

Virtual Machines

COMP755

What is a VM?

- An efficient, isolated duplicate of a real machine
- Current use includes virtual machines which have no direct correspondence to any real hardware

Categories of Virtual Machines

- Process virtual machine
 - Runs a program
- System virtual machine
 - Provides a complete system platform which supports the execution of a complete operating system

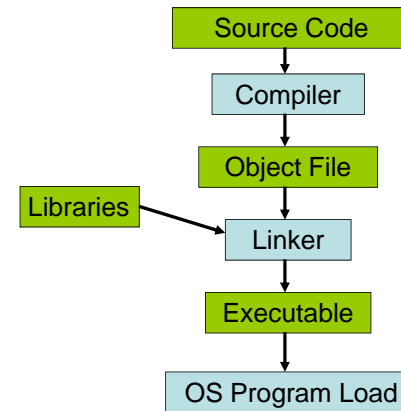
Process Virtual Machine

- Provides a platform-independent programming environment that abstracts away details of the underlying hardware or operating system
- Allows a program to execute in the same way on any platform.
- Runs as a normal application inside an OS
- Generally starts and terminates with an application program

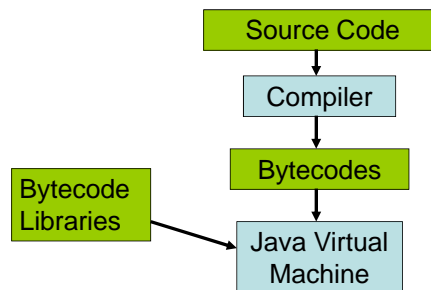
Java Virtual Machine

- The Java Virtual Machine (JVM) provides an architecture independent environment for executing Java programs.
- There are JVM available for many different platforms.
- Java was originally created by Sun Microsystems. There are now many sources for a JVM.

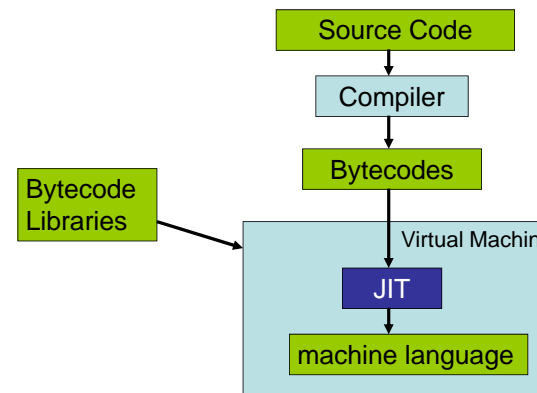
Traditional Program Creation



Traditional Java Programs



Modern Virtual Machines



Virtual Machine Advantages

- Architecture independent bytecodes can be executed in a Virtual Machine on any architecture
- Bytecodes can be compiled into machine language for any machine
- Compilers can generate code that can be sent over the network to be executed anywhere
- Virtual Machines provide additional services

Java Virtual Machine Services

- Program safety verification
- Garbage collection
- Array bounds checking
- Enforce security restrictions

Java Security

- Java applications can do just about anything
- Java applets (which usually run in a browser)
 - cannot access client files
 - cannot execute many of the System functions, such as exit
 - cannot connect to sites other than the one from which it was downloaded

Java Policies

- The basic responsibility of a Policy object is to determine whether access to a protected resource is permitted
- Java encapsulates a security policy in the `java.security.Policy` class
- Policies are kept in a `.policy` file, either the default or a user created `.policy` file
- Policy files can be created using the GUI **policytool** utility

JVM Disadvantages

- There are differences in how the underlying systems work
- Performance might not be as good as native code

Dalvik - Android Virtual Machine

- Creates environment to execute phone applications
- Interpretively executes Java programs
- After the Java program is compiled, the Java jar file is converted to a new format called Dex (Dalvik Executable)
- Android applications are written in Java, but they call libraries written in C



- Common Language Runtime
- Programming language independent
 - C, C++, C#, J#, Visual Basic
- Cross-language integration, especially cross-language inheritance
- Garbage collection
- Microsoft Visual Studio will generate CLR code that is executed in the .NET environment.

Microsoft Intermediate Language

- Microsoft intermediate language (MSIL) is generated by .Net compilers
- CPU-independent set of instructions that can be efficiently converted to native code
- Converted to CPU-specific code, by a just-in-time (JIT) compiler or by the Native Image Generator before execution
- Code verification determines if the code is type safe

Java Lawsuits

- Sun/Oracle have sued to keep Java to their specifications
- In 1997 Microsoft released a JVM with IE that had several changes; some improvements and some to avoid competition with MS software
- In 2010 Oracle sued Google claiming the Dalvik virtual machine violated Oracle Java patents



System Virtual Machines

- A system virtual machine provides a simulated environment that supports guest OS
- The VM simulates the bare hardware
- The guest OS might not know it is running in a simulated environment

Advantages of System VMs

- Multiple OS environments can co-exist on the same computer, in strong isolation from each other
- The virtual machine can provide an instruction set architecture that is somewhat different from that of the real machine

Fewer Computers

- Many servers often have low utilization
- One computer may have sufficient resources to run multiple servers
- It is often possible to run multiple server applications on one computer
- Running server applications in different virtual machines provides isolation should one of them fail

Being Green

Fewer computers saves

- Money
- Space
- Maintenance
- Electricity
- Cooling

Disadvantages of System VMs

- A virtual machine is less efficient than a real machine because it accesses the hardware indirectly
- Page faults can be challenging to support

Types of System VM

- The virtual machine system is often called the hypervisor
- The hypervisor can be booted onto the native hardware or it can run as an application under a host OS
- The hypervisor can provide the guest OS access to real devices or simulated devices.

VM Isolation

- VMs are commonly used to simultaneously run two or more server systems on the same computer
- While it may be possible to run the two server applications under the same OS on the same computer, the failure of one server may impact the other.
- With VM the failure of one guest system will not impact the others.

How VM does it

- The guest OS runs at user level
- User level instructions of the guest OS are executed on the hardware as usual
- When the guest OS executes a privileged instruction, the hardware interrupts. The hypervisor then simulates the action of the privileged instruction.

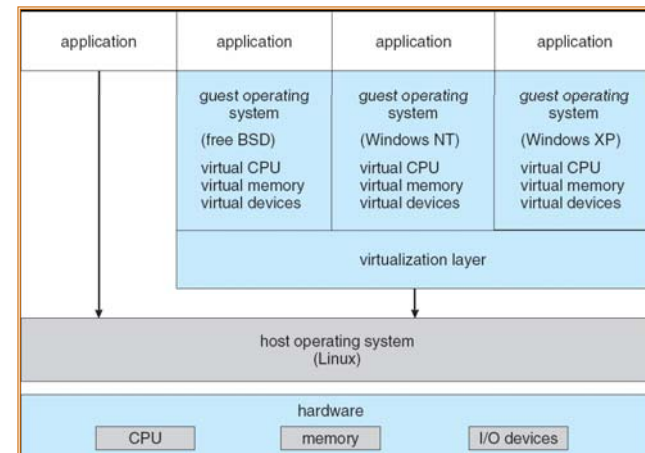
VM challenges

- Virtual memory can be a challenge for virtual machines.
- When a guest OS applications gets a page fault, should the guest OS service the interrupt or should the hypervisor.
- The hypervisor controls how much real memory the guest OS has.
- Some VM systems configure the guest OS as having a very large amount of real memory.

Emulators

- An emulator program can simulate the hardware execution.
- The simulated system does not need to be identical to the native hardware.
- Particularly useful for developing software for a hardware platform that is not yet available.

VMware Architecture



IBM VM/370

- IBM/370 was a system virtual machine created in 1972 for IBM mainframes
- Current version is z/VM, and is still widely used
- Provided virtual machines to simultaneously run a range of different OS

Control Program

- control program or hypervisor called CP
- Runs on the physical hardware, and creates the virtual machine environment
- Performs the system's resource-sharing, including device management, dispatching, virtual storage management, and other traditional operating system tasks

Virtual Machines under VM/370

- Each VM user is provided with a separate virtual machine having its own address space, virtual devices
- The virtual machine has virtual devices including "minidisks" which are files on the CP that look like small disk drives to the VM

Guest OS Possibilities

- Conversational Monitor System (CMS)
- Mainstream IBM OS
- specialized VM subsystem
 - providing services to CMS users such as spooling, inter-process communications, and specialized device support
- Non-IBM OS such as Linux
- Another copy of VM.

Conversational Monitor System (CMS)

- Lightweight, single-user operating system. Its interactive environment is comparable to that of a single-user PC
- Runs as a single user OS under VM/370
- Historically not a “strategic” IBM OS

CMS User Interface

- Users boot CMS in their virtual machine
- Each account has an OS that will boot automatically when the user logs on
- Provides a command line interface with full screen applications
- Typically used an IBM 3270 terminal

IBM 3270 terminal

- IBM’s block concept of communication
- Local buffer storage
- Some minor processing capabilities
- Generally dealt with an entire screen of data at a time
- Handled editing tasks locally, and then transmitted a set of fields (or the entire page) at once when the ENTER key was pressed.

VM / CP Interface

- Users could configure their virtual machine
- Communicated with CP via the non-virtualized DIAG (diagnose) instruction
- The virtual machines had virtual card readers, card punches and printers
- To send email to another user, your virtual card punch was connected to another users virtual card reader

REXX

- REXX was an IBM scripting language
- It runs under CMS and in the primary CMS editor Xedit
- Many of the CMS standard applications were just REXX macros
 - Users could edit the macros