Chapter 3 Operating System Structures

3.1 Resource Management
Process Management

- The creation, deletion, suspension, and resumption of processes.
- Provide mechanism for
  - process synchronization
  - process communication
  - deadlock handling

Process Management

- Program: A sequence of instructions。被動的實體(Passive entity)。
- Process: 正在執行的程式。活動的實體(Active entity)。
- Job: The collection of activities needed to do the work required.

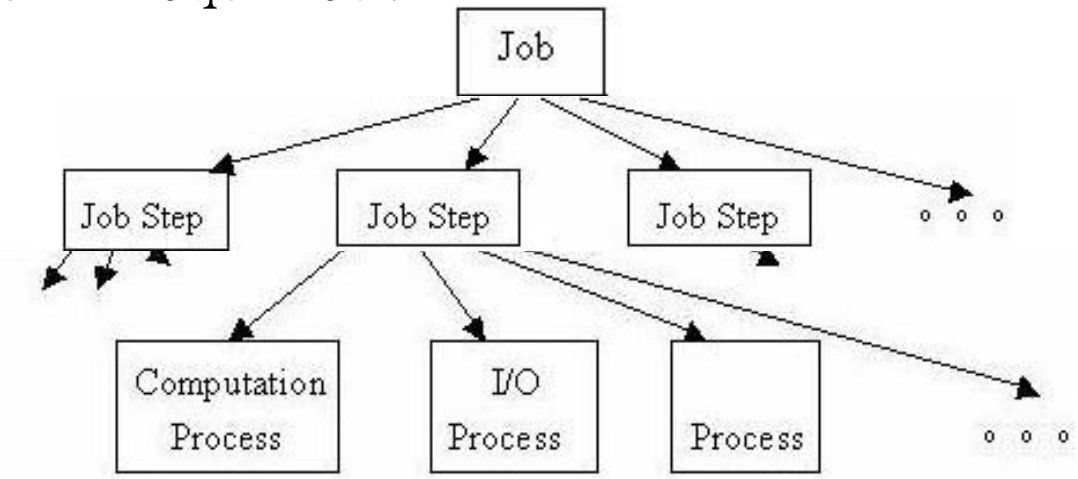


圖 3.2　工作、程式、及處理元之關係

Memory Management

- Keep track of the memory. What parts are used an by whom?
- Decide which processes gets memory, when it gets it and how much.
- Allocate and de-allocate memory.

Device Management

- Keep track of the devices, control units.
- Decide what is an efficient way to allocate the device.
- Allocate and de-allocate devices.
  - Buffer caching system
  - Device driver interface
  - Drivers for specific hardware devices (device driver)

Information Management

- Creation and deletion of file systems.
- Keep track of the information (location, used, status).
- The support of primitives for manipulating file systems.
- The mapping of files into secondary storage.

System Call(1)

3.2 System Calls

- When system call, there are three general methods are used to pass parameters to the OS:
  - Registers
  - Pass block address
  - Stack

System Call(2)

- System call can be grouped into five categories
  - Process control
  - File manipulation
  - Device manipulation
  - Information maintenance
  - Communications

System Call(3)

- Process Control
  - end, abort
  - load, execute
  - create process, terminate process
  - get process attributes, set process attributes
  - wait for time
  - wait event, signal event
  - allocate and free memory

System Call(4)

- File Manipulation
  - create file, delete file
  - open, close
  - read, write, reposition
  - get file attributes, set file attributes

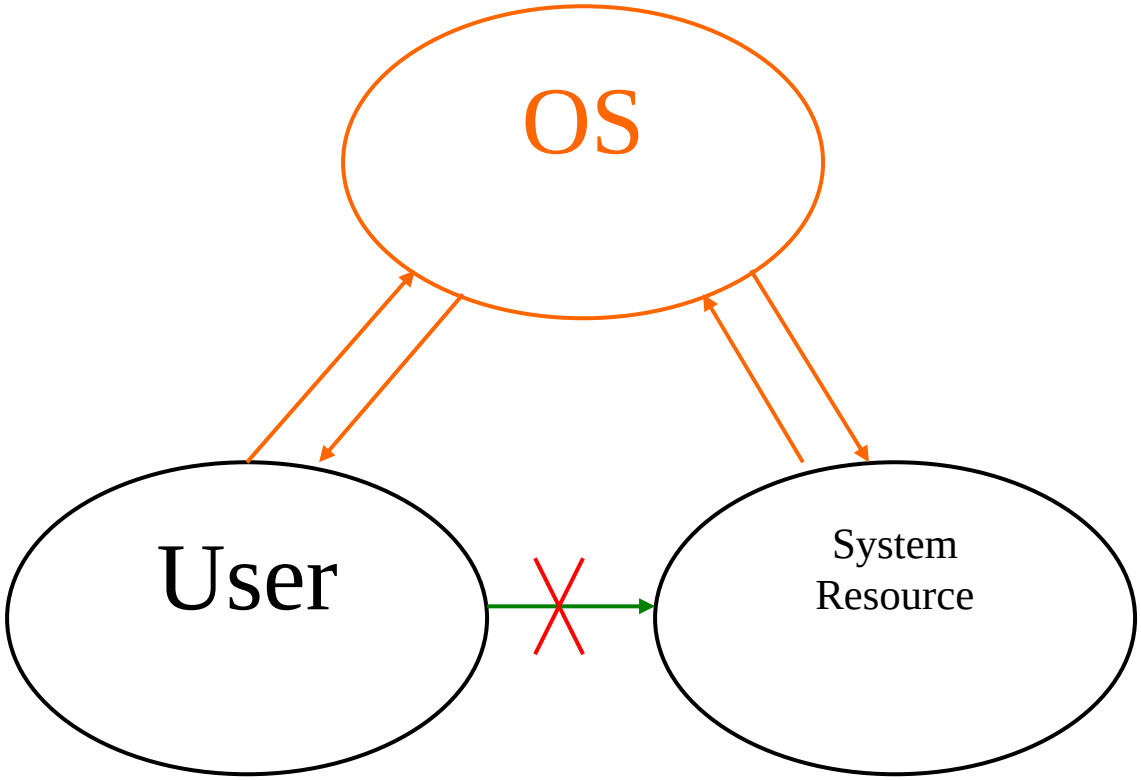
- Device Manipulation
  - request device, release device
  - read, write, reposition
  - get device attributes, set device attributes
  - logically attach or detach devices

System Call(5)

- Information Maintenance
  - get time or date, set time or date
  - get system data, set system data
  - get process, file, or device attributes
  - set process, file, or device attributes

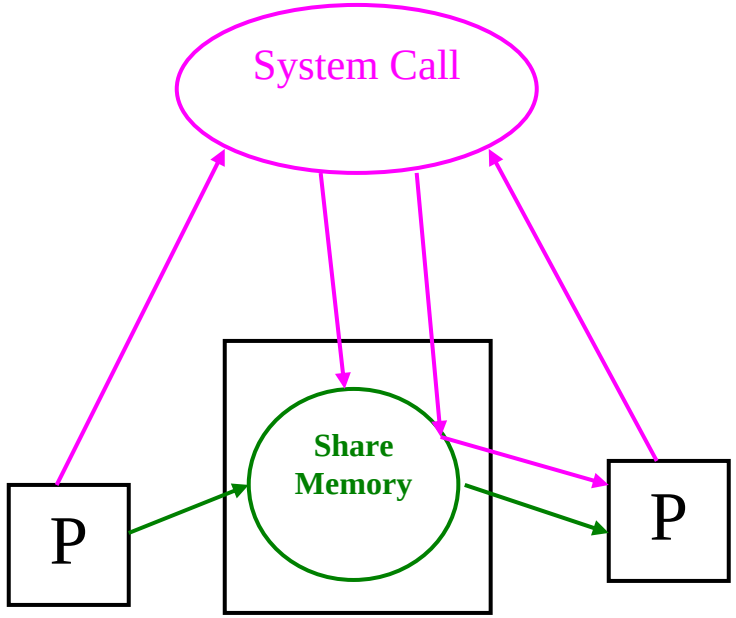
- Communication
  - creat, delete communication connection
  - send, receive messages
  - transfer status information
  - attach or detach remote devices

System Call(6)



Communication

- Message passing model
  - By using Inter Process Communication (IPC)
- Shared memory model
  - Processes use map memory system calls to gain access to regions of memory owned by other processes.



分層結構(Layer Approach)(1)

3.3 System Structure

- 原則上作業系統之各個管理及功能都是以分層結構方式組合。

- 作業系統也是由軟體及軔體所撰寫完成，而撰寫軟體常強調系統化及模組化。

分層結構(Layer Approach)(2)

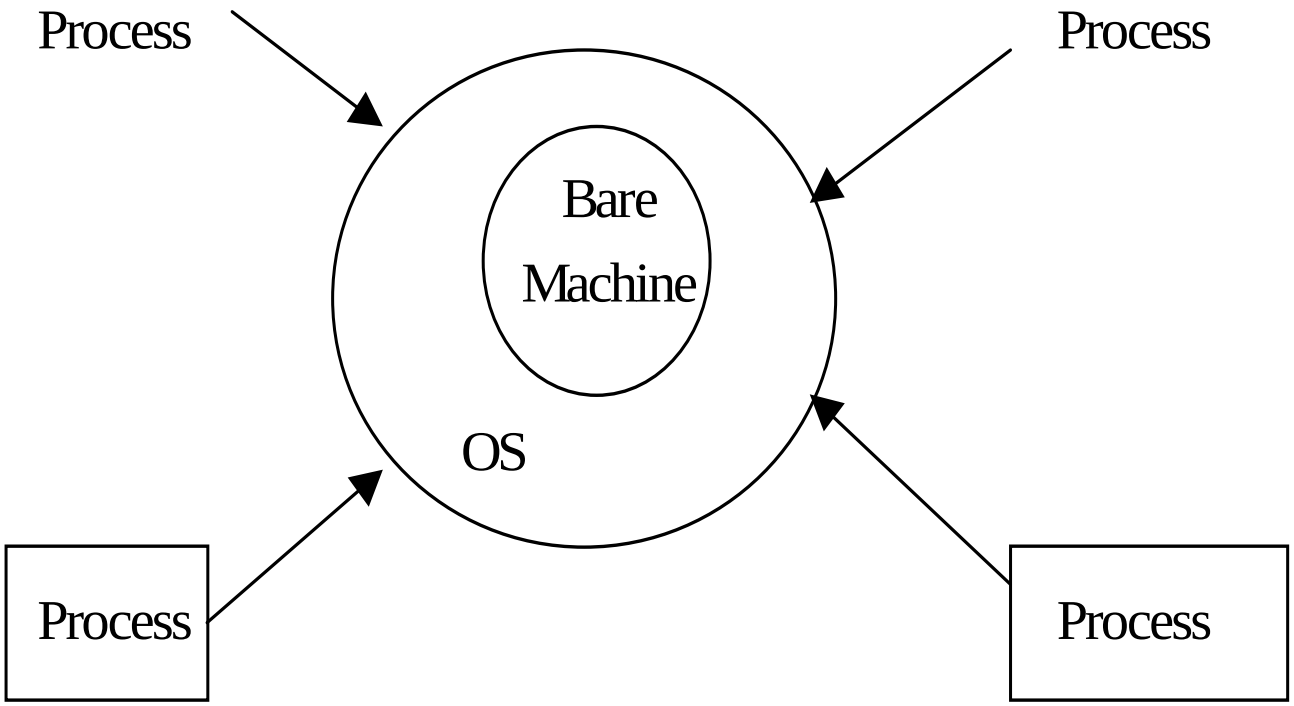


圖 1.22a　非分層結構之作業系統

分層結構(Layer Approach)(3)

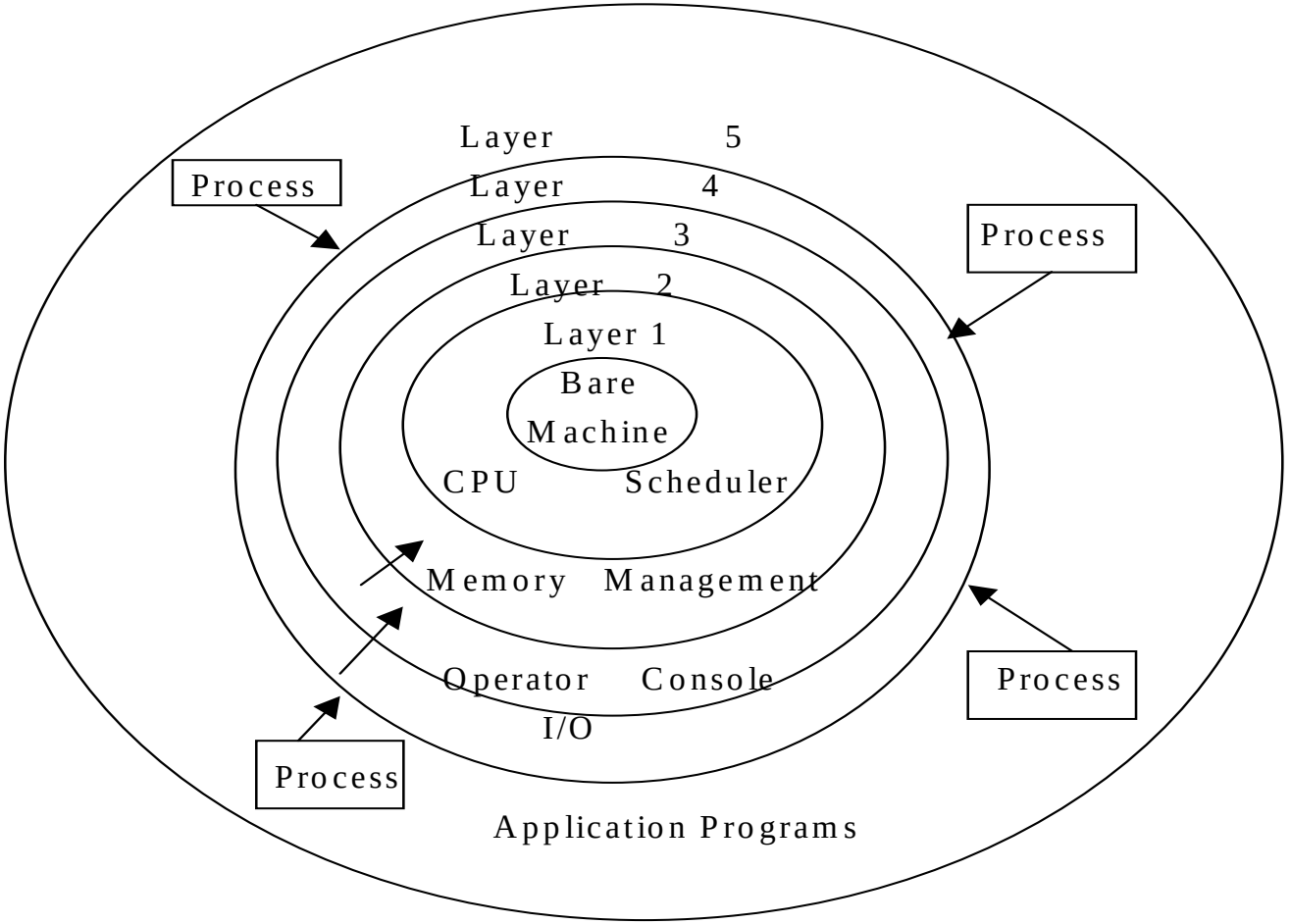


圖 1.22b 分層結構之作業系統

分層結構(Layer Approach)(4)

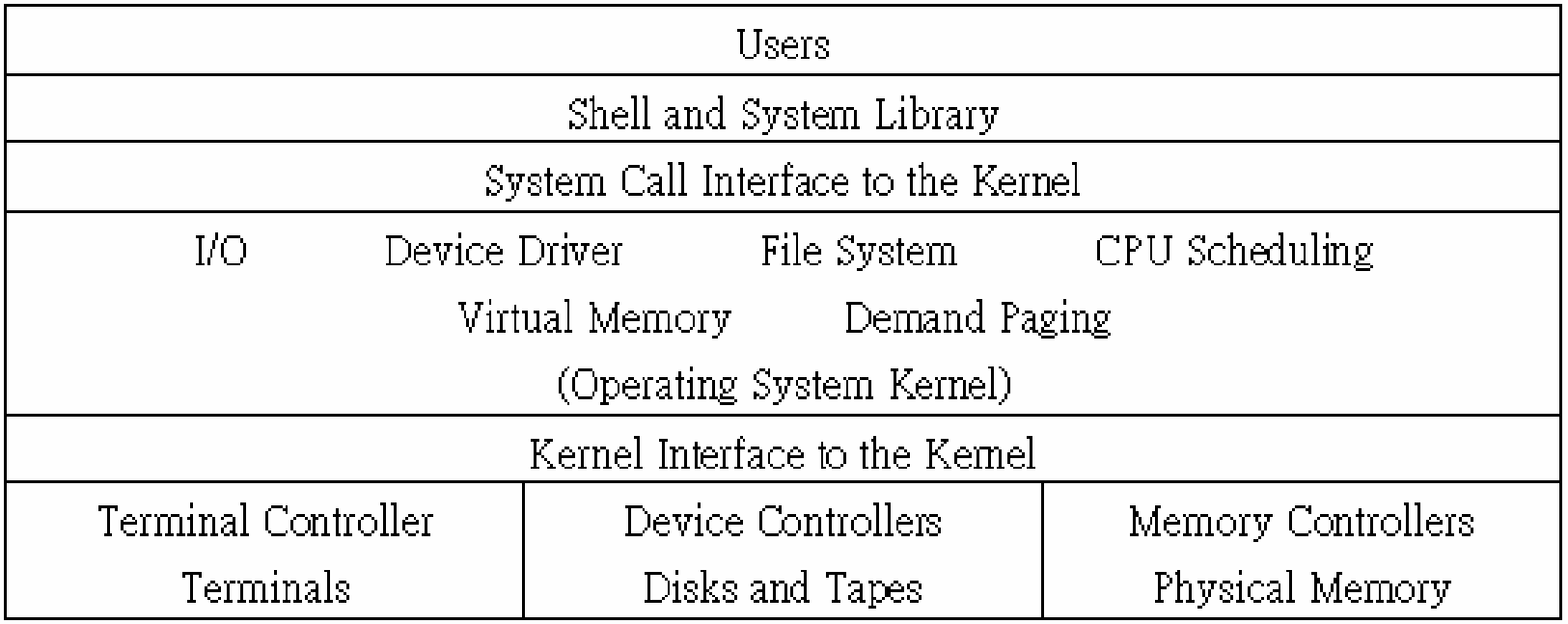


圖 1.12　UNIX 作業系統之分層結構

分層結構(Layer Approach)(5)

MS / DOS之分層結構，應用程式可以直接呼叫基本輸出 / 輸入函數，而沒有透過作業系統之系統呼叫後，再經由作業系統核心出面安排處理工作；也因爲MS / DOS作業系統的不明顯分層結構，而可以由應用程式直接控制系統資源，以致造成容易當機的問題發生。

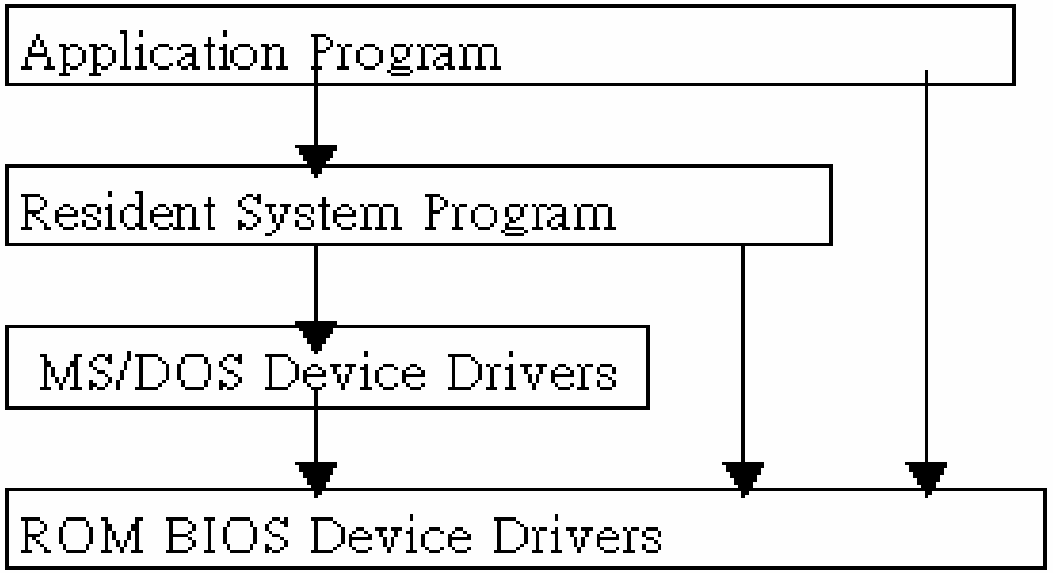
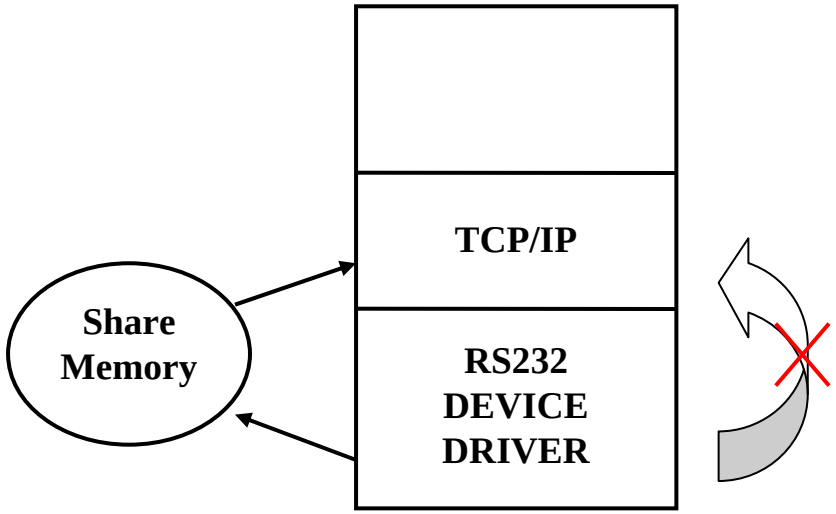


圖 1.13　MS / DOS 之分層結構

分層結構(Layer Approach)(6)

- Direct user access to low level facilities is not allowed.
- 內層不可以呼叫外層。
- 外層只能一層一層呼叫。



- 可以在不更改該介面的條件下，將某一層抽離換新，而不會影響其他幾層。

虛擬機器(Virtual Machine)(1)

3.4 Virtual Machine

- 從使用者的角度來看，每個人都認爲電腦是自己所擁有的，電腦資源都是自己所獨享的，這就是虛擬機器的概念。

- 每位使用者都擁有自己的虛擬機器，而實際上作業系統會將虛擬機器上的虛擬資源（Virtual Resources），轉換爲實際資源（Physical Resources），並去操控它們。

虛擬機器(Virtual Machine)(2)

For each user program execution on its own processor(CPU scheduling is managed by OS) with its own memory virtually, and the system support spooling facility.

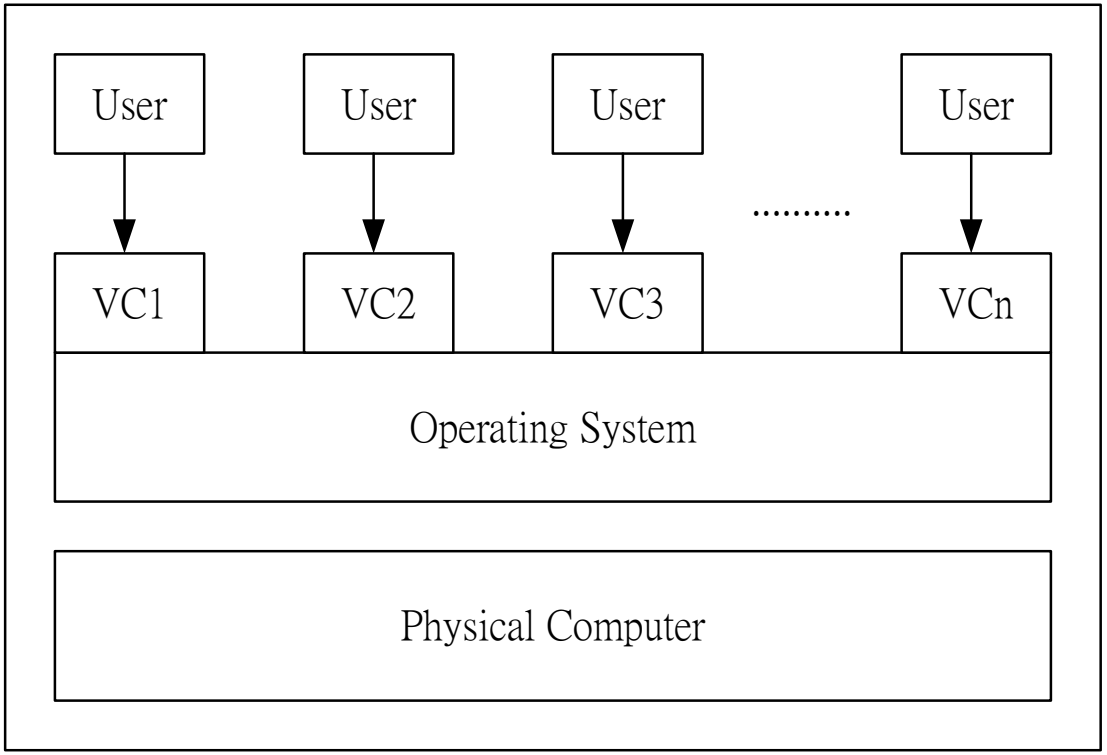


圖2.12 虛擬機器

虛擬機器(Virtual Machine)(3)

- 作業系統提供時間多工（Time Multiplex），使得許多程式可以分時使用中央處理器。

- 提供空間多工（Space Multiplex），使得許多程式可以存在主記憶體內不同位置。

虛擬機器(Virtual Machine)(4)

- 虛擬機器最簡單的講法就是說，一部電腦的作業系統是以分層結構方式完成，使用者純粹以邏輯結構來看電腦及命令電腦工作，而用不著去了解細微的機器控制動作，因爲這些細微控制及同步機制均由作業系統內各分層協調完成。

- Layer approach is taken to its logical conclusion in the concept of a virtual machine.

Java Virtual Machine(1)

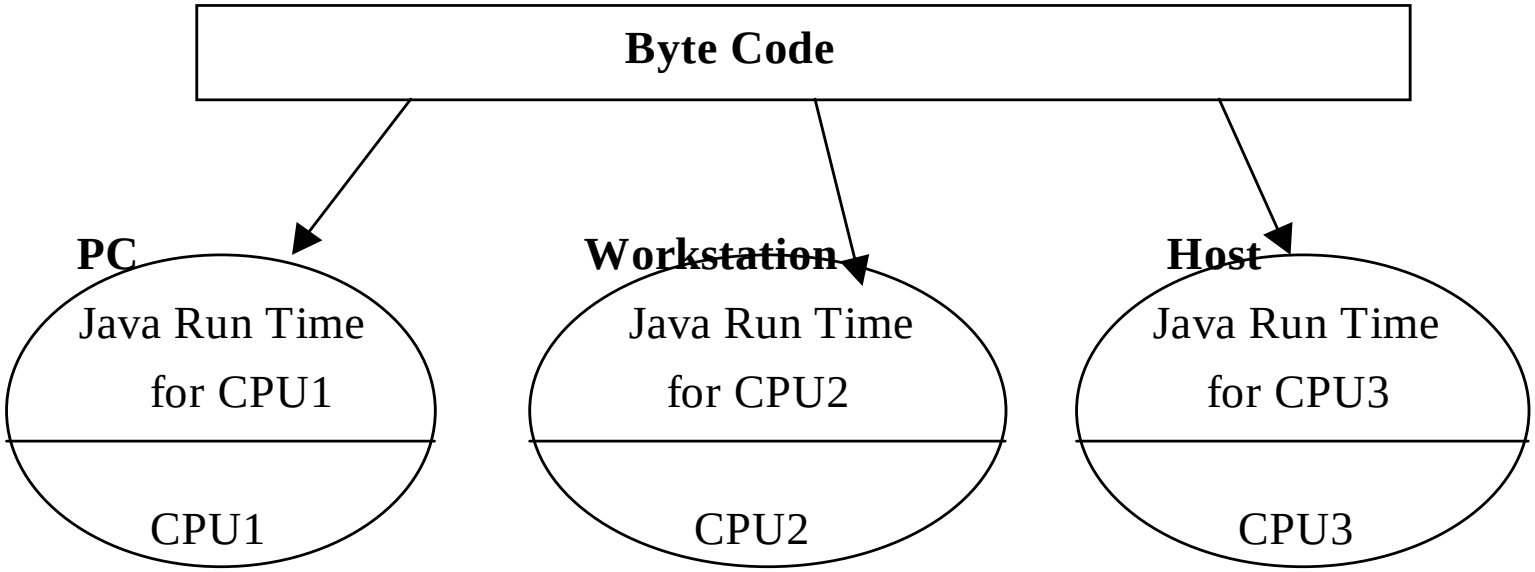


圖 2.13　Java 虛擬機器

Java Virtual Machine(2)

- Java挾其可攜帶性之優點，橫掃跨平台作業系統，它便是一種虛擬機器的代表。
- 從外層的角度來看，所有命令都與機器無關，它們是位元組碼。
- 從內層來看，不同機器的Java執行模組均不相同，因爲Java執行模組必須依據所使用的中央處理器，撰寫不同的機器碼，來命令電腦工作。

3.5 System Design and Implementation

- 在設計分層結構時，通常必須將機制（Mechanism）與政策（Policy）分開。

- 所謂機制是一組供使用之基本功能。
  - A set of basic facilities that can be used in many different ways. Determine how to do something.如何作?

- 所謂政策是使用機制功能作特定工作。
  - The use of a mechanism for a particular purpose. Decide what will be done.作什麼?

機制（Mechanism）與政策（Policy）

- Mechanism
  - CPU protection: timer, interrupt
  - Protect file: Read, Write, Execute

- Policies
  - CPU protection: how long, what to do
  - Protect file: which files can be R, W, E

- 如果我們確定機制之後，即使政策改變，只要重新定義系統中相關的參數便可以（例如時間片斷減小）；因此將機制與政策分開，可以增加系統的彈性。

System Generation

3.6 System Generation(Installation)

- The system be configured for each specific computer site. e.g. memory size, device, options,...etc.