The BIOS

BIOS (Basic input output system):
The BIOS is a small program stored in **EPROM (Erasable programmable read-only memory**) that gets the computer started once it is turned on.

The fundamental purpose of a BIOS is to **initialize and test the system hardware components** and to **load the operating system** (or key parts of it) from the hard disk **into the RAM.**

BIOS historically was used as an abstraction layer which allowed a consistent way for application programs and the operating system to interact with input-output devices. Modern PCs **do not us the BIOS after the operating system is loaded.**

The BIOS program is stored in a read-only (flash memory ROM) chip which is **the firmware for the motherboard**. It controls input and output operations that are essential for communicating with the computer.

On personal computers, for example, the BIOS contains all the code required to control **the keyboard, display screen, disk drives, serial communications** and a number of miscellaneous functions.

The BIOS is stored on ROM since **this prevents it from being corrupted by power losses** or **changed by the user** and so the BIOS software will **always be present on the computer**. Since it is a form of flash memory, it has fast access speeds. Modern computers store this on a flash chip (EPROM) so it can be updated with the latest manufacture software if desired.

Sequence of events

The AC power switch is pressed and power runs to the motherboard and connected components. The CPU initializes itself and the clock begins to tick. The CPU then locates the non-volatile ROM BIOS.

The BIOS is the first thing to run when the computer is started up. It sends a hardware reset signal along the control bus to the CPU.
The BIOS runs a POST program (power on self-test); the BIOS needs to check for hardware failures and test the CMOS and RAM.

If the CMOS battery has power, then the POST test can check to ensure all the hardware components are functioning correctly. If the POST test fails - a component is not functioning - then a pattern of beeps is sounded that refers to an error code by which a pc specialist can identify the failed component. A single beep means the system is functional.

The BIOS then checks the settings stored in a CMOS chip on the motherboard. The CMOS chip contains all the utility settings that determine how to user wants the computer to run.

The CMOS then coordinates the BIOS to run the bootloader program which is stored on the lowest number drive.

The bootstrapping sequence loads the operating system kernel before passing the control over to the operating system.

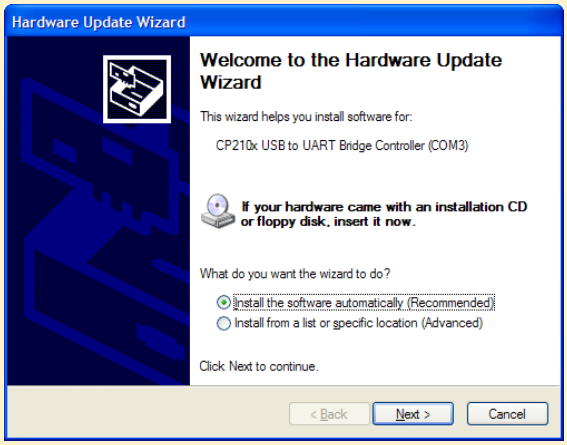
Device Drivers

A device driver is a computer program that provides a software interface to a particular piece of hardware.

This enables operating systems to access hardware functions without needing to know the details of the hardware being used.

When you attach a new printer, the computer must install the device driver program that comes with it before it will work. The driver helps the OS interpret communications between the device. The OS can sometimes do this automatically if it has a record of this particular model of printer and an accompanying driver.

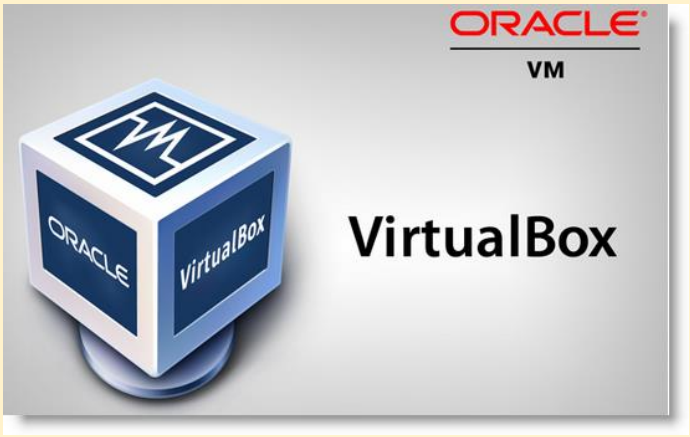
Drivers are dependent on hardware and the operating system. A driver typically communicates with the device via the system bus or communications subsystem to which the hardware connects.



A calling program will invoke a routine in the driver. This routine issues commands to the device that are hardware specific. The device will send data back to the driver and the driver will invoke routines in the original calling program.

Virtual machine = Any instance where software is used to **take on the function of the machine**, including executing **intermediate code** or running an operating system within another to emulate different hardware.

A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. The virtual machine is comprised of a set of specification and configuration files and is backed by the physical resources of a host.



Virtual machines are essentially **sandboxed environments** where the guest OS has **no knowledge** that it is being run inside a virtual machine.

The host computer can **assign certain hardware resourceless** to it such as RAM, CPU cores, storage drives and maybe access to printers **and USB devices**. The Guest OS has its **own memory manager** and scheduler. All hardware provided by the host is emulated as generic hardware by the virtual machine, meaning that the guest machine has no need for installing device drivers.

Intermediate code

System virtual machines represent entire OSs, whereas process virtual machines virtualise a **single application only.**

Process virtualisation – where intermediate code can be executed by a virtual machine.

Process virtual machines, as they are known, act as interpreters for **generic machine code instructions** known as **intermediate code**. They run intermediate code that has previously been compiled.

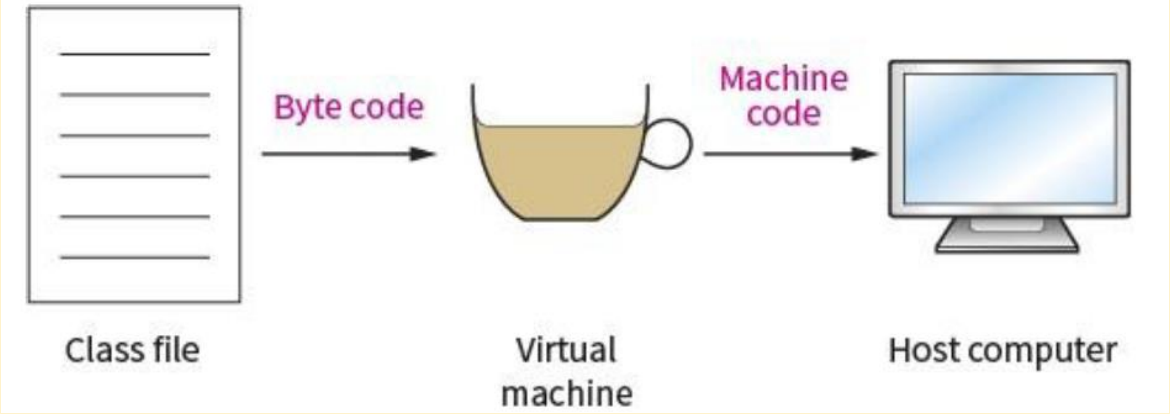
Process virtual machines (VMs) will run intermediate code that has previously been compiled.

1. **The source code is compiled** into the **intermediate code to prevent the incompatibilities** that exist when trying to run compiled code on different CPU architectures/OSs. This generic intermediate code cannot be directly run by a CPU and **is an abstraction** rather than the real instruction set.
2. To run the code, the **process virtual machine (VM)** must **interpret the code** into **machine code**. This can be executed upon by the CPU.

Java is a programming language originally developed by Sun Microsystems, now owned by Oracle. Once the **source code** has been produced, it is **compiled** into **intermediate code** known as **bytecode** (in Java's terms), which is a pseudo-executable file known as a **class file**.
Bytecode must then **be interpreted** by the Java VM (virtual process machine).
For this reason, all Java programs require the Java VM to be installed on the computer.

A host system cannot run the intermediate code (bytecode) and requires an interpreter (i.e. process VM) to convert it to machine code for the CPU.

If the bytecode tries to open a window, the process VM will have to send some system calls to the host OS to open a window on its behalf. At this `layer' access rights and security can be maintained by the process VM or OS.



Virtual machines:

Advantages	Disadvantages
Multiple OS environments can be used simultaneously on the same machine	Performance degrades quickly when multiple virtual machines are running on a host as they each deplete resources.
A virtual machine can offer instruction set architecture that of the host	A virtual machine is not as efficient at accessing the hardware as the host is.
Virtual machines can be used as a test environment. The Guest OS has no direct access to the host so there is little change of damage to the host.	Great use of secondary storage since the host system has to dedicate sections to store all the fields of the guest OS and its data.
Companies can use industrial VM software to run multiple servers on a single computer and improve security by isolating the separate servers to their own virtual machine.	Failure of the host machine could result in data loss of the virtual machine.
Changes can be made to separate virtual machine servers without impacting the rest of the network.	Industrial VM software can be very expensive.
Virtual machines allow snapshots to be taken, where their current state is recorded, so the administrator can roll back to previous snapshots if disaster strikes.	
Virtual machines can easily be copied and migrated. This can save lots of time when cloning machines.	
Since process VMs manage their own threads and have their own memory management system, the sandboxed environment offers extra security features: restrictions to accessing the hard drive or other system resources.