Command-Line Diagnostic Tools for Computer Components

The command line provides powerful tools for diagnosing and examining computer hardware components. Here are comprehensive commands for Windows (CMD and PowerShell) and Linux/macOS (Bash) that help you retrieve detailed information about each component.

System Overview Commands

Windows CMD

```
systeminfo
```

This command displays detailed configuration information about a computer and its operating system, including hardware resources, component versions, network information, and more.

Windows PowerShell

```
Get-ComputerInfo
```

This provides comprehensive system information including hardware details, operating system configuration, and network settings in a structured format.

Linux/macOS Bash

```
sudo lshw
```

Lists hardware information for the entire system. The output is detailed and hierarchical.

```
sudo dmidecode
```

Displays the Desktop Management Interface (DMI) data which provides a complete inventory of your hardware components.

CPU Information Commands

Windows CMD

```
wmic cpu get name, maxclockspeed, currentclockspeed, numberofcores, numberoflogicalprocessor
```

This displays the processor name, maximum speed, current speed, physical cores, and logical processors.

Windows PowerShell

```
Get-WmiObject Win32_Processor | Select-Object Name, MaxClockSpeed, CurrentClockSpeed, Number
```

Provides detailed CPU information in a more structured format than CMD.

```
Get-CimInstance -ClassName Win32_Processor | Format-List *
```

Shows all available CPU properties for in-depth analysis.

Linux/macOS Bash

```
cat /proc/cpuinfo
```

Displays detailed information about each CPU core in the system.

```
lscpu
```

Shows architecture information, CPU op-modes, number of cores, cache sizes, and more in a condensed format.

Memory (RAM) Information Commands

Windows CMD

```
wmic memorychip get capacity, speed, manufacturer
```

Lists each memory module with its capacity, speed, and manufacturer.

```
wmic memphysical get maxcapacity, memorydevices
```

Shows maximum supported memory capacity and number of memory slots.

Windows PowerShell

```
Get-WmiObject Win32_PhysicalMemory | Select-Object Capacity, Speed, Manufacturer, DeviceLoca
```

Provides detailed information about each memory module including its physical location.

```
Get-CimInstance -ClassName Win32_PhysicalMemoryArray
```

Shows information about the physical memory array, including location, use, and memory error correction capabilities.

Linux/macOS Bash

```
free -h
```

Displays the total amount of free and used physical and swap memory in human-readable format.

```
sudo dmidecode --type memory
```

Shows detailed information about each memory slot and installed memory modules.

Storage Device Information Commands

Windows CMD

```
wmic diskdrive get model, size, interfacetype
```

Lists each disk drive with its model, size, and interface type.

```
wmic logicaldisk get name, description, size, freespace
```

Shows logical drives with their descriptions, total size, and free space.

Windows PowerShell

```
Get-PhysicalDisk | Format-Table -AutoSize
```

Displays physical disk information including health status and operational status.

```
Get-Disk | Format-Table -AutoSize
```

Shows disk configuration information including partition style and operational status.

```
Get-Volume | Format-Table -AutoSize
```

Displays volume information including drive letter, file system, and space usage.

Linux/macOS Bash

```
lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
```

Lists block devices in a tree-like format with names, sizes, filesystem types, and mount points.

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

Shows SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes for a specific drive (replace 'sda' with your drive).

```
df -h
```

Reports file system disk space usage in human-readable format.

Motherboard and BIOS Information Commands

Windows CMD

```
wmic baseboard get product, manufacturer, version, serialnumber
```

Displays motherboard information including product name, manufacturer, version, and serial number.

```
wmic bios get manufacturer, name, version
```

Shows BIOS manufacturer, name, and version.

Windows PowerShell

```
Get-WmiObject Win32_BaseBoard | Format-List *
```

Provides comprehensive motherboard information.

```
Get-WmiObject Win32_BIOS | Format-List *
```

Shows detailed BIOS information including release date and status.

Linux/macOS Bash

```
sudo dmidecode -t baseboard
```

Displays detailed information about the motherboard.

```
sudo dmidecode -t bios
```

Shows BIOS details including vendor, version, and release date.

Graphics Card (GPU) Information Commands

Windows CMD

```
wmic path win32_VideoController get name, adapterram, driverversion
```

Lists installed graphics cards with name, memory size, and driver version.

Windows PowerShell

```
Get-WmiObject Win32_VideoController | Format-List Name, AdapterRAM, DriverVersion, VideoMode
```

Provides detailed information about graphics adapters including current display settings.

Linux/macOS Bash

```
lspci | grep -i vga
```

Lists VGA compatible devices on the PCI bus.

```
glxinfo | grep "OpenGL renderer"
```

Shows the OpenGL renderer string, which usually indicates the graphics card in use.

```
nvidia-smi
```

For NVIDIA GPUs, displays detailed information including temperature, memory usage, and power consumption.

Network Adapter Information Commands

Windows CMD

```
ipconfig /all
```

Displays all current TCP/IP network configuration values.

```
wmic nicconfig get description, macaddress, ipaddress
```

Lists network adapters with descriptions, MAC addresses, and IP addresses.

Windows PowerShell

```
Get-NetAdapter | Format-Table Name, InterfaceDescription, Status, LinkSpeed -AutoSize
```

Shows network adapters with their descriptions, status, and link speed.

```
Get-NetIPConfiguration | Format-Table InterfaceAlias, IPv4Address, IPv6Address -AutoSize
```

Displays IP configuration for each network interface.

Linux/macOS Bash

```
ifconfig -a
```

Shows all network interfaces with their configurations (may require `net-tools` package on some distributions).

```
ip addr
```

Modern replacement for ifconfig, showing all network interfaces and addresses.

```
lspci | grep -i network
```

Lists network controllers on the PCI bus.

Peripheral and USB Device Information Commands

Windows CMD

```
wmic path Win32_USBController get Name, DeviceID
```

Lists USB controllers with names and device IDs.

```
wmic path Win32_USBHub get DeviceID, PNPDeviceID
```

Shows USB hubs with device IDs.

Windows PowerShell

```
Get-PnpDevice -Class USB | Format-Table Status, Class, FriendlyName, InstanceId -AutoSize
```

Lists all USB devices with their status and friendly names.

Linux/macOS Bash

```
lsusb
```

Lists USB devices connected to the system.

```
lspci
```

Lists all PCI devices, which includes many peripheral controllers.

Temperature, Fan Speed, and Power Information Commands

Windows PowerShell

```
# Requires installation of external modules like HWiNFO or Open Hardware Monitor
Get-CimInstance -Namespace root\OpenHardwareMonitor -ClassName Sensor | Where-Object SensorT
```

This requires third-party software with WMI providers like Open Hardware Monitor.

Linux/macOS Bash

```
sensors
```

Shows temperature readings from hardware sensors (requires `lm-sensors` package).

```
sudo powermetrics
```

On macOS, displays detailed power and thermal information.

Advanced System Diagnostics Commands

Windows PowerShell

```
Get-WinEvent -LogName System -MaxEvents 100 | Where-Object {$_.LevelDisplayName -eq "Error"}
```

Shows recent system error events that may indicate hardware problems.

```
Get-CimInstance -ClassName Win32_ReliabilityRecords | Select-Object -First 10 | Format-List
```

Displays recent reliability issues including hardware failures.

Linux/macOS Bash

```
dmesg | grep -i error
```

Shows kernel messages with errors, which often indicate hardware issues.

```
sudo smartctl -t short /dev/sda && sleep 120 && sudo smartctl -l selftest /dev/sda
```

Runs a short SMART self-test on a drive and displays the results after waiting for completion.

Command Groups for Comprehensive System Analysis

Windows PowerShell Script for Complete Hardware Report

```
# Create a comprehensive hardware report
$report = @()
$report += "=== SYSTEM INFORMATION ==="
$report += Get-ComputerInfo | Format-List
$report += "`n=== PROCESSOR INFORMATION ==="
$report += Get-WmiObject Win32_Processor | Format-List
$report += "`n=== MEMORY INFORMATION ==="
$report += Get-WmiObject Win32_PhysicalMemory | Format-Table
$report += "`n=== DISK INFORMATION ==="
$report += Get-PhysicalDisk | Format-Table
$report += Get-Volume | Format-Table
$report += "`n=== VIDEO CONTROLLER INFORMATION ==="
$report += Get-WmiObject Win32_VideoController | Format-List
$report += "`n=== NETWORK ADAPTER INFORMATION ==="
$report += Get-NetAdapter | Format-Table
$report | Out-File -FilePath "$env:USERPROFILE\Desktop\HardwareReport.txt"
```

This script creates a comprehensive hardware report and saves it to your desktop.

Linux Bash Script for Complete Hardware Report

```
#!/bin/bash
# Create a comprehensive hardware report
echo "=== SYSTEM INFORMATION ===" > hardware_report.txt
sudo lshw >> hardware_report.txt
echo -e "\n=== PROCESSOR INFORMATION ===" >> hardware_report.txt
lscpu >> hardware_report.txt
echo -e "\n=== MEMORY INFORMATION ===" >> hardware_report.txt
free -h >> hardware_report.txt
sudo dmidecode --type memory >> hardware_report.txt
echo -e "\n=== DISK INFORMATION ===" >> hardware_report.txt
lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT >> hardware_report.txt
df -h >> hardware_report.txt
echo -e "\n=== GPU INFORMATION ===" >> hardware_report.txt
lspci | grep -i vga >> hardware_report.txt
if command -v nvidia-smi &> /dev/null; then
    nvidia-smi >> hardware_report.txt
fi
echo -e "\n=== NETWORK INFORMATION ===" >> hardware_report.txt
ip addr >> hardware_report.txt
echo -e "\n=== TEMPERATURE INFORMATION ===" >> hardware_report.txt
if command -v sensors &> /dev/null; then
    sensors >> hardware_report.txt
fi
echo "Hardware report saved to $(pwd)/hardware_report.txt"
```

This script creates a comprehensive hardware report in the current directory.

These command-line tools provide deep insights into your computer's hardware components. By using these commands regularly, you can monitor component health, diagnose problems, and make informed decisions about upgrades or repairs. Remember that some of these commands require administrative privileges, especially those that access low-level hardware information.