Virtual Machines

• Allow you to run a Guest Operating System on top of a Host Operating System
  – VMware (for most systems)
  – Microsoft Virtual PC, Virtual Server
  – Parallels (for Mac OS X)
Virtual Machine Usage

• Run applications written for another OS
  – Convenience for a Desktop / Laptop system
  – Efficient hw usage for servers
• Windows 7 includes a virtual machine to provide Windows XP compatibility
Virtual Machine Usage

• Simplifies Regression (Repeated) Testing of applications
  – Simply restart a saved virtual configuration to repeat a test
• Have one machine that can be used to test applications for different OSs
Virtual Machine Usage

• By having Virtual Machine configuration on a server ...
  – Can tremendously simplify deployment to desktops when all desktops are same
• Example. Call Center
Virtual Machine Requirements

• Fidelity
  – Programs run same on virtual machine as a real computer

• Performance
  – Programs should run at close to the same speed on the virtual machine

• Safety
  – Programs running on virtual machine must be unable to interfere with Host OS or other apps
Virtual Machine Requirements

• Guest Application and Guest OS must execute same whether on virtual machine or real hardware
  – Regular instructions
  – Privileged instructions
  – Sensitive instructions
    • Instructions that act differently depending on whether computer is in OS or User Mode
Virtual Machine Requirements

• Execution of Guest Machine Instructions
  – Can be done if Host & Guest have different machine instructions, but is slow
  – Best if Host & Guest are for same CPU

• Proper handling of Guest OS execution of privileged & sensitive instructions
  – Typically requires interpretation of intent of the instruction
  – Can be difficult & cumbersome
System Call Handling

Diagram showing the flow of system calls from a guest application to the host OS kernel through a virtual machine monitor and guest OS.
System Call Handling with Hardware Support

- Guest App
- Guest OS
- Virtual Machine Monitor
- Host OS Kernel
Input / Output
Input / Output with Hardware Support
## Device Mapping

<table>
<thead>
<tr>
<th>Guest OS</th>
<th>Host OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>Application Window</td>
</tr>
<tr>
<td>Main Disk</td>
<td>File</td>
</tr>
<tr>
<td>Network Card</td>
<td>Virtual Network Port</td>
</tr>
</tbody>
</table>
Efficient Device Mapping

• A custom Guest OS device driver is installed for each mapped device
  – Avoids complex interpretation of privileged instructions
  – Increases efficiency
Non-Mapped Devices

• CD, USB devices
• Virtual Machine must obtain exclusive control over these devices to avoid conflicts with applications running on other operating systems (including the Host OS)
Memory Mapping

• Requires Host OS to interpret Guest OS’s attempts to set values into page tables.
• Hardware support for virtual (extended) page tables minimizes this
Memory Mapping
Integrating Guest & Host

• Makes entire system look more like one computer rather than 2 separate entities
## Integrating Guest & Host

<table>
<thead>
<tr>
<th>Host OS</th>
<th>Guest OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>File System</td>
<td>Network Drive</td>
</tr>
<tr>
<td>Clipboard</td>
<td>Clipboard</td>
</tr>
<tr>
<td>Application Window</td>
<td>Application Window</td>
</tr>
</tbody>
</table>
Integrate Host File System

- Can be mapped as a Network Drive on the Guest
- Configurable Feature
  - Allows Guest OS/Apps to share files with other OSs (including Host)
  - Not configuring this isolates Guest system for security purposes
Integrated Clipboard

- Allows cut & paste between applications running on different operating systems
Integrated Windows

• Instead of mapping Guest Application window onto a Host window for the Guest OS...
• Guest Application window goes to Host Application window
  – Makes Guest Application appear as though it is running on the Host