

# **INT - 301: OPEN SOURCE TECHNOLOGIES**

## **A Project Report**

Submitted in partial fulfilment for the requirements of the award of the degree of

### **Bachelor of Technology Computer Science Engineering (Hons.)**

**Lovely Professional University  
Phagwara, Punjab.**



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

**Submitted by:**

**Name of the Student:** Mohammad Rasool

**Registration Number:** 11901876

**Roll No:** 03

**Section:** KE059

**CA-3**

**Q42. Using desired Open Source Software display an overview of all the hardware and operating system detail; also do live monitoring to show the temperature and current usage of various hardware components.**

To display an overview of hardware and operating system details in Linux, you can use the following open-source software:

## 1. Lshw

- It is a command-line utility that provides detailed information about the hardware in our system.

To install Lshw on Ubuntu or Debian-based systems, we can use the following command:

**sudo apt-get install lshw**

Once installed, we can use the following command to display the hardware details:

We can use this command:

**sudo lshw**

```
rsasool@rsasool-VirtualBox:~$ sudo lshw
rsasool-virtualbox
description: Computer
product: VirtualBox
vendor: innotek GmbH
version: 1.2
serial: 0
width: 64 bits
capabilities: smbios-2.5 dmi-2.5 smp vsyscall32
configuration: family=Virtual Machine uuid=42A6E396-92C2-FF4A-9A91-17A10A9F21F5
*-core
  description: Motherboard
  product: VirtualBox
  vendor: Oracle Corporation
  physical id: 0
  version: 1.2
  serial: 0
*-firmware
  description: BIOS
  vendor: innotek GmbH
  physical id: 0
  version: VirtualBox
  date: 12/01/2006
  size: 128KiB
  capacity: 128KiB
  capabilities: isa pci cdboot bootselect int9keyboard int10video acpi
*-memory
  description: System memory
  physical id: 1
  size: 5240MiB
*-cpu
  product: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
  vendor: Intel Corp.
  physical id: 2
  bus info: cpu@0
  width: 64 bits
  capabilities: fpu fpu_exception up vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx x86-64 constant tsc rep_good nopl xtopology nonstop tsc cpuid tsc_known_freq p
ni pcmlqdgq sse3 cx16 pcid sse4_1 sse4_2 movbe popcnt aes rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single ibrs_enhanced fsgsbase bmi1 bmi2 invpcid rdseed clflushopt md_clear flush_l1d arch_capabilities
*-pci
  description: Host bridge
  product: 440FX - 82441FX PMC [Natoma]
  vendor: Intel Corporation
  physical id: 100
  bus info: pci@0000:00:00.0
  version: 02
  width: 32 bits
  clock: 33MHz
*-isa
  description: ISA bridge
  product: 82371SB PIIX3 ISA [Natoma/Triton II]
  vendor: Intel Corporation
  physical id: 1
  bus info: pci@0000:00:01.0
```

```

*--isa
  description: ISA bridge
  product: 82371AB PIIX4 ISA [Matoma/Triton II]
  vendor: Intel Corporation
  physical id: 1
  bus info: pci@0000:00:01:0
  version: 00
  width: 32 bits
  clock: 33MHz
  capabilities: isa bus master
  configuration: latency=0
*--ide
  description: IDE interface
  product: 82371AB/EB/MB PIIX4 IDE
  vendor: Intel Corporation
  physical id: 1.1
  bus info: pci@0000:00:01:1
  logical name: scsi1
  version: 01
  width: 32 bits
  clock: 33MHz
  capabilities: ide isa compat mode pci native mode bus master emulated
  configuration: driver=ata_piix latency=64
  resources: irq:0 ioport:1f0(size=8) ioport:3f6 ioport:170(size=8) ioport:376 ioport:0000(size=16)
*--cdrom
  description: DVD reader
  product: CD-ROM
  vendor: VBOX
  physical id: 0.0.0
  bus info: scsi@0:0.0
  logical name: /dev/cdrom
  logical name: /dev/dvd
  logical name: /dev/sr0
  logical name: /media/rasool/VBox_GAs_7.0.6
  version: 1.0
  capabilities: removable audio dvd
  configuration: ansiversion=5 mount.fstype=iso9660 mount.options=ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400 state=mounted status=ready
*--medium
  physical id: 0
  logical name: /dev/cdrom
  logical name: /media/rasool/VBox_GAs_7.0.6
  configuration: mount.fstype=iso9660 mount.options=ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400 state=mounted
*--display
  description: VGA compatible controller
  product: SVGA II Adapter
  vendor: VMware
  physical id: 2
  bus info: pci@0000:00:02:0
  version: 00
  width: 32 bits
  clock: 33MHz
  capabilities: vga controller bus master rom
  configuration: driver=vmwgfx latency=64

```

```

*--network
  description: Ethernet interface
  product: 82540EM Gigabit Ethernet Controller
  vendor: Intel Corporation
  physical id: 3
  bus info: pci@0000:00:03:0
  logical name: enp0s3
  version: 02
  serial: 00:00:27:b4:eb:3f
  size: 100bit/s
  capacity: 10Gbit/s
  width: 32 bits
  clock: 60MHz
  capabilities: pm pciix bus master cap list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
  configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=7.3.21-k8-NAPI duplex=full ip=10.0.2.15 latency=64 link=yes mii=nbt=255 multicast=yes port=twisted pair speed=10Gbit/s
  resources: irq:19 memory:f0400000-f021ffff ioport:0020(size=8)
*--generic
  description: System peripheral
  product: VirtualBox Guest Service
  vendor: InnoTek Systemberatung GmbH
  physical id: 4
  bus info: pci@0000:00:04:0
  version: 00
  width: 32 bits
  clock: 33MHz
  configuration: drivers=vboxguest latency=0
  resources: irq:20 ioport:d040(size=32) memory:f0400000-f07fffff memory:f0800000-f0803fff
*--multimedia
  description: Multimedia audio controller
  product: 82801AA AC'97 Audio Controller
  vendor: Intel Corporation
  physical id: 5
  bus info: pci@0000:00:05:0
  version: 01
  width: 32 bits
  clock: 33MHz
  capabilities: bus master
  configuration: drivers=snd_intel8x0 latency=64
  resources: irq:21 ioport:d100(size=256) ioport:d200(size=64)
*--usb
  description: USB controller
  product: KeyLargo/Intrepid USB
  vendor: Apple Inc.
  physical id: 6
  bus info: pci@0000:00:06:0
  version: 00
  width: 32 bits
  clock: 33MHz
  capabilities: ohci bus master cap list
  configuration: drivers=ohci-pci latency=64
  resources: irq:22 memory:f0804000-f0804fff
*--usbhost
  product: OHCI PCI host controller

```

```

*--usbhost
  product: OHCI PCI host controller
  vendor: Linux 5.4.0-58-generic ohci_hcd
  physical id: 1
  bus info: usb@1
  logical name: usb1
  version: 5.04
  capabilities: usb-1.10
  configuration: driver=hub slots=12 speed=12Mbit/s
*--usb
  description: Human interface device
  product: USB Tablet
  vendor: VirtualBox
  physical id: 1
  bus info: usb@1:1
  version: 1.00
  capabilities: usb-1.10
  configuration: driver=usbhid maxpower=100mA speed=12Mbit/s
*--bridge
  description: Bridge
  product: 82371AB/EB/MB PIIX4 ACPI
  vendor: Intel Corporation
  physical id: 7
  bus info: pci@0000:00:07:0
  version: 00
  width: 32 bits
  clock: 33MHz
  capabilities: bridge
  configuration: driver=pix4_smbus latency=0
  resources: irq:9
*--sata
  description: SATA controller
  product: 82801JM/JHM ICH8M/ICH8M-E SATA Controller [AHCI mode]
  vendor: Intel Corporation
  physical id: d
  bus info: pci@0000:00:0d:0
  logical name: scsi2
  version: 02
  width: 32 bits
  clock: 33MHz
  capabilities: sata pm ahci 1.0 bus master cap list emulated
  configuration: driver=ahci latency=64
  resources: irq:21 ioport:d240(size=8) ioport:d248(size=4) ioport:d250(size=8) ioport:d258(size=4) ioport:d260(size=16) memory:f0806000-f0807fff
*--disk
  description: ATA Disk
  product: VBOX HARDDISK
  vendor: VirtualBox
  physical id: 0.0.0
  bus info: scsi@2:0.0.0
  logical name: /dev/sda
  version: 1.0
  serial: VB01398449-17b9eb01
  size: 100GiB (107GB)

```

```
vendor: VirtualBox
physical id: 0.0.0
bus info: scsi02:0.0.0
logical name: /dev/sda
version: 1.0
serial: VB61398449-f7b9ebef1
size: 100618 (10700)
capabilities: partitioned partitioned:dos
configuration: ansiversion=5 logicalsectorsize=512 sectorsize=512 signature=0888f47e
*-volume:0
  description: Windows FAT volume
  vendor: mKfs.fat
  physical id: 1
  bus info: scsi02:0.0.0.1
  logical name: /dev/sda1
  logical name: /boot/efi
  version: FAT32
  serial: 30ab-a537
  size: 518MiB
  capacity: 512MiB
  capabilities: primary bootable fat initialized
  configuration: FATs=2 filesystem=fat mount.options=rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro state=mounted
*-volume:1
  description: Extended partition
  physical id: 2
  bus info: scsi02:0.0.0.2
  logical name: /dev/sda2
  size: 99618
  capacity: 99618
  capabilities: primary extended partitioned:extended
*-logicalvolume
  description: EXT4 volume
  vendor: Linux
  physical id: 5
  logical name: /dev/sda5
  logical name: /
  version: 1.0
  serial: 847ae4ed-3473-4e7d-bfab-c394235164f9
  size: 99618
  capacity: 99618
  capabilities: journaled extended attributes large files huge files dir.nlink recover 64bit extents ext4 ext2 initialized
  configuration: created=2021-03-14 15:48:50 filesystem=ext4 lastmountpoint=/ modified=2023-04-07 15:50:11 mount.fstype=ext4 mount.options=rw,relatime,errors=remount-ro mounted=2023-04-07 15:50:13 state=mounted
*-ppp00:00
  product: PnP device PNP0303
  physical id: 3
  capabilities: pnp
  configuration: driver=1B042 kbd
*-ppp00:01
  product: PnP device PNP0F03
  physical id: 4
  capabilities: pnp
  configuration: driver=1B042 aux
rasool@rasool-VirtualBox:~$
```

## 2. Neofetch

- It is a command-line system information tool that displays the operating system, CPU, GPU, RAM, and other system details in an aesthetically pleasing way.

To install Neofetch on Ubuntu or Debian-based systems, use the following command:

**sudo apt-get install neofetch**

Once installed, we can use the following command to display the system details:

**sudo neofetch**

To monitor the temperature and current usage of various hardware components in Linux, you can use the following open-source software:

```
rasool@rasool-VirtualBox:~$ sudo neofetch
root@rasool-VirtualBox
-----
OS: Linux Mint 20.1 x86_64
Host: VirtualBox 1.2
Kernel: 5.4.0-58-generic
Uptime: 3 hours, 27 mins
Packages: 2013 (dpkg)
Shell: bash 5.0.17
Resolution: 1920x1004
DE: Cinnamon
WM: Muttter (Muffin)
WM Theme: Mint-Y-Dark (Mint-Y)
Theme: Mint-Y [GTK2/3]
Icons: Mint-Y [GTK2/3]
Terminal: gnome-terminal
CPU: Intel i5-10300H (4) @ 2.495GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 955MiB / 5015MiB

rasool@rasool-VirtualBox:~$
```

### 3. Im-sensors

- It is a command-line utility that provides information about the temperature, voltage, and fan speed sensors in your system.

To install Im-sensors on Ubuntu or Debian-based systems, use the following command:

**sudo apt-get install lm-sensors**

Once installed, we can use the following command to detect the sensors in our system:

**sudo sensors-detect**

```

rasool@rasool-VirtualBox:~$ sudo sensors-detect
# sensors-detect version 3.6.0
# System: innotek GmbH VirtualBox [1.2]
# Board: Oracle Corporation VirtualBox
# Kernel: 5.4.0-58-generic x86_64
# Processor: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz (6/165/2)

This program will help you determine which kernel modules you need
to load to use lm_sensors most effectively. It is generally safe
and recommended to accept the default answers to all questions,
unless you know what you're doing.

Some south bridges, CPUs or memory controllers contain embedded sensors.
Do you want to scan for them? This is totally safe. (YES/no): y
Silicon Integrated Systems SIS5595... No
VIA VT82C686 Integrated Sensors... No
VIA VT8231 Integrated Sensors... No
AMD K8 thermal sensors... No
AMD Family 10h thermal sensors... No
AMD Family 11h thermal sensors... No
AMD Family 12h and 14h thermal sensors... No
AMD Family 15h thermal sensors... No
AMD Family 16h thermal sensors... No
AMD Family 17h thermal sensors... No
AMD Family 15h power sensors... No
AMD Family 16h power sensors... No
Hygon Family 18h thermal sensors... No
Intel digital thermal sensor... No
Intel AMB FB-DIMM thermal sensor... No
Intel 5500/5520/X58 thermal sensor... No
VIA C7 thermal sensor... No
VIA Nano thermal sensor... No

Some Super I/O chips contain embedded sensors. We have to write to
standard I/O ports to probe them. This is usually safe.
Do you want to scan for Super I/O sensors? (YES/no): y
Probing for Super-I/O at 0x2e/0x2f
Trying family `National Semiconductor/ITE'... No
Trying family `SMSC'... No
Trying family `VIA/Winbond/Nuvoton/Fintek'... No
Trying family `ITE'... No
Probing for Super-I/O at 0x4e/0x4f
Trying family `National Semiconductor/ITE'... No
Trying family `SMSC'... No
Trying family `VIA/Winbond/Nuvoton/Fintek'... No
Trying family `ITE'... No

Some systems (mainly servers) implement IPMI, a set of common interfaces
through which system health data may be retrieved, amongst other things.
We first try to get the information from SMBIOS. If we don't find it
there, we have to read from arbitrary I/O ports to probe for such
interfaces. This is normally safe. Do you want to scan for IPMI
interfaces? (YES/no): y

```

```

Some systems (mainly servers) implement IPMI, a set of common interfaces
through which system health data may be retrieved, amongst other things.
We first try to get the information from SMBIOS. If we don't find it
there, we have to read from arbitrary I/O ports to probe for such
interfaces. This is normally safe. Do you want to scan for IPMI
interfaces? (YES/no): y
Probing for `IPMI BMC KCS' at 0xca0...      No
Probing for `IPMI BMC SMIC' at 0xca8...     No

Some hardware monitoring chips are accessible through the ISA I/O ports.
We have to write to arbitrary I/O ports to probe them. This is usually
safe though. Yes, you do have ISA I/O ports even if you do not have any
ISA slots! Do you want to scan the ISA I/O ports? (YES/no): y
Probing for `National Semiconductor LM78' at 0x290...    No
Probing for `National Semiconductor LM79' at 0x290...    No
Probing for `Winbond W83781D' at 0x290...               No
Probing for `Winbond W83782D' at 0x290...               No

Lastly, we can probe the I2C/SMBus adapters for connected hardware
monitoring devices. This is the most risky part, and while it works
reasonably well on most systems, it has been reported to cause trouble
on some systems.
Do you want to probe the I2C/SMBus adapters now? (YES/no): y
Using driver `i2c-piix4' for device 0000:00:07.0: Intel 82371AB PIIX4 ACPI

Next adapter: SMBus PIIX4 adapter at 4100 (i2c-0)
Do you want to scan it? (YES/no/selectively): y

Sorry, no sensors were detected.
Either your system has no sensors, or they are not supported, or
they are connected to an I2C or SMBus adapter that is not
supported. If you find out what chips are on your board, check
https://hwmon.wiki.kernel.org/device\_support\_status for driver status.

```

After detecting the sensors, you can use the following command to display the sensor readings:

## sudo sensors

```

rasool@rasool-VirtualBox:~$ sensors
BAT0-acpi-0
Adapter: ACPI interface
in0:          10.00 V

rasool@rasool-VirtualBox:~$

```



## 4. htop

- It is a command-line utility that provides real-time monitoring of CPU usage, memory usage, and other system details.

To install htop on Ubuntu or Debian-based systems, use the following command:

```
sudo apt-get install htop
```

Once installed, we can use the following command to display the system details:

## htop

We can use these open-source software tools to display an overview of hardware and operating system details and monitor the temperature and current usage of various hardware components in Linux.

```

1 [ ] 0.91] Tasks: 116, 289 thr: 1 running
2 [ ] 0.91] Load average: 0.08 0.11 0.04
3 [ ] 0.70] Uptime: 03:30:23
4 [ ] 0.70]
5 [ ]
6 [ ]
7 [ ]
8 [ ]
9 [ ]
10 [ ]
11 [ ]
12 [ ]
13 [ ]
14 [ ]
15 [ ]
16 [ ]
17 [ ]
18 [ ]
19 [ ]
20 [ ]
21 [ ]
22 [ ]
23 [ ]
24 [ ]
25 [ ]
26 [ ]
27 [ ]
28 [ ]
29 [ ]
30 [ ]
31 [ ]
32 [ ]
33 [ ]
34 [ ]
35 [ ]
36 [ ]
37 [ ]
38 [ ]
39 [ ]
40 [ ]
41 [ ]
42 [ ]
43 [ ]
44 [ ]
45 [ ]
46 [ ]
47 [ ]
48 [ ]
49 [ ]
50 [ ]
51 [ ]
52 [ ]
53 [ ]
54 [ ]
55 [ ]
56 [ ]
57 [ ]
58 [ ]
59 [ ]
60 [ ]
61 [ ]
62 [ ]
63 [ ]
64 [ ]
65 [ ]
66 [ ]
67 [ ]
68 [ ]
69 [ ]
70 [ ]
71 [ ]
72 [ ]
73 [ ]
74 [ ]
75 [ ]
76 [ ]
77 [ ]
78 [ ]
79 [ ]
80 [ ]
81 [ ]
82 [ ]
83 [ ]
84 [ ]
85 [ ]
86 [ ]
87 [ ]
88 [ ]
89 [ ]
90 [ ]
91 [ ]
92 [ ]
93 [ ]
94 [ ]
95 [ ]
96 [ ]
97 [ ]
98 [ ]
99 [ ]
100 [ ]

```



Created a File called “main” and using functions I had executed all 4 commands for getting info of Hardware overview & operating system details, and live monitoring of temperature and current working of various hardware components

```
function cpuusageInfo() {
echo -e "-----\n"
echo -e "-----\n"

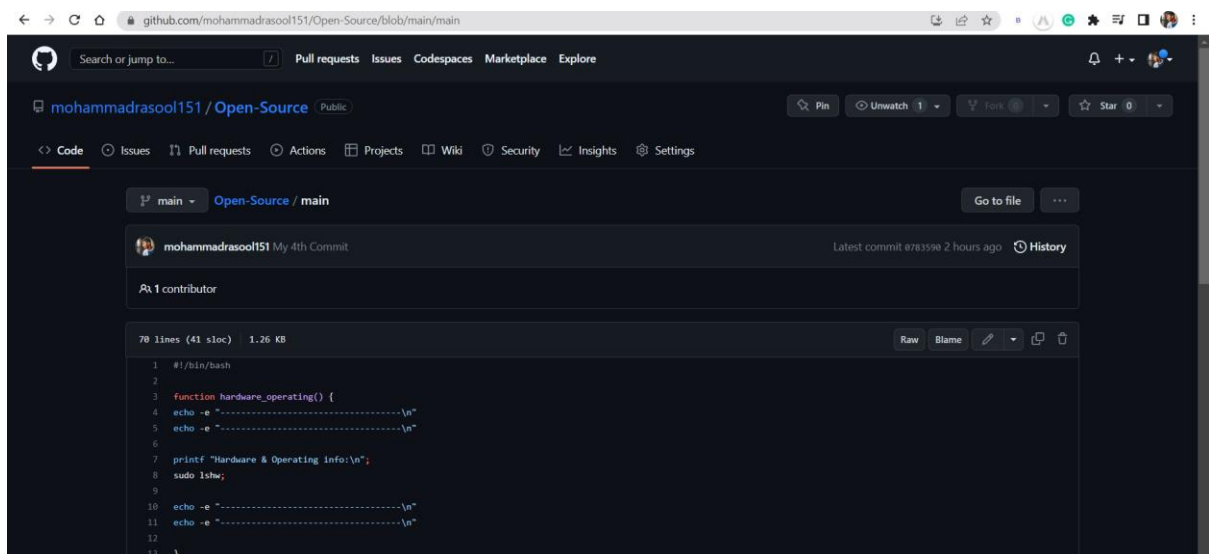
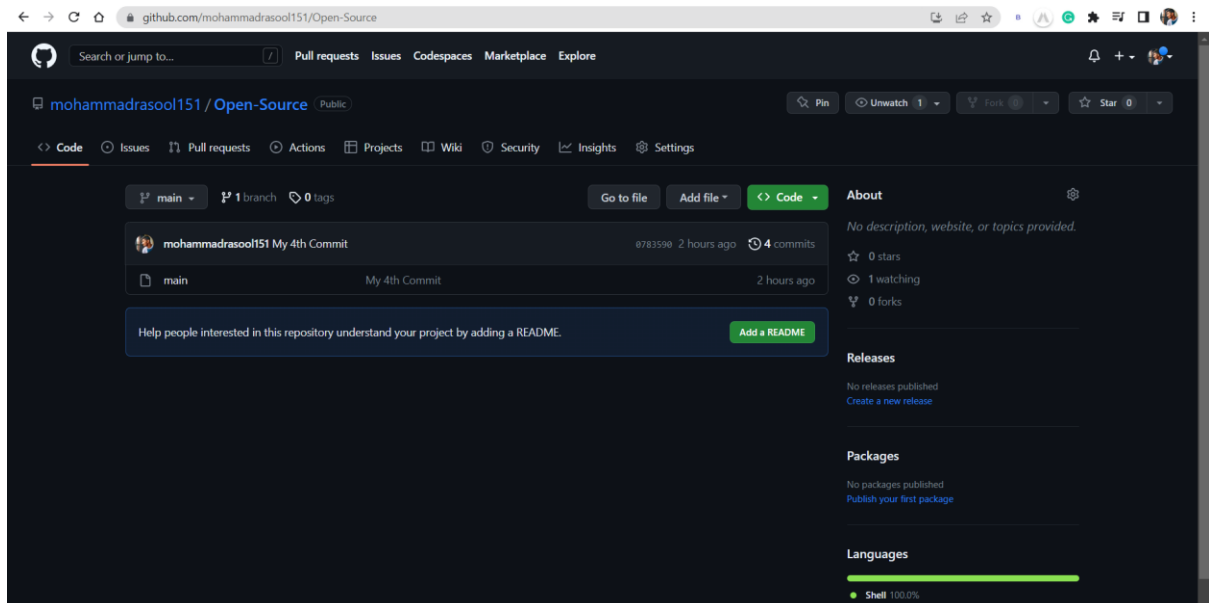
printf "Monitoring of CPU Usage:\n"
sudo htop

echo -e "-----\n"
echo -e "-----\n"

echo "Report of Hardware & operating system details..."

hardware operating
operatingsystemInfo
sensorsInfo
cpuusageInfo
}
```

# GITHUB:



```

15
16 function operatingsystemInfo() {
17     echo -e "-----\n"
18     echo -e "-----\n"
19
20     printf "Operating System Info in graphical way:\n";
21     sudo neofetch
22     echo -e "-----\n"
23     echo -e "-----\n"
24
25 }
26
27 function sensorsInfo() {
28     echo -e "-----\n"
29     echo -e "-----\n"
30
31     printf "Temperature and current usage of Hardware systems:\n";
32     sudo sensors-detect;
33     sudo sensors;
34
35     echo -e "-----\n"
36     echo -e "-----\n"
37
38 }
39
40 function cpuusageInfo() {
41     echo -e "-----\n"
42     echo -e "-----\n"
43
44     printf "Monitoring of CPU Usage:\n"
45     sudo htop
46
47     echo -e "-----\n"
48     echo -e "-----\n"
49
50 }
51

```

```

54
55 echo "Report of all Hardware & operating system details, sensors(temperature), cpu usage...!"
56 {
57     hardware_operating
58     operatingsystemInfo
59     sensorsInfo
60     cpuusageInfo
61 }
62

```

## **GIT CODE:**

```
#!/bin/bash
```

```
function hardware_operating() {  
echo -e "-----\n"  
echo -e "-----\n"  
  
printf "Hardware & Operating info:\n";  
sudo lshw;  
  
echo -e "-----\n"  
echo -e "-----\n"  
  
}
```

```
function operatingsystemInfo() {  
echo -e "-----\n"  
echo -e "-----\n"  
  
printf "Operating System Info in graphical way:\n";  
sudo neofetch  
echo -e "-----\n"  
echo -e "-----\n"  
  
}
```

```
function sensorsInfo() {  
echo -e "-----\n"  
echo -e "-----\n"
```

```
printf "Temperature and current usage of Hardware systems:\n";  
sudo sensors-detect;  
sudo sensors;
```

```
echo -e "-----\n"  
echo -e "-----\n"
```

```
}
```

```
function cpuusageInfo() {  
echo -e "-----\n"  
echo -e "-----\n"
```

```
printf "Monitoring of CPU Usage:\n"  
sudo htop
```

```
echo -e "-----\n"  
echo -e "-----\n"
```

```
}
```

```
echo "Report of all Hardware & operating system details,  
sensors(temperature), cpu usage...!"
```

```
{  
hardware_operating  
operatingsystemInfo  
sensorsInfo  
cpuusageInfo  
}
```

-Thank You