Antonius RC

SISTEM OPERASI 12
VIRTUAL MACHINE

Overview

- **Fundamental idea** – abstract hardware of a single computer into several different execution environments
  - **layered approach model**
- Several components:
  - **Host** – underlying hardware system
  - **Virtual machine manager** (**VMM**) or **hypervisor** – creates and runs virtual machines by providing interface that is ***identical*** to the host
  - **Guest** – process provided with virtual copy of the host
    - Usually an operating system
- Single physical machine can run **multiple** operating systems concurrently, each in its **own virtual machine**

VIRTUAL MACHINE (1)

- Mesin virtual memperlakukan hardware dan sistem operasi seolah-olah berada pada level yang **sama** sebagai hardware.
- Pendekatan mesin virtual menyediakan sebuah interface yang **identik** dengan seluruh hardware yang ada.
- Sistem Operasi **host** membuat **ilusi** dari banyak proses, masing-masing dieksekusi pada prosesornya sendiri dengan virtual memorinya sendiri.
- Setiap **guest** menyediakan sebuah (virtual) copy dari semua hal yang ada pada komputer
- VM ada 2: **system VM dan application VM**

VIRTUAL MACHINE (2)

- First appeared commercially in IBM mainframes in 1972
- Fundamentally, **multiple execution environments** (different operating systems) can share the same hardware
- **Protect** from each other VM
- **Commutate** with each other, other physical systems via networking
- It using "**Open Virtual Machine Format**", standard format of virtual machines, allows a VM to run within many different virtual machine (host) platforms

KEUNTUNGAN VM

- Keamanan
  - Dari virus, serangan hacker
- Kemudahan instalasi
  - Tinggal copy paste image, cocok untuk semua hardware
- Cocok digunakan dalam penelitian dan percobaan
- Mendukung green computing
  - Run multiple, different OSes on a single machine
- Murah

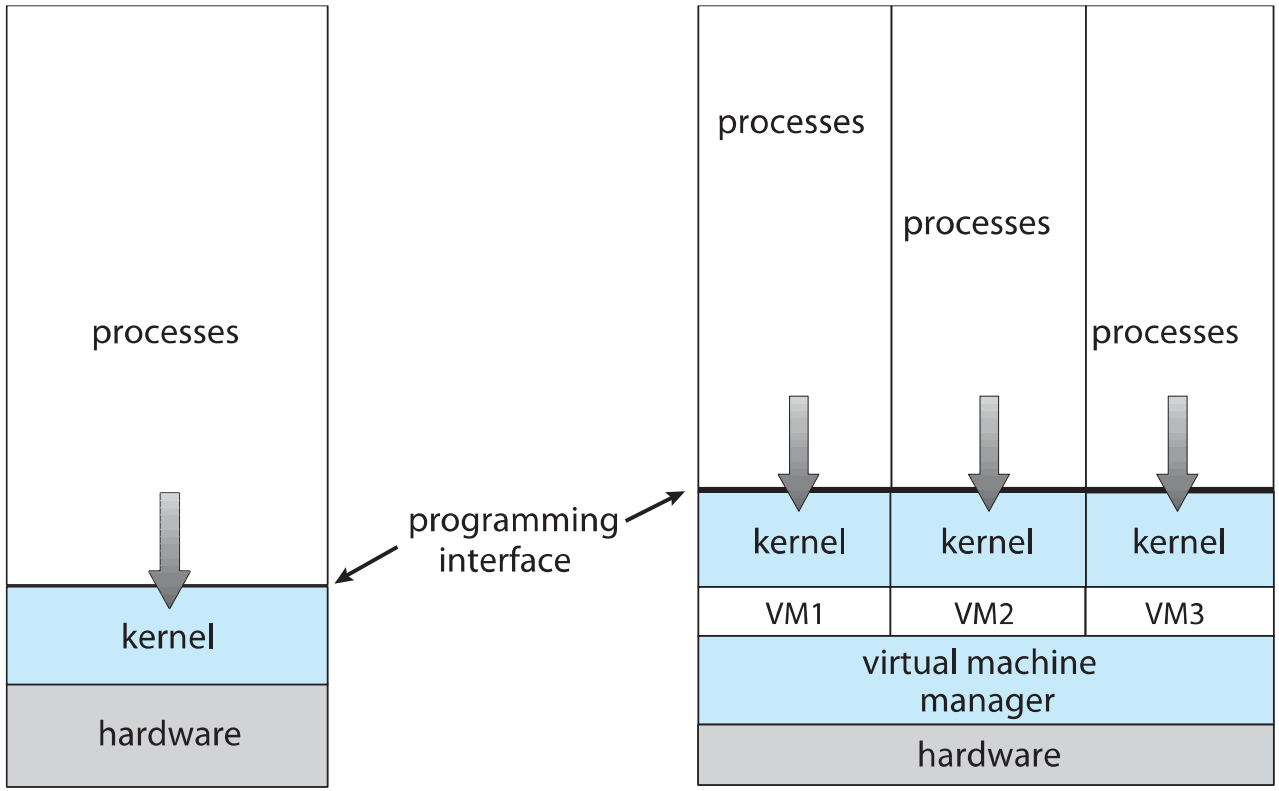
KEUNTUNGAN VM (2)

- **Templating** – create an OS + application VM, provide it to customers, use it to create multiple instances of that combination
- **Live migration** – move a running VM from one host to another!
  - No interruption of user access
- When all those features taken together -> **cloud computing**

KERUGIAN VM

- Waktu yang dibutuhkan I/O bisa lebih cepat (karena ada spooling), tapi bisa lebih lambat (karena diinterpreted)
- Tidak semua aplikasi kompatible
- Membutuhkan hardware dengan spesifikasi yang cukup tinggi

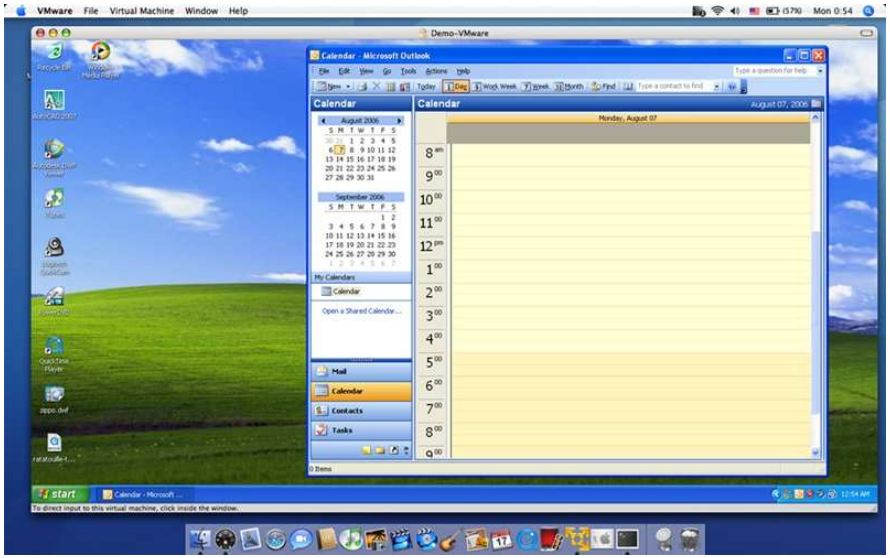
VM System Models

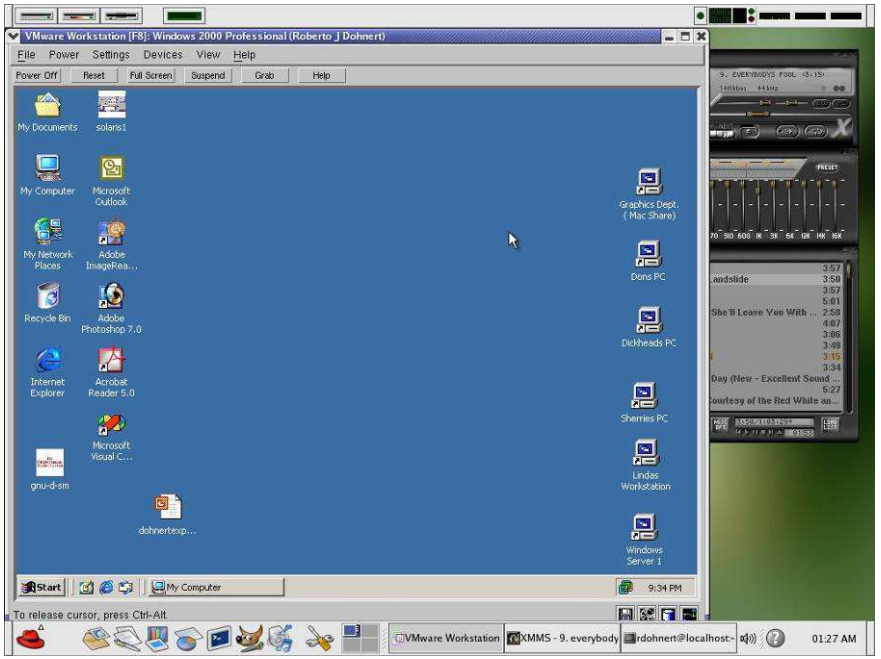


(a) Nonvirtual machine

(b) Virtual machine

CONTOH: VMWARE





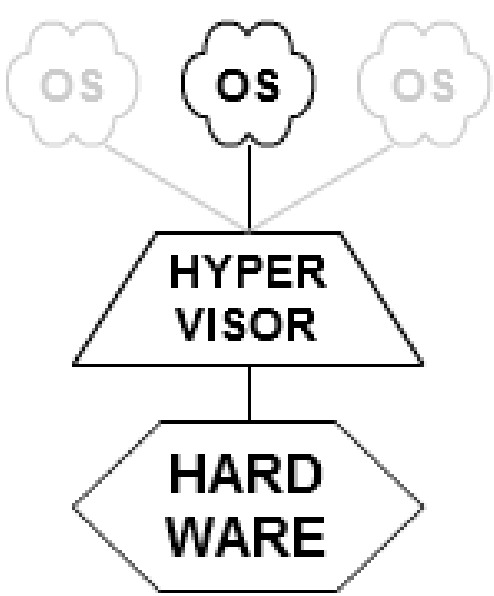
Implementation of VMMs

- **Type 0 hypervisors - Hardware-based solutions** that provide support for virtual machine creation and management via **firmware**
  - Hypervisor is unhosted
  - IBM LPARs and Oracle LDOMs are examples
- **Type 1 hypervisors (OS) –** includes **general-purpose operating systems** that provide standard functions as well as VMM functions
  - Hypervisor is controlled and assisted by a root or parent operating system
  - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM

Implementation of VMMs

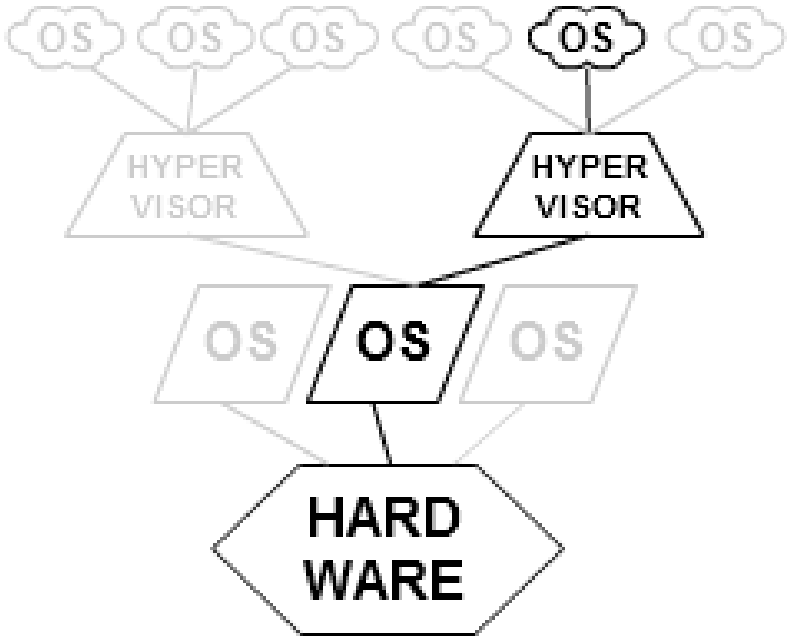
- **Type 2 hypervisors (Application) - Applications** that run on standard operating systems but provide VMM features to guest operating systems
  - hypervisor is integrated into a host OS
  - Including VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox

TYPE 1 VS TYPE 2



TYPE 1

native
(bare metal)



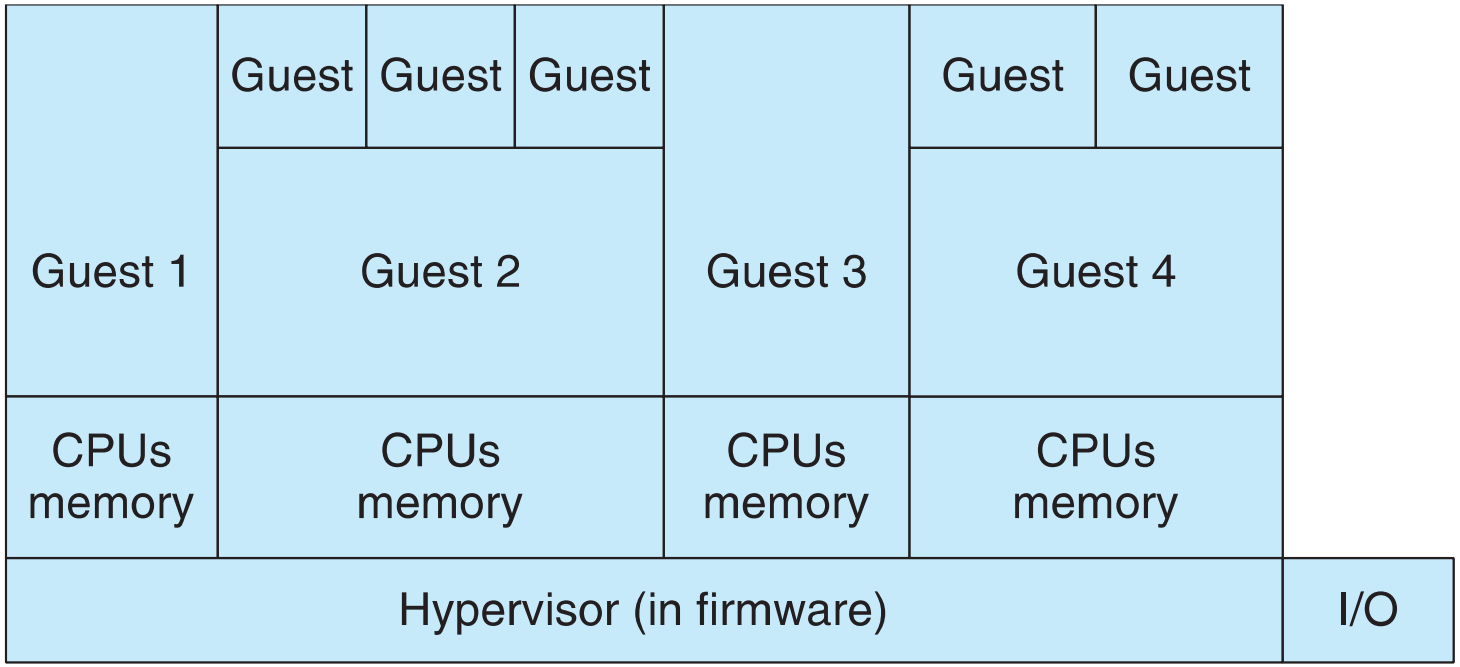
TYPE 2

hosted

Types of VMs – Type 0 Hypervisor

- Old idea, under many names by Hardware manufacturers
  - A HW feature implemented by **firmware**
  - OS need to nothing special, because VMM is in firmware
  - **Smaller** feature set than other types
  - Each guest has dedicated HW
  - Can be guest in guest VMM
- I/O a challenge as: **difficult to have enough** devices, controllers to dedicate to each guest

TYPE 0 HYPERVISOR



TYPES OF VMS – TYPE 1 HYPERVISOR

- Commonly found in **company data centers**
- **Special purpose operating systems** that run natively on HW
  - Run in **kernel mode**
  - Guests generally don't know they are running in a VM
  - Implement **device drivers** for host HW
  - In many ways, treat **guests OSes as just another process**
- Ex: RedHat Enterprise Linux with KVM, Windows with Hyper-V, Oracle Solaris

Types of VMs – Type 2 Hypervisor

- **Very little** OS involvement in virtualization
- VMM is **simply another process**, run and managed by host
  - Even the host doesn't know they are a VMM running guests
- Tend to have poorer overall performance because **can't take advantage of some HW features**
- But also a benefit because **require no changes to host OS**

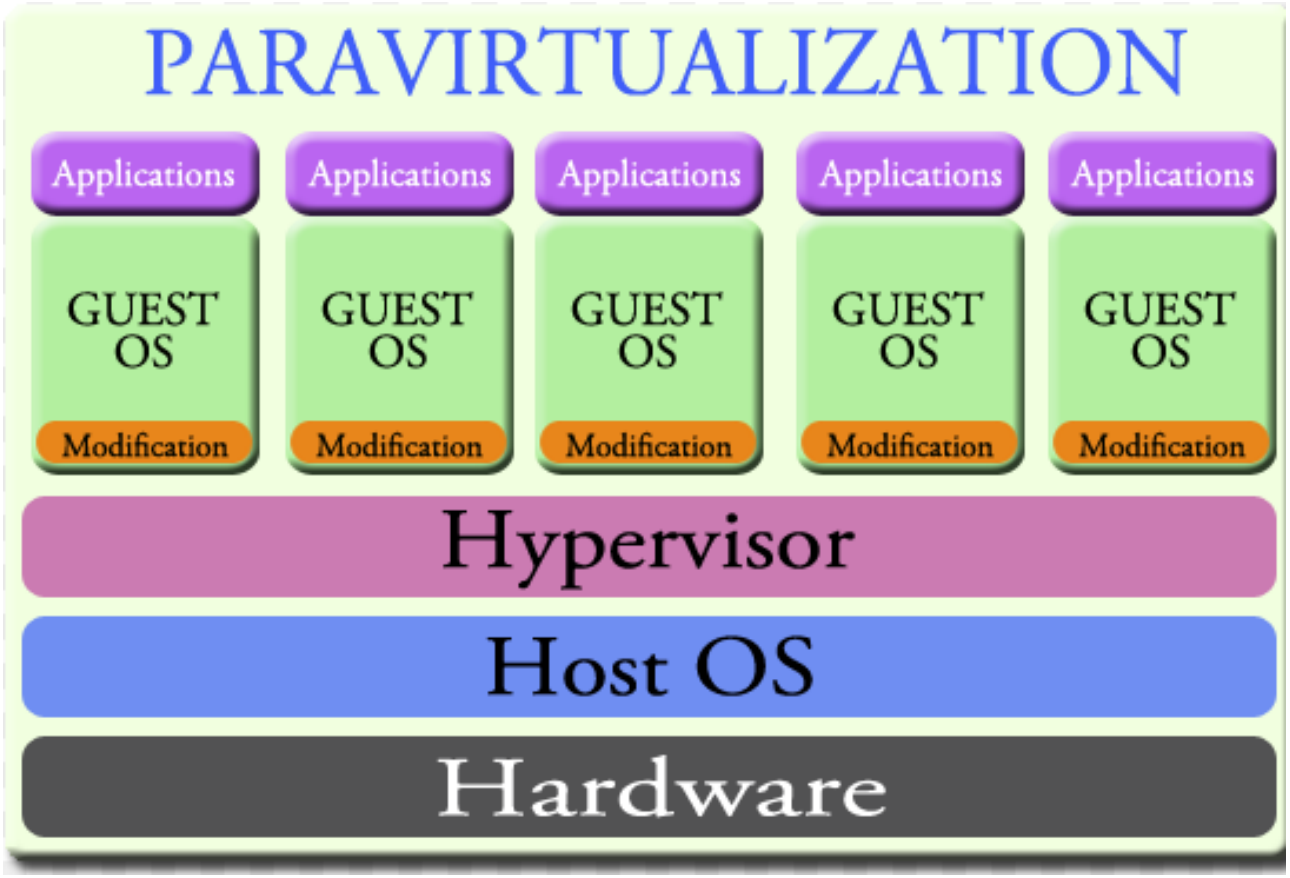
Implementation of VMMs (cont.)

- Other variations of Virtual Machine:
  - **Paravirtualization** - Technique in which the **guest operating system is modified** to work in cooperation with the VMM to optimize performance
    - Ex: VMWare dan Linux
  - **Programming-environment virtualization** - VMMs do not virtualize real hardware but instead create an **optimized virtual system framework**
    - Used by Oracle Java and Microsoft.Net by its Framework

Types of VMs – Paravirtualization

- Does not fit the definition of virtualization – VMM not presenting an **exact duplication** of underlying hardware
  - Guest had to be **modified** to use run on paravirtualized VMM
  - Less needed as hardware support for VMs grows
- Ex: Xen, Microsoft Hyper-V and VMware ESX Server

PARAVIRTUALIZATION ARCHITECTURE



Types of VMs – Programming Environment Virtualization

- Programming language is designed to run within **custom-built virtualized environment**
  - For example Oracle Java has many features that depend on running in **Java Virtual Machine** (**JVM**)
- Its using API
- Its create **new layer** between OS and programming language
- Example: JVM
  - JVM compiled to run on many systems (Windows, Linux, etc)
  - Programs written in Java run in the JVM no matter the underlying system
  - It's interpreted

Implementation of VMMs (cont.)

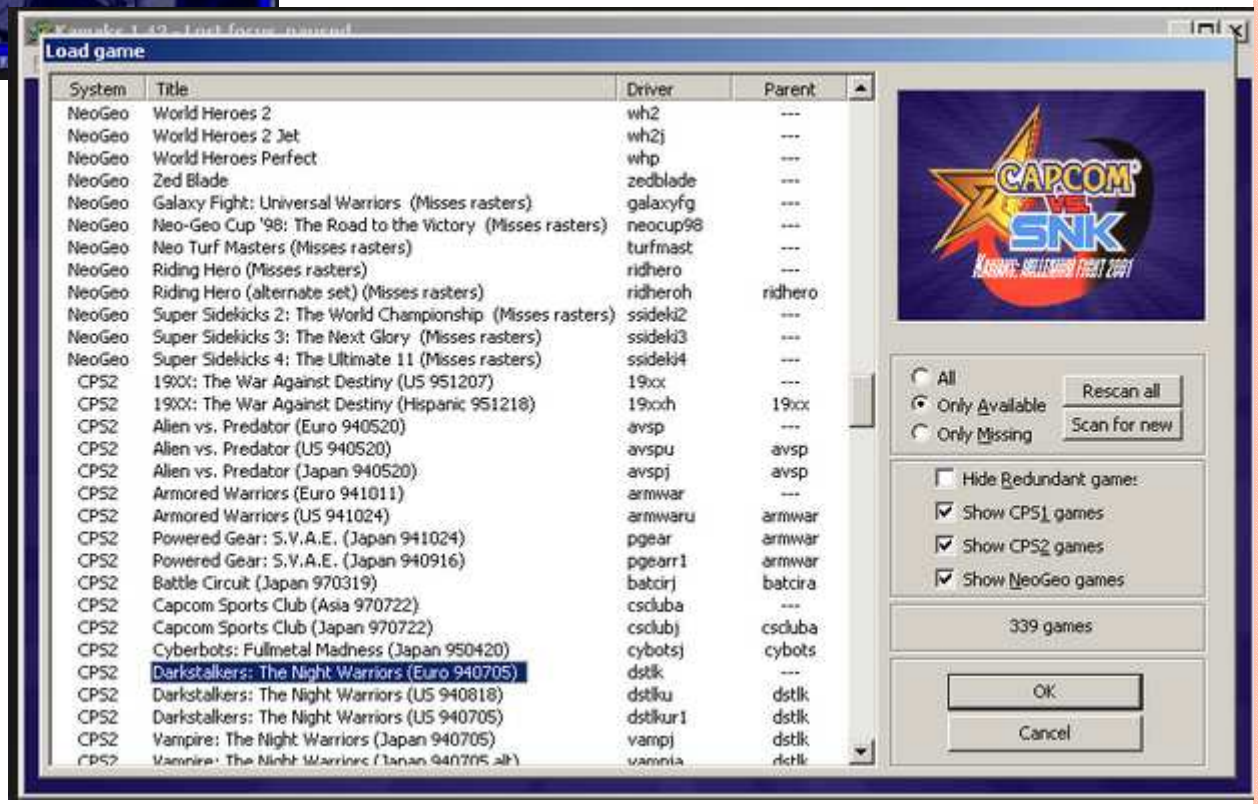
- **Emulators –** Allow applications written for one hardware environment **to run on a very different hardware** environment, such as a different type of CPU
  - NeoGeo Emulator, Android Emulator, Windows Phone Emulator, Cygwin, MAME
- **Application containment** - Not virtualization at all but rather provides virtualization-like features by **segregating applications** from the operating system, making them **more secure, manageable**
  - **Sandboxie**: runs your programs in an isolated space which prevents them from making permanent changes to other programs and data in your computer.
  - Including Oracle Solaris Zones, Sanboxie (http://www.sandboxie.com/)

Types of VMs – Emulation / Emulator

- Another (older) way for running one operating system on a different operating system
  - Virtualization requires underlying **CPU to be same as guest was compiled for**
  - Emulation allows **guest to run on different CPU**
- Necessary to **translate** all guest instructions from guest CPU to native CPU
- Useful when host system has **one architecture**, guest compiled for **other architecture**
  - Ex: CPU is new but still want to run old applications
- **Very popular** – especially in gaming where old consoles emulated on new

NeoGEO


GAMELIST (0/181)
K.O.F. '95
K.O.F. '96
K.O.F. '96 (Bootleg)
K.O.F. '97
K.O.F. '97 Plus
K.O.F. '98
K.O.F. '99
K.O.F. 2000
K.O.F. 2001
K.O.F. 2001 Plus
K.O.F. 2001 Plus s2
K.O.F. 2002
K.O.F. 2002 Magic
K.O.F. 2002 Magic s2
K.O.F. 2002 Plus
K.O.F. 2002 Plus s2
K.O.F. 2002 Plus s3
K.O.F. Special 2004
OPTIONS
CONTROLS
DIRECTORIES
VIDEO OPTIONS
AUDIO OPTIONS
MACHINE SETTINGS
SHOTS FACTORY
LAUNCH
IMPORT
EXIT
PREVIEW
EVENTS
NeoRAGEx 5.0
GenuineIntel with MMX.
DIRECT SOUND audio device detected.
SnK-NeoFighters/Team!!


Load game
System
Title
Driver
Parent
NeoGeo World Heroes 2 wh2 ---
NeoGeo World Heroes 2 Jet wh2j ---
NeoGeo World Heroes Perfect whp ---
NeoGeo Zed Blade zedblade ---
NeoGeo Galaxy Fight: Universal Warriors (Misses rasters) galaxyfg ---
NeoGeo Neo-Geo Cup '98: The Road to the Victory (Misses rasters) neocup98 ---
NeoGeo Neo Turf Masters (Misses rasters) turfmast ---
NeoGeo Riding Hero (Misses rasters) ridhero ---
NeoGeo Riding Hero (alternate set) (Misses rasters) ridheroh ridhero
NeoGeo Super Sidekicks 2: The World Championship (Misses rasters) ssideki2 ---
NeoGeo Super Sidekicks 3: The Next Glory (Misses rasters) ssideki3 ---
NeoGeo Super Sidekicks 4: The Ultimate 11 (Misses rasters) ssideki4 ---
CPS2 19XX: The War Against Destiny (US 951207) 19xx ---
CPS2 19XX: The War Against Destiny (Hispanic 951218) 19xxh 19xx
CPS2 Alien vs. Predator (Euro 940520) avsp ---
CPS2 Alien vs. Predator (US 940520) avspu avsp
CPS2 Alien vs. Predator (Japan 940520) avspj avsp
CPS2 Armored Warriors (Euro 941011) armwar ---
CPS2 Armored Warriors (US 941024) armwaru armwar
CPS2 Powered Gear: S.V.A.E. (Japan 941024) pgear armwar
CPS2 Powered Gear: S.V.A.E. (Japan 940916) pgearr1 armwar
CPS2 Battle Circuit (Japan 970319) batcirj batcira
CPS2 Capcom Sports Club (Asia 970722) cscluba ---
CPS2 Capcom Sports Club (Japan 970722) csclubj cscluba
CPS2 Cyberbots: Fullmetal Madness (Japan 950420) cybotsj cybots
CPS2 Darkstalkers: The Night Warriors (Euro 940705) dstlk ---
CPS2 Darkstalkers: The Night Warriors (US 940818) dstlku dstlk
CPS2 Darkstalkers: The Night Warriors (US 940705) dstlkur1 dstlk
CPS2 Vampire: The Night Warriors (Japan 940705) vampj dstlk
CAPCOM VS. SNK
All
Only Available
Only Missing
Rescan all
Scan for new
Hide Redundant games
Show CPS1 games
Show CPS2 games
Show NeoGeo games
339 games
OK
Cancel

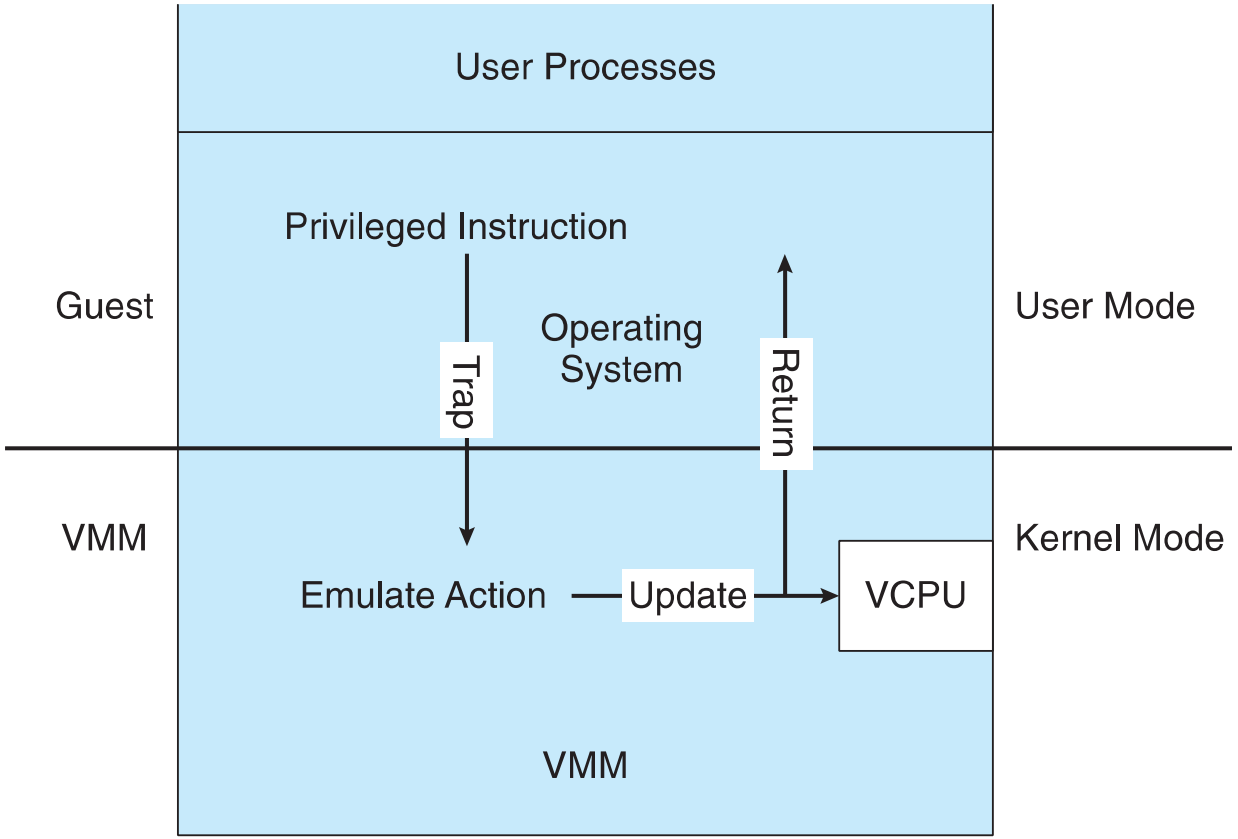
Building Block of VM

- Most VMMs implement **virtual CPU** (**VCPU**) to represent **state** of CPU per guest
- Dual mode CPU means: **guest executes in user mode**
  - Kernel runs in kernel mode, but it's not safe to let guest kernel run in kernel mode too
  - So VM needs **two modes – virtual user mode and virtual kernel mode**
  - Actions in guest that usually cause switch to kernel mode must cause switch to virtual kernel mode

Teknik 1: Trap-and-Emulate

- How does switch from virtual user mode to virtual kernel mode occur?
  - Attempting a privileged instruction in user mode causes an error -> trap
  - VMM gains control, analyzes error, executes operation as attempted by guest
  - Known as **trap-and-emulate**
- User mode code in guest runs at same speed
- But kernel mode privilege mode code runs **slower** due to trap-and-emulate

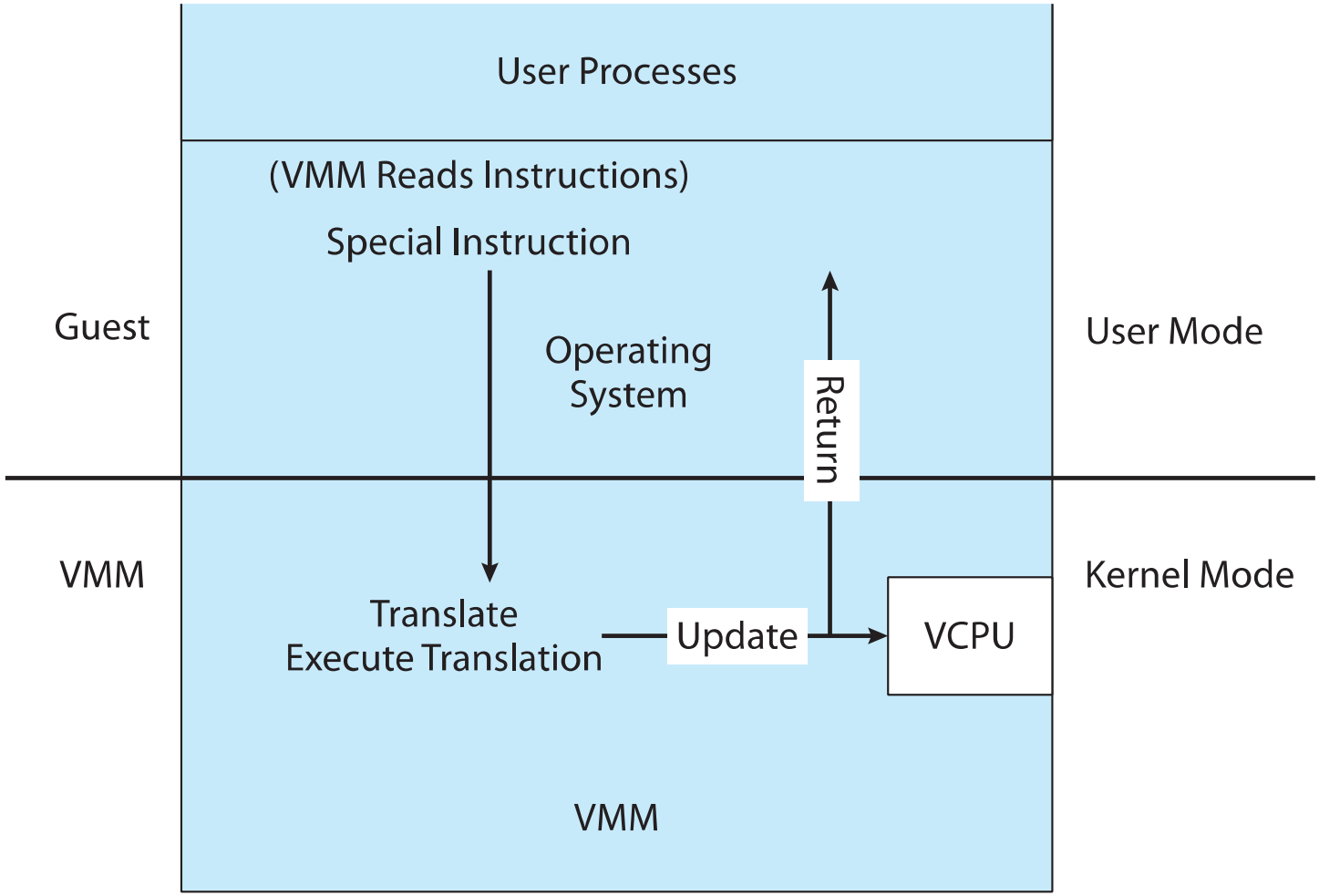
Trap-and-Emulate Virtualization Implementation



Teknik 2: Binary Translation

- Implemented by **translation of code within VMM**
- Code reads native instructions dynamically from guest, generates **native binary code** that executes in place of original code
- Performance of this method would be **poor without optimizations**
  - Products like VMware use **caching**

Binary Translation Virtualization Implementation



Building Blocks – Hardware Assistance

- All virtualization needs some **HW support**
- More support -> more feature rich, stable, better performance of guests
- Intel added new **VT-x** instructions in 2005 and AMD the **AMD-V** instructions in 2006
  - CPUs with these instructions **remove need for binary translation**
  - In guest mode, guest OS thinks it is running natively

OS Component – CPU Scheduling

- Single-CPU systems act like **multiprocessor** ones when virtualized
  - One or more virtual CPUs per guest
- Generally VMM has **one or more physical CPUs and number of threads to run on them**
  - Guests configured with certain number of VCPUs
    - Can be adjusted throughout life of VM
  - When enough CPUs for all guests -> VMM can allocate dedicated CPUs, each guest much like native operating system managing its CPUs
  - Usually not enough CPUs -> CPU **overcommitment**

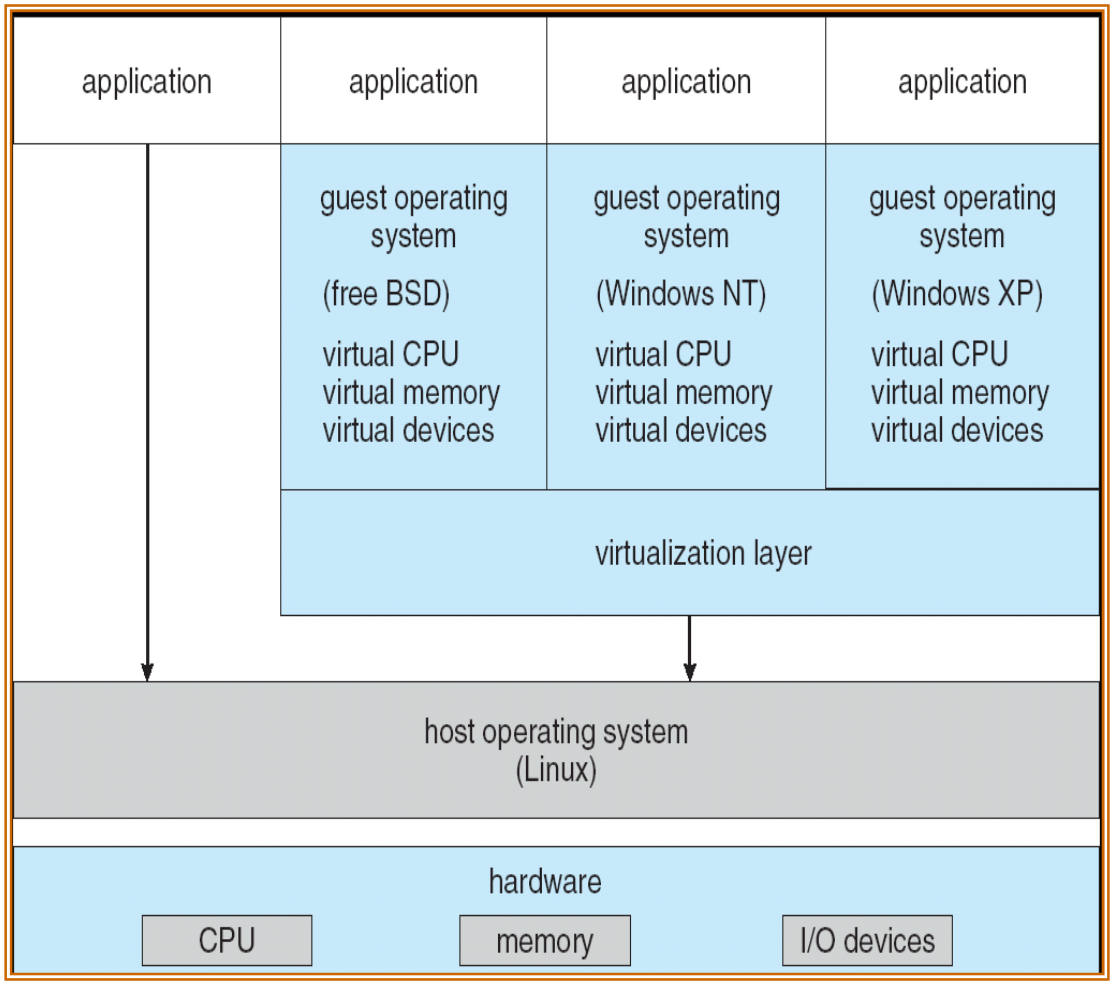
OS Component – Networking

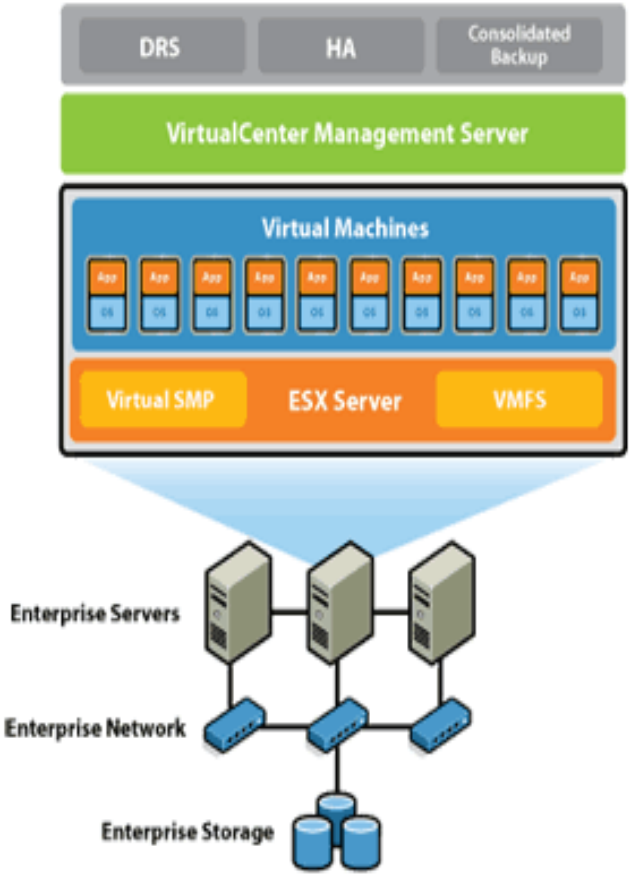
- Networking also complex as VMM and guests all need network access
  - VMM can **bridge** guest to network (allowing direct access)
  - And / or provide **network address translation** (**NAT**)
    - NAT address local to machine on which guest is running, VMM provides address translation to guest to hide its address

EXAMPLES - VMWARE

- VMware Workstation runs **on x86 or x64**, provides VMM for guests
- Runs as application on other native, installed host operating system -> **Type 2**
- Lots of guests possible, including Windows, Linux, etc all runnable concurrently (as resources allow)

VMware Architecture





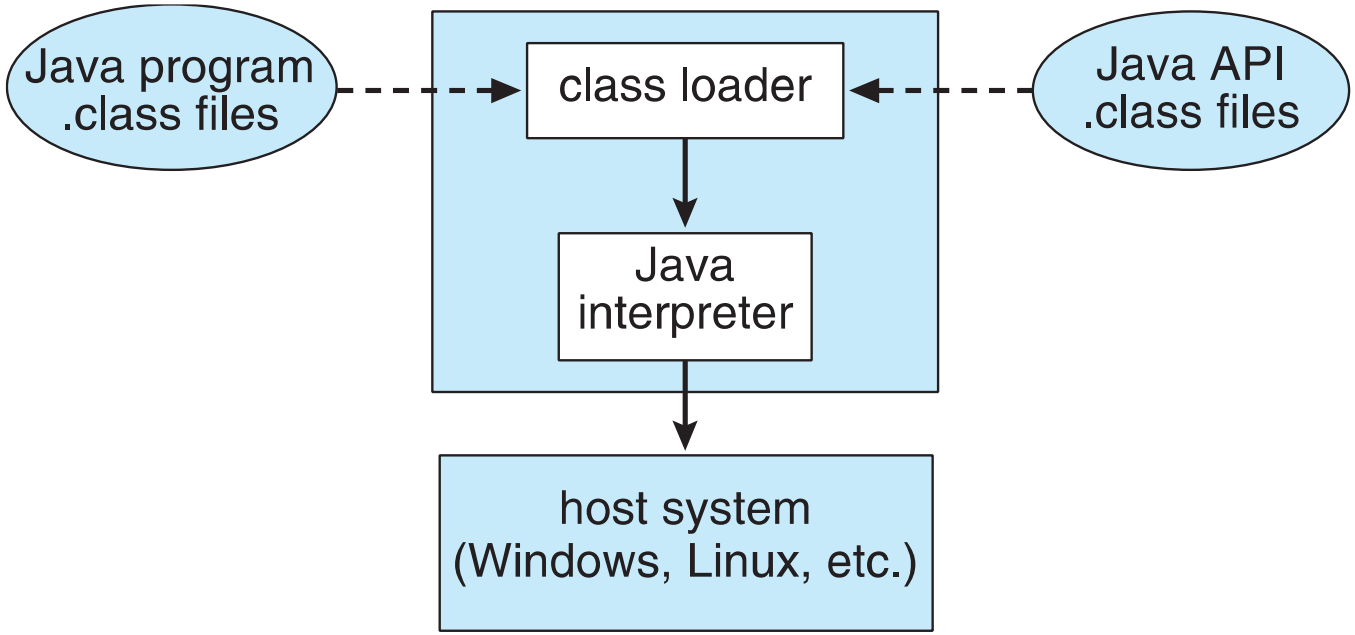
Examples – Java Virtual Machine

- Example of **programming-environment virtualization**
- Very popular language / application environment invented by Sun Microsystems in 1995
- Write once, run anywhere (WORE)
- Includes language specification (Java), API library, Java virtual machine (JVM)
- Each Java object compiled into architecture-neutral **bytecode** output (`.class`) which JVM **class loader** loads
- JVM compiled per architecture, reads bytecode and executes
- Includes **garbage collection** to reclaim memory no longer in use

JAVA VIRTUAL MACHINE

- Program Java yang telah dicompile adalah **platform-neutral bytecodes** yang dieksekusi oleh Java Virtual Machine(JVM)
- JVM terdiri dari:
  - Class loader
  - Class verificatier
  - runtime interpreter
- Mendukung **Just In-Time (JIT)** compilers yang meningkatkan performance

The Java Virtual Machine



NEXT

- **Presentasi final**
- TAS