Chapter 33: Virtual Machines

- Virtual Machine Structure
- Virtual Machine Monitor

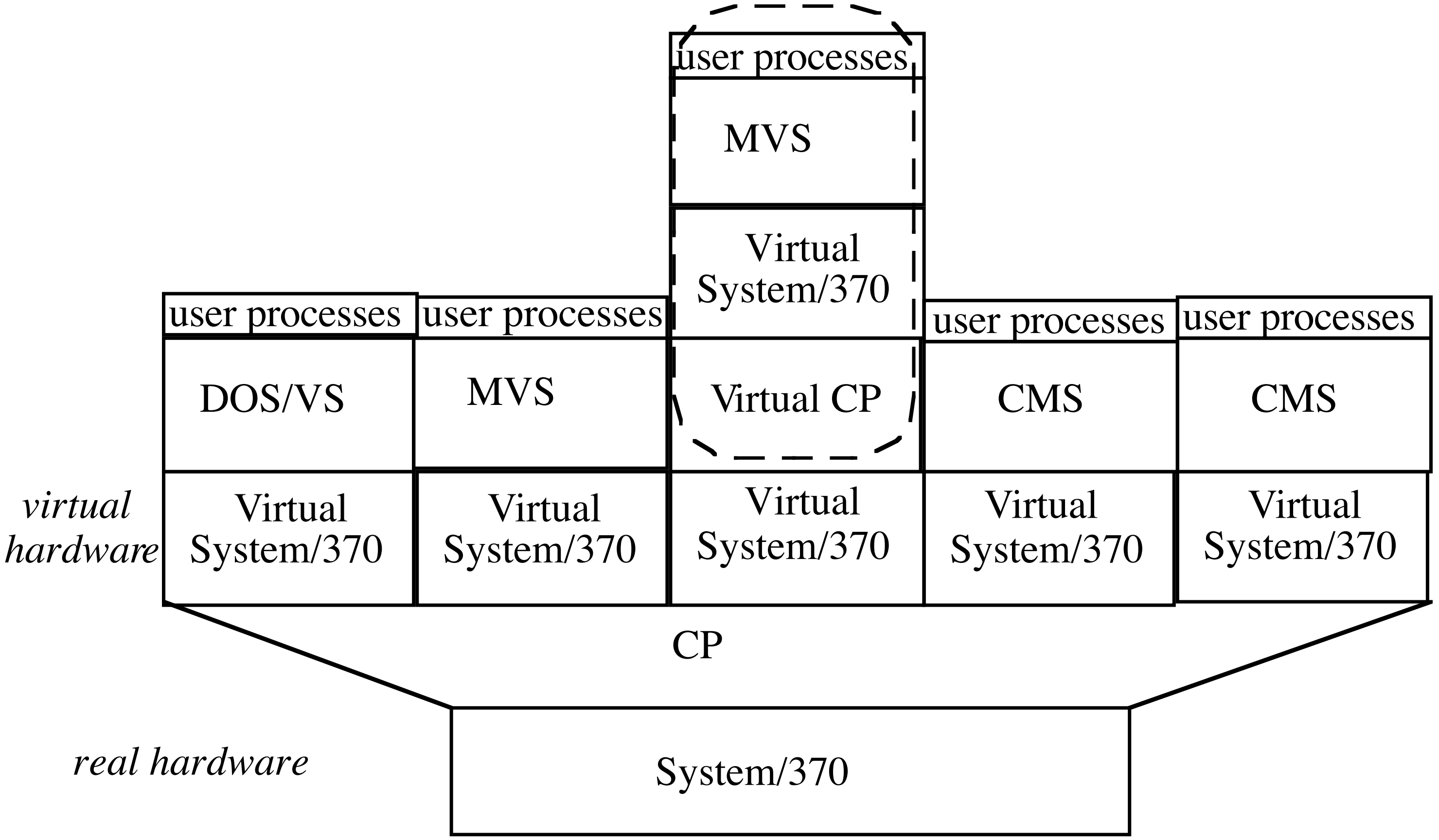
Overview

- Virtual Machine Structure
- Virtual Machine Monitor
  - – Privilege
  - – Physical Resources
  - – Paging

What Is It?

- *Virtual machine monitor* (VMM) virtualizes system resources
  - Runs directly on hardware
  - Provides interface to give each program running on it the illusion that it is the only process on the system and is running directly on hardware
  - Provides illusion of contiguous memory beginning at address 0, a CPU, and secondary storage to *each* program

Example: IBM VM/370



Adapted from Dietel, pp. 606–607

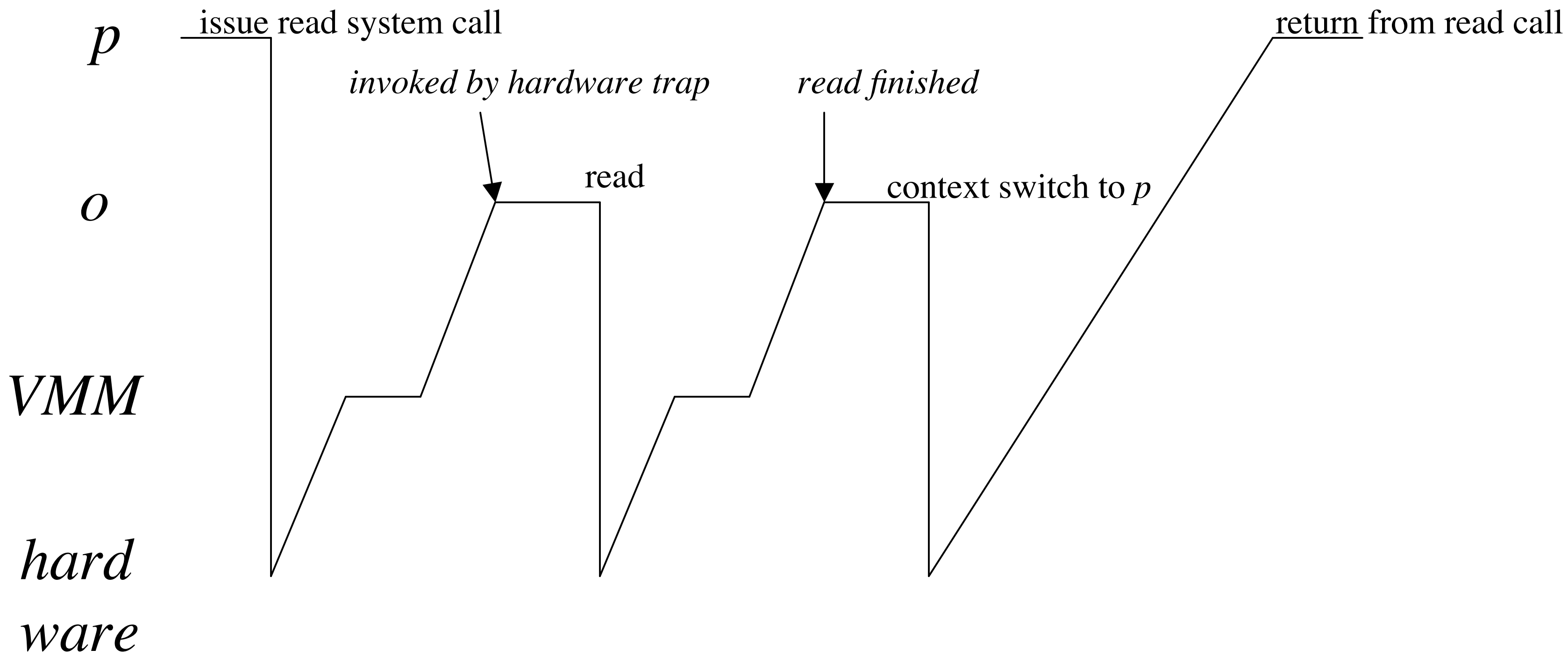
Privileged Instructions

1. VMM running operating system *o*, which is running process *p*
   - *p* tries to read—privileged operation traps to hardware
2. VMM invoked, determines trap occurred in *o*
   - VMM updates state of *o* to make it look like hardware invoked *o* directly, so *o* tries to read, causing trap
3. VMM does read
   - Updates *o* to make it seem like *o* did read
   - Transfers control to *o*

Privileged Instructions

4. *o* tries to switch context to *p*, causing trap
5. VMM updates virtual machine of *o* to make it appear *o* did context switch successfully
   - Transfers control to *o*, which (as *o* apparently did a context switch to *p*) has the effect of returning control to *p*

Privileged Instructions



Privilege and VMs

- *Sensitive instruction* discloses or alters state of processor privilege
- *Sensitive data structure* contains information about state of processor privilege

When Is VM Possible?

- Can virtualize an architecture when:
  1. All sensitive instructions cause traps when executed by processes at lower levels of privilege
  2. All references to sensitive data structures cause traps when executed by processes at lower levels of privilege

Example: VAX System

- 4 levels of privilege (user, supervisor, executive, kernel)
  - CHMK changes privilege to kernel level; sensitive instruction
    - Causes trap *except* when executed in kernel mode; meets rule 1
  - Page tables have copy of PSL, containing privilege level; sensitive data structure
    - If user level processes prevented from altering page tables, trying to do so will cause a trap; this meets rule 2

Multiple Levels of Privilege

- Hardware supports *n* levels of privilege
  - VM must also support *n* levels
  - VM monitor runs at highest level, so *n*–1 levels of privilege left!
- Solution: virtualize levels of privilege
  - Called *ring compression*

Example: VAX VMM System

- VMM at kernel level
- VMM maps virtual kernel and executive level to (real) executive mode
  - Called *VM kernel level*, *VM executive level*
  - Virtual machine bit added to PSL
    - If set, current process running on VM
  - Special register, VMPSL, records PSL of currently running VM
  - All sensitive instructions that *could* reveal level of privilege get this information from VMPSL or trap to the VMM, which then emulates the instruction

Alternate Approach

- Divide users into different classes
- Control access to system by limiting access of each class

Example: IBM VM/370

- Each control program command associated with user privilege classes
  - "G" (general user) class can start a VM
  - "A" (primary system operator) class can control accounting, VM availability, other system resources
  - "Any" class can gain or surrender access to VM

Physical Resources and VMs

- Distributes resources among VMs as appropriate
  - Each VM appears to have reduced amount of resources from real system
  - Example: VMM to create 10 VMs means real disk split into 10 minidisks
    - Minidisks may have different sizes
    - VMM does mapping between minidisk addresses, real disk addresses

Example: Disk I/O

- VM's OS tries to write to disk
  - I/O instruction privileged, traps to VMM
- VMM checks request, services it
  - Translates addresses involved
  - Verifies I/O references disk space allocated to that VM
  - Services request
- VMM returns control to VM when appropriate
  - If I/O synchronous, when service complete
  - If I/O asynchronous, when service begun

Paging and VMs

- Like ordinary disk I/O, but 2 problems
  - Some pages may be available only at highest level of privilege
    - VM must remap level of privilege of these pages
  - Performance issues
    - VMM paging its own pages is transparent to VMs
    - VM paging is handled by VMM; if VM's OS does lots of paging, this may introduce significant delays

Example: VAX/VMS

- On VAX/VMS, only kernel level processes can read some pages
  - What happens if process at VM kernel level needs to read such a page?
    - Fails, as VM kernel level is at real executive level
  - VMM reduces level of page to executive, then it works
    - Note: security risk!
      - In practice, OK, as VMS allows executive level processes to change to kernel level

Example: IBM VM/370

- Supports several different operating systems
  - OS/MFT, OS/MVT designed to access disk storage
    - If jobs being run under those systems depend on timings, delay caused by VM may affect success of job
  - If system supports virtual paging (like MVS), either MVS or VMM may cause paging
    - The VMM paging may introduce overhead (delays) that cause programs to fail that would not were the programs run directly on the hardware