

Hardware Information

LinuXpert Systems

No.5, Dr. Ambedkar St, Rathinapuram,

Nanganallur, Chennai – 600 061.

Phone: 98841 65649, 90808 01382

E-mail: baskar@linuxpert.in

TABLE OF CONTENTS

1. lshw 1

2. dmidecode 8

3. lspci 16

4. lsusb 22

5. lsscsi 29

6. hwinfo 35

7. smartctl 43

1. lshw

The `lshw` command displays detailed information about the hardware configuration of a Linux system. It can show information about the CPU, memory, disks, network interfaces, PCI devices, USB devices, BIOS, motherboard, and much more.

It is commonly used by:

- System administrators
- Hardware troubleshooting engineers
- Linux users who need complete hardware details

Syntax

```
lshw [OPTIONS]
```

Basic Example

```
sudo lshw
```

Output (shortened):

```
bharath
    description: Desktop Computer
    product: OptiPlex 7090
    vendor: Dell Inc.
    serial: ABCD123

  *-core
       description: Motherboard

  *-cpu
       description: CPU
       product: Intel Core i7-11700

  *-memory
       description: System Memory
       size: 16GiB

  *-disk
       description: SSD
       product: Samsung SSD 980
```

```
  *-network
       description: Ethernet interface
```

This displays the complete hardware inventory.

Common Options

Command	Description
`sudo lshw`	Display all hardware information
`sudo lshw -short`	Display a concise hardware summary
`sudo lshw -class cpu`	Show CPU details
`sudo lshw -class memory`	Show memory information
`sudo lshw -class disk`	Show storage devices
`sudo lshw -class network`	Show network interfaces
`sudo lshw -class display`	Show graphics card information
`sudo lshw -class bridge`	Show bridge devices
`sudo lshw -html`	Generate HTML output
`sudo lshw -xml`	Generate XML output
`sudo lshw -json`	Generate JSON output

Frequently Used Examples

1. Display a Short Summary

```
sudo lshw -short
```

Example:

```
H/W path     Device     Class      Description
============================================
/0                      system     Desktop Computer
/0/0                    memory     16GiB RAM
/0/4                    processor  Intel Core i7
/0/100                  bridge     Host bridge
/0/100/1f.2  /dev/sda   disk       1TB SSD
/0/100/1f.6  enp0s31f6  network    Ethernet
```

2. Display CPU Information

```
sudo lshw -class cpu
```

Output:

```
*-cpu
     product: Intel(R) Core(TM) i7-11700
     vendor: Intel Corp.
     physical id: 4
     bus info: cpu@0
     width: 64 bits
     clock: 2900MHz
```

---

3. Display Memory Information

```
sudo lshw -class memory
```

Output:

```
*-memory
     size: 16GiB

*-bank:0
     size: 8GiB
     slot: DIMM A

*-bank:1
     size: 8GiB
     slot: DIMM B
```

Useful for checking:

- Installed RAM
- Empty memory slots
- Memory speed (on supported hardware)

4. Display Disk Information

```
sudo lshw -class disk
```

Example:

```
*-disk
     description: SSD
     product: Samsung SSD 980
     size: 1TB
```

---

5. Display Network Information

```
sudo lshw -class network
```

Example:

```
*-network
     description: Ethernet interface
     product: Intel Ethernet Controller
     logical name: enp0s31f6
     serial: 3c:52:82:xx:xx:xx
```

---

6. Display Graphics Card Information

```
sudo lshw -class display
```

Example:

```
*-display
     description: VGA compatible controller
     product: NVIDIA GeForce RTX 3060
```

---

7. Save Output to a File

```
sudo lshw > hardware.txt
```

8. Generate an HTML Report

```
sudo lshw -html > hardware.html
```

Open it in a browser:

```
firefox hardware.html
```

---

9. Generate JSON Output

```
sudo lshw -json > hardware.json
```

Useful for automation and scripts.

Installing `lshw`

Ubuntu/Debian

```
sudo apt install lshw
```

RHEL/CentOS

```
sudo dnf install lshw
```

or

```
sudo yum install lshw
```

---

openSUSE

```
sudo zypper install lshw
```

Why Use `sudo`?

Without `sudo`, some hardware details may be unavailable.

Compare:

```
lshw
```

vs.

```
sudo lshw
```

The second command provides more complete information because it can access privileged system resources.

Common Hardware Classes

Class	Description
`cpu`	Processor
`memory`	RAM and memory banks
`disk`	HDD/SSD devices
`storage`	Storage controllers (SATA, NVMe, RAID, etc.)
`network`	Ethernet and Wi-Fi adapters
`display`	Graphics cards
`bridge`	PCI and chipset bridges
`bus`	System buses
`usb`	USB controllers and devices
`multimedia`	Audio and multimedia devices

Example:

```
sudo lshw -class usb
```

Comparison with Similar Commands

Command	Purpose
`lshw`	Comprehensive hardware inventory
`lscpu`	CPU information only
`lsblk`	Block devices (disks and partitions)
`lsusb`	USB devices
`lspci`	PCI devices
`dmidecode`	BIOS, motherboard, and memory information from DMI tables
`hwinfo`	Another comprehensive hardware information tool (if installed)
`inxi`	Human-friendly system summary (if installed)

Practical Use Cases

- Audit all hardware in a Linux system.
- Verify installed RAM, CPU, and storage after an upgrade.
- Identify exact hardware models before installing drivers.
- Troubleshoot network or graphics hardware issues.
- Generate hardware reports for documentation or support.
- Collect hardware details for inventory management.

Summary

- **Command:** `lshw`
- **Purpose:** Displays detailed hardware information about the system.
- **Requires root?** `sudo` is recommended for complete information.
- **Best options:**
  - `sudo lshw` – Full hardware inventory.
  - `sudo lshw -short` – Concise summary.
  - `sudo lshw -class <class>` – Show a specific hardware category.
  - `sudo lshw -html`, `-xml`, `-json` – Export hardware information in different formats.

2. dmidecode

The `dmidecode` command is used to **display hardware information stored in the system's DMI/SMBIOS tables**. It provides detailed information about your computer's BIOS, motherboard, processor, memory, chassis, and other hardware components.

- **Full form:**
  - **DMI** – Desktop Management Interface
  - **SMBIOS** – System Management BIOS

Unlike commands such as `lscpu` or `lsblk`, `dmidecode` reads information directly from the BIOS/UEFI firmware rather than querying the hardware itself.

Syntax

```
dmidecode [options]
```

Most uses require **root privileges**:

```
sudo dmidecode
```

What information does it show?

`dmidecode` can display details about:

- BIOS
- System manufacturer
- Product name
- Motherboard
- Processor
- Memory modules (RAM)
- Chassis
- Cache

- Power supplies
- System slots
- Serial numbers
- Asset tags

Common Examples

1. Display all hardware information

```
sudo dmidecode
```

Displays the complete DMI table.

Example:

```
BIOS Information
    Vendor: American Megatrends
    Version: P2.30
    Release Date: 08/15/2023

System Information
    Manufacturer: Dell Inc.
    Product Name: OptiPlex 7090
```

---

2. Display BIOS information

```
sudo dmidecode -t bios
```

or

```
sudo dmidecode --type bios
```

Example:

```
BIOS Information
    Vendor: Dell Inc.
    Version: 1.12.3
    Release Date: 12/10/2024
```

3. Display system information

```
sudo dmidecode -t system
```

Output:

```
Manufacturer: Lenovo
Product Name: ThinkPad T14
Serial Number: XXXXXXXX
UUID: XXXXXXXX
```

---

4. Display motherboard information

```
sudo dmidecode -t baseboard
```

Example:

```
Base Board Information
Manufacturer: ASUS
Product Name: PRIME B660M-A
Version: Rev 1.xx
```

---

5. Display processor information

```
sudo dmidecode -t processor
```

Shows:

- CPU socket
- Manufacturer
- Version
- Maximum speed
- Current speed
- Core count
- Thread count

6. Display memory information

```
sudo dmidecode -t memory
```

Shows:

- Installed RAM
- Maximum supported RAM
- Memory slots
- Error correction capability

---

7. Display each RAM module

```
sudo dmidecode -t memory | less
```

Example:

```
Memory Device
    Size: 8192 MB
    Speed: 3200 MT/s
    Manufacturer: Samsung
    Locator: DIMM_A1
```

---

8. Display chassis information

```
sudo dmidecode -t chassis
```

Shows:

```
Chassis Information
    Type: Desktop
    Manufacturer: HP
    Serial Number: XXXXX
```

9. Display cache information

```
sudo dmidecode -t cache
```

Shows:

- L1 cache
- L2 cache
- L3 cache

---

10. Display only the system serial number

```
sudo dmidecode -s system-serial-number
```

Example:

```
ABC123456
```

---

11. Display BIOS version

```
sudo dmidecode -s bios-version
```

Output:

```
1.16.0
```

---

12. Display system manufacturer

```
sudo dmidecode -s system-manufacturer
```

Output:

```
Dell Inc.
```

13. Display product name

```
sudo dmidecode -s system-product-name
```

Output:

```
Latitude 5520
```

---

14. Display BIOS release date

```
sudo dmidecode -s bios-release-date
```

Example:

```
11/24/2024
```

---

15. Save output to a file

```
sudo dmidecode > hardware_info.txt
```

Frequently Used Options

Option	Description
`-t bios`	BIOS information
`-t system`	System details
`-t processor`	CPU information
`-t memory`	Memory information
`-t cache`	CPU cache
`-t chassis`	Chassis information
`-t baseboard`	Motherboard information
`-s`	Display a specific string
`-q`	Less verbose output
`-u`	Display raw hexadecimal data
`--dump-bin file`	Dump SMBIOS data to a binary file

Useful `-s` Keywords

```
sudo dmidecode -s bios-vendor
sudo dmidecode -s bios-version
sudo dmidecode -s bios-release-date
sudo dmidecode -s system-manufacturer
sudo dmidecode -s system-product-name
sudo dmidecode -s system-version
sudo dmidecode -s system-serial-number
sudo dmidecode -s system-uuid
sudo dmidecode -s baseboard-manufacturer
sudo dmidecode -s baseboard-product-name
sudo dmidecode -s chassis-serial-number
sudo dmidecode -s processor-family
```

Common DMI Types

Type	Description
`bios`	BIOS information
`system`	Computer/system information
`baseboard`	Motherboard details
`chassis`	Chassis/case details
`processor`	CPU information
`memory`	Memory array information
`cache`	CPU cache information
`connector`	Ports and connectors
`slot`	Expansion slots (PCIe, etc.)
`oem-string`	OEM-specific strings

Practical Uses

- Identify the motherboard model.
- Check the BIOS version before updating firmware.
- Find the computer's serial number for warranty or support.
- Determine installed and maximum supported RAM.
- Verify CPU socket and core information.
- Inventory hardware across multiple systems.
- Troubleshoot hardware configuration issues.

Difference Between `dmidecode` and Similar Commands

Command	Shows
`dmidecode`	BIOS/UEFI (SMBIOS) hardware information
`lshw`	Detailed hardware detected by the kernel
`lscpu`	CPU details
`lsblk`	Block storage devices
`hwinfo`	Comprehensive hardware information (if installed)
`inxi`	Human-friendly system summary (if installed)

Advantages

- Provides detailed firmware-reported hardware information.
- Useful for system inventory and asset management.
- Can display serial numbers and UUIDs.
- Shows RAM slot information and motherboard details.
- Does not require internet access.

Limitations

- Requires root privileges for most information.
- Relies on the accuracy of SMBIOS data provided by the system firmware.
- Some fields may be blank or incorrect if the manufacturer did not populate them.
- Primarily intended for x86/x86_64 systems with SMBIOS support.

Summary

`dmidecode` is a powerful Linux utility for retrieving hardware information from the system's SMBIOS/DMI tables. It is commonly used by system administrators to inspect BIOS details, motherboard information, CPU and RAM configuration, serial numbers, and other firmware-reported hardware details. It is especially useful for hardware inventory, troubleshooting, and verifying system specifications.

3. lspci

The `lspci` command **lists all PCI (Peripheral Component Interconnect) devices** connected to your computer. It is commonly used to identify hardware such as graphics cards, network adapters, sound cards, storage controllers, USB controllers, and more.

Syntax:

```
lspci [options]
```

What is PCI?

PCI (Peripheral Component Interconnect) is a hardware bus standard that allows devices to communicate with the CPU and memory. Modern systems use **PCI Express (PCIe)**, but the `lspci` command displays both PCI and PCIe devices.

Basic Usage

1. List all PCI devices

```
lspci
```

Example Output:

```
00:00.0 Host bridge: Intel Corporation 12th Gen Core Host Bridge
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics
00:14.0 USB controller: Intel Corporation USB 3.2 Controller
00:17.0 SATA controller: Intel Corporation SATA AHCI Controller
00:1f.3 Audio device: Intel Corporation HD Audio Controller
01:00.0 Network controller: Intel Corporation Wi-Fi 6 AX201
```

Explanation

- `00:02.0` → PCI bus/device/function address
- `VGA compatible controller` → Device type
- `Intel Corporation UHD Graphics` → Manufacturer and device

Common Options

2. Show detailed information

```
lspci -v
```

Displays more details about each PCI device.

---

3. Show very detailed information

```
lspci -vv
```

Displays additional information like:

- Interrupts
- Power management
- Memory regions
- Capabilities

---

4. Show maximum detail

```
lspci -vvv
```

Displays every available detail about the hardware.

---

5. Show numeric IDs

```
lspci -n
```

Example:

```
00:02.0 0300: 8086:46a6
```

Here:

- `8086` → Intel Vendor ID
- `46a6` → Device ID

Useful when searching for driver support.

6. Show numeric IDs and names

```
lspci -nn
```

Example:

```
00:02.0 VGA compatible controller [0300]: Intel Corporation UHD Graphics
[8086:46a6]
```

This is one of the most commonly used options when troubleshooting hardware.

---

7. Show kernel driver in use

```
lspci -k
```

Example:

```
01:00.0 Network controller
    Kernel driver in use: iwlwifi
    Kernel modules: iwlwifi
```

Useful for checking:

- Which driver is loaded
- Which module supports the hardware

---

8. Display devices as a tree

```
lspci -t
```

Example:

```
-[0000:00]-+-00.0
           +-02.0
           +-14.0
           \-17.0
```

Shows the PCI bus hierarchy.

9. Show only one device

```
lspci -s 00:02.0
```

Displays information for a specific PCI device.

---

10. Search for a device

```
lspci | grep VGA
```

or

```
lspci | grep Network
```

---

11. Display hexadecimal dump of configuration space

```
sudo lspci -xxx
```

Useful for hardware debugging.

Useful Examples

Find graphics card

```
lspci | grep VGA
```

Output:

```
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics
```

Find Wi-Fi adapter

```
lspci | grep Network
```

Output:

```
01:00.0 Network controller: Intel Corporation Wi-Fi 6 AX201
```

Find Ethernet controller

```
lspci | grep Ethernet
```

Output:

```
02:00.0 Ethernet controller: Realtek RTL8168
```

Find storage controller

```
lspci | grep SATA
```

Output:

```
00:17.0 SATA controller: Intel Corporation SATA AHCI Controller
```

Check which driver is used by the graphics card

```
lspci -k | grep -A3 VGA
```

Example:

```
00:02.0 VGA compatible controller
    Kernel driver in use: i915
    Kernel modules: i915
```

Commonly Used Options Summary

Option	Description
`lspci`	List all PCI devices
`-v`	Verbose output
`-vv`	More detailed output
`-vvv`	Maximum detail
`-n`	Numeric vendor/device IDs
`-nn`	Numeric IDs and names
`-k`	Show kernel driver and modules
`-t`	Show PCI bus tree
`-s <device>`	Show a specific device
`-xxx`	Dump configuration space (root required)

Typical Use Cases

- Identify graphics cards (NVIDIA, AMD, Intel)
- Detect Wi-Fi and Ethernet adapters
- Identify SATA, NVMe, RAID, and USB controllers
- Verify that Linux detects newly installed PCIe hardware
- Check which kernel driver is handling a device
- Troubleshoot hardware detection and driver issues

Related Commands

Command	Purpose
`lsusb`	List USB devices
`lshw`	Detailed hardware information
`lsblk`	List storage block devices
`hwinfo`	Comprehensive hardware information (if installed)
`dmidecode`	BIOS and motherboard information
`dmesg`	View kernel hardware detection messages
`lscpu`	Display CPU information
`lsmod`	List loaded kernel modules

Notes

- `lspci` is provided by the **pciutils** package.
- On Debian/Ubuntu, install it with:

  ```
  sudo apt install pciutils
  ```

- Most information is available to regular users, but some advanced details (such as configuration space dumps with `-xxx`) require `sudo`.
- `lspci` reports only **PCI/PCIe devices**. For USB hardware, use `lsusb`; for a full hardware inventory, use `lshw`.

4. lsusb

The `lsusb` command is a Linux utility used to **display information about USB buses and the USB devices connected to the system**. It helps you identify USB hardware such as flash drives, keyboards, mice, webcams, printers, Bluetooth adapters, USB Wi-Fi adapters, and more.

It is part of the **usbutils** package.

Syntax

```
lsusb [OPTIONS]
```

Basic Example

```
lsusb
```

Example output:

```
Bus 001 Device 002: ID 8087:0026 Intel Corp.
Bus 001 Device 003: ID 046d:c534 Logitech USB Receiver
Bus 002 Device 004: ID 0781:5567 SanDisk Cruzer Blade
```

Explanation:

- **Bus 001** → USB bus number
- **Device 003** → Device number assigned by the kernel
- **ID 046d:c534** → Vendor ID : Product ID
- **Logitech USB Receiver** → Device description

Commonly Used Options

1. Show all USB devices

```
lsusb
```

Lists every connected USB device.

2. Display detailed information

```
lsusb -v
```

Shows detailed information including:

- USB version
- Device class
- Vendor
- Product
- Endpoints
- Power requirements
- Serial number
- Interfaces

Example:

```
sudo lsusb -v
```

Some details require root privileges.

---

3. Show device hierarchy (tree view)

```
lsusb -t
```

Example:

```
/:  Bus 02.Port 1: Dev 1, Class=root_hub
    |__ Port 2: Dev 3, Class=Mass Storage
    |__ Port 3: Dev 4, Class=Human Interface Device
```

Useful for understanding which device is connected to which USB port.

4. Display a specific device

```
lsusb -d 046d:c534
```

Shows only the specified vendor/product.

Format:

```
VendorID:ProductID
```

---

5. Show devices from a specific vendor

Example:

```
lsusb -d 046d:
```

Displays only Logitech devices.

---

6. Show numeric IDs only

```
lsusb -n
```

Example:

```
Bus 001 Device 003: ID 046d:c534
```

Only numeric IDs are displayed.

---

7. Display USB topology with speeds

```
lsusb -tv
```

Shows:

- Device tree
- USB speed
- Driver
- Interface

Very useful for troubleshooting USB devices.

Finding a USB Flash Drive

Plug in the drive and run:

```
lsusb
```

Example:

```
Bus 002 Device 005: ID 0781:5567 SanDisk Corp. Cruzer Blade
```

The device is successfully detected by the USB subsystem.

Finding a USB Wi-Fi Adapter

```
lsusb
```

Example:

```
Bus 001 Device 004: ID 0bda:c811 Realtek Semiconductor Corp.
```

This identifies the USB Wi-Fi adapter.

Finding a Webcam

```
lsusb
```

Example:

```
Bus 001 Device 006: ID 1bcf:2c99 Sunplus Innovation Webcam
```

Find Vendor and Product IDs

```
lsusb
```

Output:

```
Bus 001 Device 003: ID 046d:c534 Logitech USB Receiver
```

Here:

- **Vendor ID = 046d**
- **Product ID = c534**

These IDs are often used to search for compatible drivers or identify unknown devices.

Combine with `grep`

Find Logitech devices:

```
lsusb | grep Logitech
```

Find Realtek devices:

```
lsusb | grep Realtek
```

Find storage devices:

```
lsusb | grep -i storage
```

Check Whether a USB Device Is Detected

If a USB device isn't working:

1. Plug it in.
2. Run:

```
lsusb
```

If the device appears, Linux has detected it at the USB hardware level. If it does not appear, the issue could be:

- A faulty USB port
- A bad cable
- Insufficient power
- A hardware failure

You can also check kernel messages:

```
dmesg | tail -20
```

Install `lsusb`

Ubuntu/Debian

```
sudo apt install usbutils
```

Fedora

```
sudo dnf install usbutils
```

RHEL/CentOS

```
sudo yum install usbutils
```

Arch Linux

```
sudo pacman -S usbutils
```

Frequently Used `lsusb` Commands

Command	Description
`lsusb`	List all connected USB devices
`lsusb -v`	Show detailed device information
`lsusb -t`	Show USB device tree
`lsusb -tv`	Tree view with verbose details
`lsusb -d vendor:product`	Show a specific USB device
`lsusb -n`	Display only numeric vendor/product IDs
`lsusb | grep Logitech`	Search for Logitech USB devices

When to Use `lsusb`

Use `lsusb` when you need to:

- Verify that Linux detects a USB device.
- Identify the vendor and product IDs of USB hardware.
- Troubleshoot USB peripherals such as keyboards, mice, storage devices, webcams, Wi-Fi adapters, or Bluetooth adapters.
- Inspect the USB bus topology and connection speeds.
- Gather information needed to install or debug device drivers.

`lsusb` is one of the most useful first-step diagnostic commands for USB hardware on Linux systems.

5. lsscsi

The `lsscsi` command **lists all SCSI (Small Computer System Interface) devices** connected to a Linux system. Despite its name, it also displays many modern storage devices such as **SATA, SAS, USB storage, and some NVMe devices** because the Linux kernel often exposes them through the SCSI subsystem.

It is commonly used by Linux administrators to identify:

- Hard disk drives (HDDs)
- Solid-state drives (SSDs)
- Optical drives (CD/DVD/Blu-ray)
- Tape drives
- RAID controllers
- Storage enclosures

Syntax

```
lsscsi [OPTIONS]
```

Basic Example

```
lsscsi
```

Example output:

```
[0:0:0:0]    disk    ATA      Samsung SSD 870  2B6Q  /dev/sda
[1:0:0:0]    disk    ATA      ST1000DM010      CC43  /dev/sdb
[2:0:0:0]    cd/dvd  ASUS     DRW-24D5MT       1.00  /dev/sr0
```

Explanation

Column	Description
`[0:0:0:0]`	Host:Channel:Target:LUN (SCSI address)
`disk`	Device type
`ATA`	Vendor/Interface
`Samsung SSD 870`	Device model
`2B6Q`	Firmware version
`/dev/sda`	Linux device name

Understanding the SCSI Address

Example:

```
[2:0:1:0]
```

This means:

```
Host    = 2
Channel = 0
Target  = 1
LUN     = 0
```

These values help identify exactly where the device is attached.

Common Options

1. Show all storage devices

```
lsscsi
```

Lists every detected SCSI device.

2. Show device sizes

```
lsscsi -s
```

Example:

```
[0:0:0:0] disk ATA Samsung SSD 870 /dev/sda 500GB
```

Useful for checking disk capacity.

---

3. Show generic SCSI devices

```
lsscsi -g
```

Example:

```
[0:0:0:0] disk ATA Samsung SSD 870 /dev/sda /dev/sg0
```

Here:

- `/dev/sda` → Block device
- `/dev/sg0` → Generic SCSI device

The generic device is used by low-level SCSI utilities.

---

4. Display transport information

```
lsscsi -t
```

Example:

```
[0:0:0:0] disk sata:1
```

Shows how the device is connected:

- SATA
- SAS
- USB
- Fibre Channel
- iSCSI

5. Show WWN (World Wide Name)

```
lsscsi -w
```

Useful in SAN (Storage Area Network) environments.

---

6. Display hexadecimal values

```
lsscsi -H
```

Shows certain identifiers in hexadecimal format.

---

7. List NVMe devices (supported versions)

```
lsscsi -N
```

Shows NVMe drives along with SCSI devices if supported.

Practical Examples

List only disks

```
lsscsi | grep disk
```

Output:

```
[0:0:0:0] disk ATA Samsung SSD 870 /dev/sda
[1:0:0:0] disk ATA ST1000DM010 /dev/sdb
```

Find optical drives

```
lsscsi | grep cd
```

Output:

```
[2:0:0:0] cd/dvd ASUS DRW-24D5MT /dev/sr0
```

Display disks with sizes

```
lsscsi -s
```

Output:

```
[0:0:0:0] disk ATA Samsung SSD 870 /dev/sda 500GB
[1:0:0:0] disk ATA ST1000DM010 /dev/sdb 1TB
```

Display generic devices

```
lsscsi -g
```

Output:

```
[0:0:0:0] disk ATA Samsung SSD /dev/sda /dev/sg0
```

Installing `lsscsi`

Ubuntu/Debian

```
sudo apt install lsscsi
```

RHEL/CentOS/Rocky/AlmaLinux

```
sudo dnf install lsscsi
```

or

```
sudo yum install lsscsi
```

---

openSUSE

```
sudo zypper install lsscsi
```

Difference Between `lsscsi` and Similar Commands

Command	Purpose
`lsscsi`	Lists SCSI storage devices with SCSI addressing
`lsblk`	Lists block devices, partitions, and mount points
`fdisk -l`	Displays partition tables
`blkid`	Shows filesystem UUIDs and types
`lshw -class disk`	Detailed hardware information about storage devices
`lsusb`	Lists USB devices
`lspci`	Lists PCI devices

When to Use `lsscsi`

Use `lsscsi` when you need to:

- Identify all storage devices attached to the system.
- Map Linux device names (`/dev/sdX`) to their SCSI addresses.
- Troubleshoot storage hardware or SCSI-related issues.
- Verify whether a newly connected disk is detected.
- View transport information (SATA, SAS, USB, Fibre Channel, iSCSI).
- Work with SAN or enterprise storage systems.

Summary

- **Command:** `lsscsi`
- **Purpose:** Lists SCSI storage devices and their details.
- **Common options:**
  - `-s` → Show disk sizes.
  - `-g` → Show generic SCSI device nodes (`/dev/sg*`).
  - `-t` → Show transport information.
  - `-w` → Show World Wide Names (WWNs).
  - `-N` → Include NVMe devices (if supported).
- **Typical use cases:** Storage inventory, troubleshooting, SAN administration, and mapping disks to SCSI addresses.

6. hwinfo

`hwinfo` is a **hardware information utility** that detects and displays detailed information about the hardware installed in a Linux system. It is commonly used for **system administration, troubleshooting, hardware inventory, and driver verification**.

It provides much more detailed information than commands like `lshw` or `lspci` for many hardware components.

Syntax

```
hwinfo [OPTIONS]
```

What `hwinfo` Displays

It can detect information about:

- CPU
- RAM
- Motherboard
- BIOS/UEFI
- Hard disks and SSDs
- USB devices
- PCI devices
- Graphics cards
- Network adapters
- Wireless devices
- Bluetooth
- Sound cards
- Printers
- Monitors
- Storage controllers

- Partitions
- Power management
- Sensors (on supported systems)

Common Examples

1. Display all hardware information

```
hwinfo
```

Example

```
cpu:
  Intel Core i7-1165G7

memory:
  16 GB

disk:
  Samsung SSD 980 1TB

network:
  Intel Wi-Fi 6 AX201
```

---

2. Display CPU information

```
hwinfo --cpu
```

Example

```
Processor: Intel Core i5-12400
Clock Speed: 2.5 GHz
Cores: 6
Threads: 12
```

Useful for checking:

- CPU model
- Number of cores
- Clock speed
- Cache
- Supported features

3. Display memory information

```
hwinfo --memory
```

Shows

- Installed RAM
- Memory modules
- Capacity
- Memory slots
- Speed

Example

```
Size: 16384 MB
Type: DDR4
Speed: 3200 MT/s
```

---

4. Display disk information

```
hwinfo --disk
```

Shows

- HDD
- SSD
- NVMe drives
- Vendor
- Model
- Capacity

Example

```
Model: Samsung SSD 980
Size: 1 TB
```

5. Display network devices

```
hwinfo --network
```

Example

```
Ethernet Controller
Intel I219-V

Wireless
Intel AX201
```

---

6. Display graphics card

```
hwinfo --gfxcard
```

Example

```
VGA compatible controller
NVIDIA RTX 3060
```

---

7. Display monitor information

```
hwinfo --monitor
```

Shows

- Resolution
- Manufacturer
- Supported modes

---

8. Display USB devices

```
hwinfo --usb
```

Example

```
USB Mouse

USB Keyboard

USB Webcam
```

9. Display PCI devices

```
hwinfo --pci
```

Lists

- Graphics card
- Network card
- SATA controller
- Audio controller

---

10. Display BIOS information

```
hwinfo --bios
```

Example

```
Vendor: American Megatrends
Version: 2.21
Date: 2025-01-18
```

---

11. Display storage controller

```
hwinfo --storage-ctrl
```

Shows

- SATA controller
- RAID controller
- NVMe controller

---

12. Display sound card

```
hwinfo --sound
```

Useful when troubleshooting audio problems.

13. Display Bluetooth devices

```
hwinfo --bluetooth
```

Shows installed Bluetooth hardware.

---

14. Display printer information

```
hwinfo --printer
```

---

15. Display short hardware information

```
hwinfo --short
```

Example

```
cpu:
Intel Core i7

disk:
Samsung SSD

network:
Intel Ethernet
```

Useful for a quick hardware summary.

---

16. Save hardware information to a file

```
hwinfo > hardware.txt
```

Later view it using

```
less hardware.txt
```

Frequently Used Options

Option	Description
`--cpu`	CPU information
`--memory`	RAM information
`--disk`	Disk drives
`--network`	Network devices
`--gfxcard`	Graphics cards
`--monitor`	Monitor information
`--usb`	USB devices
`--pci`	PCI devices
`--bios`	BIOS information
`--sound`	Sound devices
`--bluetooth`	Bluetooth hardware
`--printer`	Printers
`--storage-ctrl`	Storage controllers
`--short`	Short summary
`--help`	Help information

Install `hwinfo`

Ubuntu/Debian

```
sudo apt install hwinfo
```

Fedora

```
sudo dnf install hwinfo
```

RHEL/CentOS (EPEL may be required)

```
sudo dnf install hwinfo
```

or

```
sudo yum install hwinfo
```

openSUSE

```
sudo zypper install hwinfo
```

Practical Uses

- Verify newly installed hardware.
- Troubleshoot missing or malfunctioning devices.
- Check whether Linux has detected all hardware.
- Gather hardware inventory before upgrades.
- Identify the correct device drivers.
- Generate a detailed hardware report for support.

Difference Between `hwinfo`, `lshw`, and `lspci`

Command	Purpose	Best For
`hwinfo`	Comprehensive hardware detection	Detailed hardware inventory and diagnostics
`lshw`	Hardware tree with configuration details	General hardware overview
`lspci`	PCI bus devices only	Graphics cards, network cards, storage controllers
`lsusb`	USB devices only	USB peripherals
`dmidecode`	BIOS/DMI information	Motherboard, BIOS, RAM details
`inxi`	User-friendly system summary	Quick system information

Example workflow

If your Wi-Fi adapter is not working:

```
hwinfo --network
```

This lets you verify whether Linux detects the network hardware, its model, driver status, and related details. You can then use `lspci` (for PCI devices) or `lsusb` (for USB adapters) to inspect the device further if needed.

7. smartctl

`smartctl` is a Linux command-line utility used to **monitor, analyze, and manage the health of storage devices** such as HDDs, SSDs, NVMe SSDs, and USB drives (if supported). It is part of the **smartmontools** package and works with **S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology)**.

It is one of the most important tools for diagnosing disk failures before they happen.

Syntax

```
smartctl [OPTIONS] DEVICE
```

Example:

```
smartctl -a /dev/sda
```

- `-a` → Display all SMART information.
- `/dev/sda` → Target disk.

What SMART Is

SMART is a built-in monitoring system inside hard disks and SSDs.

It continuously records:

- Temperature
- Read/write errors
- Reallocated sectors
- Power-on hours
- Start/stop count
- Wear level (SSD)
- Bad blocks
- Self-test results

Instead of Linux checking the disk, **the disk itself monitors its own health.**

Installing smartctl

Ubuntu/Debian

```
sudo apt install smartmontools
```

RHEL/CentOS

```
sudo dnf install smartmontools
```

or

```
sudo yum install smartmontools
```

Commonly Used Options

Option	Meaning
`-a`	Show all SMART information
`-H`	Display overall health status
`-i`	Show device information
`-A`	Display SMART attributes
`-c`	Show SMART capabilities
`-x`	Display extended information
`-t short`	Run short self-test
`-t long`	Run extended self-test
`-l selftest`	Show self-test log
`-l error`	Show error log
`-s on`	Enable SMART
`-s off`	Disable SMART

Example 1: Check Overall Disk Health

```
sudo smartctl -H /dev/sda
```

Output:

```
SMART overall-health self-assessment test result: PASSED
```

Possible results

- PASSED
- FAILED

Example 2: Show Complete SMART Information

```
sudo smartctl -a /dev/sda
```

Shows

- Device model
- Serial number
- Firmware version
- SMART status
- Temperature
- Error logs
- Self-test history
- Health attributes

Example 3: Display Disk Information

```
sudo smartctl -i /dev/sda
```

Example output

```
Device Model: Samsung SSD 870 EVO
Serial Number: S6PXXXXXXXX
Firmware Version: SVT01B6Q
Capacity: 500 GB
```

Useful when identifying disks.

Example 4: View SMART Attributes

```
sudo smartctl -A /dev/sda
```

Example

```
ID# ATTRIBUTE_NAME          VALUE WORST THRESH RAW_VALUE

5   Reallocated_Sector_Ct   100   100    10      0
9   Power_On_Hours          98    98      0      5623
194 Temperature_Celsius     70    65      0      36
```

Important SMART Attributes

ID	Attribute	Meaning
5	Reallocated Sector Count	Bad sectors replaced
9	Power-On Hours	Total operating hours
10	Spin Retry Count	Failed spin-up attempts
187	Reported Uncorrectable Errors	Unrecoverable read/write errors
188	Command Timeout	Timed-out commands
190	Airflow Temperature	Drive temperature
194	Temperature	Current drive temperature
197	Current Pending Sector	Unstable sectors awaiting reallocation
198	Offline Uncorrectable	Unrecoverable sectors
199	UDMA CRC Error Count	Communication errors, often due to SATA cable/controller issues
241/242	Total LBAs Written/Read	Total data written/read (commonly SSDs)

Example 5: Check Temperature Only

```
sudo smartctl -A /dev/sda | grep Temperature
```

Example

```
194 Temperature_Celsius 36
```

Example 6: Enable SMART

```
sudo smartctl -s on /dev/sda
```

Example 7: Run a Short Self-Test

```
sudo smartctl -t short /dev/sda
```

Typical duration

```
2 minutes
```

After it finishes:

```
sudo smartctl -l selftest /dev/sda
```

Example

```
Completed without error
```

Example 8: Run a Long Self-Test

```
sudo smartctl -t long /dev/sda
```

May take:

- HDD: 1–5 hours (or more for large drives)
- SSD: Often shorter, depending on the model

Check the results:

```
sudo smartctl -l selftest /dev/sda
```

Example 9: Show Error Log

```
sudo smartctl -l error /dev/sda
```

Example

```
Error 1 occurred at disk power-on lifetime 5120 hours
```

Useful for troubleshooting disk problems.

Example 10: NVMe SSD

```
sudo smartctl -a /dev/nvme0n1
```

Shows information specific to NVMe drives, such as:

- Percentage Used
- Available Spare
- Critical Warning
- Temperature
- Data Units Read/Written
- Media Errors

Example 11: USB Drive (if supported)

```
sudo smartctl -a -d sat /dev/sdb
```

The `-d sat` option tells `smartctl` to communicate with SATA drives behind many USB-to-SATA adapters. Not all adapters support SMART passthrough.

Checking All Drives

List available disks:

```
lsblk
```

Then inspect each:

```
sudo smartctl -a /dev/sda
sudo smartctl -a /dev/sdb
sudo smartctl -a /dev/nvme0n1
```

Exit Status

`smartctl` returns useful exit codes:

- `0` → No errors; SMART status is good.
- Non-zero values → Indicate conditions such as command failures, SMART errors, or a failing health assessment. The exit status is a bitmask, so multiple conditions can be reported simultaneously.

Real-World Use Cases

- Check if a hard disk is failing.
- Monitor SSD wear and remaining lifespan.
- Diagnose SATA cable issues (e.g., rising UDMA CRC Error Count).
- Verify drive temperature.
- Run periodic self-tests as preventive maintenance.
- Troubleshoot I/O errors or filesystem corruption.
- Monitor servers and storage systems proactively.

Related Commands

Command	Purpose
`lsblk`	List block devices
`fdisk -l`	Display partition tables
`blkid`	Show filesystem UUIDs and labels
`df -h`	Show filesystem disk usage
`du -sh`	Show directory sizes
`badblocks`	Scan a disk for bad blocks
`fsck`	Check and repair filesystems
`nvme`	Manage and inspect NVMe SSDs

Summary

- **Purpose:** Monitor and diagnose the health of HDDs, SSDs, and NVMe drives using SMART.
- **Package:** `smartmontools`
- **Most common command:**

  ```
  sudo smartctl -a /dev/sda
  ```

- **Best for:** Detecting failing disks early, viewing SMART attributes, monitoring temperatures and SSD wear, running built-in self-tests, and investigating storage-related hardware issues.