

Virtual Machines for Backward Compatibility

(and a lot more)

Explained and Demonstrated

DACS General Meeting – Oct. 1, 2013

Bruce Preston

Why Virtual Machines?

Here are some of the ways that you might utilize a virtual machine.

- **Support “Legacy” applications**
- **Support for obsolete hardware**
- **Training Environment**
- **Experimentation – try before you commit**
- **Security – Isolation**
- **Development – Client/Server**
- **Installation and Regression Testing**
- **Backup of a critical machine**

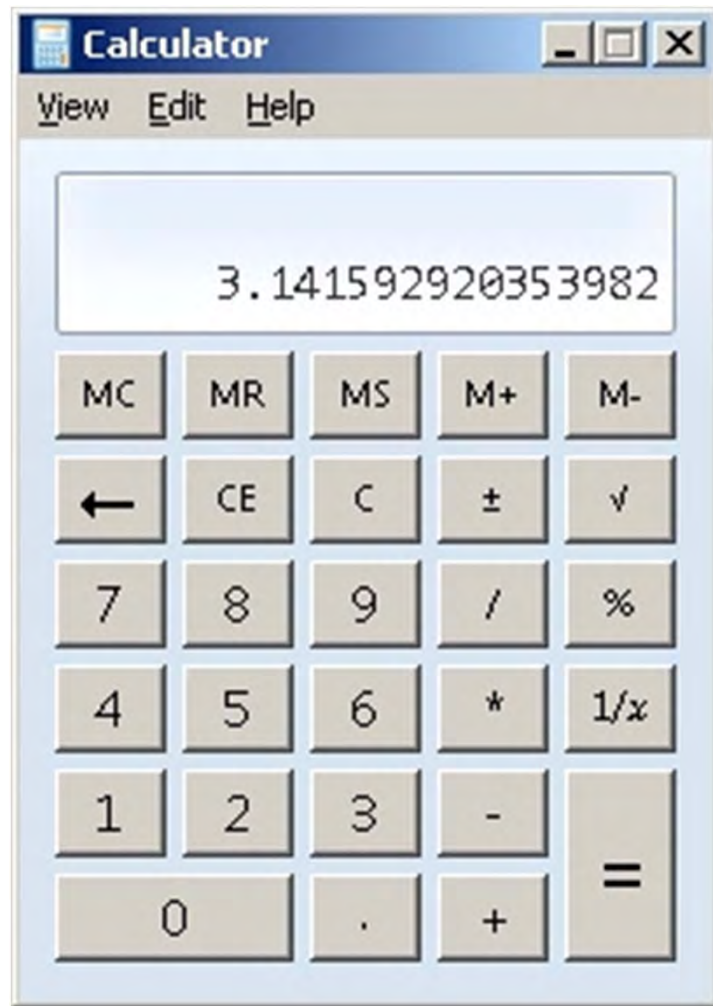
What is Virtual?

Virtual Device refers to any device that performs the same function as a physical device but does so using virtual resources.

A ***Virtual Resource*** has no physical components, it is constructed completely from software.

A ***Virtual Machine***, therefore, consists only of virtual devices, i.e. it exists only in executing software.

Is This Calculator Real?



It behaves as if it:

has a display,

has buttons,

can do calculations etc.

But it does not have a physical existence – no case, no battery, no buttons, no display, no circuit board - it is all software.

You may have been using a ***VIRTUAL CALCULATOR*** for years and never thought about it.

I can 'dual boot' – so why bother?

That's a good question.

Let's look a bit at how a computer works, starting with the boot process and going on to device management and ultimately, virtualization.

Booting 101

In a regular machine, the 'bootstrap loader' in the computer's BIOS looks at the bootable device(s) for an *active partition* which indicates that it has an operating system. It then loads the operating system from that device. The sequence of devices examined can be controlled, but traditionally was

- Floppy drive
- Optical drive (CD or DVD)
- Hard Disk (IDE, SCSI, SATA)
- Network (rarely seen, diskless workstations etc.)

Booting from USB devices was added around 2000.

Booting 102

When booting a machine, you load one and only one operating system.

Later on *boot managers* became available. These let you select which operating system is to be loaded. In Windows it was controlled by the BOOT.INI file, a file hidden in the root of the active partition.

For example, you could have had Windows XP as your primary system, but boot a beta release of the next generation OS. However while these two operating systems co-existed on the hard drive (in different partitions) they do NOT run simultaneously. It is one or the other.

Booting 103

Around the time that Apple converted to using Intel processors, they also released *Boot Camp* which was a boot manager. Again, it was one OS (Mac) or the other (typically Windows.)

Booting 104

The GNU Project (***GNU's Not Unix***) created GNU GRUB (*short for GNU **GR**and **U**nified **B**ootloader*) which is a very flexible package.

GNU GRUB is widely implemented and supports the selection of multiple operating systems. The limit how many active partitions the machine supports, but still only one may be loaded at any given time. It is often included as part of a Linux installation.

Early Virtualization

In the early mainframe days, when a program needed to make use of a comparatively slower resource such as a disk drive, tape drive, card reader, card punch, or a printer – processing stopped until the slow device delivered or accepted the data.

Perhaps the first virtual device implemented was a virtual printer. In this case the information was spooled to disk and then printed by another low-priority task. The low-priority task would be activated when the main processing stream was interrupted by the need to reference a resource.

Early Virtualization

Even with virtual devices the processor in a mainframe wasn't worked hard.

- IBM took several approaches to virtualization, starting with the System/360 family. The development of virtual storage such as MVS (c. 1974) in turn led to MVS/VM – virtual machines, each with their own virtual storage.**
- Concurrently and independently Dennis Ritchie and Ken Thompson of AT&T Bell Labs developed the C programming language and using it the UNIX operating system which made extensive use of virtual device technology.**

Software/Hardware Interface

In the early days of personal computers (e.g. CP/M etc.) it was often necessary to “patch” the OS and/or the application program to support devices such as printers.

WordPerfect Corporation had a dedicated staff on an 800 number just to assist with printer installation.

OFFICE



Epson FX-80
dot-matrix
parallel



NEC 3550 Spinwriter
thimble
serial

HOME



Anadex
dot-matrix
parallel



Brother
daisy wheel
parallel

Hardware Abstraction Layer (HAL)

Newer OSs such as Windows make use of a mechanism whereby application programs write to an *abstract device* and then an installed 'device driver' *attached to the operating system rather than the application program* implements device-specific capabilities.

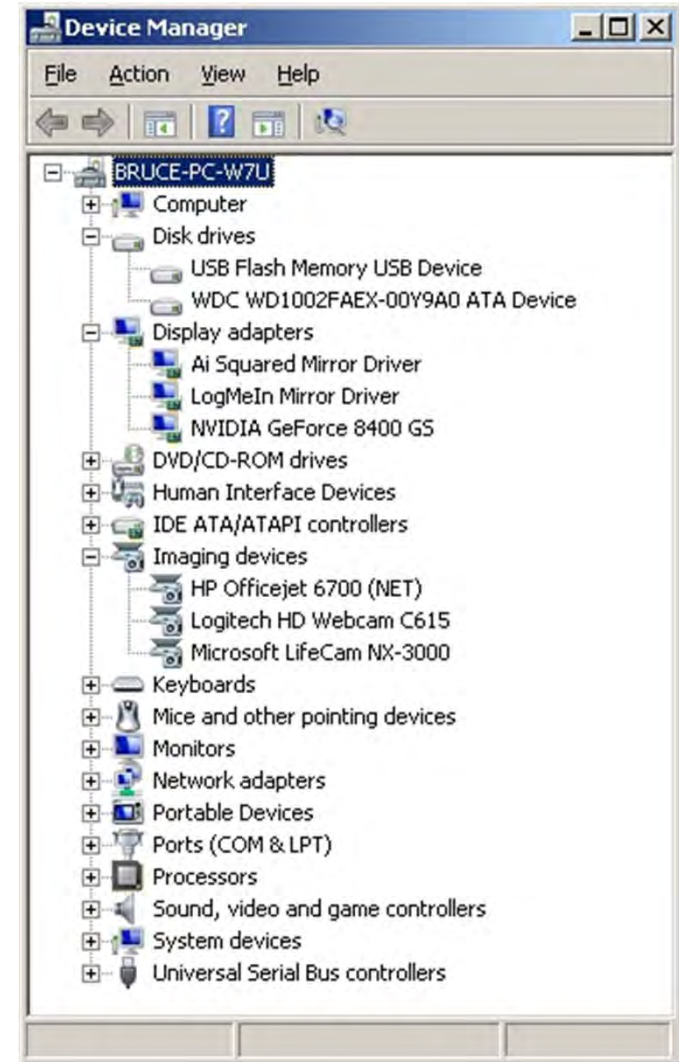


When you add a device you typically add a device driver which links to the abstract device in your operating system.

HAL Devices Link to Physical Devices and Their Drivers



Device Manager lists HAL devices, click the “+” to display the physical device(s) linked to the abstract device.



Conceptual Diagram of a Physical Computer

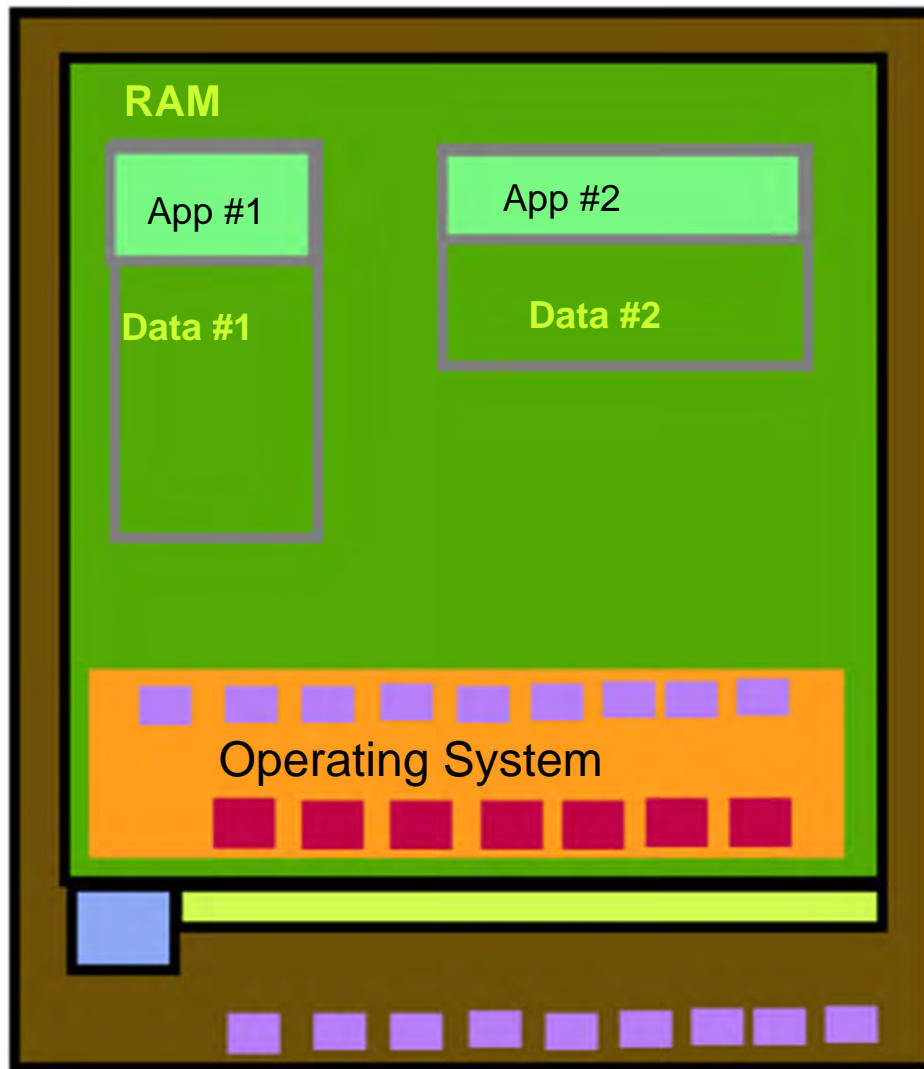


Illustration shows two applications - their program code and associated protected data are located in the system's RAM

Hardware Abstraction Layer

Device drivers from hardware vendors
CPU and BUS

Connections to physical devices

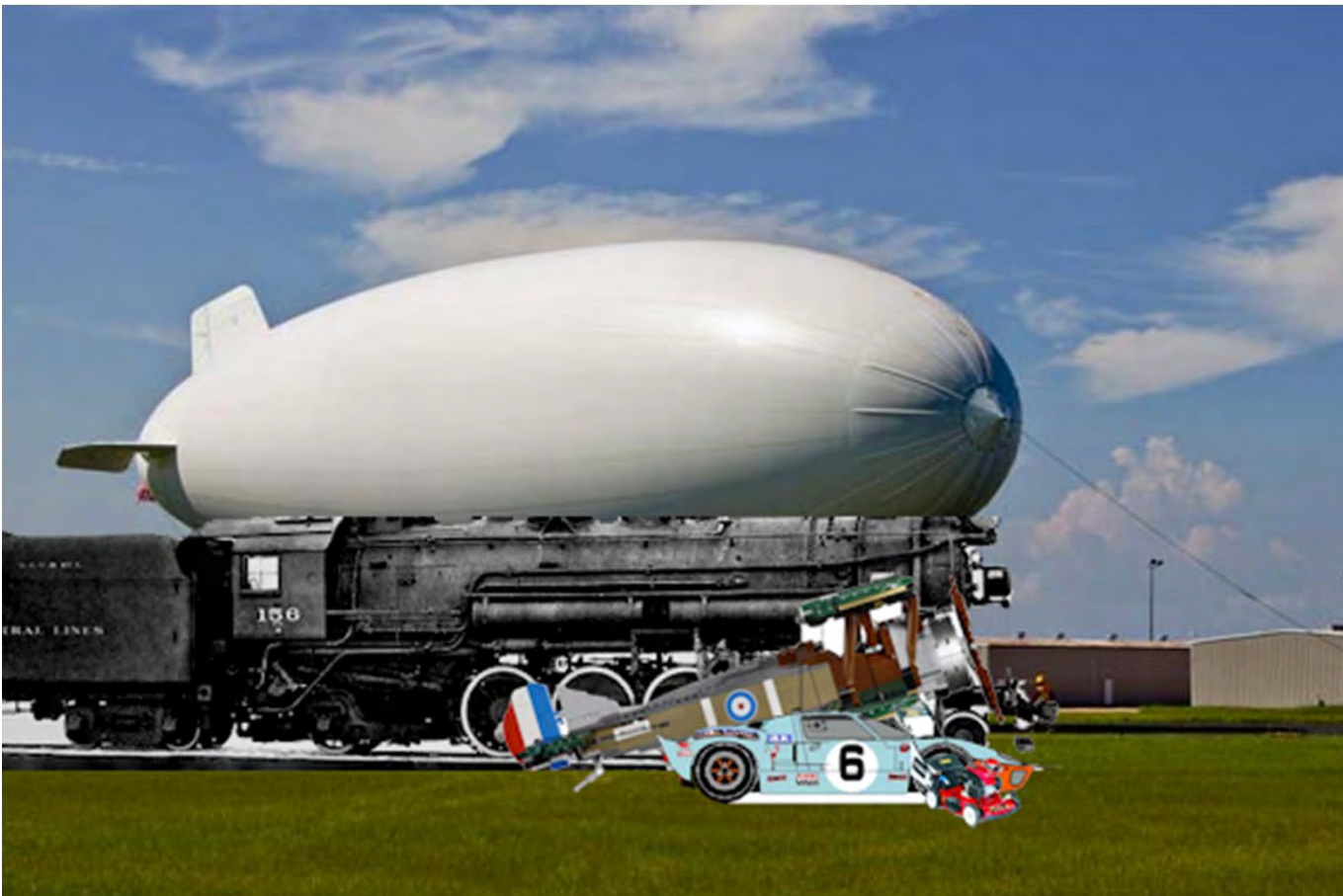
A Command to a Device “Start Your Engines!”



Command to an Abstract Device

We have widely differing physical devices, yet each can respond properly to a command such as “Start Your Engine!”

Each physical device’s “driver” knows what to do.



Blimp

Steam Locomotive

Sopwith Camel

Ford GT40

Lawn Mower

Introducing the Hypervisor

The Hypervisor is software that

- **Assembles and configures a virtual machine from virtual components.**
- **Links the VM's virtual devices to physical devices on the host.**
- **Manages the operation of the resulting virtual machine(s).**

The Hypervisor is said to “host” the “guest” virtual machines.

Hypervisor Types

In the micro computer arena there are two common ways to implement a host environment.

Type 1: “Bare Metal Hypervisor” – consists of a host operating system specifically written to act as a host to virtual machines. This operating system does nothing other than host virtual machine(s). This is most often used in the enterprise environment. *You do not run any applications within the host environment, only VM(s).*

Type 2: “Hosted Hypervisor” is an application that runs on a standard host operating system (typically Windows, Mac, Linux) and in turn hosts virtual machine(s). *You may run applications in the host environment, in parallel with VM(s).*

The hypervisor may host multiple VMs which may run simultaneously.

Common Sources of Hypervisors

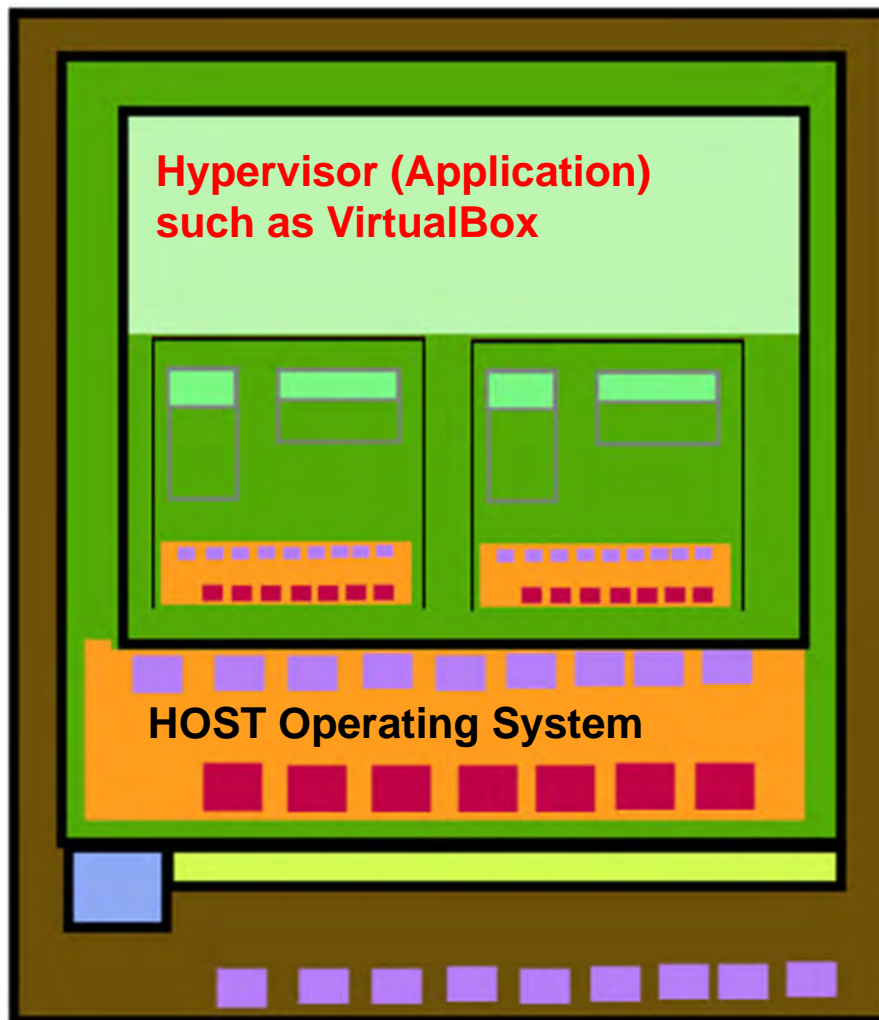
- **VMware** – provides enterprise level Type 1 and Type 2. It is available for multiple platforms.
- **Parallels Desktop** – Type 2 implementation for Mac that is often used to host Windows VMs.
- **Microsoft Hyper-V Server 2012** – Type 1, free, primarily used to host Microsoft servers and/or Virtual Desktop Infrastructure - thin client access to virtual PCs.
- **Microsoft VirtualPC** – Type 2, free, supports multiple VMs and is hosted by “Pro” versions of Windows since XP. It is often used to support “legacy” Windows applications.
- **VirtualBox** – Type 2, free – developed by Sun Microsystems which was acquired by Oracle. Oracle now provides support to the *open source* VirtualBox.org which maintains and publishes it. Host platforms are Windows, Mac, Linux and Sun Solaris.

Let's Focus on VirtualBox

Did you notice the presence of “open source” in the description of VirtualBox? It is free, and an open system with a large user community for support.

VirtualBox - Just create an account with Oracle (to join the peer-to-peer support forum and access to the downloads) then download the appropriate hypervisor for your host system, install it and you are ready to implement and use virtual machine(s). The rest of the presentation will thus focus upon VirtualBox.

Virtual Hosting



A hypervisor is an application program. VMs are entirely made up of software and data within the hypervisor's data space! Each has its own "guest" Operating System.

The device drivers in the guest OS talk to the hardware abstraction layer of the host OS through which they ultimately access physical or virtual devices. A virtual disk looks to be a complete file system to the VM, but is a single file to the host OS.

Note that there is no virtual CPU in these VMs, the host's CPU does all of the work.

Connections to physical devices

How to Create a VM - 1

Select and install the appropriate VirtualBox hypervisor application on the “host” computer. The host may be Windows (32 or 64 bit,) Mac (intel based only,) Linux, or Solaris.

My host machine has an Asus motherboard, an Intel Core-i5 3570 @ 3.4GHz processor, and 16GB of RAM. It is running a Linux distribution, specifically Ubuntu Desktop 12.04 LTS.

The VirtualBox release is Linux “AMD64”. This version works on Intel as well, the main thing is that it is 64-bit.

Some processor chips have instructions specifically designed to support hypervisors which VirtualBox will detect and use if present.

How to Create a VM - 2

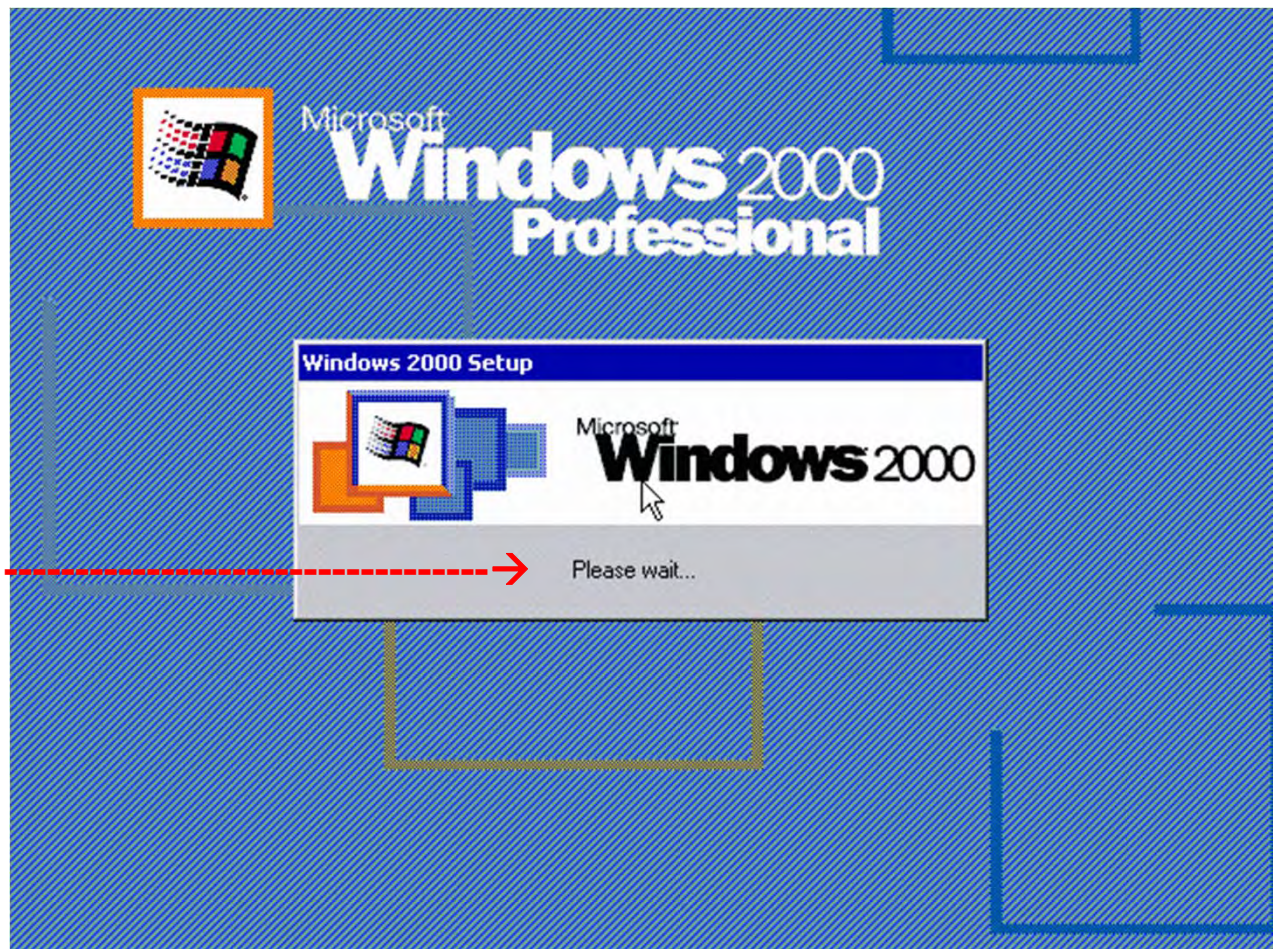
- **Launch VirtualBox**
- **Click the NEW button**
- **Fill in the wizard's forms (next slides)**

How to Create a VM - 3

- **Give your machine a name.**
- **Select type of OS from list (or “other”.)**
- **Specify how much RAM to allocate.**
- **Specify type of disk (use default unless you need otherwise.)**
- **Specify how much space the virtual disk should be given.**
- **Put bootable media (floppy, CD, DVD, flash) containing desired OS in the appropriate host drive.**
- **Click the “Start” button.**

How to Create a VM - 4

You will see the standard operating system installation process within VirtualBox's window on the host computer.



NOTE THE PLEASE WAIT

How to Create a VM - 5

- **Proceed normally. Most installs will do all of the work for you, i.e. FDISK, FORMAT, COPY, configuration dialogs etc.**
- **Older operating systems (e.g. DOS 6.22 and earlier) will require you to perform steps manually, such as:**
 - **Identify the drive to be used (your virtual drive).**
 - **FDISK and boot again to make it available.**
 - **FORMAT /S the disk.**
 - **Create folder structure if needed.**
 - **Copy OS files from source media to the drive.**
 - **Adjust configuration (CONFIG.SYS, AUTOEXEC.BAT etc.) as desired.**

How to Create a VM - 6

- **Remove the installation media when done.**
- **Restart the VM.**

You now have a bare machine.

- **For 32/64 bit Windows, add “guest additions” from VirtualBox.org for the specific OS to get better video driver, access to shared folder(s) on the host, copy/paste to/from host, etc. as needed.**
- **Add application programs etc.**

VirtualBox “Guest Additions”

VirtualBox provides a ‘Guest Additions’ download .ISO file which consists of components to increase the usability of *Windows* guest operating systems. It includes:

- **Better virtual display adapter – adjustable resolution, color depth, etc.**
- **Transparent capture of keyboard and mouse.**
- **simplified folder sharing between host and guest.**
- **Bidirectional shared clipboard between guest(s) and host.**
- **Guest application as a window in host desktop.**

Note: This is not applicable for 16-bit Windows such as Windows 3.11 (Windows for Workgroups)

How to Create a VM - 7

- **If the host recognizes a USB device (such as a flash drive, etc.) then you must add a *filter* to the guest operating system's configuration (in VirtualBox Manager) linking it to the USB device.**
- **Each filter is unique to a specific USB device. (If you have 3 different flash drives then you may need 3 filters!)**
- **The hypervisor will then make the USB resource on the host available to the guest machine.**

NETWORKING CONSIDERATIONS

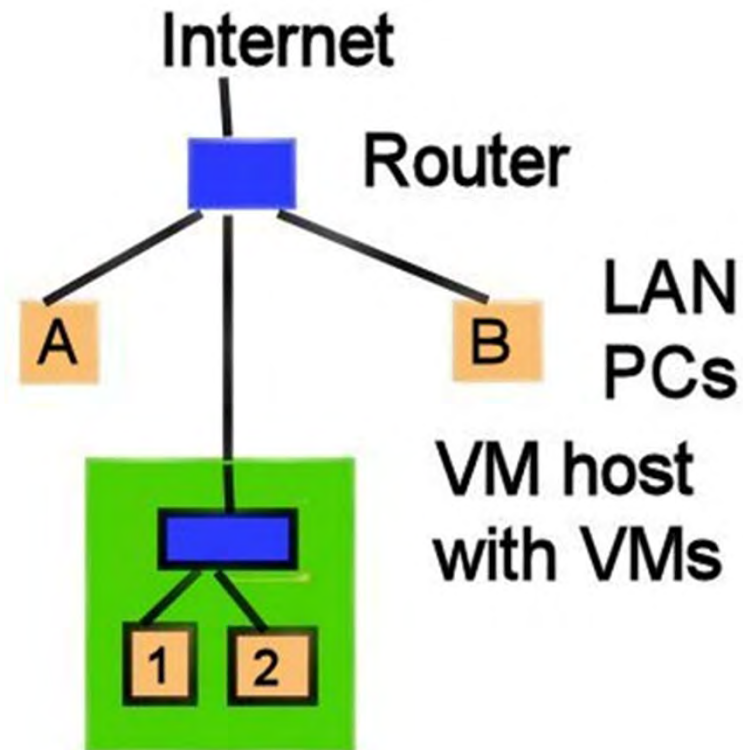
The hypervisor includes a virtual NAT router/firewall and a DHCP server. **By default** each VM's NIC will get its own IP address and can thus "see" the Internet as well as *other VMs within the LAN defined by the hypervisor.*

Alternatively you may elect *bridge mode* and assign an IP address to a VM's network adapter using either the standard methodology for static IP, or use an external DHCP server.

Thus you may establish peer-to-peer networking, sharing files and devices such as printers, external drives, USB flash memory, etc. with other machines (physical or VM) in your LAN.

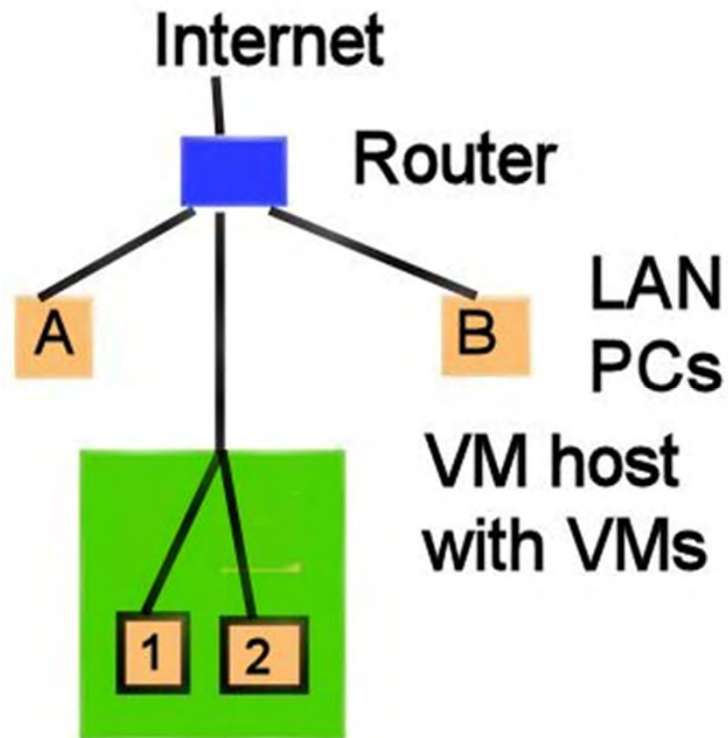
Note: All VM network traffic passes through the host machine's network adapter(s).

NAT and DHCP within VM Host



VMs 1 & 2 can see each other and the Internet, but they can not see or be seen by A or B and thus can not make use A or B's resources such as printers.

Static IP or from External DHCP Server



VMs 1 & 2 can see and be seen by each other as well as A & B and thus VMs may use A & B resources.

The network adapter's mode is a property of the VM, not the hypervisor.

Disk2VHD utility

- Clones a working Windows XP or newer machine to a VHD (Virtual Hard Disk)
- VirtualBox will work with a VHD.
- **HOWEVER**, I have had problems with booting – typically display adapter problems cause BSOD.
- Unless you tweak the registry on the source machine you will also have to reactivate the Windows installation as it detects that it is on a new machine.

Sources

<http://www.virtualbox.org>

Installation binaries, documentation, support forums, etc.

<http://vetusware.com>

A huge archive site of 'abandonware' – old operating systems (DOS, OS/2 etc.,) and applications such as Windows 3.11, Microsoft Works, dBase II, Norton Commander, etc.

www.vetusware.com home page

VETUSWARE.COM - the biggest free abandonware collection in the universe - Mozilla Firefox

File Edit View History Bookmarks Tools Help

WWW VETUSWARE.COM - the biggest free abando... +

WWW www.vetusware.com

Google

VETUSWARE.COM
the biggest free abandonware downloads collection in the universe

DOS forever. You: guest [login](#) [register](#)

[DOS-related books on Amazon.com! — Buy link here!](#)

OS		System		DBMS	
5 recently added:	5 most popular:	5 recently added:	5 most popular:	5 recently added:	5 most popular:
Windows 286 2.0	MS DOS 6.00(Sources included) (7037)	Turbo Pascal for Windows 1.5 FR	Norton Commander 5.5 (11097)	Foxbase+ 1.0.0 1.0.0	Borland dBASE IV 2.0 (3510)
Mandrake Linux 8.0	MS DOS Boot Disk 7.10 (6738)	VMDEBUG Source Code 1.32A-2B	Norton Commander 4.0 (5602)	Crystal Reports Pro 3.0	FoxBASE 2.00 (1903)
PC-GEOS v1.0 1.0 FR	OS/2 2.1 (5753)	Turbo Pascal 3.01A Source Code 3.01A	Norton Commander for Windows 2.01 (5274)	Clipper 5.2	dBASE Compiler 1.0 (1710)
OS/286 DOS Extender by Ergo Computing Inc. 2.1.A6	MD DOS Boot Disk 7.10 (5559)	VMDEBUG 1.32A-2B	Visual Basic 3.0 (5160)	Access 2.0 PT	CA-Visual Objects for Windows + upgrade 1.0 (1675)
386bsd 0.1	DOS32 Advanced 7.33 (4456)	IDA 3.7	DOS4GW 2.01 (4849)	dBASE III Plus 1.1	Foxpro for SCO Unix 2.6 (1542)

Office		Utility		Communication	
5 recently added:	5 most popular:	5 recently added:	5 most popular:	5 recently added:	5 most popular:
VI Editor for DOS 1.0r	WordStar 4.00 (5415)	Virus Protection System 4.58a	XTree Gold 4 (4070)	MajorBBS 5.31	LAN Manager 2.2c (2207)
Tasword PC	Microsoft Word for Windows 2.0c (4196)	McAfee VirusScan 2.5 for DOS 2.5	Disk Dupe Pro 7.0 (3984)	WinFax Pro 3.0 3.0	LapLink Pro 4.00 (2027)
Microsoft Word 1.00 PC	Works 3.0 (3823)	Advanced CAB Repair 1.0 (2070)	UUENCODE UUENCODE 3.4	Microsoft Internet Explorer 1.0 1.0	Lapl ink 6 (1472)
World Demo 1.00	Microsoft Word for Windows 1 (3586)	CheckIt 3.0 (1759)	UltraCompressor 2.37	Terminal Server Client 4.00	NeoTrace Pro 3.25 (1425)
Microsoft Word 1.15 for DOS 1.15	Assistant Series 2.00 (3294)	pkUnzip 2.5 (1327)	nneron 117	AOL 1.0 for Windows 1.0	LANtastic 6.0 (1272)

Games	Multimedia	Other
5 recently added:	5 recently added:	5 recently added:
5 most popular:	5 most popular:	5 most popular:

How to Access a VM

- **Launch VirtualBox on the host OS.**
- **Start the VM(s) that you want.**
- **“Login” to the VM(s) if required by the guest operating system(s).**
- **You may move back and forth between VMs. Use the *hostkey* which by default is the RIGHT Control key. *Hostkey + DEL* is the way to pass Ctrl-Alt-Del to a guest OS. *If the host is Windows, Ctrl-Alt-Del would be intercepted by the host!***
- **For a Windows guest OS, if you have installed “guest additions” you may resize the VM’s screen, make it full screen, or a window in the host’s desktop, etc.**

Running a “Legacy” application

Microsoft Works was a popular entry-level ‘suite’ consisting of a simple word processor, spreadsheet and database. It was bundled with many early DOS and Windows machines. It has been abandoned by Microsoft, and the file formats are not understood by current applications.

So a VM running an appropriate DOS or Windows and Works would enable you to retrieve data held hostage by the obsolete and discontinued file formats.

Running a “Legacy” Application

Some Windows XP (and earlier) applications interfaced directly with the hardware, which while good for processing speed, may have led to compatibility or security issues with later operating systems. With the introduction of Vista, Microsoft blocked such access. As a result some applications would not install or run.

Examples:

- Microsoft Visual Studio wouldn't run on the early Vista beta releases.
- Adobe PageMaker needed too many changes, so Adobe froze it at the XP SP3 level and published InDesign as a replacement.

Will It Run?

Most applications will run properly if run in a VM with a contemporary operating system. But there may be other factors involved that could cause a problem.

A classic example is the original Microsoft Flight Simulator. It was written for the original IBM PC, and made extensive use of the CPU clock (4.77MHz) for anything time-sensitive, such as speed and control responsiveness.

If your host's CPU is running at 3.4GHz, that is roughly 700 times faster. Even making allowances for the minimal virtualization overhead, it is still too fast for human interaction.



It would be like attaching the first stage a Saturn V rocket to your Cessna 171. If you start at Chicago's Meigs Field the runway points directly at the Sears Tower and you will crash into it the instant you release the aircraft's brakes.

Will It Run?

Hardware requiring specialized cards such as a software-specific video capture card, SCSI ZIP drive via parallel printer port, etc. will be a problem as you can not install physical component boards in a virtual machine.

Even if you can physically install it in the host machine there may not be a suitable device driver for it at the host OS's level.

Check with the VirtualBox.org forum to see if anyone has already figured out how to do it such as using a custom developed driver.

“Sheep Dip” Machine

Create a ‘clean’ machine, install an anti-virus package, and some basic applications such as LibreOffice. Then immediately make a copy of the virtual disk.

Use this VM for:

- **Guests:** Such as grand children who do who-knows-what.
- **Downloading:** Run through your anti-virus and only if clean move to production machine.
- **Browsing:** You might innocently visit a compromised sites.
- **Public e-mail address:** Such as a business that must receive many ‘unsolicited’ messages such as a placement company that receives resumes etc.

If the machine becomes compromised, just over-write the virtual disk with the backup copy and update the A/V database.

Installation Demonstration

Windows 95 and newer installations typically take about 40 minutes. However DOS 6 only takes about 5 minutes, including the VirtualBox wizard. So we will create a DOS VM.

If time permits: Windows 3.11 (Windows for Workgroups) is a 16-bit *application* that runs on top of DOS, and it takes about 15 minutes to install.