IF YOU WANT TO USE A LAPTOP, SIT ON THE LEFT SIDE OF THE CLASSROOM
Virtual Machines
Monolithic kernels vs. Microkernels
operating systems enforce modularity on a single machine using virtualization in order to enforce modularity + build an effective operating system

1. programs shouldn’t be able to refer to (and corrupt) each others’ memory → virtual memory

2. programs should be able to communicate → bounded buffers (virtualize communication links)

3. programs should be able to share a CPU without one program halting the progress of the others → threads (virtualize processors)
operating systems enforce modularity on a single machine using virtualization

in order to enforce modularity + build an effective operating system

1. programs shouldn’t be able to refer to (and corrupt) each others’ memory

2. programs should be able to communicate

3. programs should be able to share a CPU without one program halting the progress of the others

**today:** running multiple OSes at once (and dealing with kernel bugs)
Virtual Machines

- virtual machine running guest OS
- virtual machine running guest OS
- physical hardware
Virtual Machines

problem: how to (safely) share physical hardware?
Virtual Machines

VMM runs in kernel-mode on hardware

- virtual machine running guest OS
- virtual machine running guest OS
- virtual machine monitor (VMM)
- physical hardware
virtual machine monitor (VMM)

guest OS

virtual hardware

U/K
PTR
page table
...

virtual hardware

U/K
PTR
page table
...

physical hardware

U/K, PTR, page table, ...
virtual machine monitor (VMM)

VMM’s goal: virtualize hardware
Virtual machine monitor (VMM)

Guest OS

Virtual hardware

U/K, PTR, page table, ...

Virtual hardware

U/K, PTR, page table, ...

Virtual machine monitor (VMM)

Physical hardware

U/K, PTR, page table, ...

Guest virtual address

Guest physical address

Host physical address
VMM

guest OS

virtual hardware

guest OS page table

PTR

VMM

physical hardware

guest OS page table

guest OS virtual -> guest OS physical
vmm

guest OS

virtual hardware

guest OS page table

guest OS page table

load PTR

guest OS page table

guest OS virtual -> guest OS physical

VMM

physical hardware
guest OS

virtual hardware

 PTR

guest OS page table

VMM

load PTR

intercept!

guest OS page table

guest virtual -> guest physical

physical hardware
VMM

guest OS

virtual hardware

guest OS page table  PTR

load PTR

intercept!

VMM

guest OS page table

physical hardware

guest OS page table

guest virtual -> guest physical

guest OS page table
guest OS

virtual hardware

guest OS page table

PTR

load PTR

intercept!

VMM

VMM page table

guest OS page table + VMM page table

physical hardware

VMM page table

guest OS page table

guest virtual ->
guest physical

guest physical ->
host physical
VMM

**guest OS**

**virtual hardware**

- **guest OS page table**
- **PTR**

VMM

**guest OS page table**

**VMM page table**

**host page table**

**physical hardware**

**host page table**

**load PTR**

**intercept!**

**guest virtual -> guest physical**

**VMM page table**

**guest physical -> host physical**
guest OS

virtual hardware

guest OS page table

PTR

VMM

guest OS page table + VMM page table → host page table

host page table

physical hardware

PTR → host page table

load PTR

intercept!

guest virtual → guest physical

VMM page table

guest physical → host physical

host page table

guest virtual → host physical
In modern hardware, the physical hardware is aware of both page tables, and performs the translation from guest virtual to host physical itself.
virtual machine monitor (VMM)

VMM’s goal: virtualize hardware
VMM’s goal: virtualize hardware
VMM’s goal: virtualize hardware
<table>
<thead>
<tr>
<th>ID</th>
<th>Product</th>
<th>Comp</th>
<th>Assignee</th>
<th>Status</th>
<th>Resolution</th>
<th>Summary</th>
<th>Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>206597</td>
<td>Memory M</td>
<td>Page All</td>
<td>akpm</td>
<td>NEW</td>
<td>---</td>
<td>We recommend using synthetic urine before that of a friend. For starters, if your friend has smoked pot a few weeks before and has forgotten.</td>
<td>11:43:35</td>
</tr>
<tr>
<td>206591</td>
<td>Alternat</td>
<td>rt</td>
<td>alt-trees_rt</td>
<td>NEW</td>
<td>---</td>
<td>4.19.98-rt40 BUG: scheduling while atomic(vmwgfx)</td>
<td>01:06:32</td>
</tr>
<tr>
<td>206593</td>
<td>Drivers</td>
<td>Video(0t</td>
<td>drivers_video-other</td>
<td>NEW</td>
<td>---</td>
<td>AMDGPU Strange aspect ratio (monitor quirk)</td>
<td>05:26:33</td>
</tr>
<tr>
<td>206589</td>
<td>Drivers</td>
<td>Sound(AL</td>
<td>perex</td>
<td>NEW</td>
<td>---</td>
<td>No way of getting sound with the ALC294 from the Asus Zenbook (Flip) Series</td>
<td>20:21:57</td>
</tr>
<tr>
<td>206587</td>
<td>Platform</td>
<td>x86-64</td>
<td>platform_x86_64</td>
<td>NEW</td>
<td>---</td>
<td>x86/mce: Do not log spurious corrected mce errors</td>
<td>Tue 16:28</td>
</tr>
<tr>
<td>206595</td>
<td>Tools</td>
<td>Trace-cm</td>
<td>tools_tracecmd_kernelshark</td>
<td>NEW</td>
<td>---</td>
<td>libtraceevent: Fail to parse some ftrace events from kvmmu system</td>
<td>08:34:36</td>
</tr>
<tr>
<td>206599</td>
<td>Tools</td>
<td>Trace-cm</td>
<td>tools_tracecmd_kernelshark</td>
<td>NEW</td>
<td>---</td>
<td>libtraceevent: Fail to parse some ftrace events due to not defined helper functions</td>
<td>12:39:26</td>
</tr>
</tbody>
</table>

7 bugs found.
monolithic kernels: no enforced modularity within the kernel itself

Basic interprocess communication, virtual memory, scheduling, file server, device drivers, network, …
**microkernels**: enforce modularity by putting subsystems in user programs

![Diagram of microkernels]

- Application
  - application IPC
  - device driver
  - network
- Basic interprocess communication, virtual memory, scheduling
- Hardware

6.033 | spring 2020 | lacurts@mit.edu
Virtual Machines allow us to run multiple isolated OSes on a single physical machine, similar to how we used an OS to run multiple programs on a single CPU. VMs must handle the challenges of virtualizing the hardware (examples: virtualizing memory, the U/K bit).

Monolithic kernels provide no enforced modularity within the kernel. Microkernels do, but redesigning monolithic kernels as microkernels is challenging.