

강사소개



- 이상희 강사 (MCT, VCI, MCITP, MCTS, MCSA:Security, MCSE:Security, Messaging, Virtualization, Private Cloud, MCDBA, MCSA, MCITP, CCNA, CCNP, CTT, HPCP, EMCP A, VCP, VCP-DT, CCA 등)
- [Email: instructor@daum.net](mailto:instructor@daum.net) or instructor@naver.com
- Project Career
 - S 반도체 전체 MS Active Directory 설계 및 컨설팅
 - D 자동차 Active Directory 설계 및 컨설팅 수행
 - 한국국제학교 Microsoft 기반 인트라넷 Restructuring 및 message infra tuning
 - Y 대학의료원 U-hospital Project 참여(스마트카드 인증시스템 구축 및 데이터베이스모델링)+ DB Server Clustering
 - K 항공 MS Infra 구현프로젝트 참여 (AD Design + Exchange Clustering) 등등
 - NH Microsoft Hyper-V 기반 가상화 컨설팅
 - 그외 다수 Microsoft, Vmware, Citrix 가상화 및 클라우드 관련 기술 교육 및 컨설팅

1. 가상화 개요

index

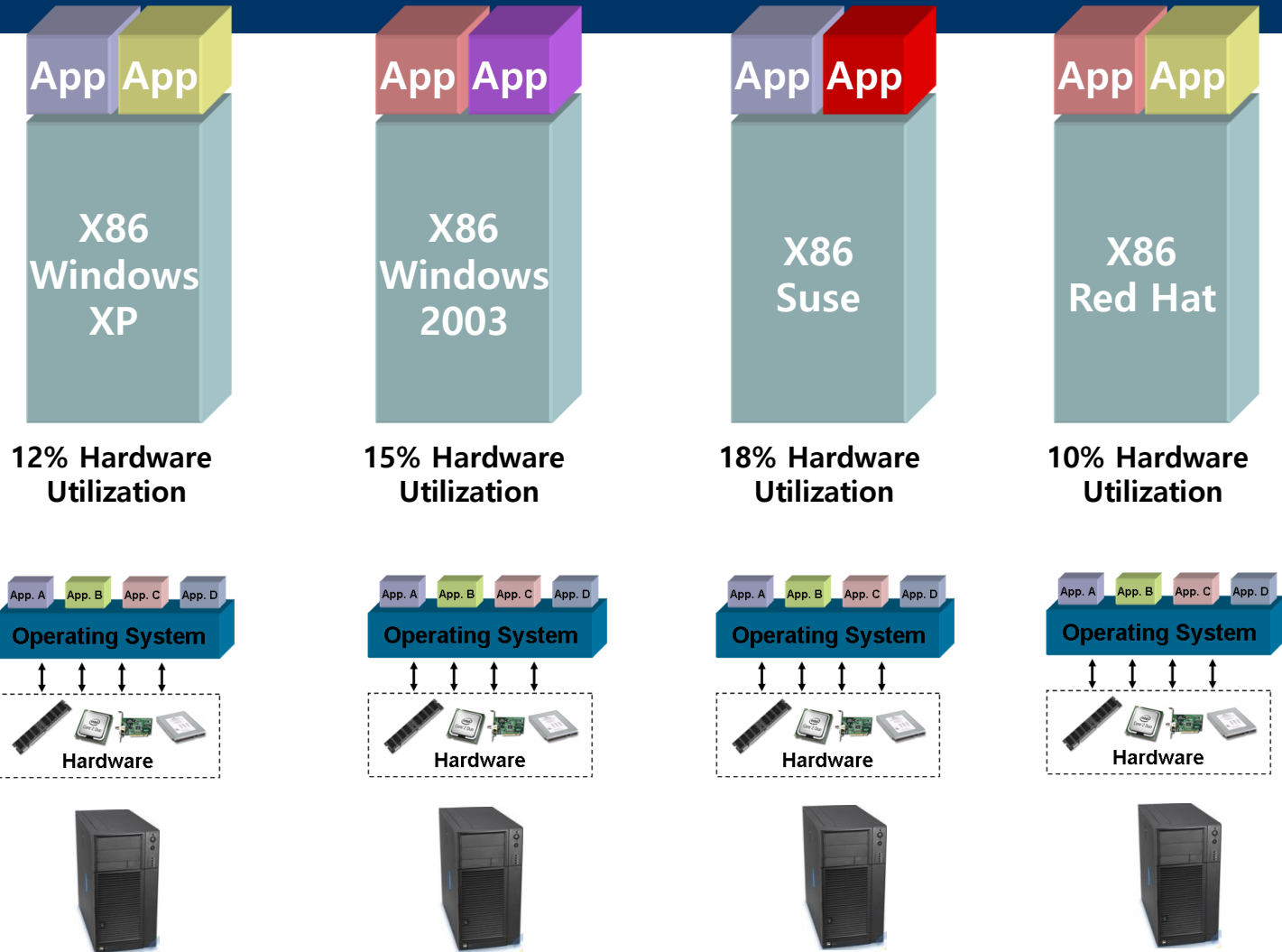
1.1 가상화?

1. 가상화?

가상화 개요

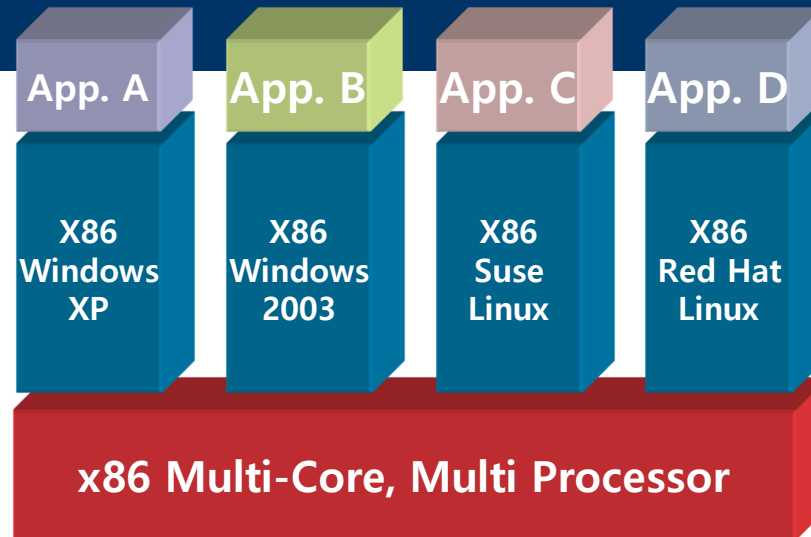
- Virtualization?
 - The abstraction of computer resources (CPU, 메모리, 스토리지, 네트워크 등)
 - Platform/**Resource**/application/desktop/cluster Virtualization
 - 가상메모리 (Virtual Memory), 가상네트워크 (VLAN) 등이 널리 알려진 가상화 기술의 예
 - 가상 머신 (VM, Virtual Machine) 이라는 용어와 더불어 "시스템 가상화"의 의미로 최근에는 널리 활용되고 있음
- Resource Virtualization
 - Virtualization of specific system resources, such as storage volumes, name spaces & network resources
 - Server V. : creation of a separate server E.g) Xen, KVM
 - Storage V.: abstracting logical storage from physical storage
 - Network V.: virtualized network address space
 - I/O V.: e.g) vNICs, vHBAs
- 서버 가상화, 스토리지 가상화, 네트워크 가상화, 서비스 가상화

1. 가상화? (Cont.)

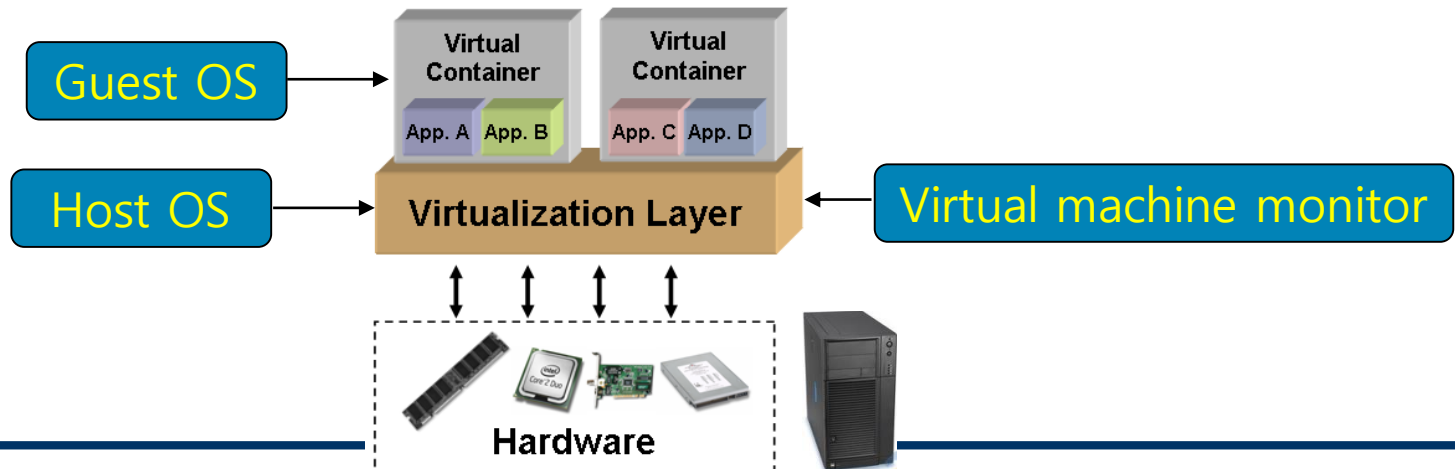


1. 가상화? (Cont.)

가상화의 장점



70% Hardware Utilization



1. 가상화? (Cont.)

서버 가상화 개요

- 개요

- 단위 서버 가상화는 서버 내부의 자원을 공유하고 그 활용률을 증대시키려는 목적을 나타낸다.
- 서버 가상화는 하나의 큰 물리적 서버 자원을 여러 대의 논리적 서버로 나누어 활용하는 'Scale Up' 형태와 여러 대의 물리적 서버를 하나의 논리적 서버로 활용하는 'Scale Out' 형태로 나눌 수 있다.

- Scale Up 가상화

- 물리적으로 하나의 서버를 논리적으로 여러 개의 서버로 활용
- 총 소유비용을 절감할 수 있도록 하고 서버 활용율을 증대시킴으로써 IT 자산을 보다 더 잘 활용할 수 있도록 지원

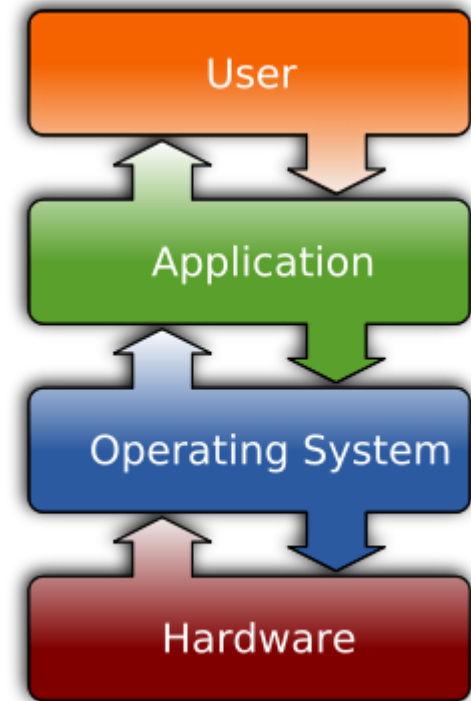
- Scale Out 가상화

- 물리적으로 여러 개의 서버를 논리적으로 하나의 서버로 활용
- 필요에 따라 애플리케이션에 대한 동적 자원 재분배를 지원함으로써 기업의 대응성과 운영 속도 제고 - 변화가 심한 사업 주기에 대처하고 급작스런 수요 증대 시점을 관리하는 데 유리

1. 가상화? (Cont.)

서버 가상화의 원리

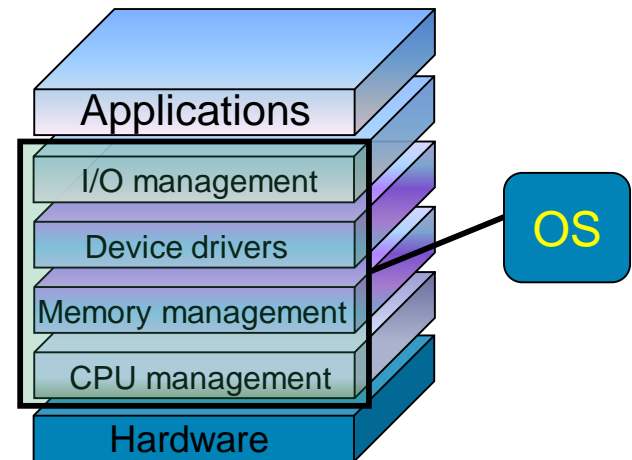
- 컴퓨터 하드웨어나 사용자 사이에서 프로그램을 번역하고 처리하는 역할을 한다.
- OS는 컴퓨터 하드웨어를 감시하고 관리하는 프로그램 집단이다.
- OS는 애플리케이션 프로그램과 하드웨어나 주변 장치들간에 자료를 원활하게 주고 받을 수 있도록 운용을 담당한다.



1. 가상화? (Cont.)

서버 가상화의 원리 : System View of the OS

- OS 는 **resource allocator** 역할을 한다.
 - Manages all resources
 - CPU time, memory space, storage space, I/O, etc.
 - 효율적이고 공정한 자원 사용을 위해 충돌을 중재.
- OS 는 **control program**이다.
 - 컴퓨터의 오류나 부적절한 사용을 막기 위해 응용 프로그램 실행을 제어합니다.



1. 가상화? (Cont.)

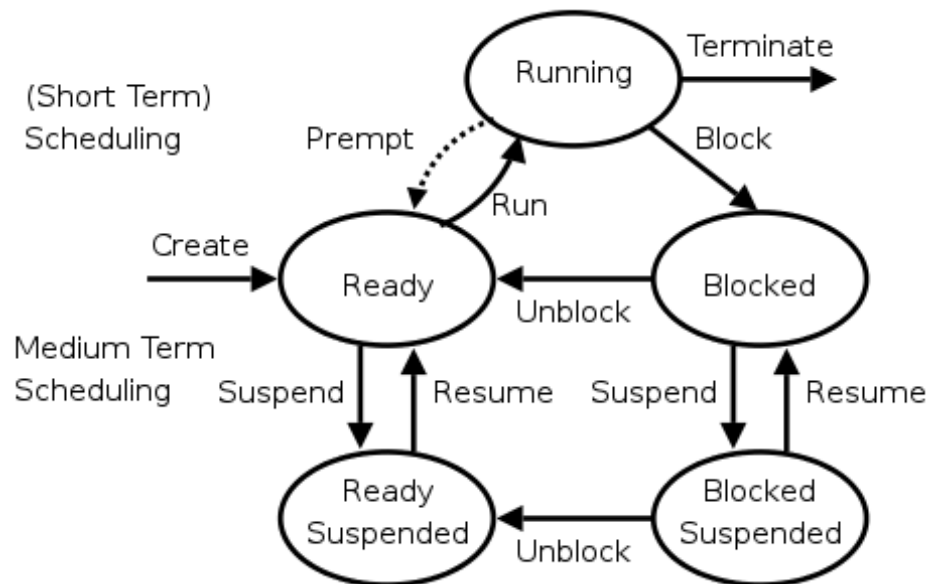
OS 아키텍처 : Kernel

- Kernel = 응용 프로그램이 기능을 수행하는 데 필요한 자원을 위한 추상화 계층을 제공합니다.
- Resources = 프로세서, 메모리, I/O 장치와 같은 하드웨어
- Kernel은 애플리케이션에서 요구하는 모든 요청들을 처리하는 인터럽트 처리기와 어떤 프로그램들이 어떤 순서로 커널의 처리시간을 공유할 것인지를 결정하는 스케줄러가 포함되어 있다.

1. 가상화? (Cont.)

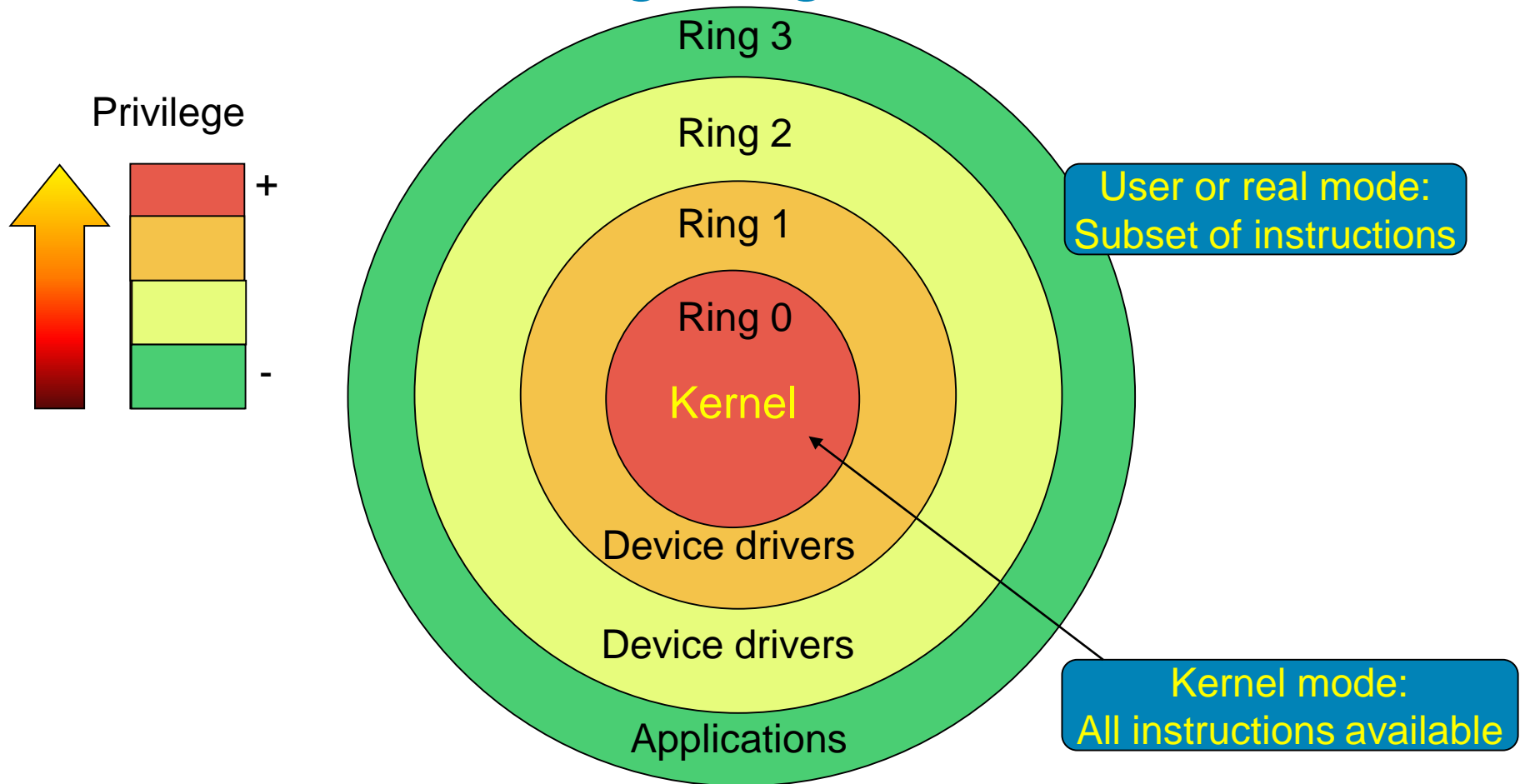
OS 아키텍처 : user-land processes

- Process = resource management의 단위
- 하나의 CPU는 동시에 하나의 Process를 처리한다.
- OS 커널은 모든 프로세스가 이들 사이를 전환하여 실행되도록합니다(pseudo-parallelism)
- 프로세스는 자체 주소 공간, 변수, 열린 파일, 신호 및 신호 처리기를 가지고 있습니다.



1. 가상화? (Cont.)

x86 Protection/Privilege Rings



1. 가상화? (Cont.)

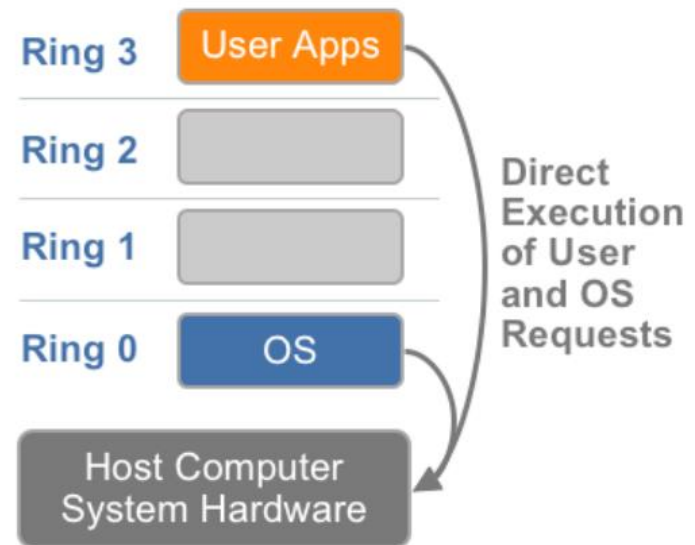
x86 Protection/Privilege Rings

- x86 CPU의 두가지 모드
 - Kernel Mode: 운영체제 핵심 커널로서 하드웨어 장치를 직접 제어한다.
 - User Mode: 사용자 프로세스나 일반적인 애플리케이션들이 작동된다.
- 사용자 모드에 있는 프로세스나 애플리케이션들이 하드웨어에 접근하기 위해서는 반드시 커널 모드의 허락을 받고 사용해야만 한다.
- 이러한 권한과 관련하여 x86 CPU에서는 4단계의 레벨이 존재한다. (Protection/Privilege Ring)
 - Ring 0 : 운영체제의 커널이 위치하며 일반적으로 하드웨어를 직접 접근하고 조종할 수 있는 권한을 갖게 된다.
 - Ring 3 : 사용자 애플리케이션들이나 프로세스들이 위치한다. Ring 3 사용자 모드의 일반 애플리케이션들이 하드웨어 접근하기 위해서는 Ring 0에 위치한 커널의 허락을 받지 않고서는 접근이 불가능하다. 이러한 권한을 설정하고 사용자 모드의 애플리케이션들이 함부로 하드웨어 접근하지 못하게 함으로써 결국 운영체제는 안정성을 갖추게 된다.

1. 가상화? (Cont.)

CPU Virtualization

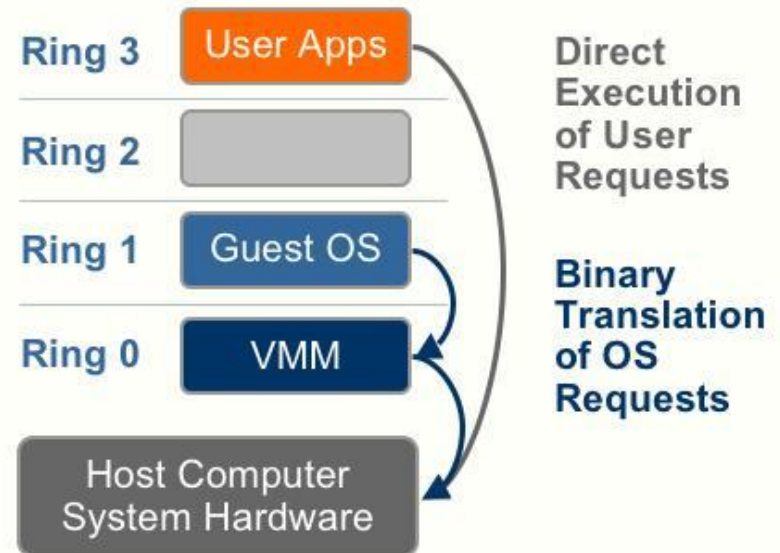
- x86 privilege level 아키텍처 without virtualization
 - x86 CPU은 4개의 Ring 아키텍처로 되어 있다.
 - 전통적인 운영체제에서는 Ring 0 는 커널이 위치하게 되어있다.
 - Ring 0 : most privileged mode (kernel code)
 - Ring 1 : 사용하지 않음
 - Ring 2 : 사용하지 않음
 - Ring 3 : user mode (applications)



1. 가상화? (Cont.)

CPU Virtualization

- Full Virtualization using Binary Translation
 - VMKernel (VMM)이 하드웨어 리소스를 조종하는 핵심 하이퍼바이저 커널이기 때문에 Ring 0에 위치한다면, 게스트 운영체제는 Ring 1에 위치하게 된다.
 - 게스트 운영체제에서 CPU 하드웨어에 직접 명령을 던져야 하는 프리빌리지 코드들을 바이너리 변환(Binary Translation) 기법을 통해 중간에 가로채어 에뮬레이션화해서 접근을 조정한다.
 - 명령을 scan하여 virtualizable 명령 (user 명령)은 그냥 실행한다.
 - non-virtualizable 명령 (Kernel 명령)에 대해서는 virtualizable 명령으로 dynamically하게 변경한다.

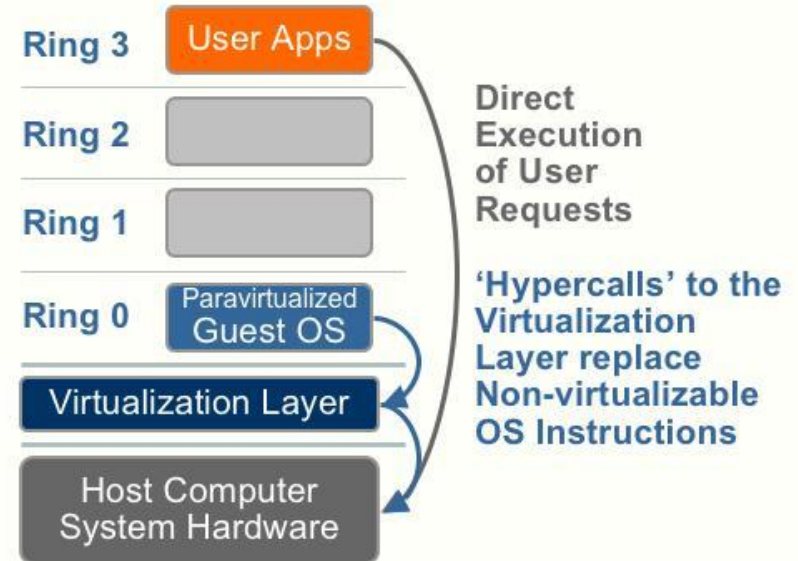


1. 가상화? (Cont.)

CPU Virtualization

- OS Assisted Virtualization or Para virtualization
 - 게스트 운영체제는 하이퍼바이저에서 인식할 수 있는 특별한 시스템 콜 명령(Hypercall)을 가지고 있어야 한다.
 - Para virtualization 은 게스트 OS들이 하이퍼바이저에 맞게 수정되어야 한다. 이것이 단점이다. 하지만, Paravirtualization 은 가상화되지 않은 시스템 성능에 가까운 성능을 보인다.

과거에는 Para Virtualization의 속도가 월등히 빨랐다. 하지만 최근에 CPU에서 Intel VT나 AMD V 기술을 제공한 후 Full-Virtualization의 속도도 Para-Virtualization 못지 않게 빨라졌다.

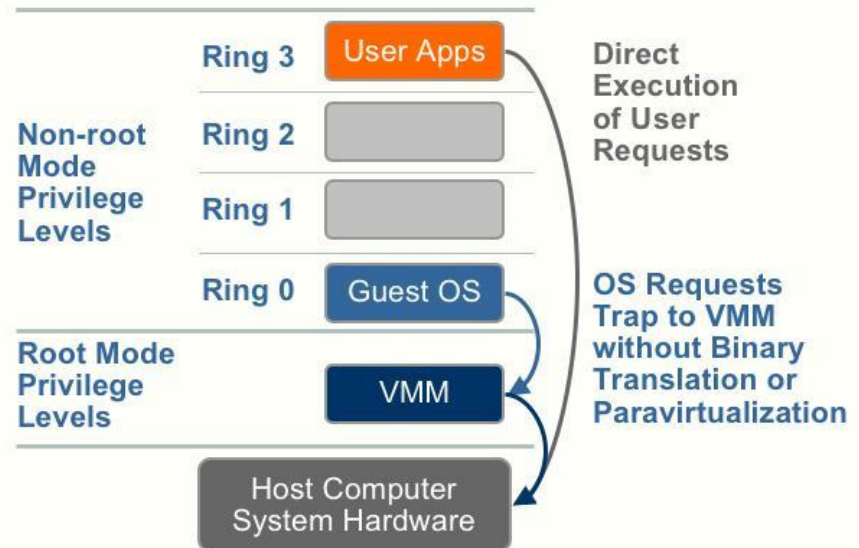


1. 가상화? (Cont.)

CPU Virtualization

- Hardware Assisted Virtualization
 - 2007년부터 CPU 제조사들이 가상화를 지원하기 시작함
 - 소프트웨어적인 가상화 시도의 한계를 하드웨어로 지원함
 - Intel과 AMD가 non-virtualizable 명령코드를 처리하기 위한 모드를 추가함 (Intel VT-x와 AMD-V 기술)

VMM이 Ring -1로 빠지면서 게스트 운영체제가 제약 없이 Ring 0의 하드웨어 접근 권한을 갖게 됨으로써 paravirtualization 가상화 시스템에서도 Windows 게스트 운영체제가 전혀 커널 수정 없이 바로 작동이 가능하게 된 것이다.

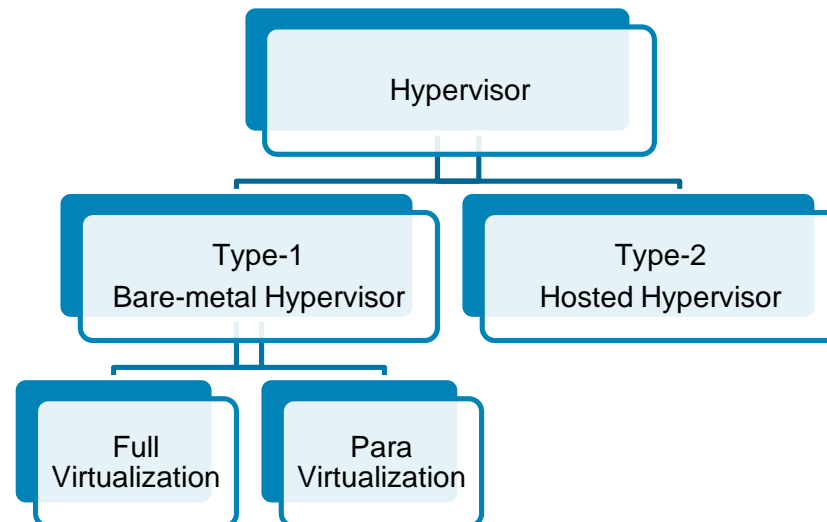


1. 가상화? (Cont.)

하이퍼바이저

- 개요

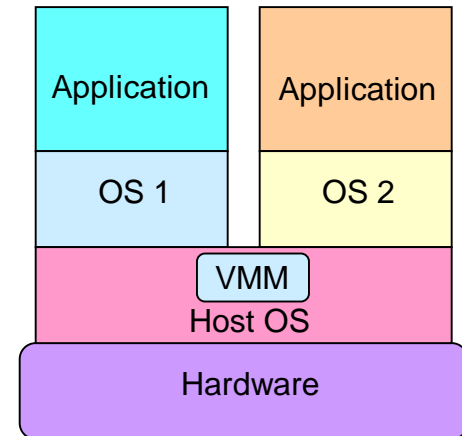
- 하이퍼바이저(hypervisor)는 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행하기 위한 논리적 플랫폼(platform)이다.
- 가상 머신 모니터(virtual machine monitor, VMM)라고도 부르며 하이퍼바이저의 유형은 일반적으로 두 가지로 나뉜다.



1. 가상화? (Cont.)

Hosted Hypervisor (Type-2)

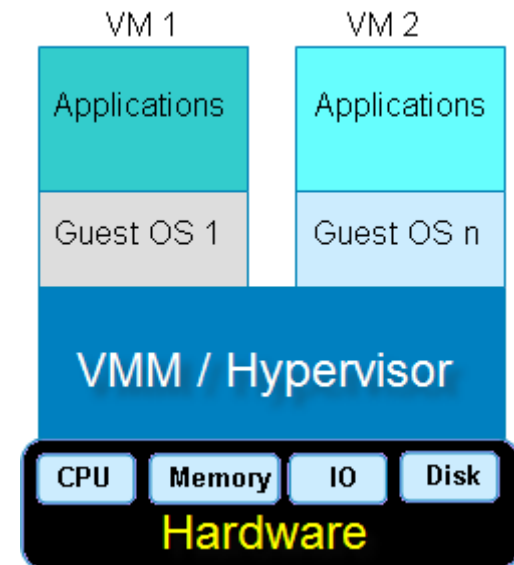
- OS 위에서 실행되는 하이퍼바이저
- VMM inside the Host OS
 - Kernel-mode driver
- Multiple Guest OS support
- VMM emulates hardware for guest OSs
- Systems
 - Microsoft Virtual Server,
 - VMware Workstation & Server
 - Oracle VirtualBox



1.가상화? (Cont.)

Bare-metal Hypervisor (Type-1)

- VMM runs on 'bare metal'
- 하드웨어 바로 위에 실행되는 Hypervisor
- 리소스 제어와 격리성을 제공함
- Examples
 - VMware ESX, Microsoft Hyper-V
 - Citrix XenServer



1. 가상화? (Cont.)

Bare-metal Hypervisor : Para-Virtualization

- Guest OS modified to make calls ("hypercalls") to or receive events from VMM
 - Xen은 반가상화(Para-virtualization) 기법을 사용하는데, 운영체제가 직접 하드웨어를 제어하지 않고 중간 레이어인 Hypervisor 에게 요청하고 하이퍼바이저는 하드웨어에 요청해서 응답받는 수직 구조를 가지고 있다.
 - 전가상화 방식에 비해서 호출이 적게 일어나기 때문에 성능은 상대적으로 높다. 그러나 Xen위에 동작하는 운영체제는 커널의 일부 소스가 수정되어야 하는 단점이 있고, Xen이 보편화되어서 수정된 운영체제를 쉽게 구할 수 있다.
- Systems
 - Xen
 - Guest OS: XenoLinux, NetBSD, FreeBSD, Solaris 10, Windows (in progress)

1. 가상화? (Cont.)

Bare-metal Hypervisor : Full Virtualization

- Guest OS unmodified

- 전가상화(Full-Virtualization) 기법을 사용하면, 운영체제의 커널 수정없이 에뮬레이터 돌리듯이 작동 시킬 수 있다.
- 운영체제가 하드웨어(CPU의 Virtualization Technology)에 요청을 하면, 중간 레이어인 hypervisor에 하드웨어 제어 요구를 하고 hypervisor는 다시 하드웨어에 요구하는 Z 방식의 호출 구조가 성능을 떨어뜨릴 수 있다.

- Examples

- VMware ESX, Microsoft Hyper-V
- IBM z/VM
- Linux KVM (Kernel VM)

