

`uname -a`
Kernel and machine

`lscpu`
CPU model and cores

`hostnamectl`
Host, OS and kernel

`nproc`
Usable CPU count

`uptime`
Up time and load

`dmesg -T --level=err,warn`
Recent kernel problems

`free -h`
Memory and swap

`vmstat 1 5`
Live CPU, memory, I/O

What is it? OS, kernel, host → `uname` / `hostnamectl` → **How busy?** load, CPU, memory → `uptime` / `vmstat` / `free` → **What hardware?** CPU, PCI, USB → `lscpu` / `lspci` / `lsusb` → **What went wrong?** kernel log → `dmesg`

CONCEPTS

Most of these tools just format files under /proc and /sys, so the same data is available to scripts.

Load average counts runnable tasks: compare it with nproc, not with 1.

free's available column, not free, tells how much memory new programs can still get; cache is reclaimable.

SYSTEM AND OS

`uname -a`
Kernel name, release, arch and host

`uname -r`
Kernel release only

`uname -m`
Machine architecture (x86_64, aarch64)

`uname -srm`
Pick fields: kernel, release, machine

`hostnamectl`
Hostname, OS, kernel, virtualization

`hostname`
Print the current hostname

`hostnamectl --json=pretty`
Same facts as JSON for scripts

`sudo hostnamectl set-hostname <name>`
Set the static hostname

`cat /etc/os-release`
Distro name and version (ID, VERSION_ID)

`. /etc/os-release && echo $PRETTY_NAME`
Read the distro name in a script

`systemd-detect-virt`
Say whether in a VM or container

`cat /proc/version`
Kernel build string and compiler

UPTIME AND LOAD

`uptime`
Time up, users, load averages (1/5/15 min)

`uptime -p`
Uptime in words

`uptime -s`
Boot time as a timestamp

UPTIME AND LOAD (cont.)

`who -b`
Last boot time

`w`
Who is logged in and what they run

`journalctl --list-boots`
Boots the journal remembers

`cat /proc/loadavg`
Load averages plus running tasks

MEMORY

`free -h`
Memory and swap, human-readable

`free -h -s <sec>`
Repeat every <sec> seconds

`free -h -c <n>`
Stop after <n> reports

`free -m`
Sizes in MiB

`free -h -t`
Add a RAM + swap total row

`free -h -w`
Split buff and cache columns

`free -L`
Whole report on one line

`grep -E 'MemAvailable|SwapFree' /proc/meminfo`
Read exact kernel counters

`vmstat -s`
Memory and event counters, one per line

CPU

`lscpu`
Model, sockets, cores, threads, caches

`nproc`
CPUs usable by this process

`nproc --all`
Installed CPUs, ignoring limits

`lscpu -e`
One row per logical CPU

`lscpu -J`
CPU facts as JSON

`lscpu -C`
Cache sizes per level

`grep -m1 'model name' /proc/cpuinfo`
Quick CPU model name

VMSTAT

`vmstat <sec> <count>`
Sample every <sec>, stop after <count>

`vmstat 1`
Sample every second until Ctrl+C

`vmstat -w <sec>`
Wide output for big numbers

`vmstat -t <sec>`
Add a timestamp column

`vmstat -S M <sec>`
Show sizes in MB

`vmstat -d`
Per-disk read and write totals

`vmstat -a`
Active and inactive memory

vmstat columns

Column	Read it as
`r / b`	Runnable / blocked tasks; r above nproc means CPU-bound
`si / so`	Swap in / out; non-zero means memory pressure
`bi / bo`	Blocks read / written on disks
`us sy id wa`	CPU user, system, idle, I/O wait

The first vmstat line is an average since boot; judge the lines after it.

HARDWARE

`lspci`
List PCI devices (GPU, NIC, controllers)

`lspci -nnk`
IDs and the kernel driver in use

`lspci -v`
Verbose details per device

`lspci -s <slot>`
Only one slot, e.g. 00:02.0

`lspci -t`
Bus tree

`lsusb`
List USB devices

`lsusb -t`
USB tree with drivers and speeds

`lsblk`
Disks and partitions as a tree

KERNEL LOG (DMESG)

`dmesg -T`
Log with readable timestamps

KERNEL LOG (DMESG) (cont.)

`dmesg --level=err,warn`
Only errors and warnings

`dmesg -w`
Follow new messages live

`dmesg -H`
Human mode: pager, colour, relative time

`dmesg | tail -n 30`
Latest messages

`dmesg -T | grep -i <word>`
Search the log

Many distros restrict dmesg to root (kernel.dmesg_restrict); use sudo, or journalctl -k.

USERS AND SESSIONS

`whoami`
Name of the current user

`id`
UID, GID and all groups

`id <user>`
Same for another user

`id -nG`
Group names only

`id -u`
Numeric UID, for scripts

`who`
Logged-in sessions

RECIPES

Quick host report

```
uname -srm
grep PRETTY_NAME /etc/os-release
uptime -p
nproc
free -h | head -2
```

Is it CPU, memory or disk?

```
uptime            # load vs nproc
vmstat 1 5        # r, si/so, wa columns
free -h           # available memory
dmesg -T --level=err,warn | tail
```

Why did it reboot?

```
journalctl --list-boots | tail -n 3
uptime -s
sudo journalctl -b -1 -e   # end of previous boot
dmesg -T | grep -iE 'oom|panic'
```

Is the hardware seen?

```
lspci -nnk | grep -iA3 'vga\|net'
lsusb -t
dmesg -T | tail -n 20   # after plugging in
```