


CSE


Extended OS



CSE

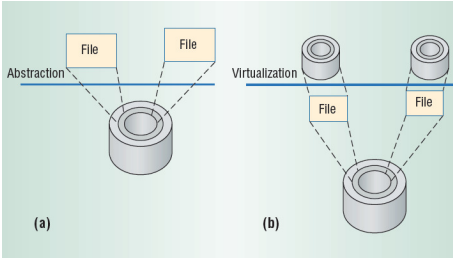
Virtual Machines

References:
 Smith, J.E.; Ravi Nair; , "The architecture of virtual machines,"
Computer , vol.38, no.5, pp. 32- 38, May 2005




CSE

Abstraction & Virtualisation

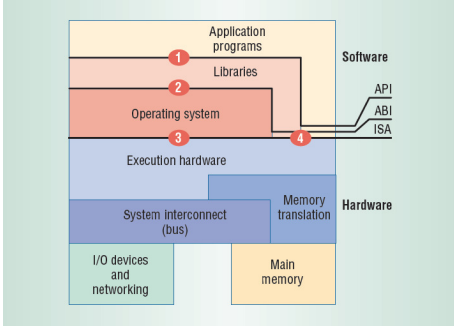



(a) (b)



CSE


Interface Levels

CSE

Instruction Set Architecture


- Interface between software and hardware
- Divided between privileged and un-privileged parts



CSE

Application Binary Interface

- Interface between programs hardware + OS
- Consists of system call interface + un-privileged ISA



Application Programming Interface

- Interface between programs hardware + OS
- Consists of library calls + un-privileged ISA
 - Syscalls usually called through library.

Process versus System Virtual Machine

OS is an extended virtual machine

- Multiplexes the "machine" between applications
 - Time sharing, multitasking, batching
- Provided a higher-level machine for
 - Ease of use
 - Portability
 - Efficiency
 - Security
 - Etc....

JAVA – Higher-level Virtual Machine

- write a program once, and run it anywhere
 - Architecture independent
 - Operating System independent
- Language itself was clean, robust, garbage collection
- Program compiled into bytecode
 - Interpreted or just-in-time compiled.
 - Lower than native performance

JIT

A → IA-32

Conventional versus Emulation/Translation

CSE

Issues

- Legacy applications
- No isolation nor resource management between applets
- Security
 - Trust JVM implementation? Trust underlying OS?
- Performance compared to native

THE UNIVERSITY OF NEW SOUTH WALES

CSE

Is the OS the “right” level of extended machine?

- Security
 - Trust the underlying OS?
- Legacy application and OSs
- Resource management of existing systems suitable for all applications?
- What about activities requiring “root” privileges

THE UNIVERSITY OF NEW SOUTH WALES

CSE

Virtual Machine Monitors

- Provide scheduling and resource management
- Extended “machine” is the actual machine interface.

THE UNIVERSITY OF NEW SOUTH WALES

CSE

IBM VM/370

The diagram illustrates the IBM VM/370 architecture. At the top, 'Virtual 370s' are shown as a layer above three 'CMS' (Control Macro System) instances. Below the CMS instances is the 'VM/370' layer, and at the bottom is the '370 Bare hardware'. Annotations indicate 'System calls here' at the top right, 'I/O instructions here' and 'Trap here' on the left side, and another 'Trap here' on the right side.

THE UNIVERSITY OF NEW SOUTH WALES

CSE

Advantages

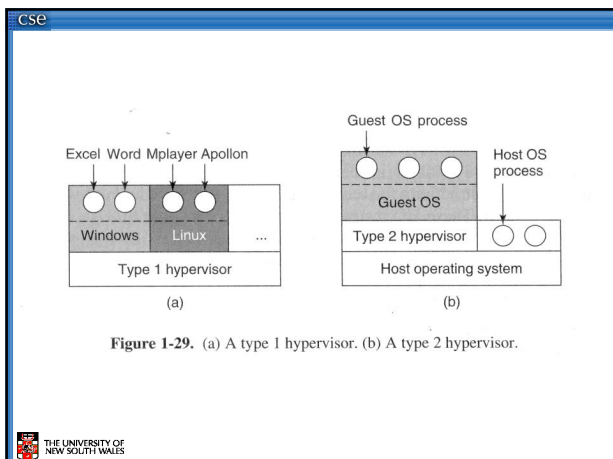
- Legacy OSes (and applications)
- Server consolidation
- Concurrent OSes
 - Linux – Windows
 - Primary – Backup
 - High availability
- Test and Development
- Security
 - VMM (hopefully) small and correct
- Performance near bare hardware
 - For some applications

THE UNIVERSITY OF NEW SOUTH WALES

CSE

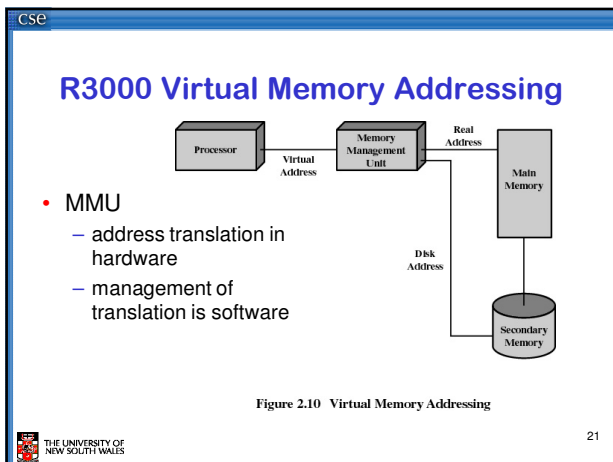
The diagram classifies VMs into two main categories: Process VMs and System VMs, separated by a vertical dashed line. Process VMs are further divided into 'Same ISA' (Multiprogrammed systems) and 'Different ISA' (Dynamic translators). System VMs are divided into 'Same ISA' (Classic system VMs) and 'Different ISA' (Whole-system VMs). Below these are implementation types: 'Same-ISA dynamic binary optimizers' and 'High-level-language VMs' under Process VMs; and 'Hosted VMs' and 'Codesigned VMs' under System VMs.

THE UNIVERSITY OF NEW SOUTH WALES



Virtual R3000???

- Interpret
 - System/161
 - slow
 - JIT dynamic compilation
- Run on the real hardware??



R3000 Address Space Layout

	0xFFFFFFFF	kseg2
	0xC0000000	
	0xA0000000	kseg1
	0x80000000	kseg0
		kuseg
	0x00000000	

- kuseg:
 - 2 gigabytes
 - MMU translated
 - Cacheable
 - user-mode and kernel mode accessible

R3000 Address Space Layout

	0xffffffff	kseg2
	0xc0000000	
	0xa0000000	kseg1
	0x80000000	kseg0
		kuseg
	0x00000000	

- kseg0:
 - 512 megabytes
 - Fixed translation window to physical memory
 - 0x80000000 - 0x9ffffff virtual = 0x00000000 - 0x1ffffff physical
 - MMU not used
 - Cacheable
 - Only kernel-mode accessible
 - Usually where the kernel code is placed

Physical Memory

R3000 Address Space Layout

	0xffffffff	kseg2
	0xc0000000	
	0xa0000000	kseg1
	0x80000000	kseg0
		kuseg
	0x00000000	

- kseg1:
 - 512 megabytes
 - Fixed translation window to physical memory
 - 0xa0000000 - 0xbffffff virtual = 0x00000000 - 0x1ffffff physical
 - MMU not used
 - NOT cacheable
 - Only kernel-mode accessible
 - Where devices are accessed (and boot ROM)

Physical Memory

R3000 Address Space Layout

- kseg2:
 - 1024 megabytes
 - MMU translated
 - Cacheable
 - Only kernel-mode accessible

0xEFFFFFFF	kseg2
0xC0000000	kseg1
0xA0000000	kseg0
0x80000000	
0x00000000	kuseg

Issues

- Privileged registers (CP0)
- Privileged instructions
- Address Spaces
- Exceptions (including syscalls, interrupts)
- Devices

Blank slide with CSE logo.

Handwritten notes:

- mfco
- mfco
- mfco r1, CO.EPC
- CP0 Register (cb, ca, ca)
- VM
- Linux user mode

Handwritten diagram showing a vertical stack of three boxes:

- VMM
- Linux
- Apps

Blank slide with CSE logo.

