LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA, PUNJAB

LOVELY PROFESSIONAL UNIVERSITY

Using any Open Source Software display advanced information on internal and external hardware, also display operating system.

Name : Zuabir Ahmad
Reg no: 11908946
Roll no : 62
Course Code : INT301
Section : KE009

Submitted to:

Rajeshwar Sharma Sir

Introduction

Modern computer systems consist of various internal and external hardware components that work together to provide the desired functionality. Some of these components include the CPU, memory, hard disk drive, graphics card, sound card, network card, and many more. Additionally, the operating system (OS) serves as the interface between the hardware and software components and manages the resources of the computer system.

Open Source Software (OSS) provides various tools and applications that can be used to display advanced information on both internal and external hardware components, as well as the operating system. These tools can help users diagnose hardware issues, identify performance bottlenecks, and monitor system resources.

Examples of OSS tools for hardware information include lshw, hwinfo, and dmidecode, which can provide detailed information on the hardware components installed in a computer system. These tools can display information such as the device model, vendor, driver, firmware version, and other details that are useful for hardware diagnostics.

For operating system details, tools such as uname, lsb_release, and /etc/os-release can be used to display information such as the OS name, version, kernel version, and other relevant details. Additionally, system monitoring tools like top, htop, and nmon can be used to monitor system resources such as CPU usage, memory usage, disk usage, and network activity in real-time.

Overall, OSS tools provide a powerful set of options for displaying advanced information on hardware and operating systems, making them a valuable resource for system administrators, developers, and power users.

Objective

The objective of using Open Source Software to display advanced information on internal and external hardware, as well as the operating system, is to gain a better understanding of the system's capabilities and potential limitations. By analyzing this information, users can diagnose hardware issues, optimize system performance, and identify any potential bottlenecks that may be affecting system performance.

Specifically, the objectives of using OSS for hardware information include:

Identifying the specific hardware components installed in the system and their characteristics.

1. Understanding the capabilities and limitations of each hardware component.
2. Diagnosing hardware issues and determining potential solutions.
3. Optimising system performance by adjusting hardware settings or upgrading components.
4. Regarding the operating system, the objectives of using OSS include:
5. Identifying the OS name, version, and kernel version.
6. Understanding the features and capabilities of the OS
7. Identifying potential security vulnerabilities and applying necessary updates or patches.
8. Monitoring system resources to optimise performance and ensure stability

The objective of using OSS to display advanced information on hardware and the operating system is to gain a comprehensive understanding of the system's capabilities and limitations, allowing for optimal performance and effective troubleshooting.

Description

There are various Open Source Software (OSS) tools that can be used to display advanced information on internal and external hardware components, as well as the operating system. These tools can provide detailed information on the hardware components installed in a computer system and the OS running on it.

One such tool is lshw, which stands for "list hardware". It is a command-line tool that displays detailed information on the hardware components installed in a system, including the CPU, memory, motherboard, graphics card, network card, and other devices. Lshw provides information such as the device model, vendor, driver, firmware version, and other details that are useful for hardware diagnostics.

Another OSS tool for hardware information is hwinfo, which provides a comprehensive report on the system's hardware components. It displays information on the CPU, memory, storage devices, graphics card, network adapters, sound card, and other devices. Hwinfo provides detailed information on each component, including the device model, vendor, driver, firmware version, and other details.

Dmidecode is another command-line tool that displays information on the hardware components installed in a system. It reads data from the system's BIOS and displays information such as the device model, vendor, serial number, and other details.

Regarding the operating system, tools like uname, lsb_release, and /etc/os-release can be used to display information such as the OS name, version, kernel version, and other relevant details. These tools provide essential information for system

administrators, developers, and power users, helping them understand the OS's features and capabilities and identifying potential security vulnerabilities.

In conclusion, using OSS tools to display advanced information on internal and external hardware components and the operating system is essential for diagnosing hardware issues, optimising system performance, and identifying potential security vulnerabilities. These tools provide a comprehensive overview of the system's capabilities and limitations, allowing users to make informed decisions and take necessary actions to ensure optimal performance and stability.

Scope

The scope of using Open Source Software to display advanced information on internal and external hardware, as well as the operating system, is broad and can be applied to various scenarios.

For example, the tools can be used by system administrators to diagnose and troubleshoot hardware issues, identify potential bottlenecks that may be affecting system performance, and optimise system performance by adjusting hardware settings or upgrading components.

Developers can also use these tools to understand the hardware requirements for their software and ensure that their software runs optimally on a given system.

Computer enthusiasts and power users can also use these tools to gain a better understanding of their system's capabilities and limitations, optimize performance, and diagnose issues.

Additionally, using Open Source Software for hardware and OS information can be useful in security assessments to identify potential vulnerabilities and ensure that the system is running the latest security patches.

In summary, the scope of using Open Source Software to display advanced information on hardware and the operating system is vast and can be applied in various scenarios, including system administration, software development, computer enthusiasts, and security assessments.

System Description

To display advanced information on internal and external hardware and the operating system using Open Source Software, a computer system is required. The system can be a desktop, laptop, server, or any other type of computer that is running an operating system.

The Open Source Software tools that are used to display hardware and OS information can be installed on various operating systems, including Linux, Windows, and macOS. However, for the purpose of this description, we will assume that the system is running a Linux-based operating system such as Ubuntu, Debian, or CentOS.

The system should have sufficient hardware resources to run the Open Source Software tools effectively. The amount of system resources required depends on the specific tools and the size and complexity of the system being analyzed. In general, a system with at least 4GB of RAM and a multi-core processor will suffice.

To display hardware and OS information, the Open Source Software tools can be installed using the system's package manager or by downloading and compiling the source code. Once installed, the tools can be run from the command line, and the output can be displayed in the terminal or saved to a file for further analysis.

It is important to note that some of the tools used to display hardware and OS information require elevated privileges, such as root or administrator access. Therefore, it is recommended to use these tools with caution and only with appropriate permissions.

In conclusion, to display advanced information on internal and external hardware and the operating system using Open Source Software, a computer system with sufficient hardware resources and a Linux-based operating system is required. The Open Source Software tools can be installed and run from the command line with appropriate permissions.

Target system description

The target system for using Open Source Software to display advanced information on internal and external hardware and the operating system can be any computer system that is running an operating system. This includes desktops, laptops, servers, and virtual machines.

The target system should be running a supported operating system that is compatible with the Open Source Software tools being used. For example, if the tools are designed for Linux-based operating systems, then the target system should be running a Linux-based OS such as Ubuntu, Debian, or CentOS.

The target system should have sufficient hardware resources to run the Open Source Software tools effectively. The amount of system resources required depends

on the specific tools and the size and complexity of the system being analyzed. In general, a system with at least 4GB of RAM and a multi-core processor will suffice.

Access to the target system may require elevated privileges, such as root or administrator access, depending on the specific tools and the information being retrieved. Therefore, it is important to ensure that appropriate permissions are granted before attempting to access the system.

In addition, it is important to ensure that the use of Open Source Software tools to retrieve information from the target system does not violate any applicable laws, regulations, or policies. The use of these tools for unauthorized purposes, such as hacking or espionage, is illegal and can result in severe legal consequences.

In summary, the target system for using Open Source Software to display advanced information on internal and external hardware and the operating system can be any computer system running a compatible operating system. The system should have sufficient hardware resources, appropriate permissions should be granted, and the use of the tools should comply with all applicable laws and policies.

Assumption and Dependencies

Assumptions:

The computer system has functioning hardware and software components.
The user has sufficient knowledge of the computer system and command line interface.
The Open Source Software tools being used are compatible with the operating system and hardware of the computer system.
The user has appropriate permissions to install and run the Open Source Software tools.

Dependencies

The Open Source Software tools being used must be available and installed on the computer system.
The computer system must have sufficient hardware resources to run the Open Source Software tools effectively.
The Open Source Software tools may depend on other software packages and libraries, which must also be installed on the computer system.
Access to certain hardware and software components may require elevated privileges, such as root or administrator access.
It is important to note that the assumptions and dependencies may vary depending on the specific Open Source Software tools being used and the target computer

system. Therefore, it is recommended to carefully review the documentation and requirements for the specific tools being used.

Data Set use

The data set used to display advanced information on internal and external hardware and the operating system can vary depending on the specific Open Source Software tools being used and the target computer system.

In general, the tools can gather data from a variety of sources, including system logs, hardware sensors, and system information files. This data can include:

- Processor and memory information
- Storage device details
- Network interface information
- Graphics card information
- System and kernel logs
- Running processes and services
- Installed software and packages
- System configuration files

The specific data set used may depend on the analysis being performed and the information required by the user. For example, if the analysis is focused on hardware components, then the data set may include details on the CPU, RAM, and storage devices. If the analysis is focused on software components, then the data set may include information on running processes and installed software packages.

It is important to ensure that the data being collected is relevant to the analysis being performed and that it is collected in a manner that does not compromise the integrity of the system being analysed. Care should be taken to ensure that the tools being used do not cause any unintended consequences or data loss.

Analysis Report

To use hwinfo in Ubanatu

```
iamzubair@iamzubair-virtual-machine: ~
iamzubair@iamzubair-virtual-machine:~$ sudo apt-get install hwinfo
[sudo] password for iamzubair:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hwinfo is already the newest version (21.72-1).
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 87 not upgraded.
```

1. To display information about the CPU

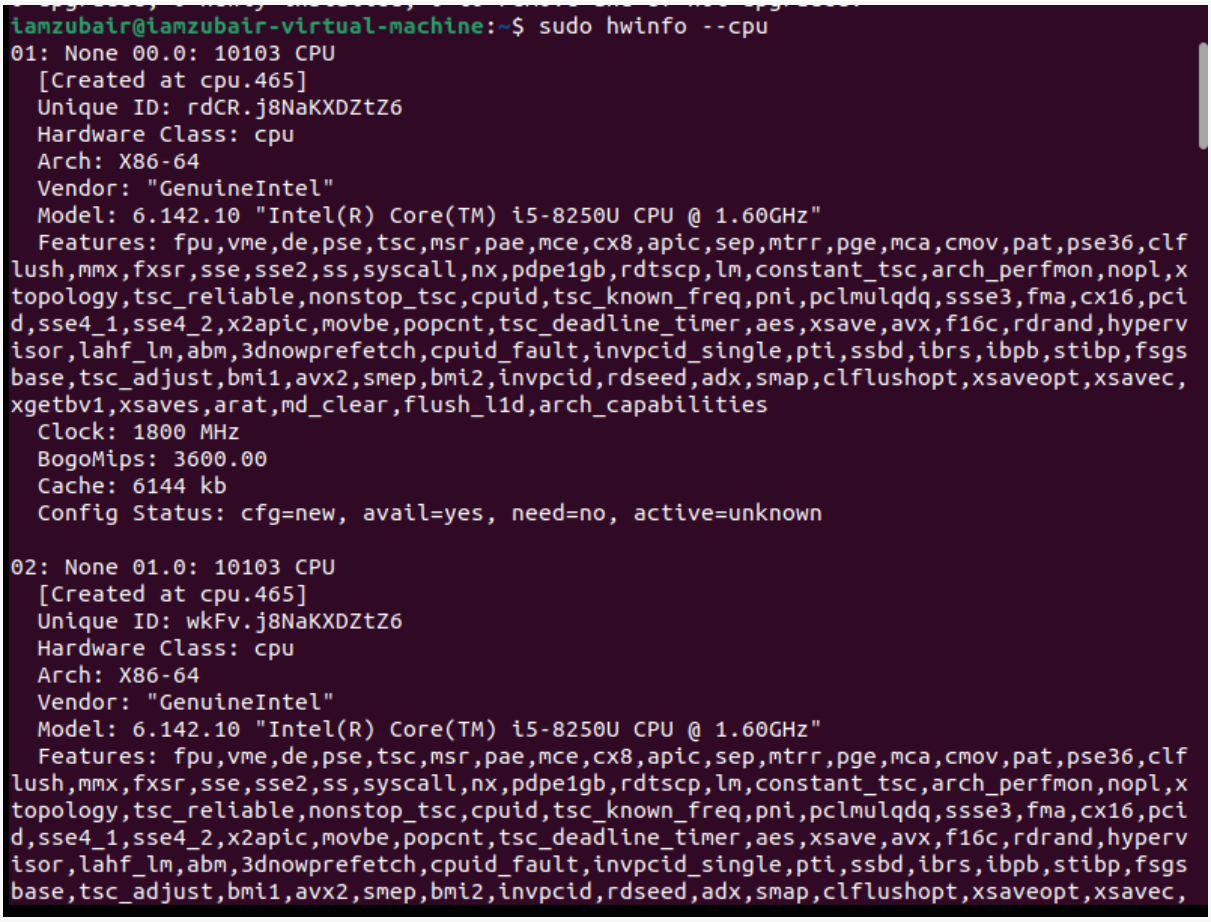
```
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --cpu
01: None 00.0: 10103 CPU
  [Created at cpu.465]
  Unique ID: rdCR.j8NaKXDZtZ6
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.142.10 "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz"
  Features: fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clf
lush,mmx,fxsr,sse,sse2,ss,syscall,nx,pdpe1gb,rdtscp,lm,constant_tsc,arch_perfmon,nopl,x
topology,tsc_reliable,nonstop_tsc,cpuid,tsc_known_freq,pni,pclmulqdq,ssse3,fma,cx16,pci
d,sse4_1,sse4_2,x2apic,movbe,popcnt,tsc_deadline_timer,aes,xsave,avx,f16c,rdrand,hyperv
isor,lahf_lm,abm,3dnowprefetch,cpuid_fault,invpcid_single,pti,ssbd,ibrs,ibpb,stibp,fsgs
base,tsc_adjust,bmi1,avx2,smep,bmi2,invpcid,rdseed,adx,smap,clflushopt,xsaveopt,xsavec,
xgetbv1,xsaves,arat,md_clear,flush_l1d,arch_capabilities
  Clock: 1800 MHz
  BogoMips: 3600.00
  Cache: 6144 kb
  Config Status: cfg=new, avail=yes, need=no, active=unknown

02: None 01.0: 10103 CPU
  [Created at cpu.465]
  Unique ID: wkFv.j8NaKXDZtZ6
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.142.10 "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz"
  Features: fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clf
lush,mmx,fxsr,sse,sse2,ss,syscall,nx,pdpe1gb,rdtscp,lm,constant_tsc,arch_perfmon,nopl,x
topology,tsc_reliable,nonstop_tsc,cpuid,tsc_known_freq,pni,pclmulqdq,ssse3,fma,cx16,pci
d,sse4_1,sse4_2,x2apic,movbe,popcnt,tsc_deadline_timer,aes,xsave,avx,f16c,rdrand,hyperv
isor,lahf_lm,abm,3dnowprefetch,cpuid_fault,invpcid_single,pti,ssbd,ibrs,ibpb,stibp,fsgs
base,tsc_adjust,bmi1,avx2,smep,bmi2,invpcid,rdseed,adx,smap,clflushopt,xsaveopt,xsavec,
```

2. To display information about the memory.

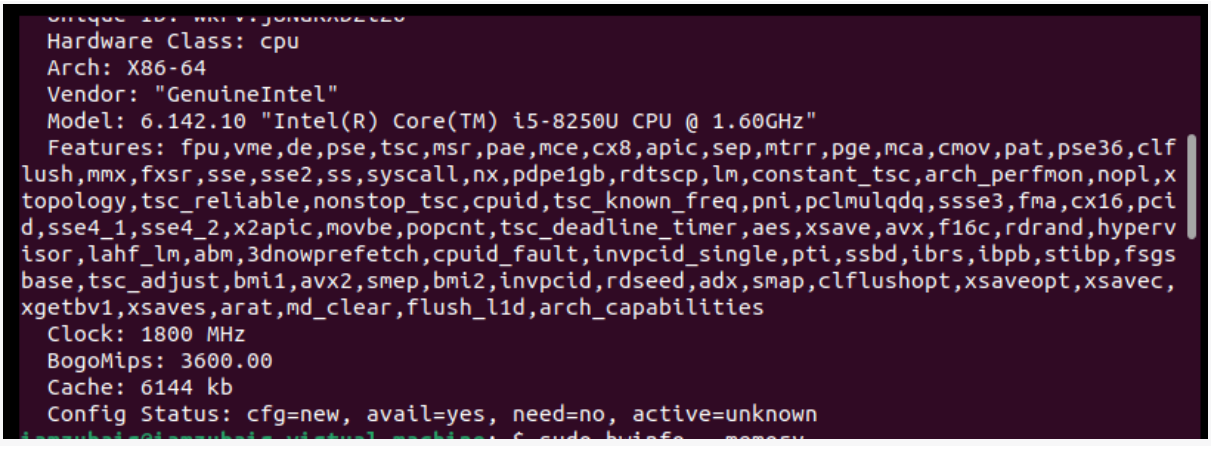
```
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --memory
01: None 00.0: 10102 Main Memory
  [Created at memory.74]
  Unique ID: rdCR.CxwsZFjVASF
  Hardware Class: memory
  Model: "Main Memory"
  Memory Range: 0x00000000-0x7949dfff (rw)
  Memory Size: 1 GB + 896 MB
  Config Status: cfg=new, avail=yes, need=no, active=unknown
```

3.To display information about the CPU



```
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --cpu
01: None 00.0: 10103 CPU
  [Created at cpu.465]
  Unique ID: rdCR.j8NaKXDZtZ6
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.142.10 "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz"
  Features: fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clf
lush,mmx,fxsr,sse,sse2,ss,syscall,nx,pdpe1gb,rdtscp,lm,constant_tsc,arch_perfmon,nopl,x
topology,tsc_reliable,nonstop_tsc,cpuid,tsc_known_freq,pni,pclmulqdq,ssse3,fma,cx16,pci
d,sse4_1,sse4_2,x2apic,movbe,popcnt,tsc_deadline_timer,aes,xsave,avx,f16c,rdrand,hyperv
isor,lahf_lm,abm,3dnowprefetch,cpuid_fault,invpcid_single,pti,ssbd,ibrs,ibpb,stibp,fsgs
base,tsc_adjust,bmi1,avx2,smep,bmi2,invpcid,rdseed,adx,smap,clflushopt,xsaveopt,xsavec,
xgetbv1,xsaves,arat,md_clear,flush_l1d,arch_capabilities
  Clock: 1800 MHz
  BogoMips: 3600.00
  Cache: 6144 kb
  Config Status: cfg=new, avail=yes, need=no, active=unknown

02: None 01.0: 10103 CPU
  [Created at cpu.465]
  Unique ID: wkFv.j8NaKXDZtZ6
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.142.10 "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz"
  Features: fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clf
lush,mmx,fxsr,sse,sse2,ss,syscall,nx,pdpe1gb,rdtscp,lm,constant_tsc,arch_perfmon,nopl,x
topology,tsc_reliable,nonstop_tsc,cpuid,tsc_known_freq,pni,pclmulqdq,ssse3,fma,cx16,pci
d,sse4_1,sse4_2,x2apic,movbe,popcnt,tsc_deadline_timer,aes,xsave,avx,f16c,rdrand,hyperv
isor,lahf_lm,abm,3dnowprefetch,cpuid_fault,invpcid_single,pti,ssbd,ibrs,ibpb,stibp,fsgs
base,tsc_adjust,bmi1,avx2,smep,bmi2,invpcid,rdseed,adx,smap,clflushopt,xsaveopt,xsavec,
```



```
  Hardware Class: cpu
  Arch: X86-64
  Vendor: "GenuineIntel"
  Model: 6.142.10 "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz"
  Features: fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clf
lush,mmx,fxsr,sse,sse2,ss,syscall,nx,pdpe1gb,rdtscp,lm,constant_tsc,arch_perfmon,nopl,x
topology,tsc_reliable,nonstop_tsc,cpuid,tsc_known_freq,pni,pclmulqdq,ssse3,fma,cx16,pci
d,sse4_1,sse4_2,x2apic,movbe,popcnt,tsc_deadline_timer,aes,xsave,avx,f16c,rdrand,hyperv
isor,lahf_lm,abm,3dnowprefetch,cpuid_fault,invpcid_single,pti,ssbd,ibrs,ibpb,stibp,fsgs
base,tsc_adjust,bmi1,avx2,smep,bmi2,invpcid,rdseed,adx,smap,clflushopt,xsaveopt,xsavec,
xgetbv1,xsaves,arat,md_clear,flush_l1d,arch_capabilities
  Clock: 1800 MHz
  BogoMips: 3600.00
  Cache: 6144 kb
  Config Status: cfg=new, avail=yes, need=no, active=unknown
```

4. To display information about the storage devices.

```
  Config Status: cfg=new, avail=yes, need=no, active=unknown
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --storage
01: None 00.0: 0102 Floppy disk controller
  [Created at floppy.112]
  Unique ID: rdCR.3wRL2_g4d2B
  Hardware Class: storage
  Model: "Floppy disk controller"
  I/O Port: 0x3f2 (rw)
  I/O Ports: 0x3f4-0x3f5 (rw)
  I/O Port: 0x3f7 (rw)
  DMA: 2
  Config Status: cfg=new, avail=yes, need=no, active=unknown

15: PCI 10.0: 0100 SCSI storage controller
  [Created at pci.386]
  Unique ID: 37TO.741NuwlerHD
  SysFS ID: /devices/pci0000:00/0000:00:10.0
  SysFS BusID: 0000:00:10.0
  Hardware Class: storage
  Model: "VMware LSI Logic Parallel SCSI Controller"
  Vendor: pci 0x1000 "Broadcom / LSI"
  Device: pci 0x0030 "53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI"
  SubVendor: pci 0x15ad "VMware, Inc."
  SubDevice: pci 0x1976 "LSI Logic Parallel SCSI Controller"
  Revision: 0x01
  Driver: "mptspi"
  Driver Modules: "mptspi"
  I/O Ports: 0x1400-0x14ff (rw)
  Memory Range: 0xfeba0000-0xfebbffff (rw,non-prefetchable)
  Memory Range: 0xfebc0000-0xfebdffff (rw,non-prefetchable)
  Memory Range: 0xc0000000-0xc0003fff (ro,non-prefetchable,disabled)
  IRQ: 17 (23223 events)
  Module Alias: "pci:v00001000d00000030sv000015ADsd00001976bc01sc00i00"
  Driver Info #0:
    Driver Status: mptspi is active
    Driver Activation Cmd: "modprobe mptspi"
  Config Status: cfg=new, avail=yes, need=no, active=unknown

46: PCI 07.1: 0101 IDE interface (ISA Compatibility mode controller, supports both chan
nels switched to PCI native mode, supports bus mastering)
  [Created at pci.386]
  Unique ID: ESW+.2kE1GEmwX42
  SysFS ID: /devices/pci0000:00/0000:00:07.1
  SysFS BusID: 0000:00:07.1
  Hardware Class: storage
  Model: "VMware Virtual Machine Chipset"
  Vendor: pci 0x8086 "Intel Corporation"
  Device: pci 0x7111 "82371AB/EB/MB PIIX4 IDE"
  SubVendor: pci 0x15ad "VMware, Inc."
  SubDevice: pci 0x1976 "Virtual Machine Chipset"
  Revision: 0x01
  Driver: "ata_piix"
  Driver Modules: "ata_piix"
  I/O Ports: 0x1f0-0x1f7 (rw)
  I/O Port: 0x3f6 (rw)
  I/O Ports: 0x170-0x177 (rw)
  I/O Port: 0x376 (rw)
  I/O Ports: 0x1060-0x106f (rw)
  Module Alias: "pci:v00008086d00007111sv000015ADsd00001976bc01sc01i8A"
  Driver Info #0:
    Driver Status: pata_acpi is active
    Driver Activation Cmd: "modprobe pata_acpi"
  Config Status: cfg=new, avail=yes, need=no, active=unknown

53: PCI 204.0: 0106 SATA controller (AHCI 1.0)
  [Created at pci.386]
  Unique ID: UVH2.UQtfkxJ89D8
  Parent ID: 7EWs.3XzZP_1GzlF
  SysFS ID: /devices/pci0000:00/0000:00:11.0/0000:02:04.0
  SysFS BusID: 0000:02:04.0
  Hardware Class: storage
  Device Name: "sata0"
  Model: "VMware SATA AHCI controller"
  Vendor: pci 0x15ad "VMware, Inc."
  Device: pci 0x07e0 "SATA AHCI controller"
  SubVendor: pci 0x15ad "VMware, Inc."
  SubDevice: pci 0x07e0
  Driver: "ahci"
  Driver Modules: "ahci"
  Memory Range: 0xfd5ee000-0xfd5eefff (rw,non-prefetchable)
  Memory Range: 0xfd510000-0xfd51ffff (ro,non-prefetchable,disabled)
  IRQ: 56 (1150 events)
  Module Alias: "pci:v000015ADd000007E0sv000015ADsd000007E0bc01sc06i01"
  Driver Info #0:
    Driver Status: ahci is active
    Driver Activation Cmd: "modprobe ahci"
  Config Status: cfg=new, avail=yes, need=no, active=unknown
  Attached to: #48 (PCI bridge)
```

5. To display information about the graphics card.

```
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --gfxcard
27: PCI 0f.0: 0300 VGA compatible controller (VGA)
  [Created at pci.386]
  Unique ID: _+Pw.jBKePf3JQB5
  SysFS ID: /devices/pci0000:00/0000:00:0f.0
  SysFS BusID: 0000:00:0f.0
  Hardware Class: graphics card
  Model: "VMware VMWARE0405"
  Vendor: pci 0x15ad "VMware, Inc."
  Device: pci 0x0405 "VMWARE0405"
  SubVendor: pci 0x15ad "VMware, Inc."
  SubDevice: pci 0x0405
  Driver: "vmwgfx"
  Driver Modules: "vmwgfx"
  I/O Ports: 0x1070-0x107f (rw)
  Memory Range: 0xe8000000-0xefffffff (ro,non-prefetchable)
  Memory Range: 0xfe000000-0xfe7fffff (rw,non-prefetchable)
  Memory Range: 0x000c0000-0x000dffff (rw,non-prefetchable,disabled)
  IRQ: 16 (2606 events)
  I/O Ports: 0x3c0-0x3df (rw)
  Module Alias: "pci:v000015ADd00000405sv000015ADsd00000405bc03sc00i00"
  Driver Info #0:
    XFree86 v4 Server Module: vmware
  Config Status: cfg=new, avail=yes, need=no, active=unknown

Primary display adapter: #27
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --netcard
```

6. To display information about the network devices.

```
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --netcard
47: PCI 201.0: 0200 Ethernet controller
  [Created at pci.386]
  Unique ID: qtsV.5dU8kR7eh2C
  Parent ID: 7EWs.3XzZP_1GzlF
  SysFS ID: /devices/pci0000:00/0000:00:11.0/0000:02:01.0
  SysFS BusID: 0000:02:01.0
  Hardware Class: network
  Device Name: "Ethernet0"
  Model: "VMware PRO/1000 MT Single Port Adapter"
  Vendor: pci 0x8086 "Intel Corporation"
  Device: pci 0x100f "82545EM Gigabit Ethernet Controller (Copper)"
  SubVendor: pci 0x15ad "VMware, Inc."
  SubDevice: pci 0x0750 "PRO/1000 MT Single Port Adapter"
  Revision: 0x01
  Driver: "e1000"
  Driver Modules: "e1000"
  Device File: ens33
  Memory Range: 0xfd5c0000-0xfd5dffff (rw,non-prefetchable)
  Memory Range: 0xfdff0000-0xfdffffff (rw,non-prefetchable)
  I/O Ports: 0x2000-0x3fff (rw)
  Memory Range: 0xfd500000-0xfd50ffff (ro,non-prefetchable,disabled)
  IRQ: 19 (579 events)
  HW Address: 00:0c:29:57:f6:b5
  Permanent HW Address: 00:0c:29:57:f6:b5
  Link detected: yes
  Module Alias: "pci:v00008086d0000100Fsv000015ADsd00000750bc02sc00i00"
  Driver Info #0:
    Driver Status: e1000 is active
    Driver Activation Cmd: "modprobe e1000"
  Config Status: cfg=new, avail=yes, need=no, active=unknown
  Attached to: #46 (PCI bridge)
```

7.To display information about the operating system.

```
  Attached to: #46 (PCI bridge)
iamzubair@iamzubair-virtual-machine:~$ sudo hwinfo --os
Usage: hwinfo [OPTIONS]
Probe for hardware.
Options:
    --<HARDWARE_ITEM>
        This option can be given more than once. Probe for a particular
        HARDWARE_ITEM. Available hardware items are:
        all, arch, bios, block, bluetooth, braille, bridge, camera,
        cdrom, chipcard, cpu, disk, dsl, dvb, fingerprint, floppy,
        framebuffer, gfxcard, hub, ide, isapnp, isdn, joystick, keyboard,
        memory, mmc-ctrl, modem, monitor, mouse, netcard, network, partition,
        pci, pcmcia, pcmcia-ctrl, pppoe, printer, redasd,
        reallyall, scanner, scsi, smp, sound, storage-ctrl, sys, tape,
        tv, uml, usb, usb-ctrl, vbe, wlan, xen, zip
    --short
        Show only a summary. Use this option in addition to a hardware
        probing option.
    --listmd
        Normally hwinfo does not report RAID devices. Add this option to
        see them.
    --only DEVNAME
        This option can be given more than once. If you add this option
        only entries in the device list matching DEVNAME will be shown.
        Note that you also have to specify --<HARDWARE_ITEM> to trigger
        any device probing.
    --save-config SPEC
        Store config  for a particular device below /var/lib/hardware.
        SPEC can be a device name, an UDI, or 'all'. This option must be
        given in addition to a hardware probing option.
    --show-config UDI
        Show saved config data for a particular device.
    --map
```

```
    --only DEVNAME
        This option can be given more than once. If you add this option
        only entries in the device list matching DEVNAME will be shown.
        Note that you also have to specify --<HARDWARE_ITEM> to trigger
        any device probing.
    --save-config SPEC
        Store config  for a particular device below /var/lib/hardware.
        SPEC can be a device name, an UDI, or 'all'. This option must be
        given in addition to a hardware probing option.
    --show-config UDI
        Show saved config data for a particular device.
    --map
        If disk names have  changed (e.g. after a kernel update) this
        prints a list of disk name mappings. Note  that  you must have
        used --save-config at some point before for this can work.
    --debug N
        Set debug level to N. The debug info is shown only in the log
        file. If you specify a log file, the debug level is implicitly
        set to a reasonable value (N is a bitmask of individual flags).
    --verbose
        Increase verbosity. Only together with --map.
    --log FILE
        Write log info to FILE.
        Don't forget to also specify --<HARDWARE_ITEM> to trigger any
        device probing.
    --dump-db N
        Dump hardware data base. N is either 0 for the external data
        base in /var/lib/hardware, or 1 for the internal data base.
    --version
        Print libhd version.
    --help
        Print usage.
iamzubair@iamzubair-virtual-machine:~$ s
```

References:

1. https://www.thewindowsclub.com/hwinfo32-hardware-information-tool#:~:text=HWiNFO%20is%20a%20software%20that,in%20detail%20in%20this%20article.
2. https://www.lifewire.com/hwinfo-review-2625766
3. https://www.lifewire.com/hwinfo-review-2625766
4. https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/13_IOSystems.html
5. https://www.umsl.edu/~joshik/msis480/chapt05.htm
6. https://www.techtarget.com/searchnetworking/definition/hardware