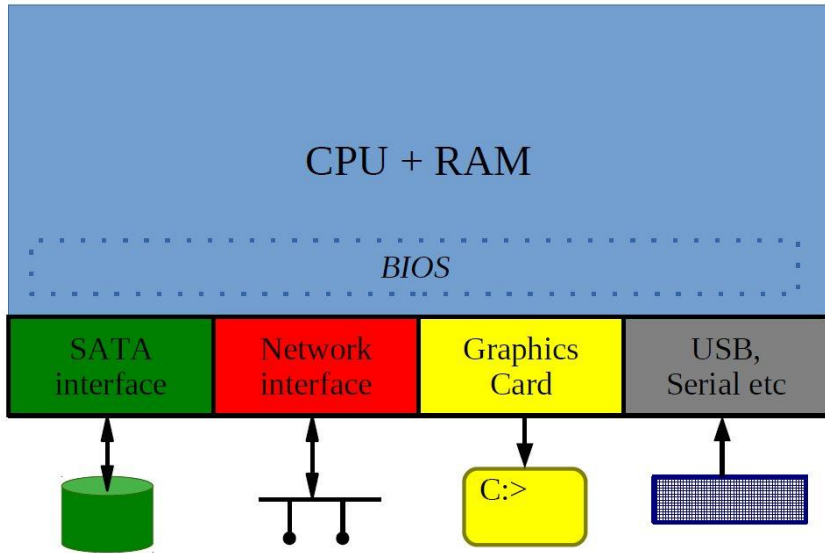




KVM and Libvirt

Day 02, Session 2.1

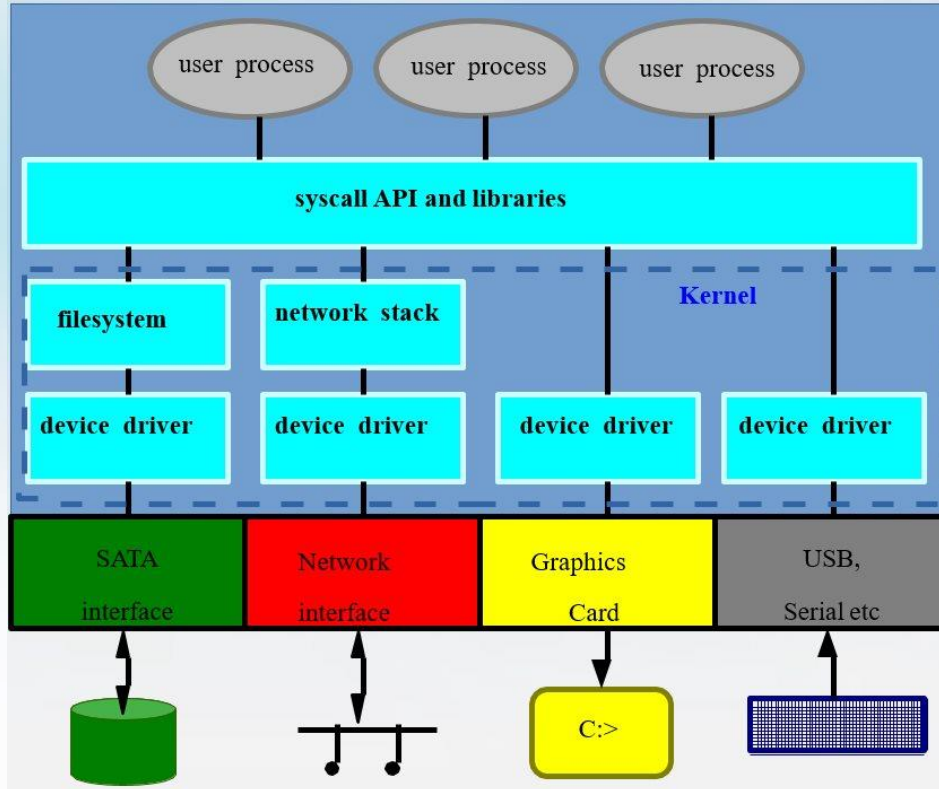
What's a PC



Boot up sequence

- A small program (the BIOS) runs when machine is switched on
- It uses the hardware to load an operating system
 - boot from hard drive, USB/CD-ROM, network...
- Modern operating systems then ignore the BIOS from that point onwards

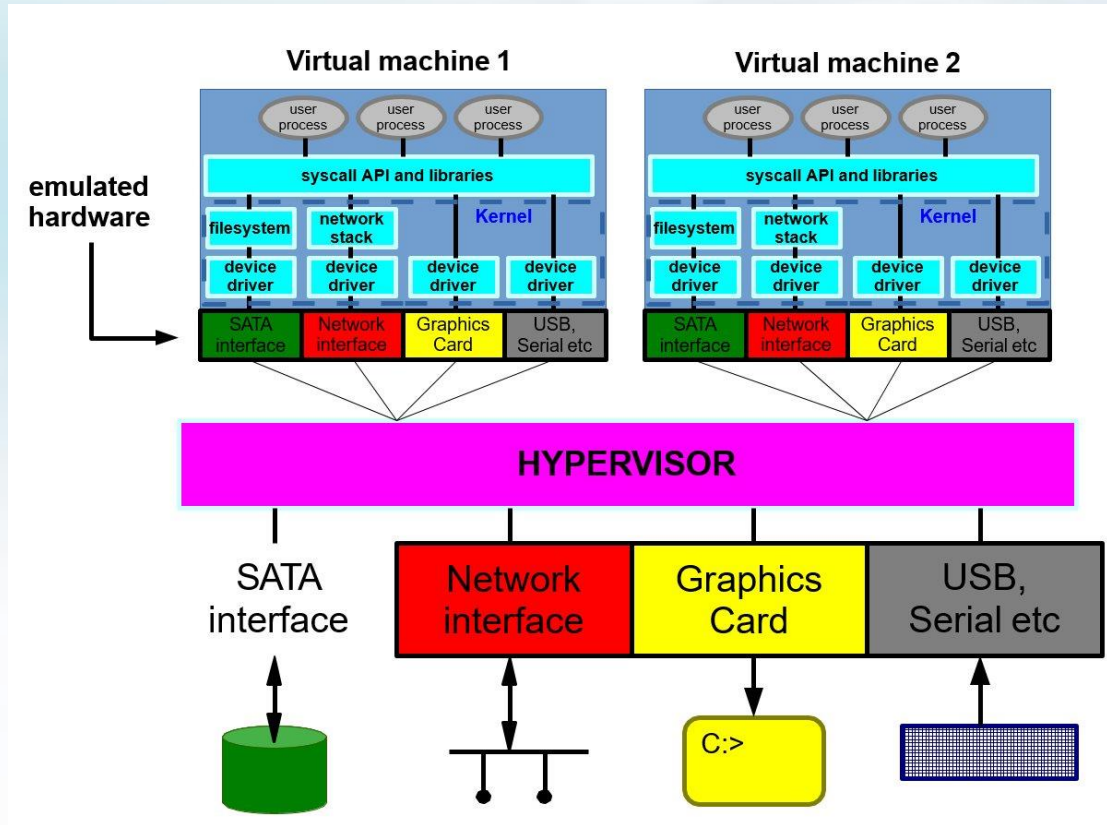
The next slide shows a machine after it has booted up



What we need

- To emulate a PC we must emulate all the components of the PC
 - hard disk interface, network card
 - graphics card, keyboard, mouse
 - clock, memory management unit etc
- We want multiple instances to co-exist and not be able to interfere with each other
 - access to memory must also be controlled
- The software to do this is called a ***hypervisor***

Virtualization



Type of Virtualization



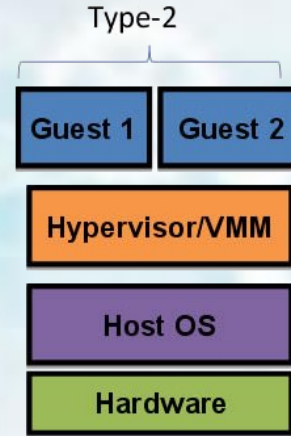
Examples:

Xen

VMware ESX

IBM pHype / LPARs

Microsoft Hyper-V



Examples:

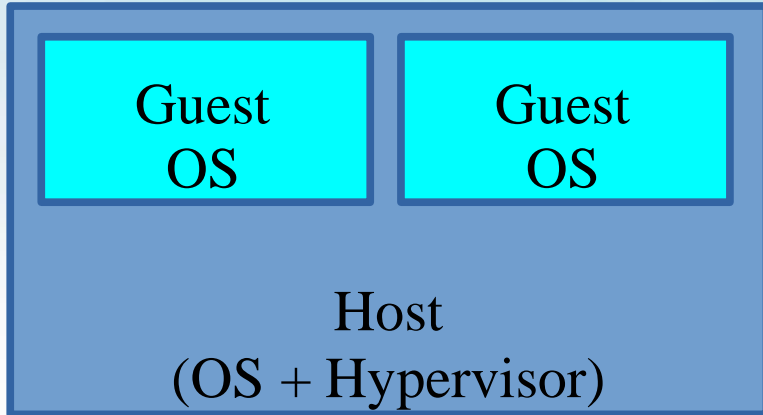
KVM (Linux)

VMware Workstation

VMware Server

Microsoft Virtual PC

Virtualization terminology



- The host is the machine running the emulation
- The guest is the emulated (virtual) machine
- One host could be running many guests

Kernel Virtual Machine (KVM)

- A hypervisor built into the Linux Kernel, based on QEMU
- It *requires* VT-x or AMD-V to run
- The host must be Linux

Checking

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

If **0** it means that your CPU doesn't support hardware virtualization.

If **1** or more it does - but you still need to make sure that virtualization is enabled in the BIOS.

Choosing a hypervisor

- **Step 1: Understand your needs**
 - Flexibility
 - Scalability
 - Availability
 - Reliability
 - Efficiency
 - Reliable support
- **Step 2: Understand the features**
- **Step 3: Investigate the ecosystem**
- **Step 4: Compare costs**
- **Step 5: Test for yourself**

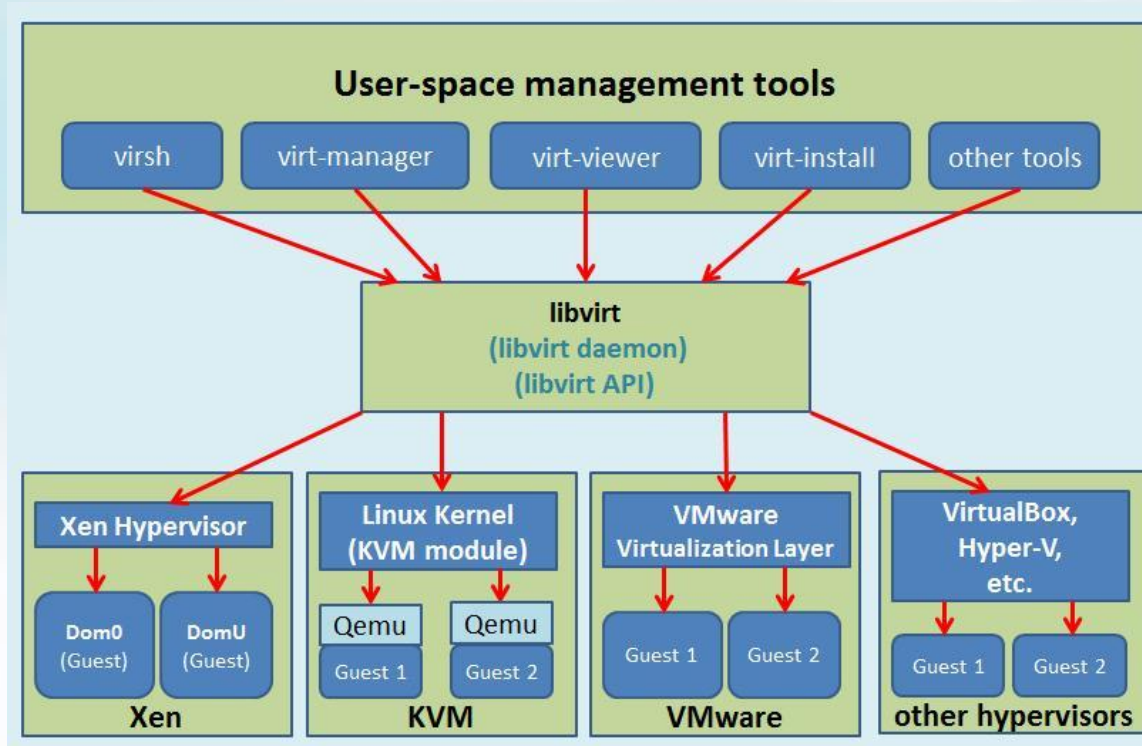
Libvirt

- ❑ A toolkit to interact with virtualization software.
- ❑ A long term stable API.
- ❑ Supports a wide range of hypervisors

Features:

- VM management
- Remote machine support
- Storage management
- Network interface management
- Virtual NAT and Route based networking

Libvirt Process



libvirt Limitations

- No simple web interface included
- virt-manager *can* talk to remote hypervisors, but virt-manager itself only runs under Linux
 - so you may end up running a VNC desktop into the Linux box, just to run virt-manager there
- XML format is unique to libvirt
 - different to OVF, VMX etc
 - too hard to write from scratch!
- libvirt's storage management is difficult

“virtinst” Package

install the packages

```
apt-get install bridge-utils qemu-kvm virtinst -y
```

- **Libvirt Main Clients**

- virsh (Command Line)
- virt-manager (Graphical X11)



Any Questions ?

Thank you