ie
SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

OPERATING SYSTEMS & PARALLEL COMPUTING

Operating system principles &
computer architecture

SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Initial Objectives

- To describe the basic organization of computer systems and operating systems.
- To give an overview of the many types of computing environments.
- To explore varied types of operating systems.
- To provide a grand tour of the major components of operating systems.
- To describe the services an operating system provides to users, processes, and other systems.
- To discuss the various ways of structuring an operating system.

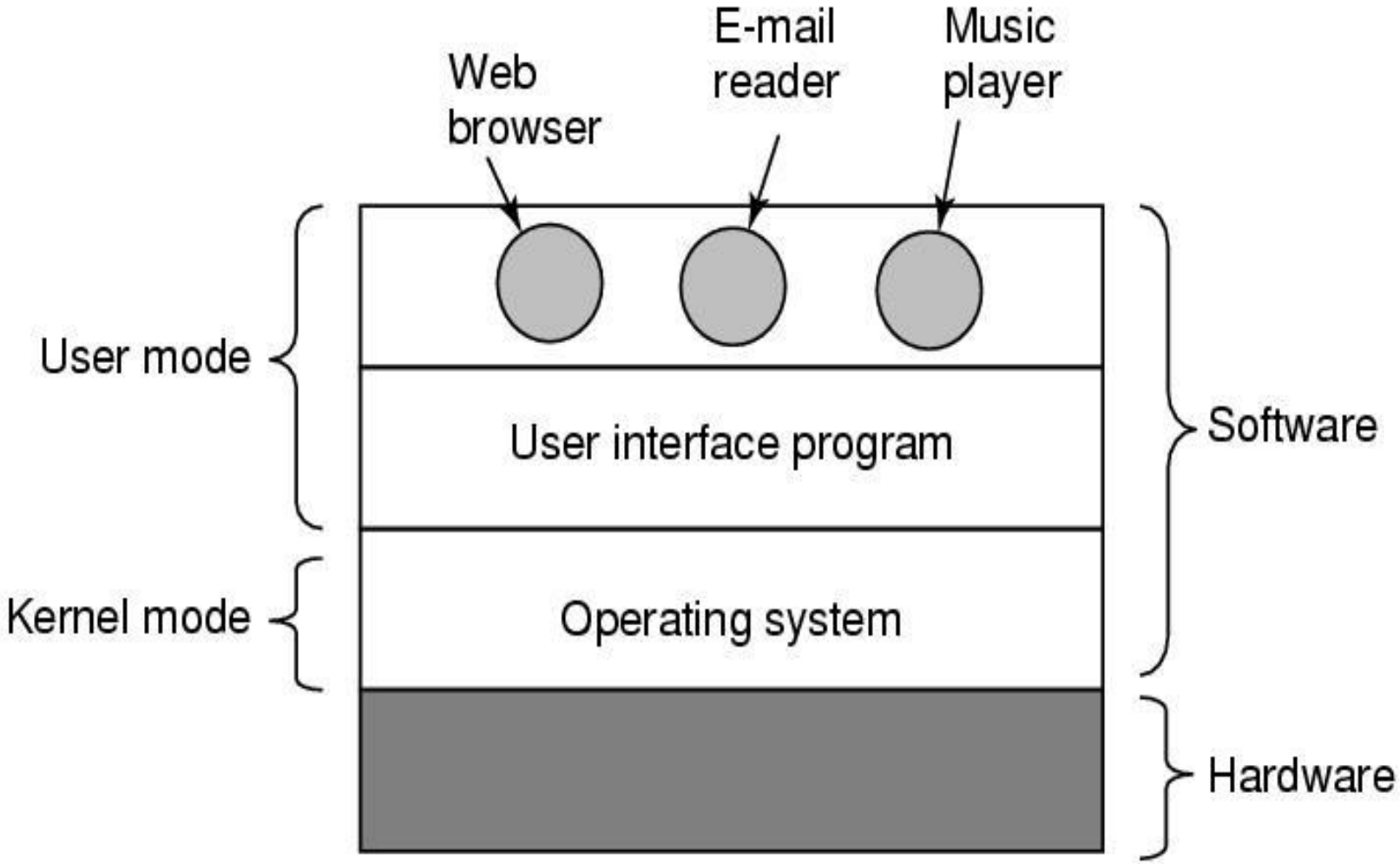
What is an Operating System (1)?

- A modern computer consists of:
  - One or more processors
  - Main memory
  - Disks
  - Printers
  - Various input/output devices.
- Managing all these varied components requires a layer of software – the **Operating System (OS).**

What is an Operating System (2)?

- An Operating System is a program that acts as an intermediary/interface between a user of a computer and the computer hardware.
- OS goals:
  - Control/execute user/application programs.
  - Make the computer system convenient to use.
  - Ease the solving of user problems.
  - Use the computer hardware in an efficient manner.

Where does the OS fit in?



ie
SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Services provided by an OS

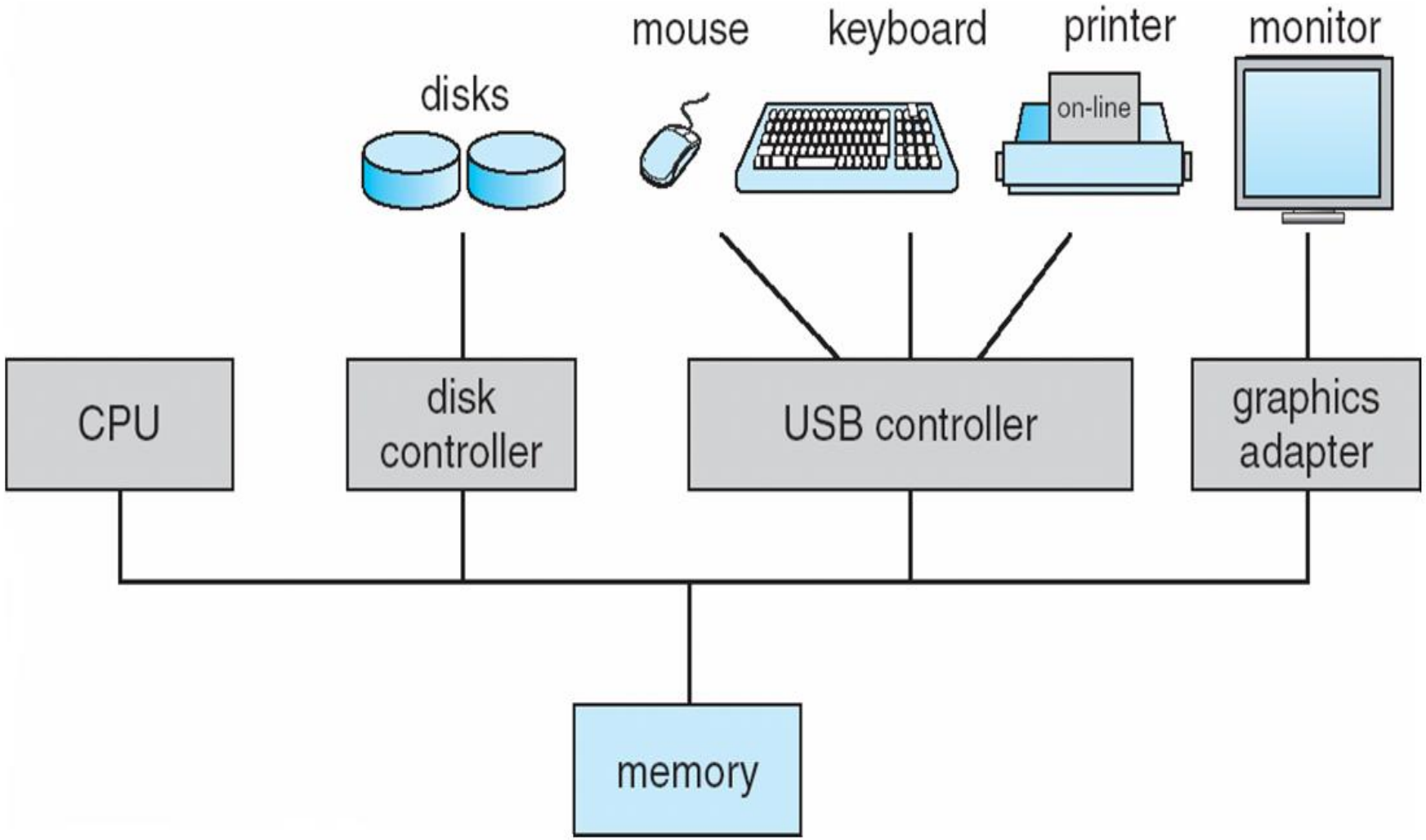
- Facilities for program creation
  - editors, compilers, linkers, debuggers, etc.
- Program execution
  - loading in memory, I/O and file initialization.
- Access to I/O and files
  - deals with the specifics of I/O and file formats.
- System access
  - resolves conflicts for resource contention.
  - protection in access to resources and data.

ie
SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Why are Operating Systems Important?

- Important to understand and know how to correctly use when writing user applications.
- Large and complex systems that have a high economic impact and result in interesting problems of management.
- Few actually involved in OS design and implementation but nevertheless many general techniques to be learned and applied.
- Combines concepts from many other areas of Computer Science: Architecture, Languages, Data Structures, Algorithms, etc.

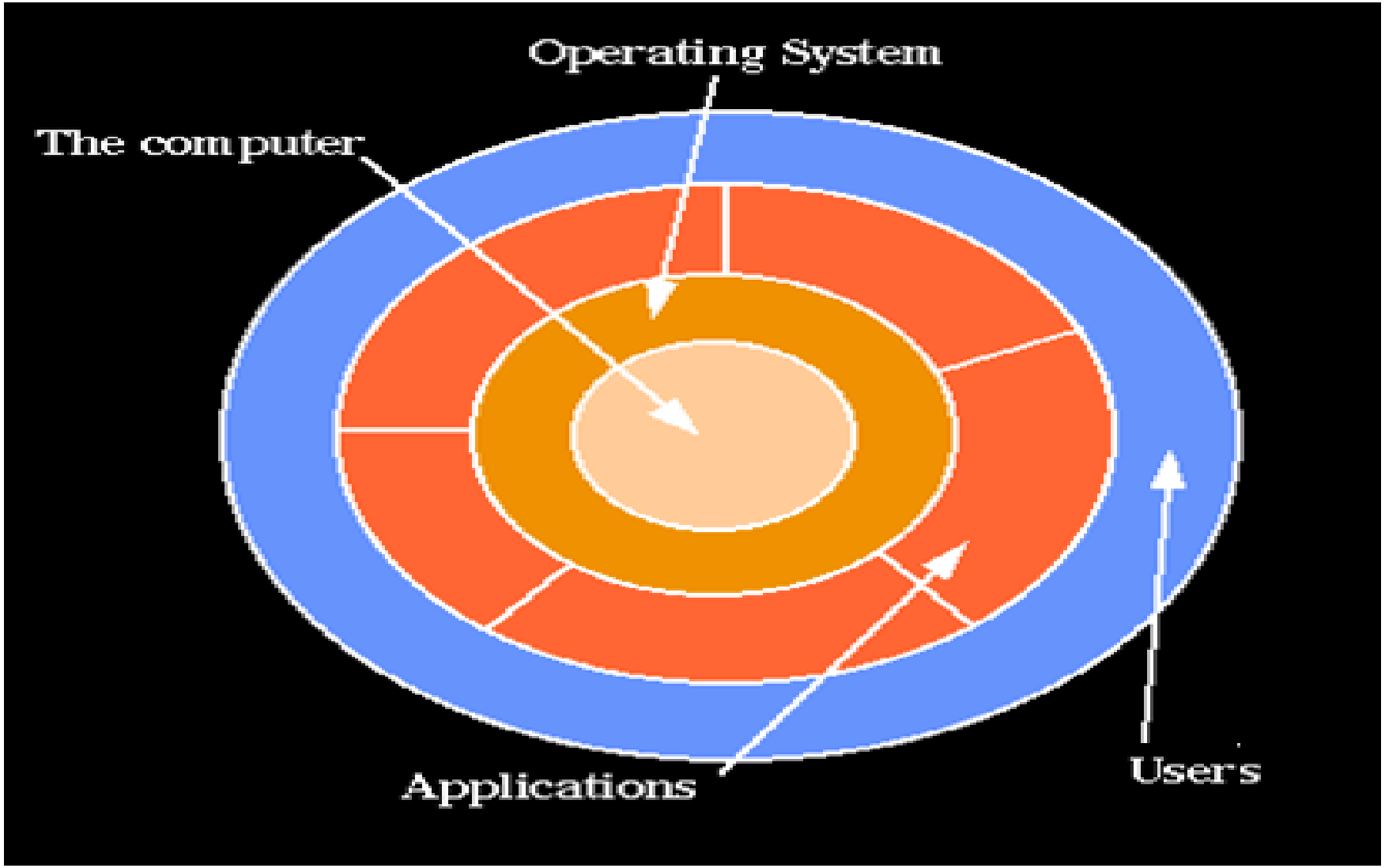
Computer Hardware Organization



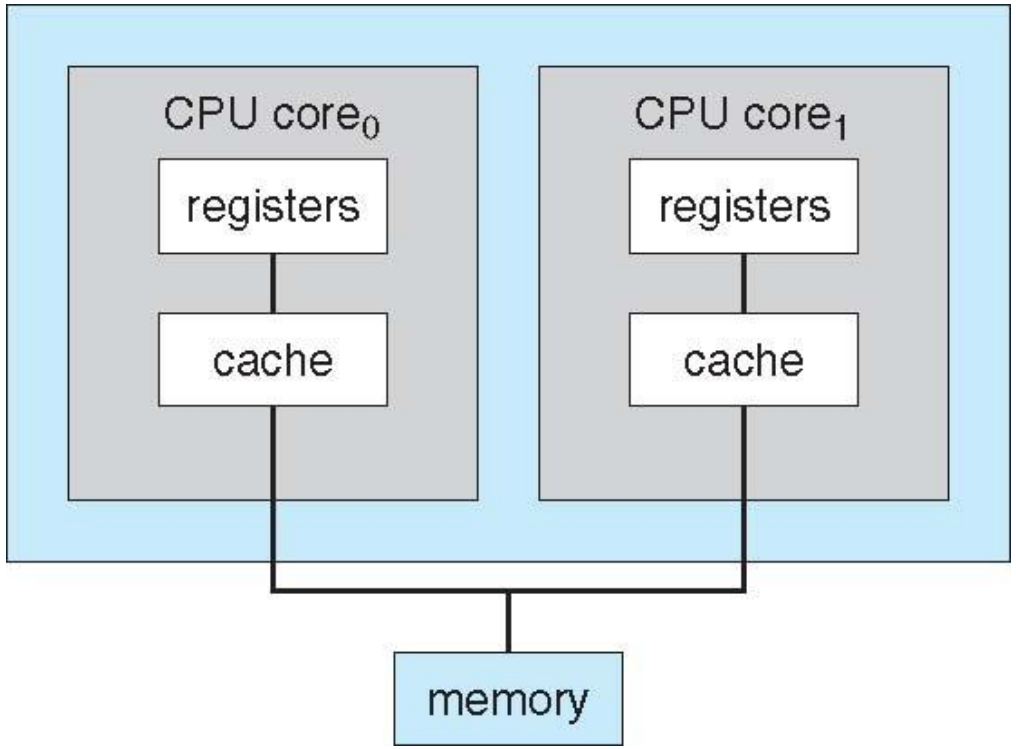
Computer System Components

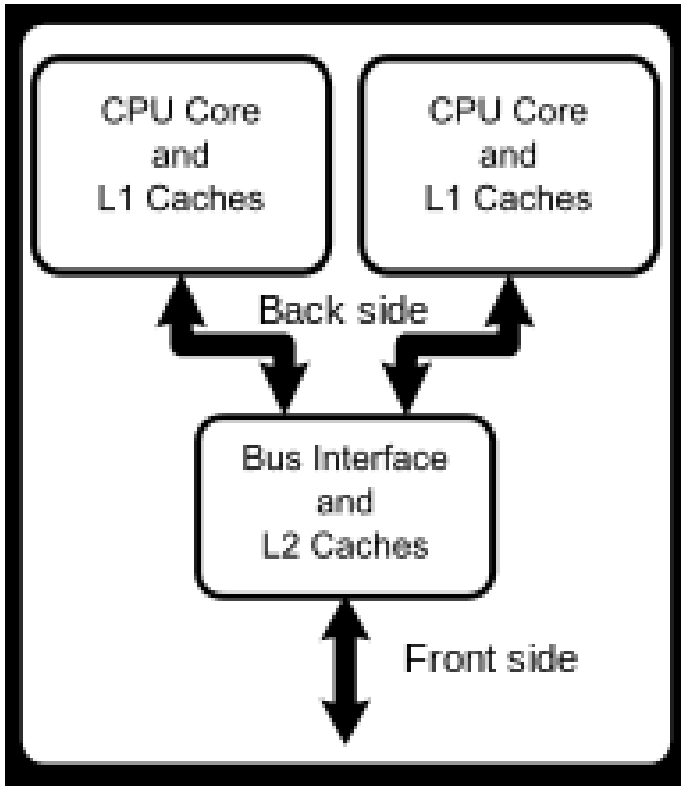
1. Hardware – provides basic computing resources (CPU, Memory, I/O devices, Communication).
2. Operating System – controls and coordinates use of the hardware among various application programs for various users.
3. System & Application Programs – ways in which the system resources are used to solve computing problems of the users (Word processors, Compilers, Web browsers, Database systems, Video games).
4. Users – (People, Machines, other computers).

Hierarchical view of computer system

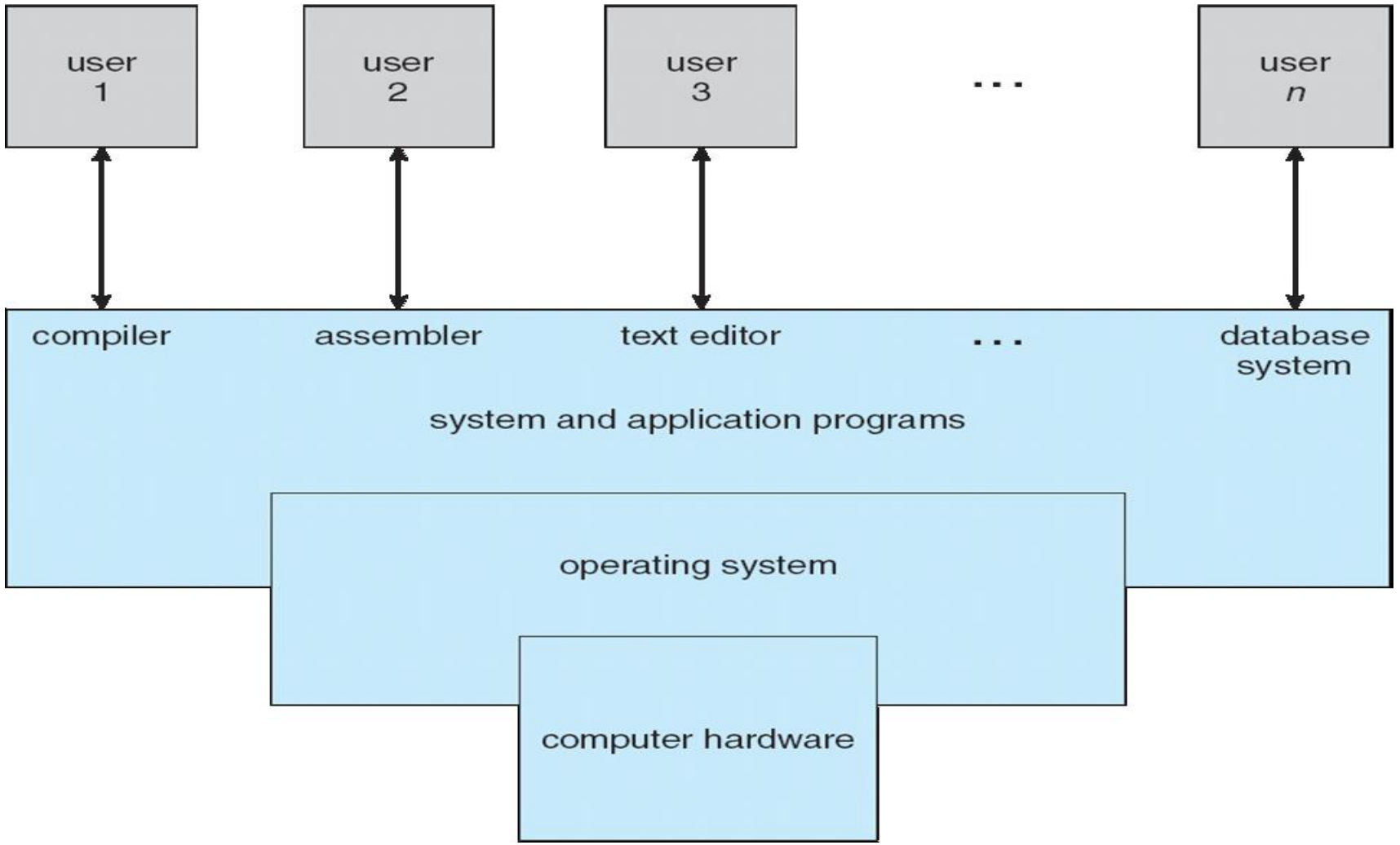


A Dual-Core Design

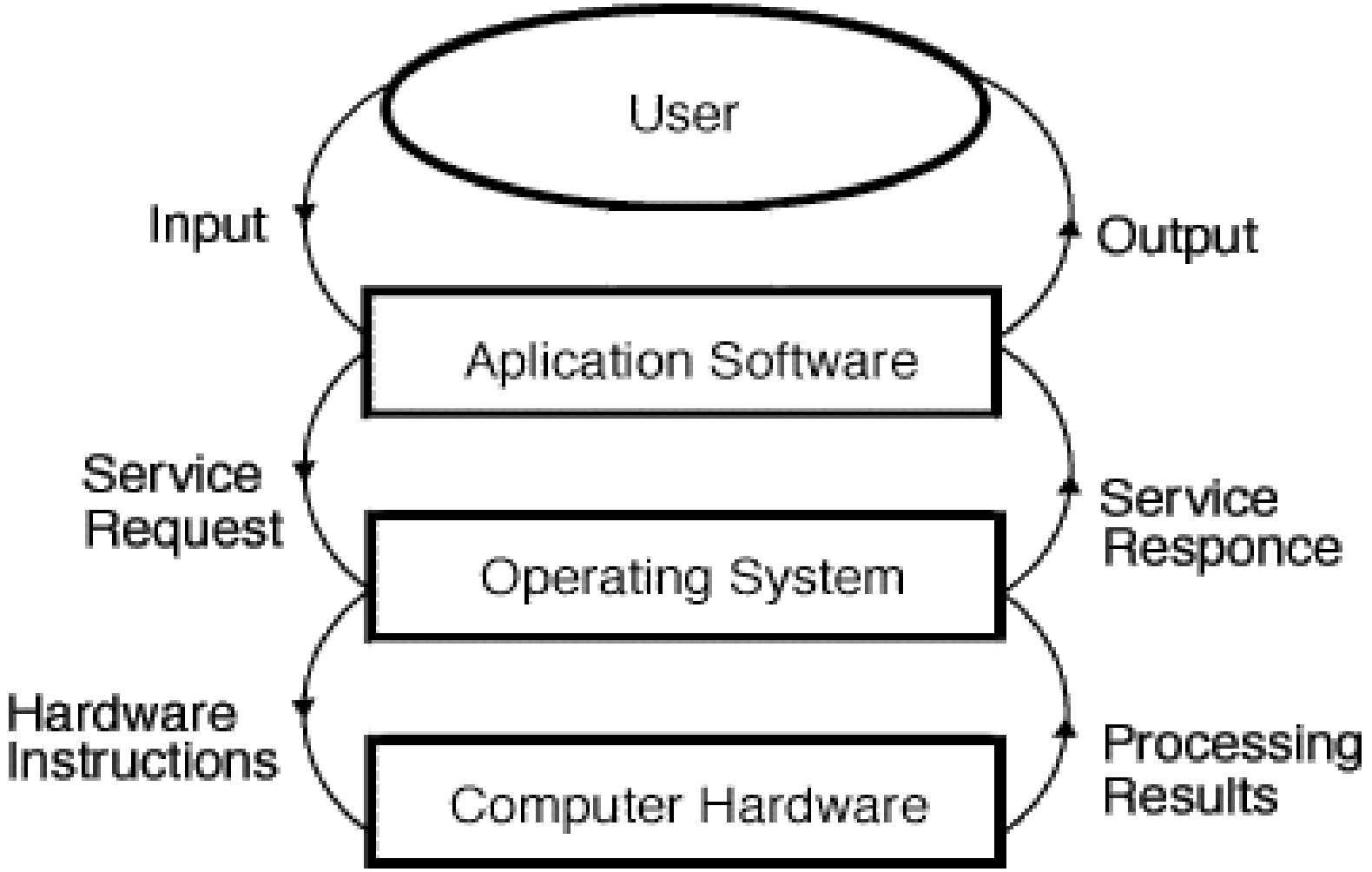




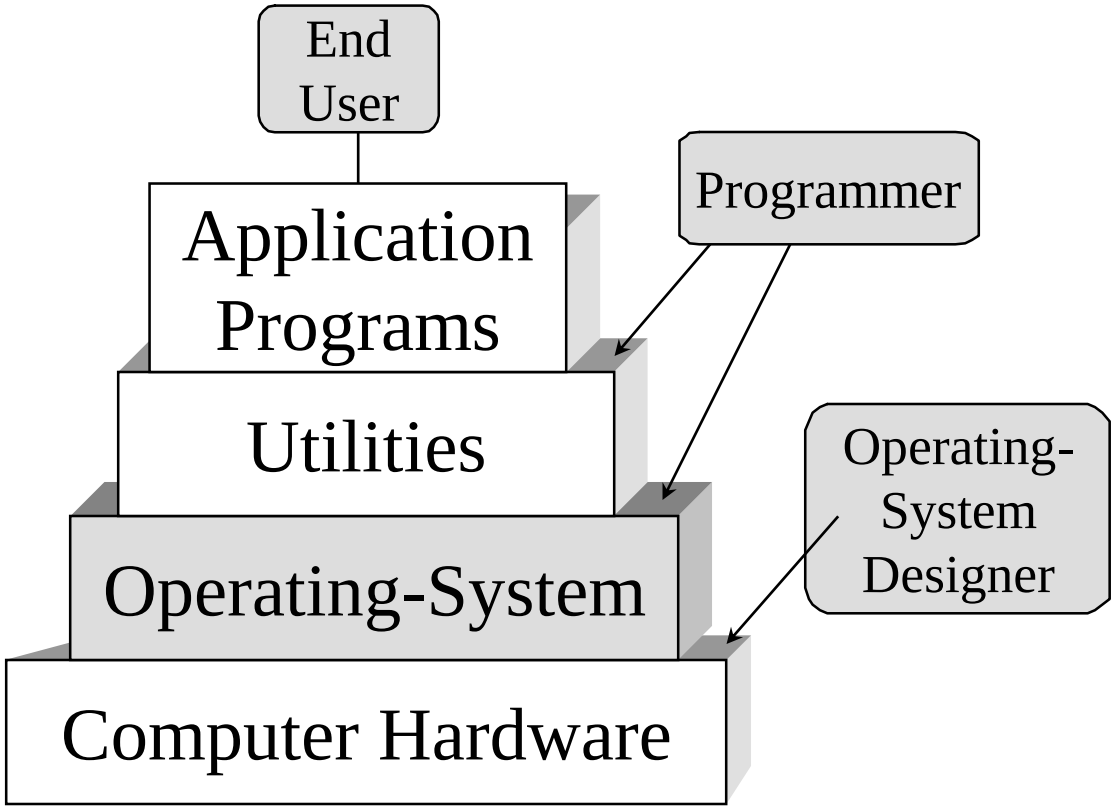
Static View of System Components



Dynamic View of System Components



Layers of a Computer System



What Operating Systems Do?

- Depends on the point of view.
- Users want convenience, ease of use and good performance
  - Don't care about resource utilization.
- But a shared computer such as mainframe or minicomputer must keep all users happy.
- Users of dedicate systems such as workstations have dedicated resources but frequently use shared resources from servers.
- Handheld computers are resource poor, optimized for usability and battery life.
- Some computers have little or no user interface, such as embedded computers in devices and automobiles.

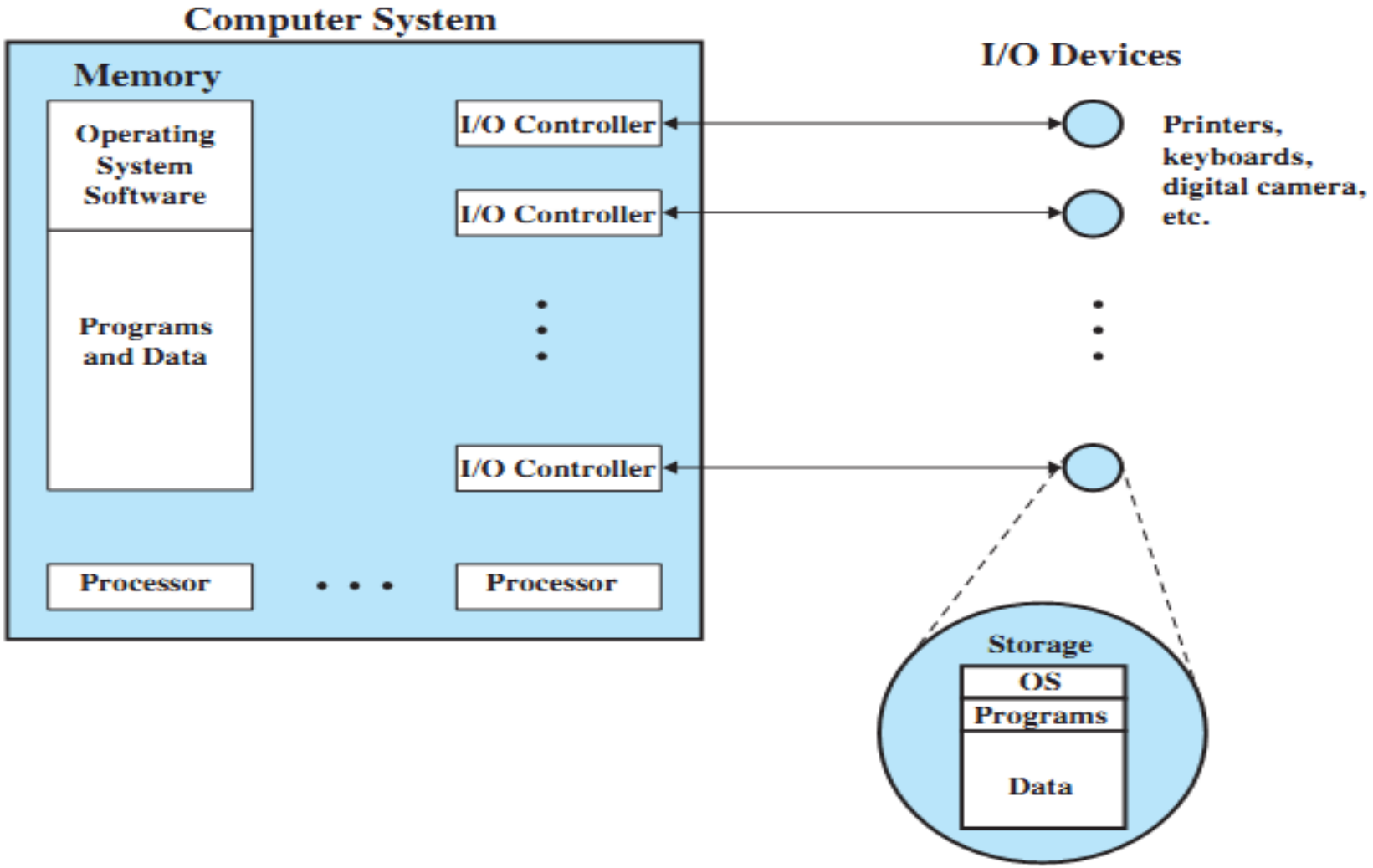
Views of an Operating System

- There are three classical views (in literature):
  1. Resource Manager – manages and allocates resources.
  2. Control program – controls the execution of user programs and operations of I/O devices.
  3. Command Executer – Provides an environment for running user commands.
- But one more modern view: the Operating System as a Virtual Machine.

1. Resource Manager

- Resource Manager:
  - Manages and protects multiple computer resources: CPU, Processes, Internal/External memory, Tasks, Applications, Users, Communication channels, etc…
  - Handles and allocates resources to multiple users or multiple programs running at the same time and space (e.g., processor time, memory, I/O devices).
  - Decides between conflicting requests for efficient and fair resource use (e.g., maximize throughput, minimize response time).
- Sort of a bottom-up view.

OS as a Resource Manager

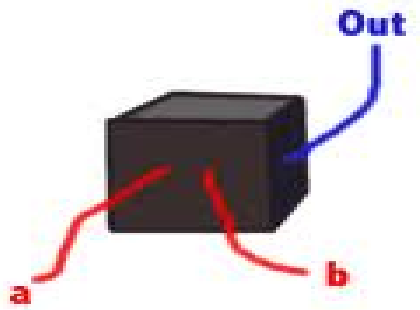


Resource Manager oriented OS names

- DEC RSX – Resource Sharing eXecutive
- MIT Multics – MULTiplexed Information and Computing Services
- IBM MFT/MVT – Multiple Fixed/Variable Tasks
- IBM MVS – Multiple Virtual Storage
- DEC VMS – Virtual Memory System
- MVS TSO – Time Sharing Option
- CTSS – Compatible Time Sharing System
- IBM VM – Virtual machine

2. Control Program

- Control Program:
  - Manages all the components of a complex computer system in an integrated manner.
  - Controls the execution of user programs and I/O devices to prevent errors and improper use of computer resources.
  - Looks over and protects the computer: Monitor, Supervisor, Executive, Controller, Master, Coordinator ….
- Sort of a black box view.



Control program oriented OS names

- Unisys MCP – Master Control Program
- DR CP/M – Control Program/Microcomputer
- IBM VM/CP – VM Control Program
- IBM AIX – Advanced Interactive eXecutive
- DEC RSX – Resource Sharing eXecutive

SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

3. Command Executer

- Command Executer:
  - Interfaces between the users and machine.
  - Supplies services/utilities to users.
  - Provides the users with a convenient CLI (Command Language Interface), also called a Shell (in UNIX), for entering the user commands.
- Sort of a top-down view.

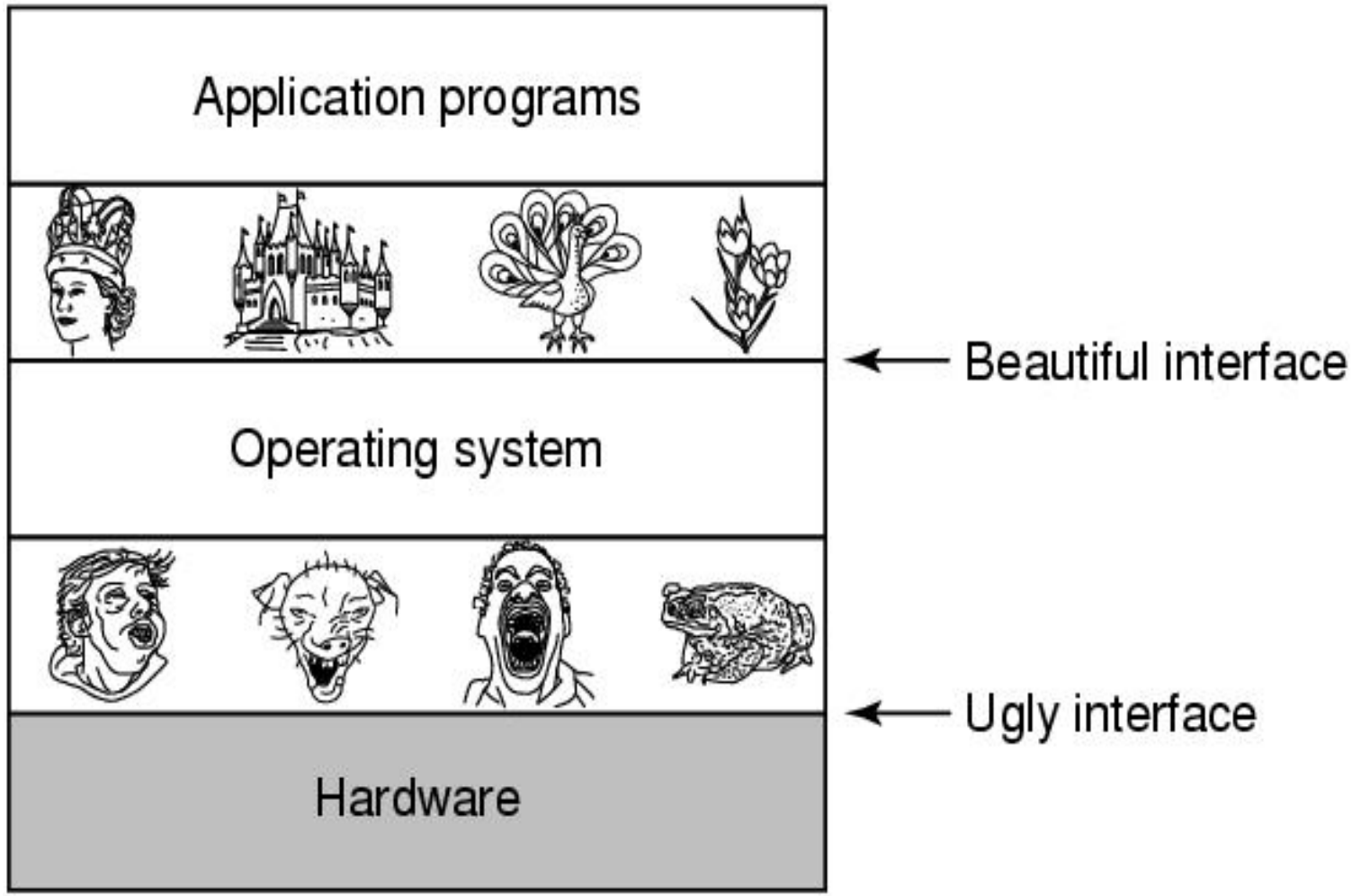
Command Executer oriented OS names

- IBM AIX – Advanced Interactive Executive
- IBM VM/CMS – Conversational monitor System

Modern view: Virtual Machine (1)

- Operating System as a Virtual Machine:
  - An interface between the user and hardware that hides the details of the hardware (e.g., I/O).
  - Constructs higher-level (virtual) resources out of lower-level (physical) resources (e.g., files).
  - **Definition**: OS is a collection of software enhancements, executed on the bare hardware, culminating in a high-level virtual machine that serves as an advanced programming environment.
    - virtual machine = software enhancement = extended machine = abstract machine = layer = level = ring.

Modern view: Virtual Machine (2)



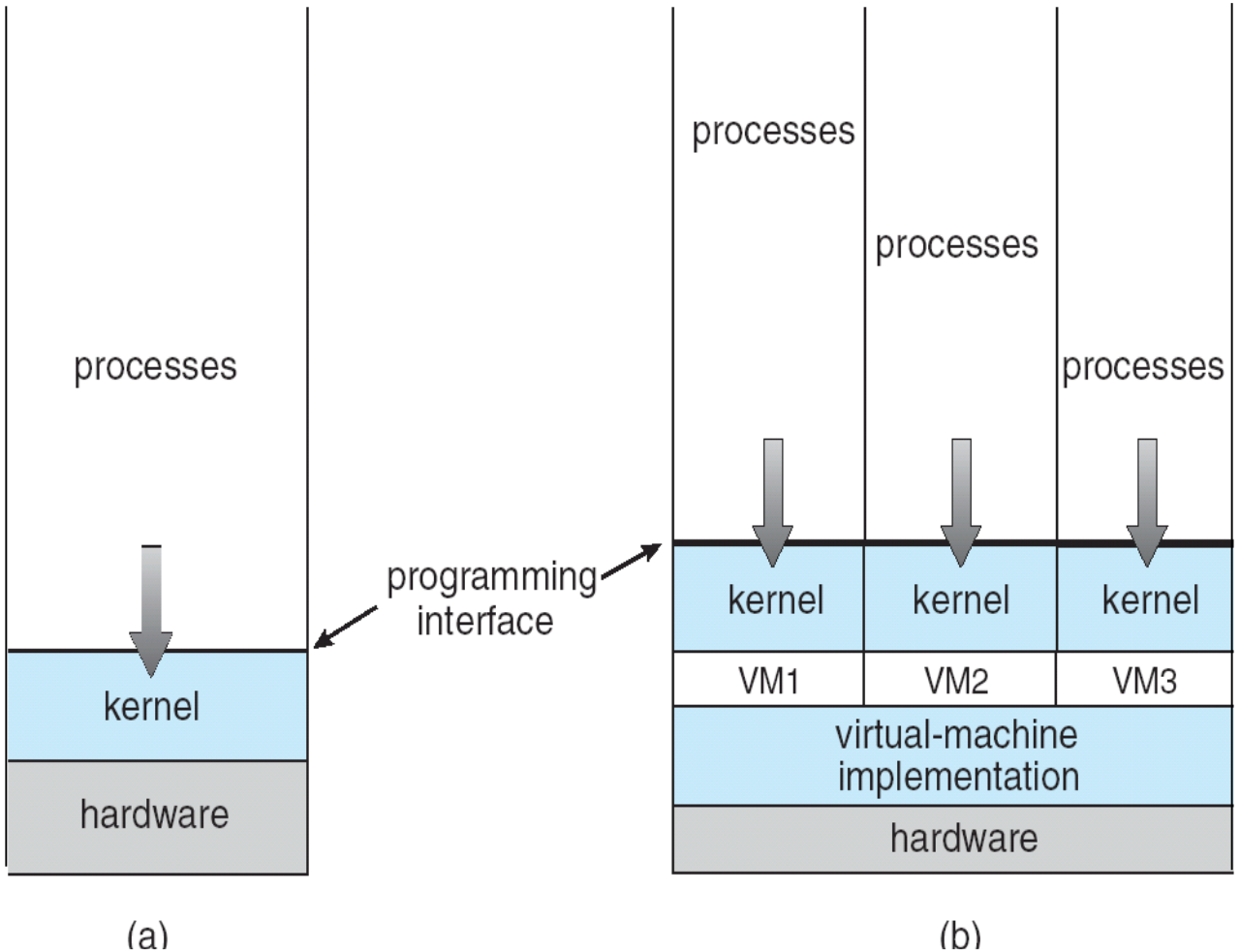
Virtual Machines (1)

- A Virtual Machine (VM) takes the layered and microkernel approach to its logical conclusion.
- It treats hardware and the operating system kernel as though they were all hardware.
- A virtual machine provides an interface identical to the underlying bare hardware.
- The operating system host creates the illusion that a process has its own processor and (virtual memory).
- Each guest provided with a (virtual) copy of underlying computer.

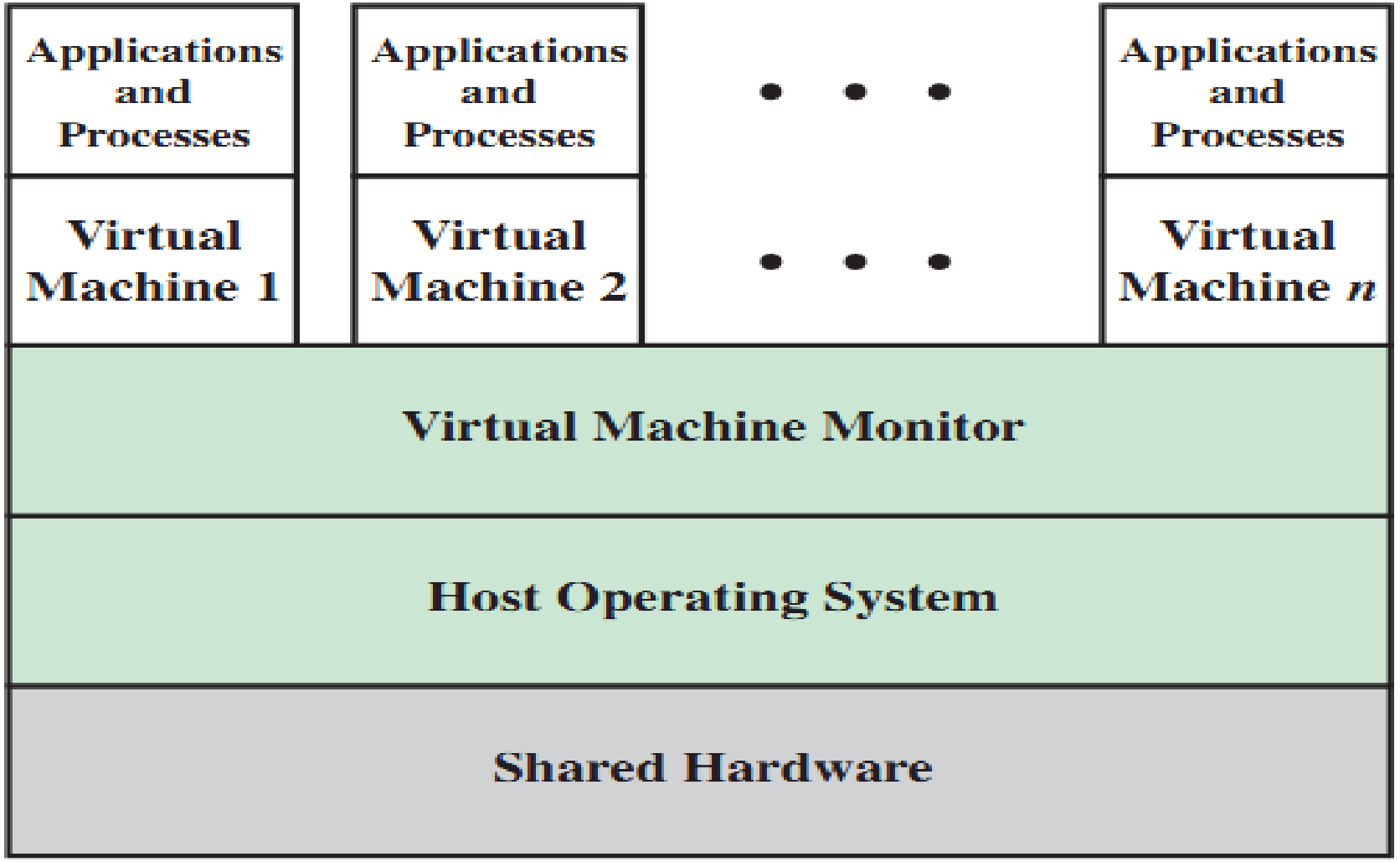
Virtual Machines (2)

- The resources of the physical computer are shared to create the virtual machines:
  - CPU scheduling can create the appearance that users have their own processor.
  - Spooling and a file system can provide virtual card readers and virtual line printers.
  - A normal user time-sharing terminal serves as the virtual machine operator's console.

on Bare Machine Implementation VM



VM Implementation on Host OS

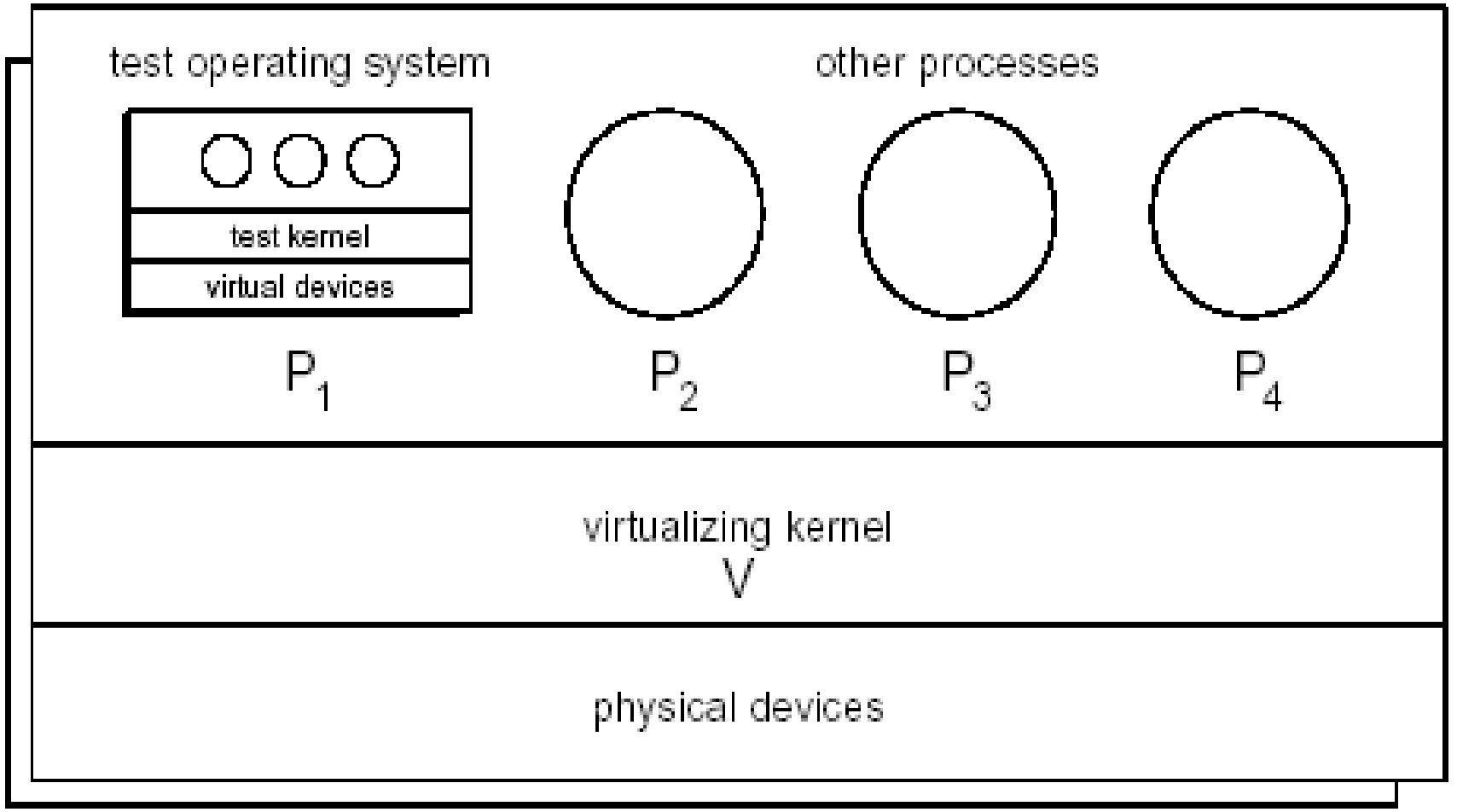


ie
SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

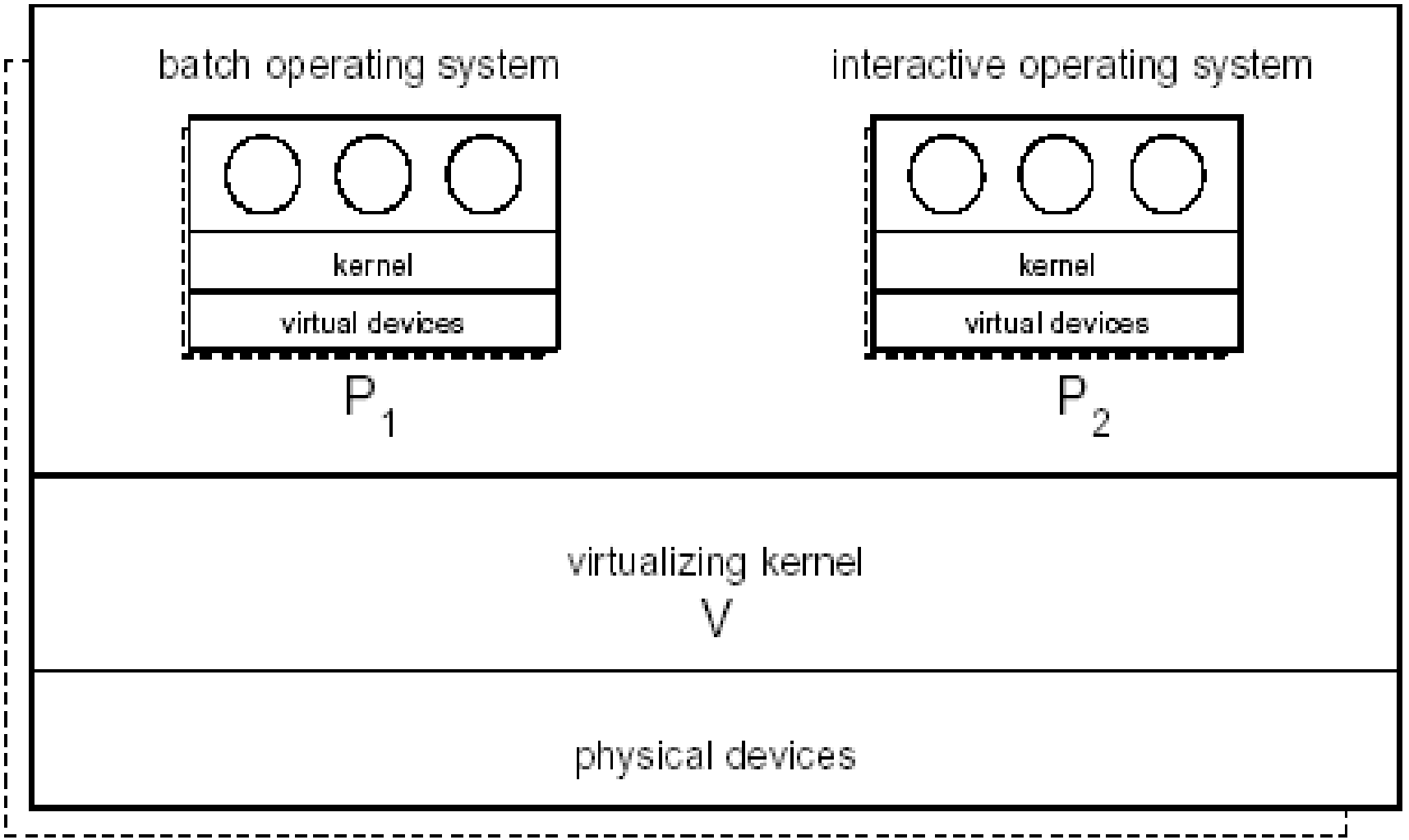
Advantages/Disadvantages of VMs

- The VM concept provides complete protection of system resources since each virtual machine is isolated from all other virtual machines. This isolation permits no direct sharing of resources.
- A VM system is a perfect vehicle for OS research and development. System development is done on the virtual machine, instead of on a physical machine and so does not disrupt normal system operation.
- The VM concept is difficult to implement due to the effort required to provide an exact duplicate to the underlying machine.

Testing a new Operating System



Integrating two Operating Systems



Virtual Machines History and Benefits

- First appeared commercially in IBM mainframes in 1972.
- Fundamentally, multiple protected execution environments (different operating systems) can share the same hardware.
- Protect from each other.
- Some sharing of file can be permitted, controlled.
- Commutate with each other, other physical systems via networking.
- Useful for development and testing.
- Consolidation of many low-resource use systems onto fewer busier systems.
- “Open Virtual Machine Format”, standard format of VMs, allows a VM to run within many different VM (host) platforms.

Emulation vs. Virtualization

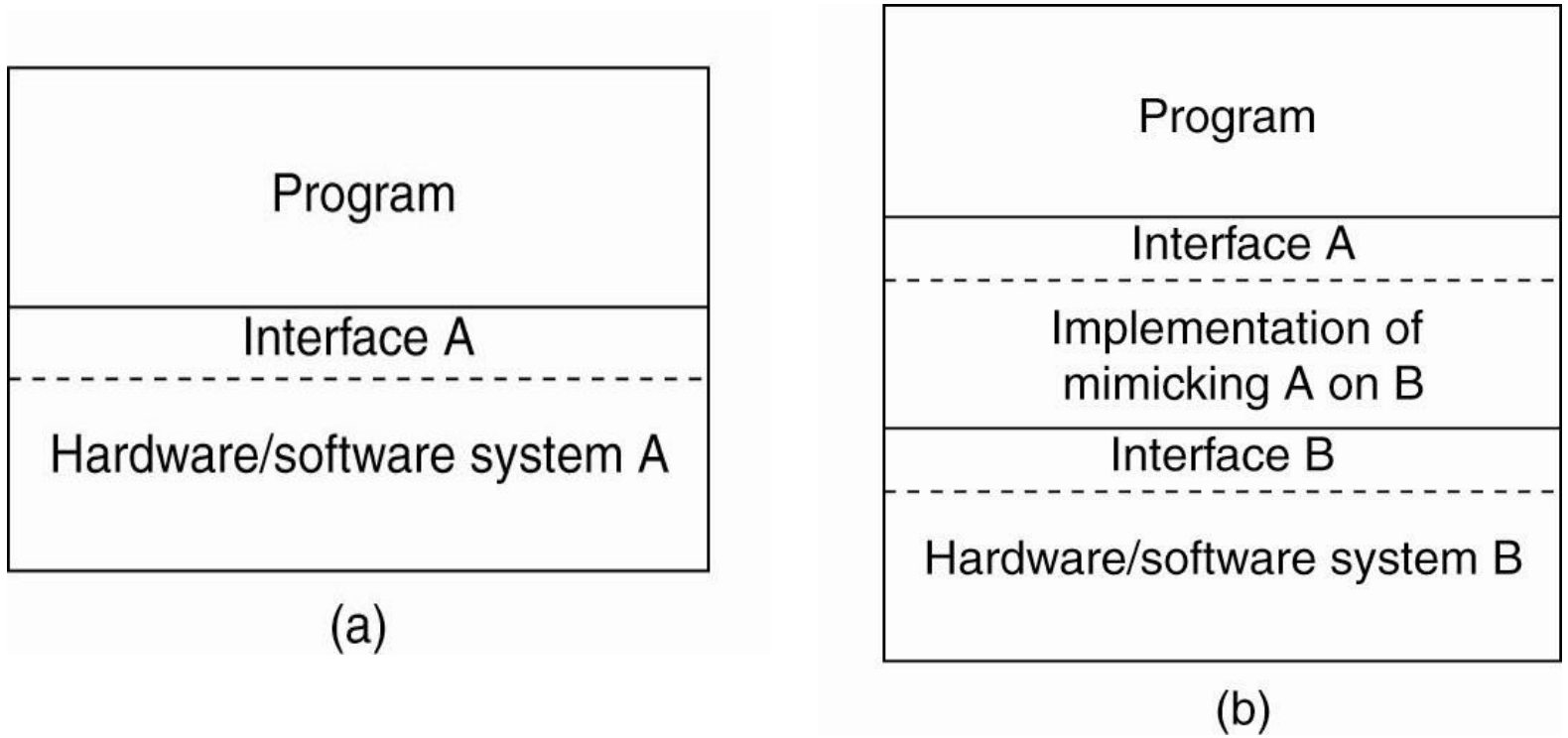
- Emulation – when source CPU type different from target type (i.e., PowerPC to Intel x86):
  - Generally slowest method.
  - When computer language not compiled to native code – Interpretation.
- Virtualization – OS natively compiled for CPU, running guest OSes also natively compiled:
  - Consider VMware running WinXP guests, each running applications, all on native WinXP host OS.
  - VMM (virtual machine Manager) provides virtualization services.

ie
SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Virtualization Examples

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility:
  - Apple laptop running Mac OS X host, Windows as a guest.
  - Developing apps for multiple OSes without having multiple systems.
  - QA testing applications without having multiple systems.
  - Executing and managing compute environments within data centers.
- VMM can run natively, so they are also the host:
  - There is no general purpose host then (VMware ESX and Citrix XenServer).

The Role of Virtualization



(a) General organization between a program, interface, and system.

(b) General organization of virtualizing system A on top of system B.

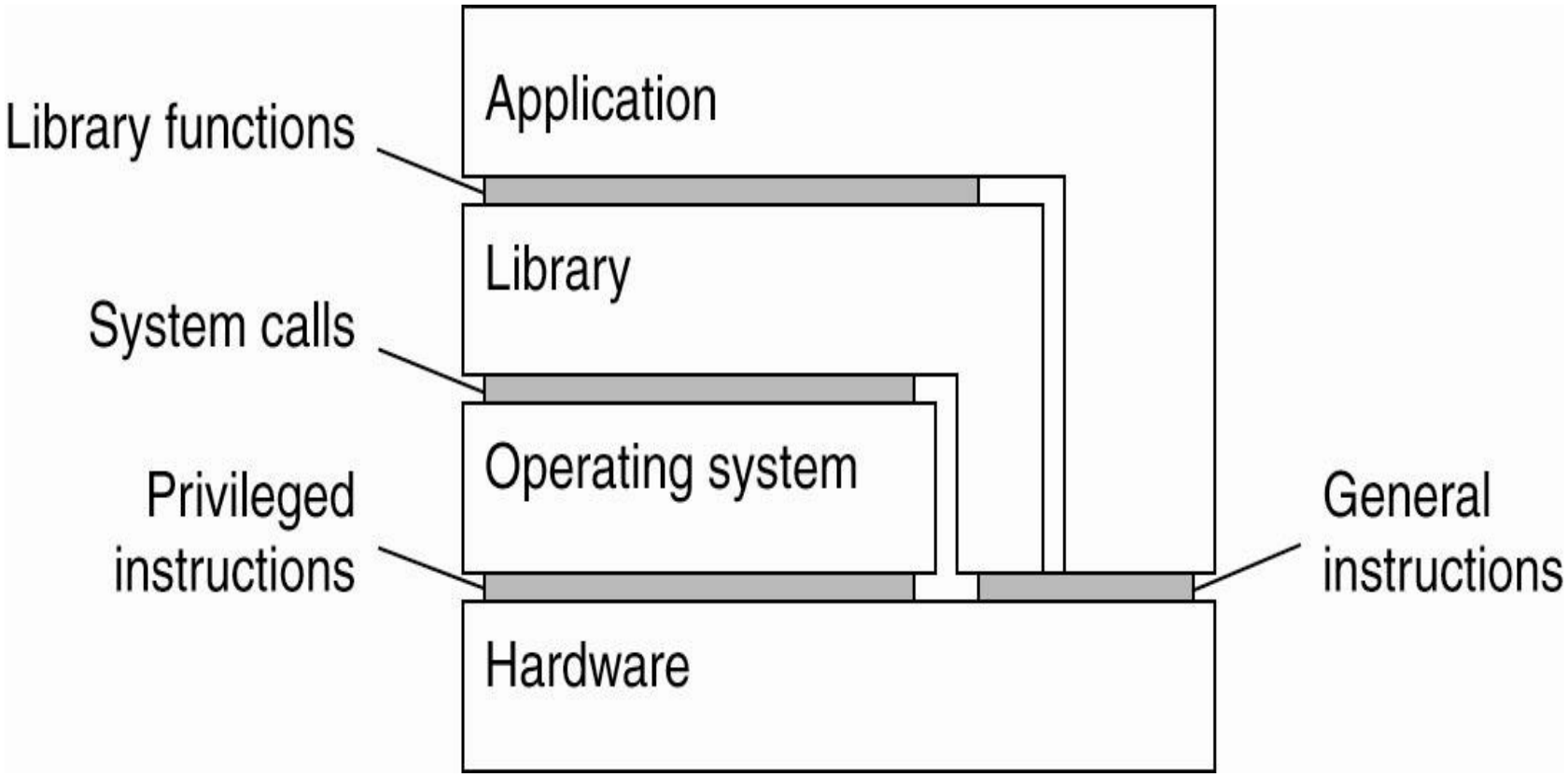
Architectures of Virtual Machines (1)

- There are interfaces at different levels.
- An interface between the hardware and software, consisting of machine instructions
  - that can be invoked by any program.
- An interface between the hardware and software, consisting of machine instructions
  - that can be invoked only by privileged programs, such as an operating system.

Architectures of Virtual Machines (2)

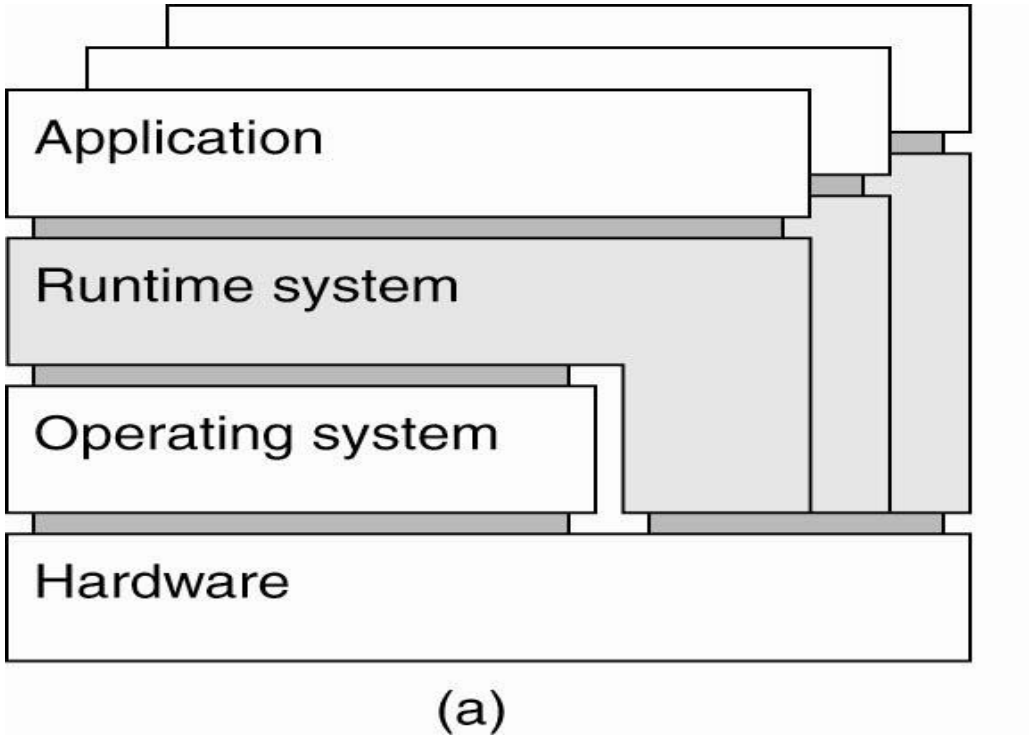
- An interface consisting of system calls as offered by an operating system.
- An interface consisting of library calls:
  - generally forming what is known as an Application Programming Interface (API).
  - In many cases, the aforementioned system calls are hidden by an API.

Architectures of Virtual Machines (3)



Various interfaces offered by computer systems

Process Virtual Machine

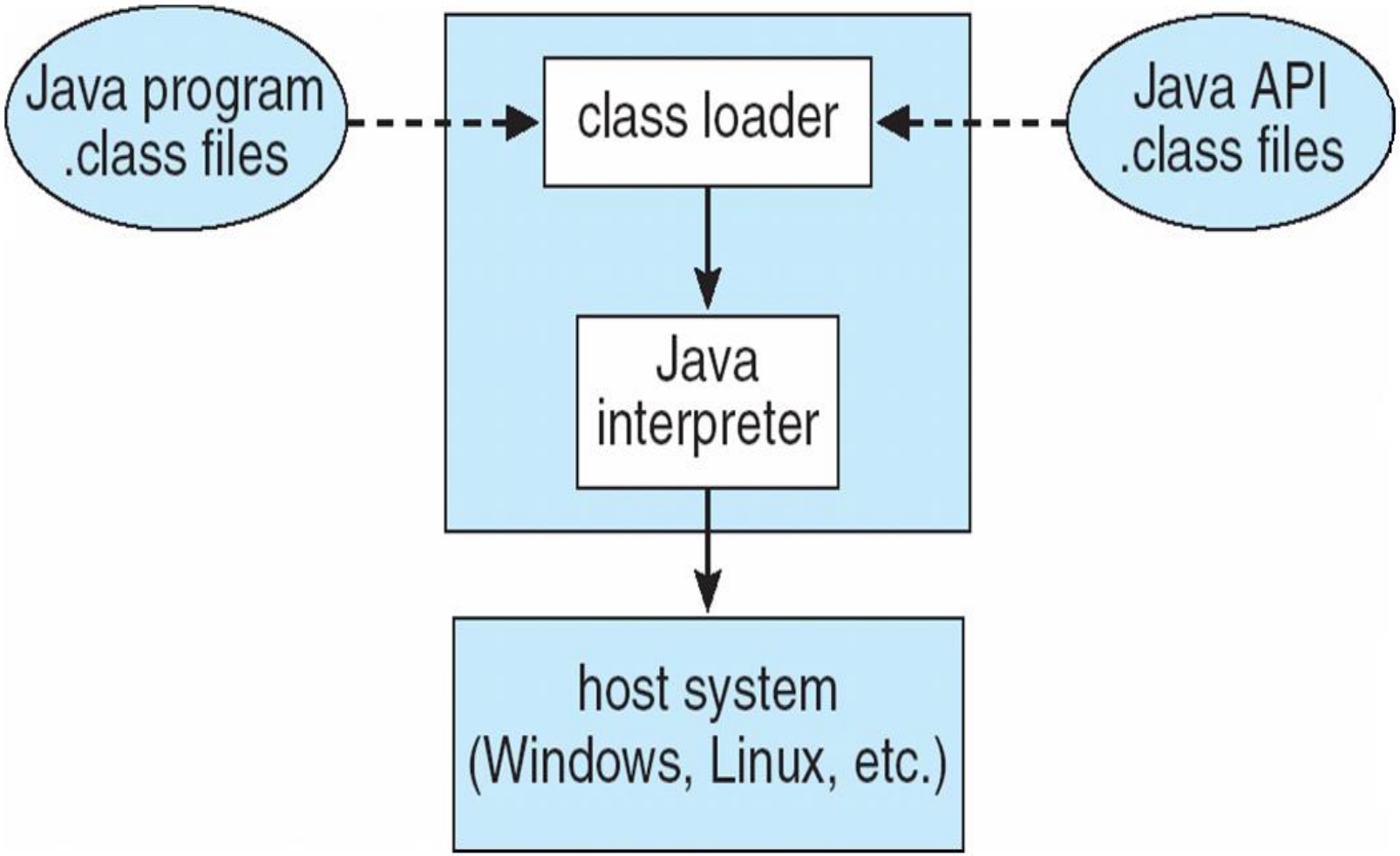


(a) A process virtual machine, with multiple instances of (application, runtime) combinations.

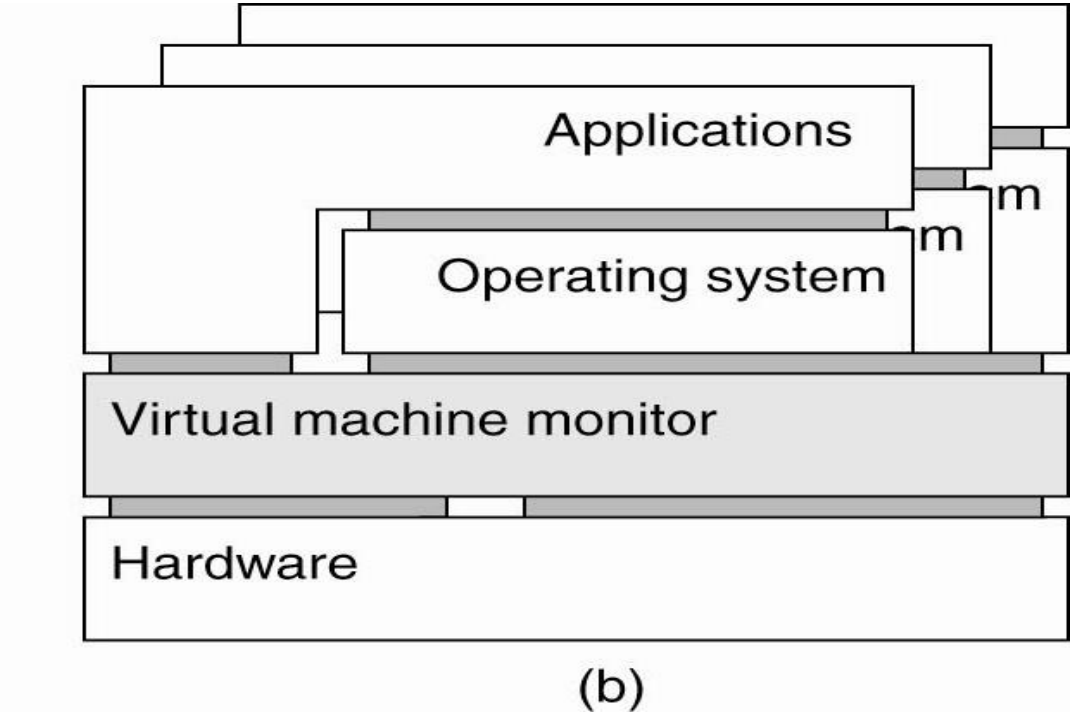
Java Virtual Machine

- Compiled Java programs are platform-neutral bytecodes executed by a Java Virtual Machine (JVM).
- JVM consists of:
  - class loader
  - class verifier
  - runtime interpreter
- Just-In-Time (JIT) compilers increase performance.

The Java Virtual Machine

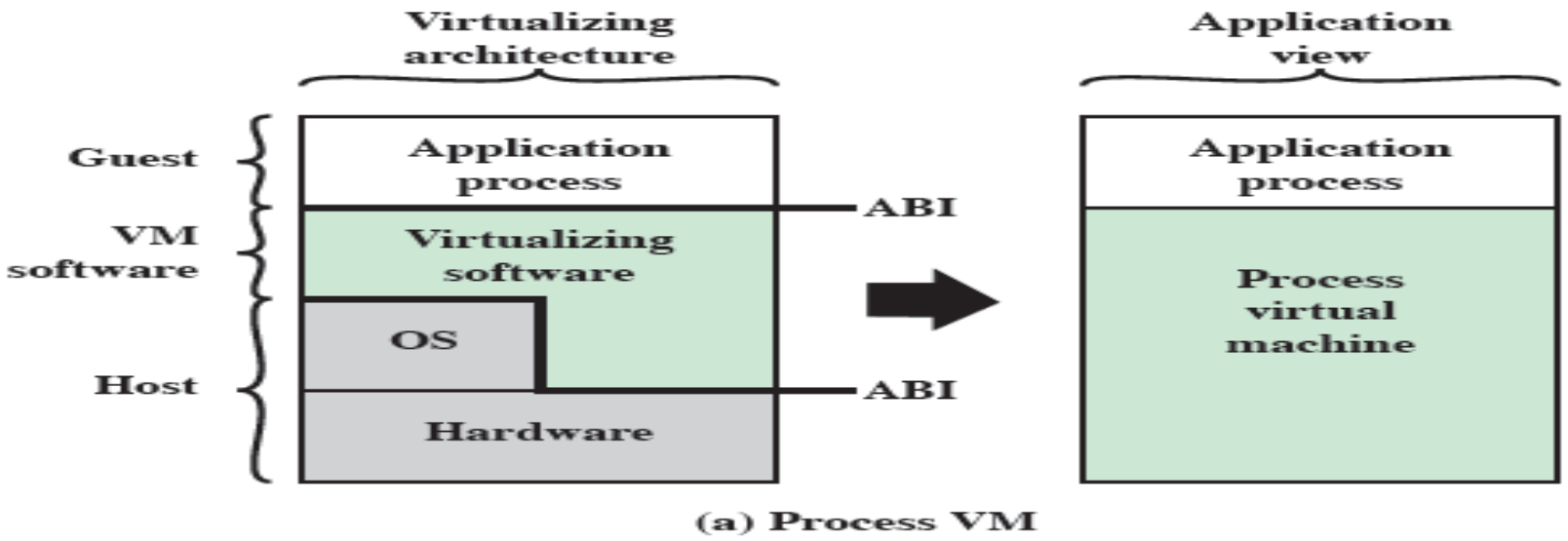


Hypervisor/VMM (Virtual Machine Monitor)

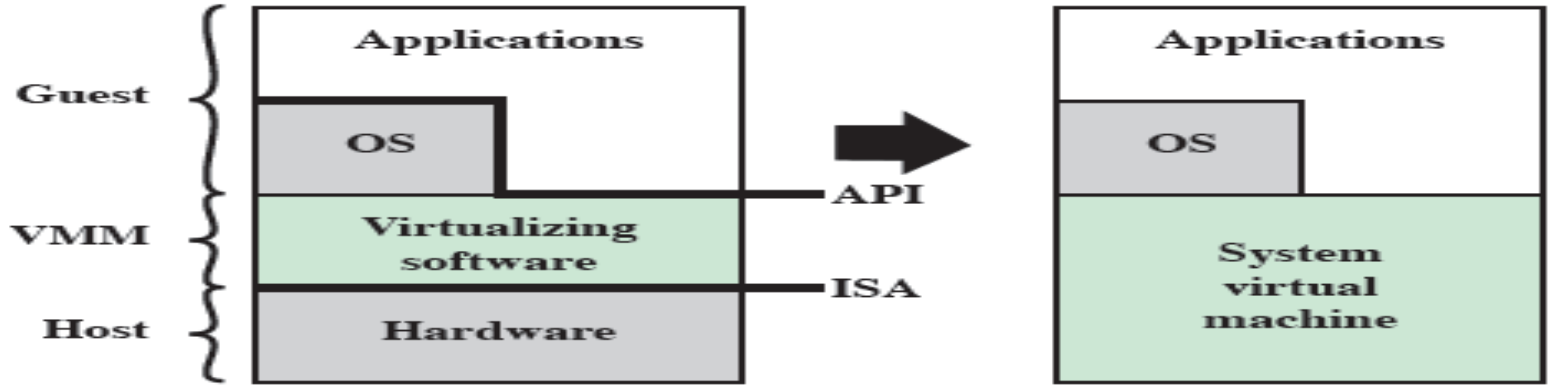


(b) A virtual machine monitor, with multiple instances of (applications, operating system) combinations.

Process and System Virtual Machines

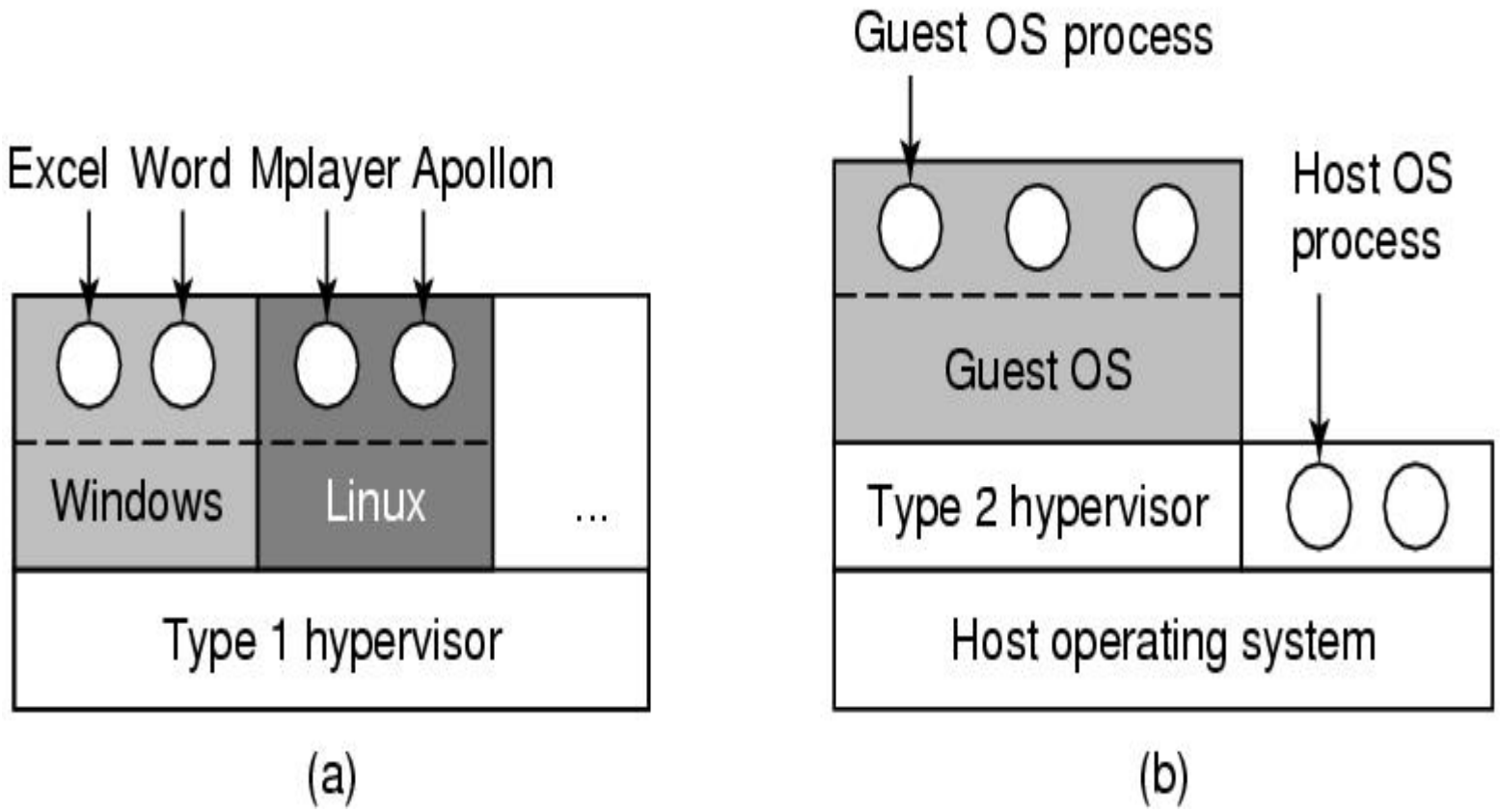


(a) Process VM



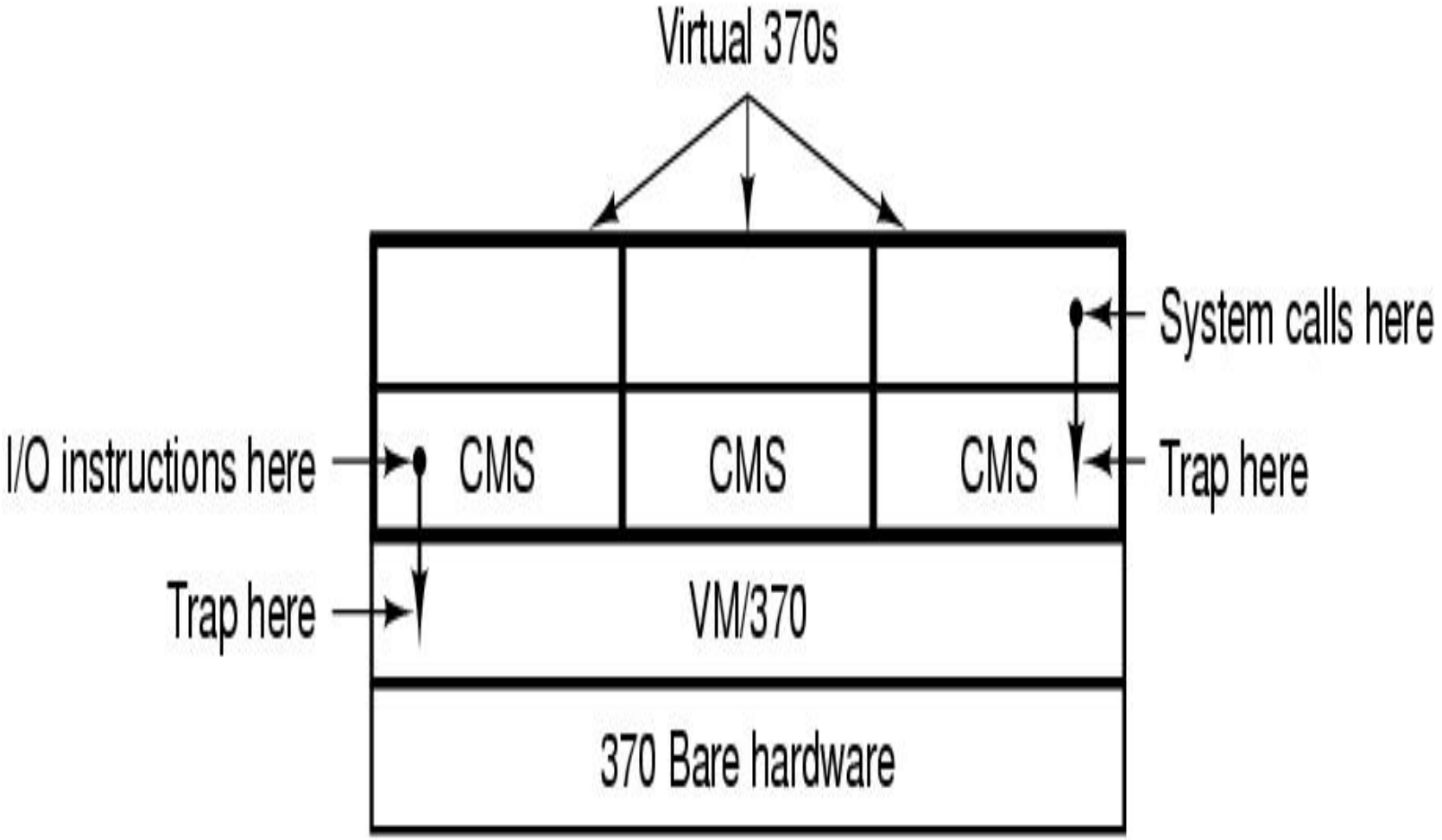
(b) System VM

Types of Hypervisors

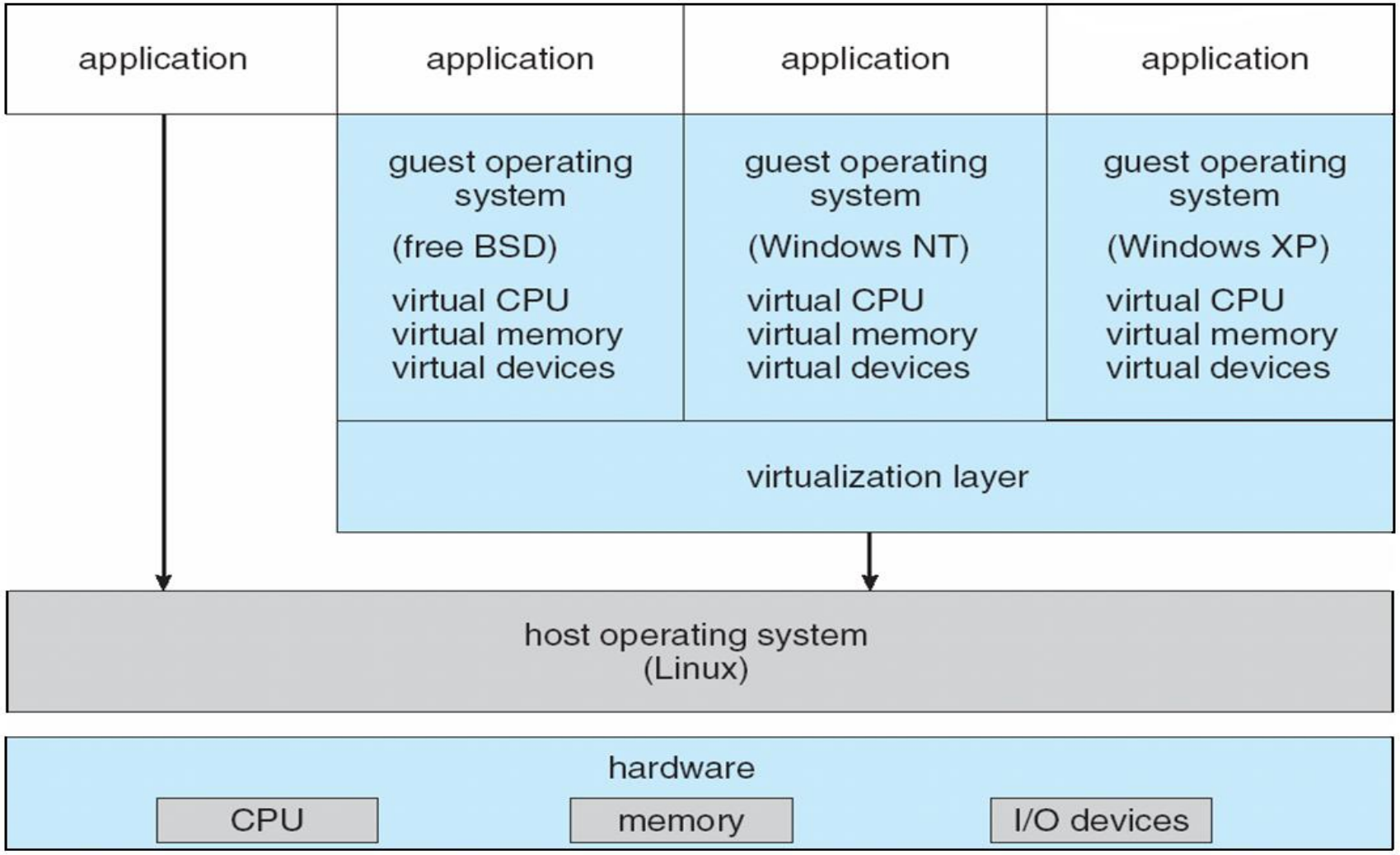


(a) A type 1 hypervisor. (b) A type 2 hypervisor

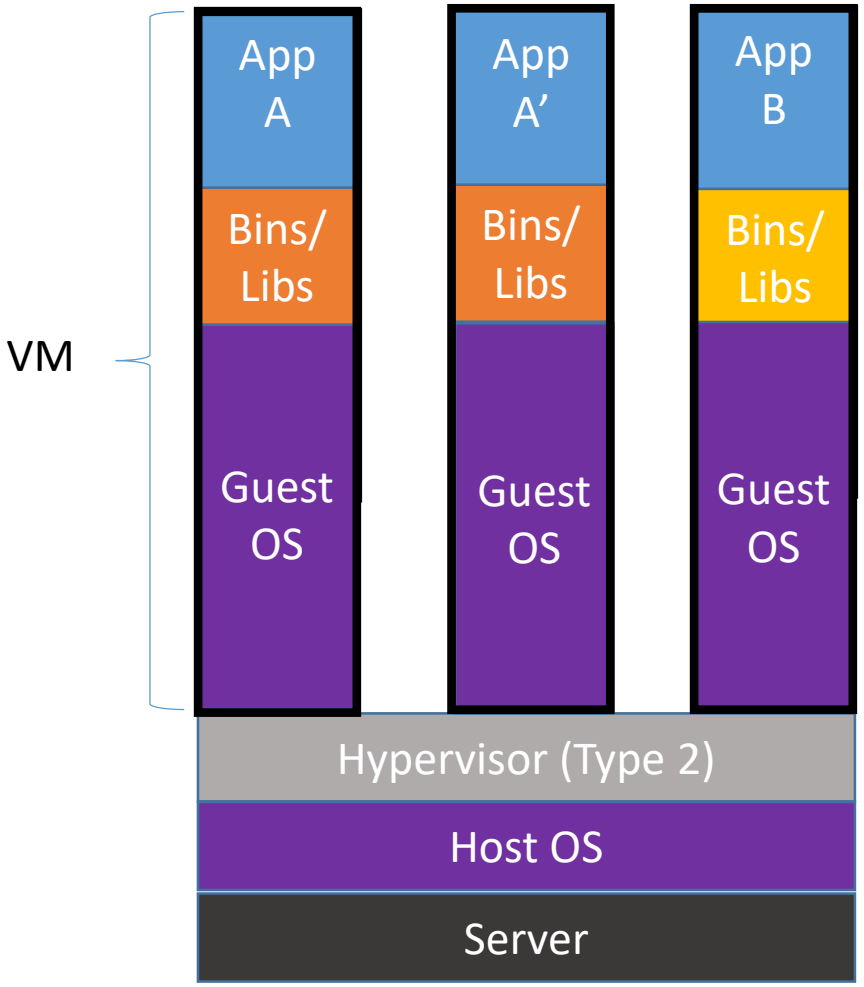
VM/370 with CMSs



VMware Architecture

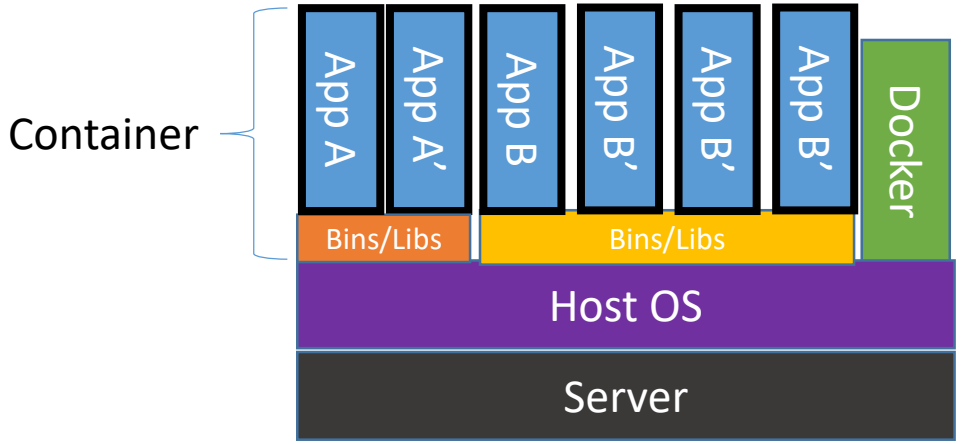


Containers vs. VMs



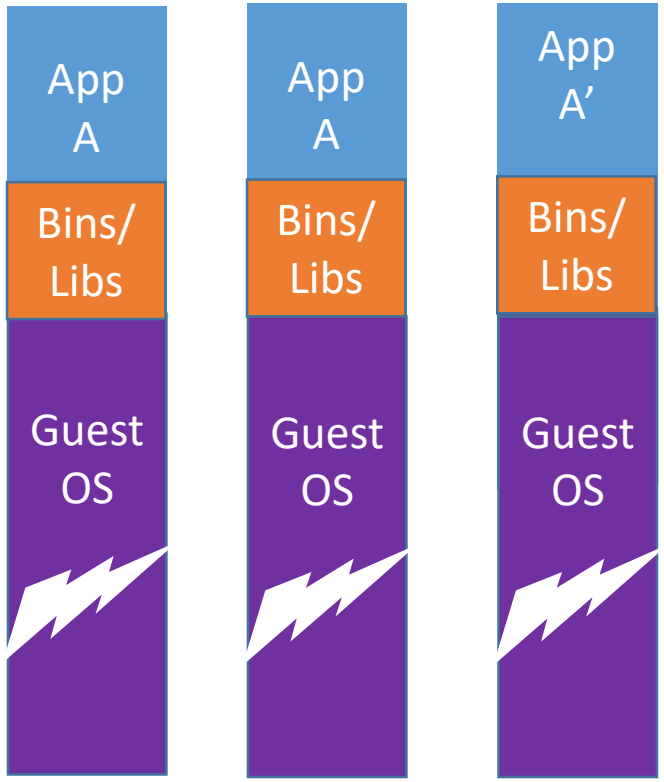
Containers are isolated, but share OS and, where appropriate, bins/libraries

…result is significantly faster deployment, much less overhead, easier migration, faster restart



Why are Docker containers lightweight?

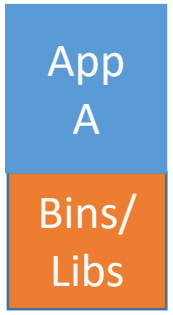
VMs



VMs

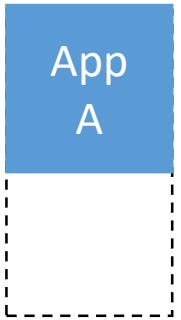
Every app, every copy of an app, and every slight modification of the app requires a new virtual server

Containers



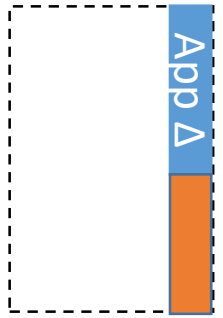
Original App

(No OS to take up space, resources, or require restart)



Copy of App

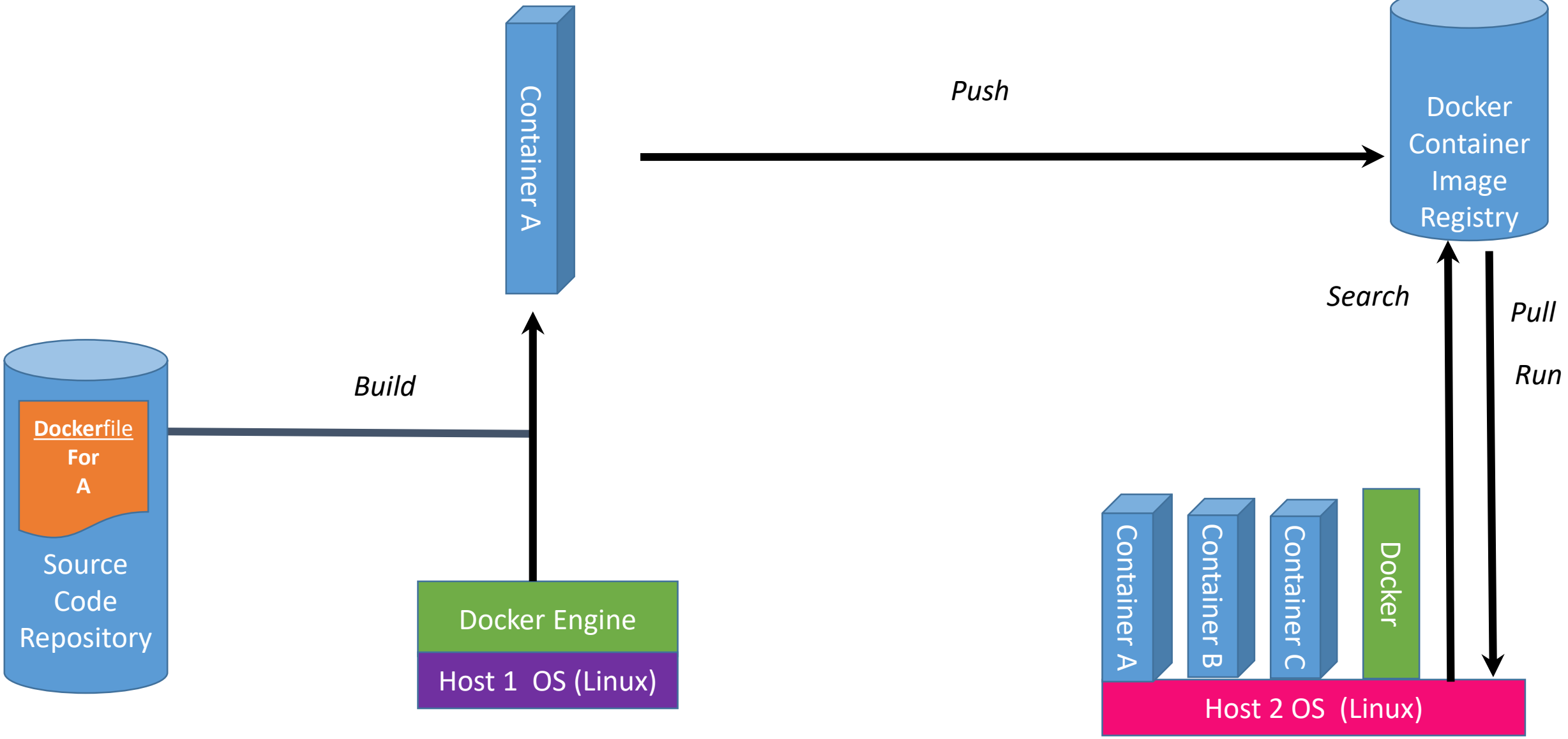
No OS. Can Share bins/libs



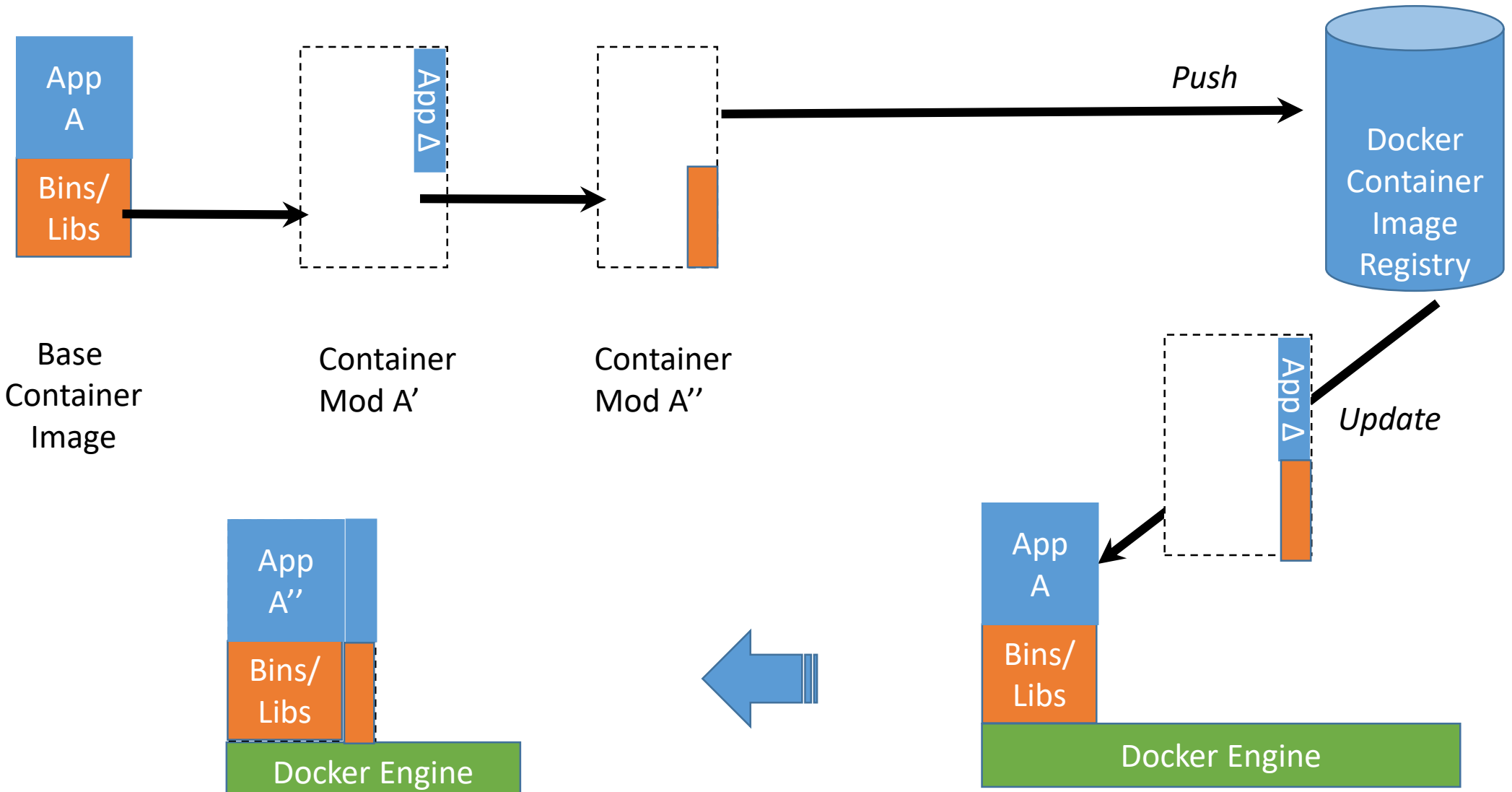
Modified App

Copy on write capabilities allow us to only save the diffs Between container A and container A'

What are the basics of the Docker system?



Changes and Updates



Host is now running A''

Host running A wants to upgrade to A''.
Requests update. Gets only diffs

Definition of Operating System

- There is no universally accepted definition.
- “Everything a vendor ships when you order an operating system” is good approximation but varies widely.
- “The one program running at all times on the computer” is the **Kernel.**
- Everything else is either a system program (ships with the operating system) or an application program.

SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY