

Chương 1

1.1 Giới thiệu

1.2 Phần cứng máy tính

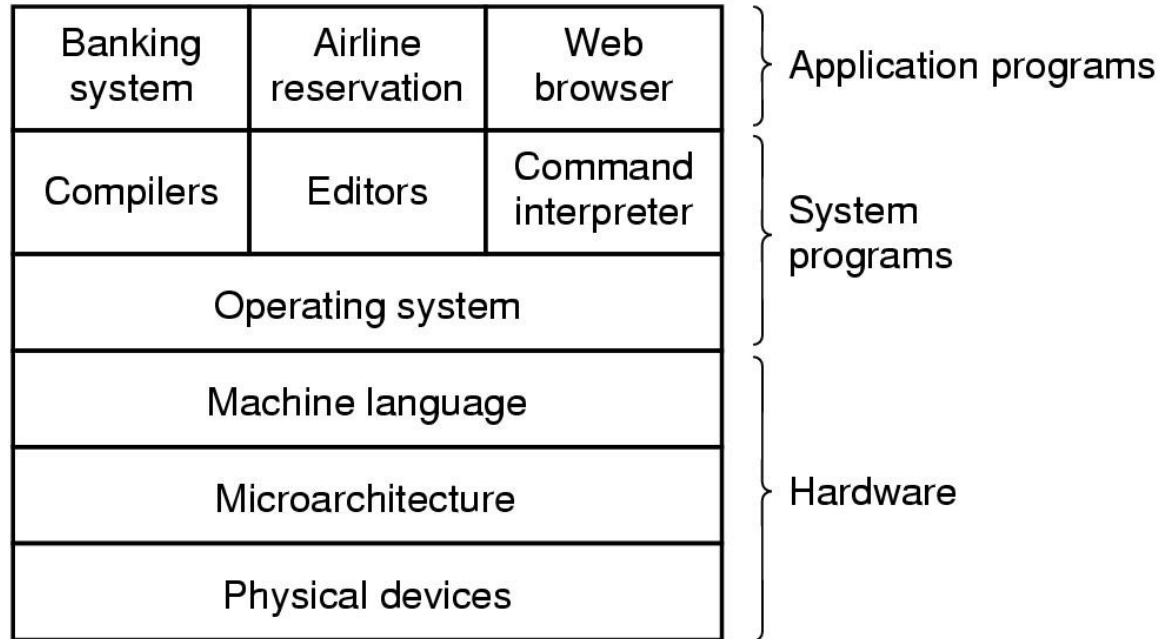
1.3 Các khái niệm về hệ điều hành

1.1 Giới thiệu

- Hệ điều hành là gì
- Lịch sử của các hệ điều hành
- Các loại hệ điều hành

Hệ điều hành là gì?

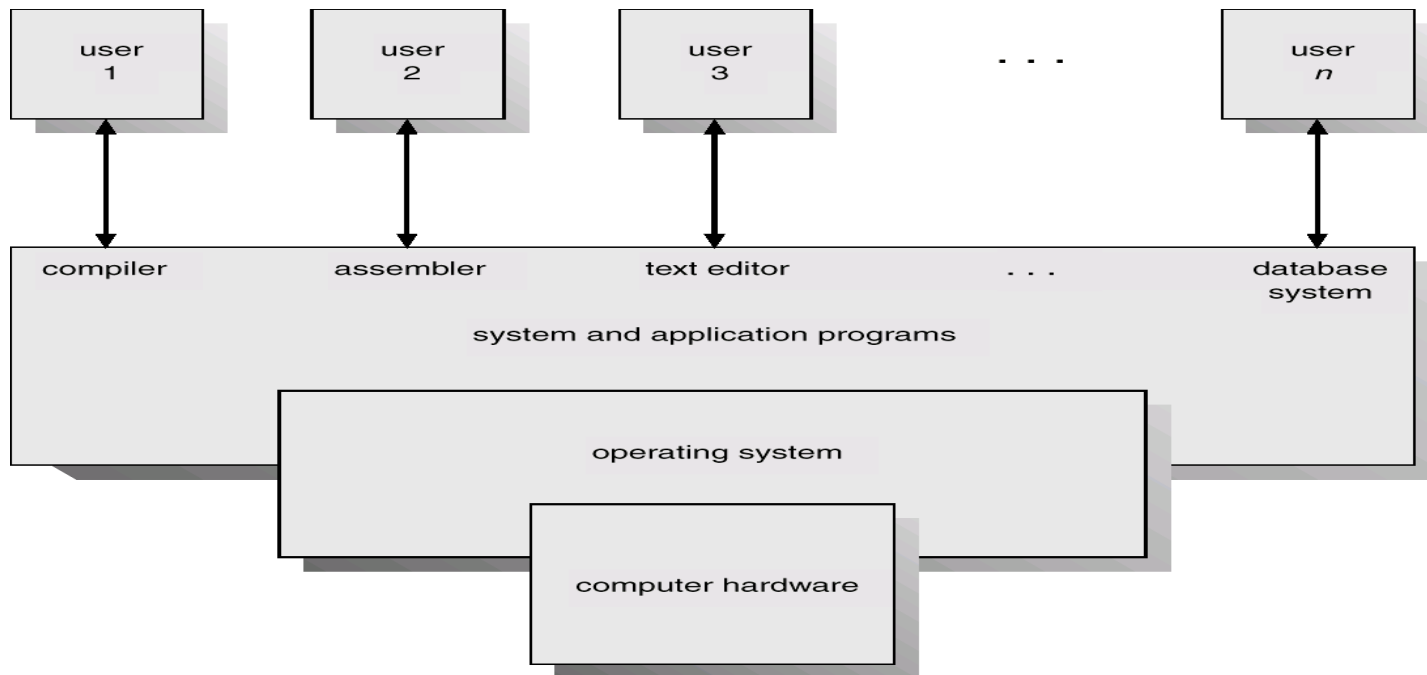
Hệ thống máy tính



- Một hệ thống máy tính bao gồm:
 - hardware
 - system programs
 - application programs

Hệ điều hành là gì?

- Mô hình thực hiện của các hệ thống đa user



Hệ điều hành là gì

- Hệ điều hành là một máy mở rộng
 - Các chi tiết phức tạp ẩn sâu bên trong hệ thống
 - Cung cấp cho người dùng một máy ảo, tạo ra giao diện tiện lợi sao cho dễ sử dụng hơn
- Hệ điều hành là một manager quản lý Tài nguyên
 - Mỗi chương trình cần cấp một khoản thời gian để sử dụng tài nguyên
 - Mỗi chương trình cần có một không gian trên tài nguyên

Lịch sử của các Hệ điều hành

Lịch sử của các hệ điều hành

- Thế hệ I 1945 - 1955
 - vacuum tubes, plug boards
- Thế hệ II 1955 - 1965
 - transistors, batch systems
- Thế hệ III 1965 – 1980
 - ICs and multiprogramming
- Thế hệ IV 1980 – ngày nay
 - personal computers

Lịch sử của các Hệ điều hành

Thế hệ I 1945 - 1955

- Máy tính: ENIAC, UNIVAC...
- Hệ điều hành: Không có HĐH,
- Ngôn ngữ máy,
- Tập trung thiết kế, xây dựng, lập trình, điều hành và bảo dưỡng trên cùng một máy.

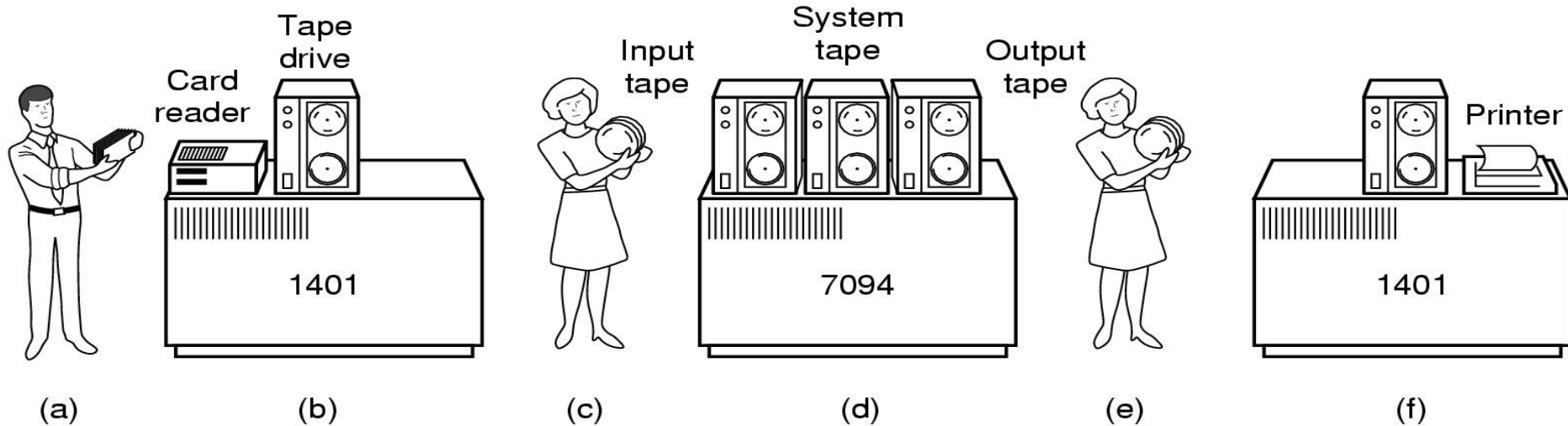
Lịch sử của Hệ điều hành

Thế hệ II 1955 – 1965 (1)

- Máy tính: IBM 1401, IBM 7094...
- Hệ điều hành : FMS (Fortran Monitor System), IBSYS for Computer 7094
- Các hệ thống xử lý lô (Batch Systems)
 - Chức năng của tiền xử lý lô
 - Cấu trúc của loại công việc FMS
- Tách biệt giữa thiết kế, xây dựng, lập trình, điều hành và bảo dưỡng trên mỗi cá nhân.

Lịch sử của Hệ điều hành

Thế hệ II 1955 – 1965 (2)

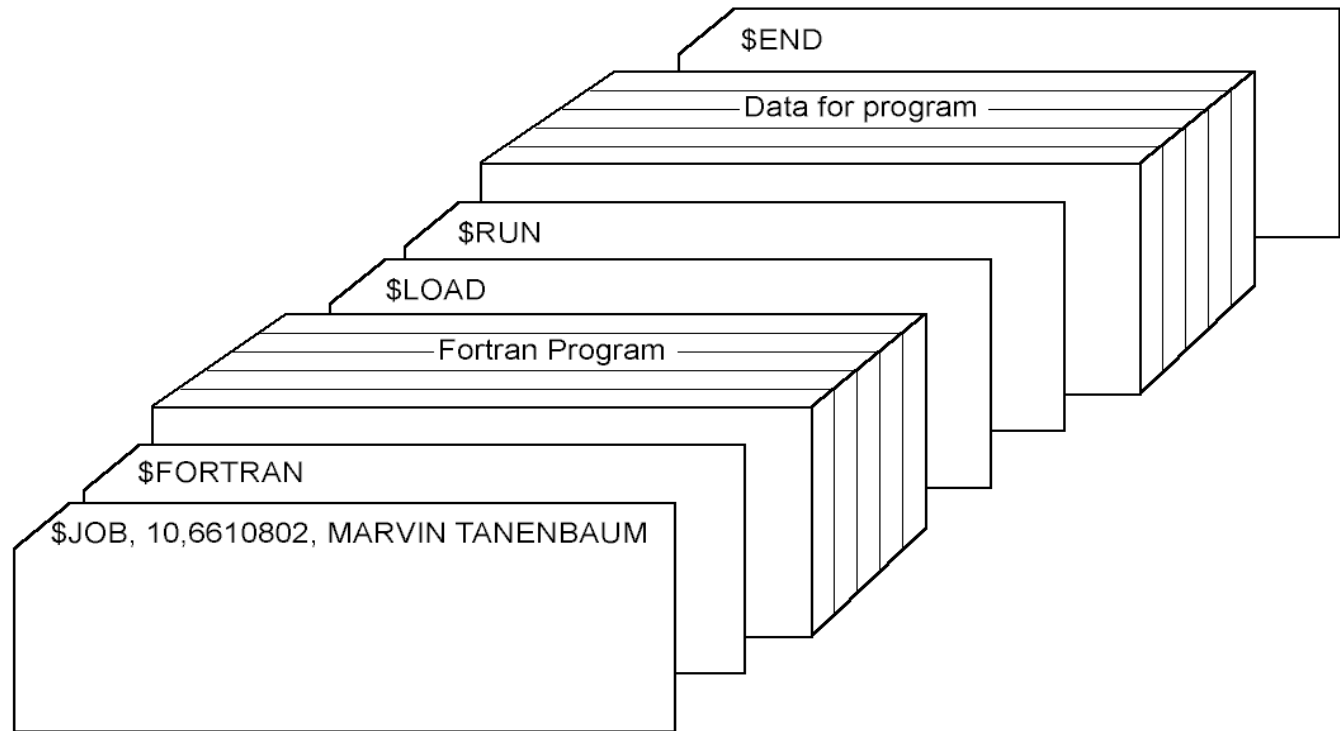


Tiền xử lý lô

- mang các card đến 1401
- đọc các card đến tape
- đặt tape vào 7094 để tính toán
- đặt tape vào 1401 để in ra

Lịch sử của Hệ điều hành

Thế hệ II 1955 – 1965 (3)



- Cấu trúc của loại công việc FMS – thế hệ II

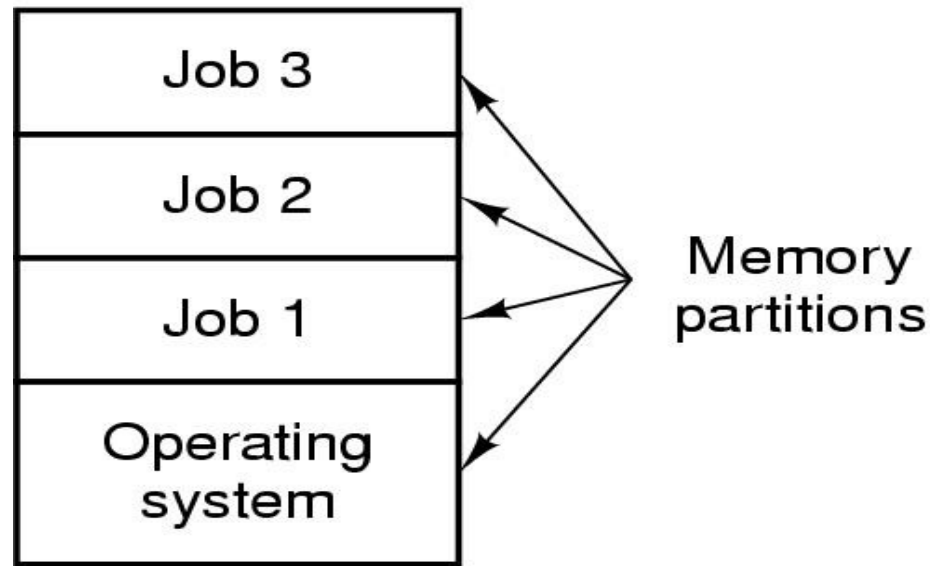
Lịch sử của Hệ điều hành

Thế hệ III 1965 – 1980 (1)

- Máy tính: System/360, IBM370, IBM4300...
- Hệ điều hành:
 - OS/360: làm việc cho tất cả các model
 - Đa nhiệm
 - Chia sẻ thời gian:
 - CTSS (Compatible Time Sharing System),
 - MULTICS (MULTiplexed Information and Computing Service),
 - Unix

Lịch sử của Hệ điều hành

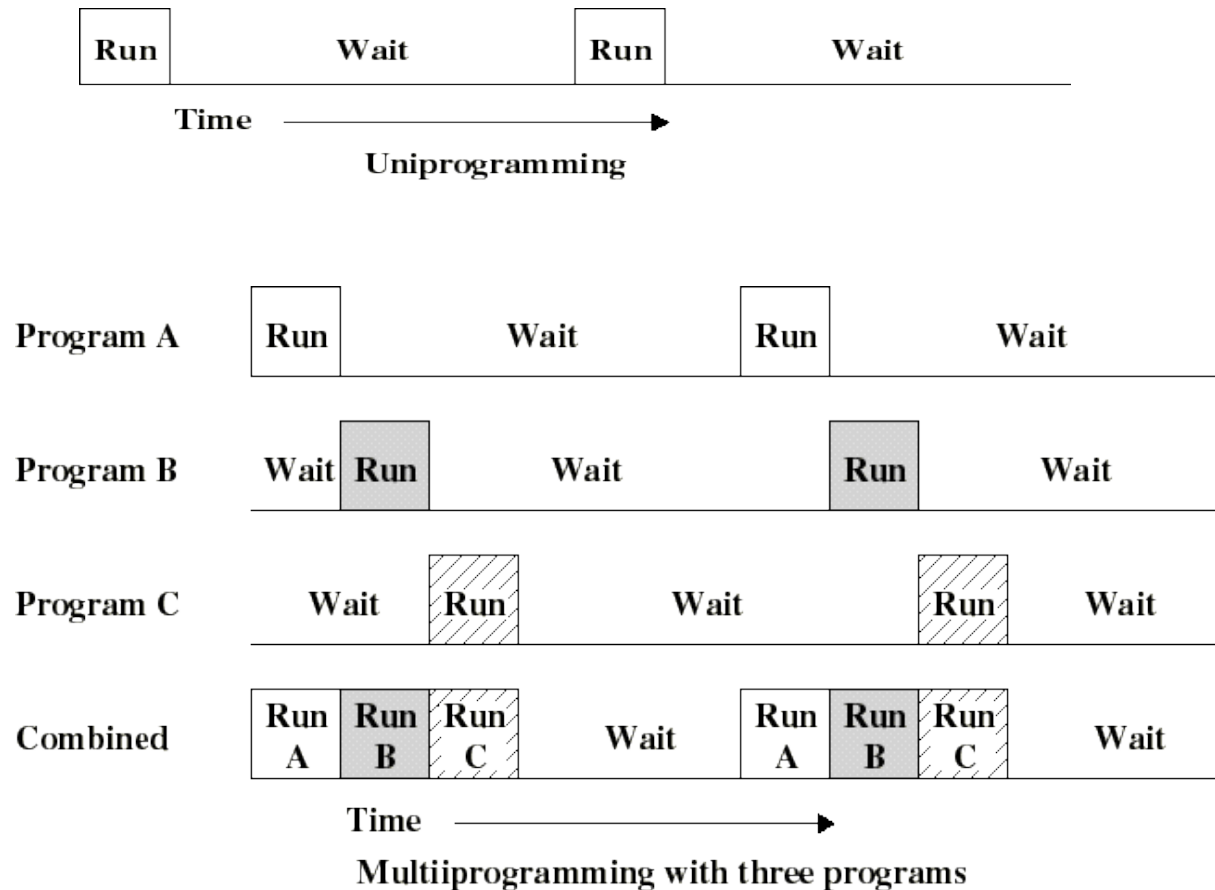
Thế hệ III 1965 – 1980 (2)



- Hệ thống đa nhiệm (Multiprogramming system)
 - Có 3 job trong bộ nhớ

Lịch sử của Hệ điều hành

Thế hệ III 1965 – 1980 (3)



Lịch sử của Hệ điều hành

Thế hệ IV 1980 – ngày nay

- Computers: IBM PC 80x86, Macintosh...
- Operating System:
 - 1977: CP/M (Control Program for Microcomputer)
 - 1980: DOS (Disk Operating System)
 - GUI with Macintosh
 - 1985-1995: Window 3.x
 - 1995: Window 95
 - 1996: Window NT 4.0
 - 1999: Window 2000
 - Window 2003
 - Unix

Các loại Hệ điều hành

Các loại Hệ điều hành

- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Real-time operating systems
- Embedded operating systems
- Smart card operating systems

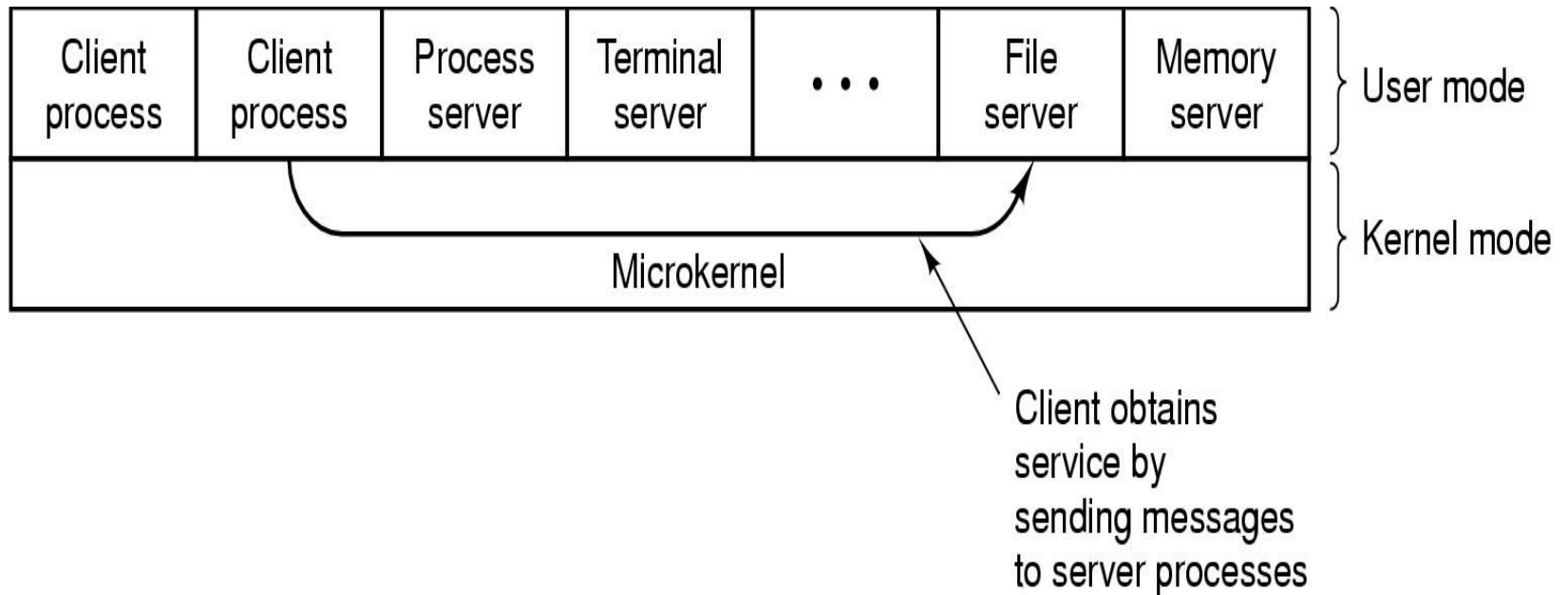
Các loại Hệ điều hành

Mainframe operating systems

- Xử lý lô (batch)
- Đa nhiệm (multiprogrammed)
- Chia sẻ thời gian (time-sharing)
- Đa tác vụ (multitasking)
- Ứng dụng: Web Server High-End; Các Server cho các giao dịch Business-To Business
- Ví dụ: OS/390

Các loại Hệ điều hành

Server operating systems



The Operating System Zoo

Multiprocessor operating systems

- Đa bộ xử lý (Multiple CPU)
- Chia sẻ computer bus, clock
- Ích lợi:
 - High system throughput
 - High availability
 - Multiprocessor system and Multicomputer system

Các loại Hệ điều hành

Personal computer operating systems

- Nhiều thiết bị Vào/Ra
- Giao tiếp với người dùng
- Có nhiều Hệ điều hành (MS Windows, Mac OS, Solaris, Linux,...).

Các loại Hệ điều hành

Real-time operating systems

- Thời gian là tham số chính
- Hai kiểu của hệ thống thời gian thực
 - Hard real-time system: dành cho các hệ thống điều khiển tiến trình công nghiệp...
 - Soft real-time system: dành cho các hệ thống multimedia

Các loại Hệ điều hành

Embedded operating systems

- Thiết bị số PDA (Personal digital assistant): Palmtop, Pocket-PC, Cellular phones, control devices
- Hạn chế kích thước bộ nhớ, tốc độ CPU, kích thước màn hình, nguồn điện.
- Hệ điều hành: PalmOS, Windows CE (Consumer Electronic)

Các loại Hệ điều hành

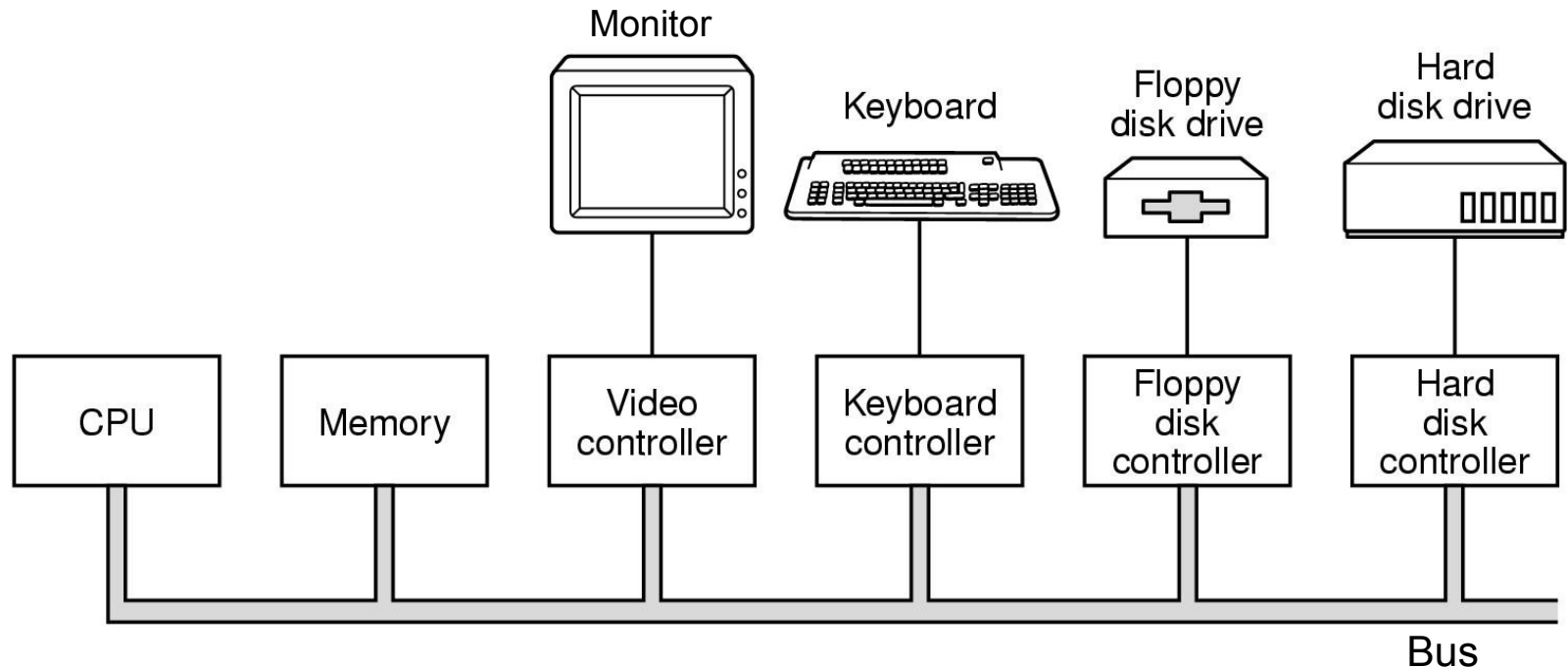
Smart card operating systems

- Các chip CPU ngay trên Card
- Đòi hỏi khắc khe về xử lý nguồn và bộ nhớ
- Các ứng dụng:
 - Đơn chức năng: electronic payments
 - Đa chức năng: proprietary systems
 - Java oriented: holds interpreter JVM

1.2 Sơ lược phần cứng máy tính

- CPU
- Memory
- I/O Devices
- Buses

Sơ lược phần cứng máy tính



- Các thành phần của một máy tính

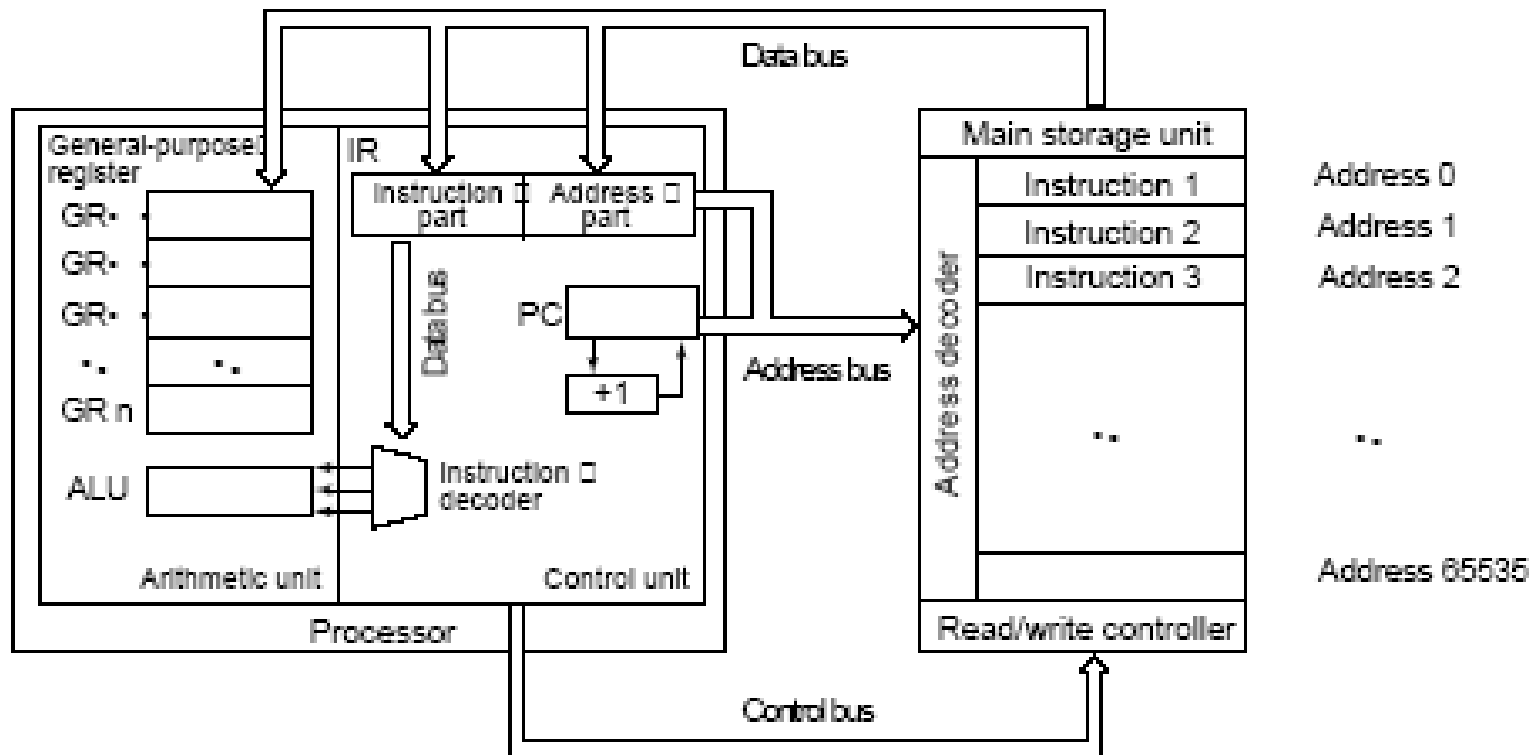
Sơ lược phần cứng máy tính

The CPU (1)

- PC Program Counter
- SP Stack Pointer
- PSW Program Status Word
- General Registers
- Instruction Cycle
- Pipeline
- Superscalar
- System Call

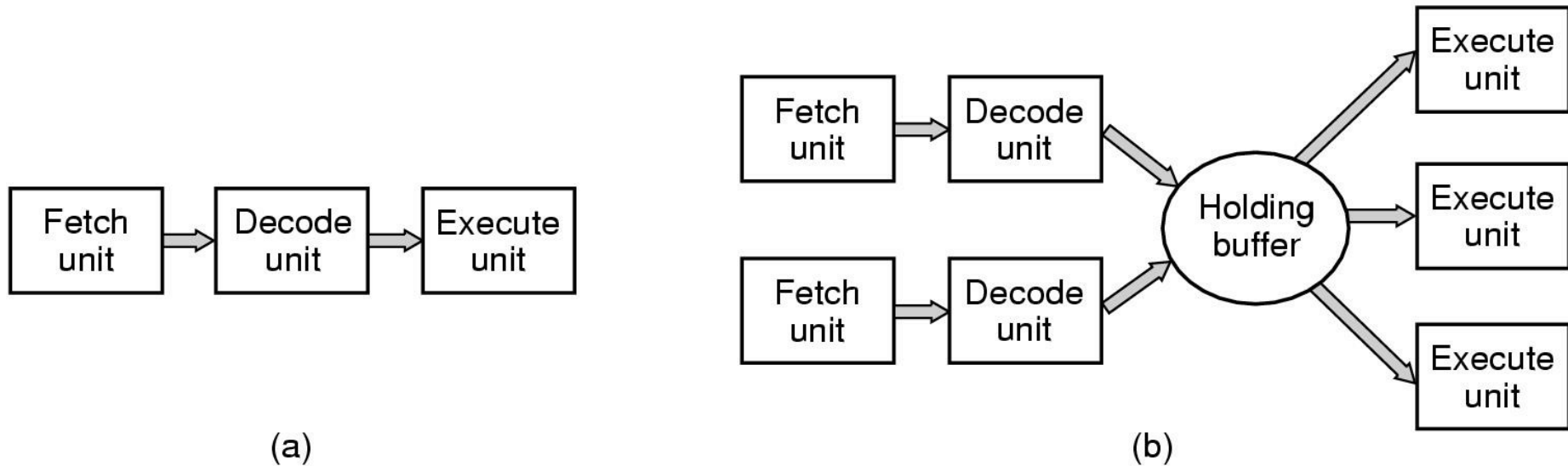
Sơ lược phần cứng máy tính

The CPU (2)



Sơ lược phần cứng máy tính

The CPU (3)



(a) A three-stage pipeline

(b) A superscalar CPU

Sơ lược phần cứng máy tính

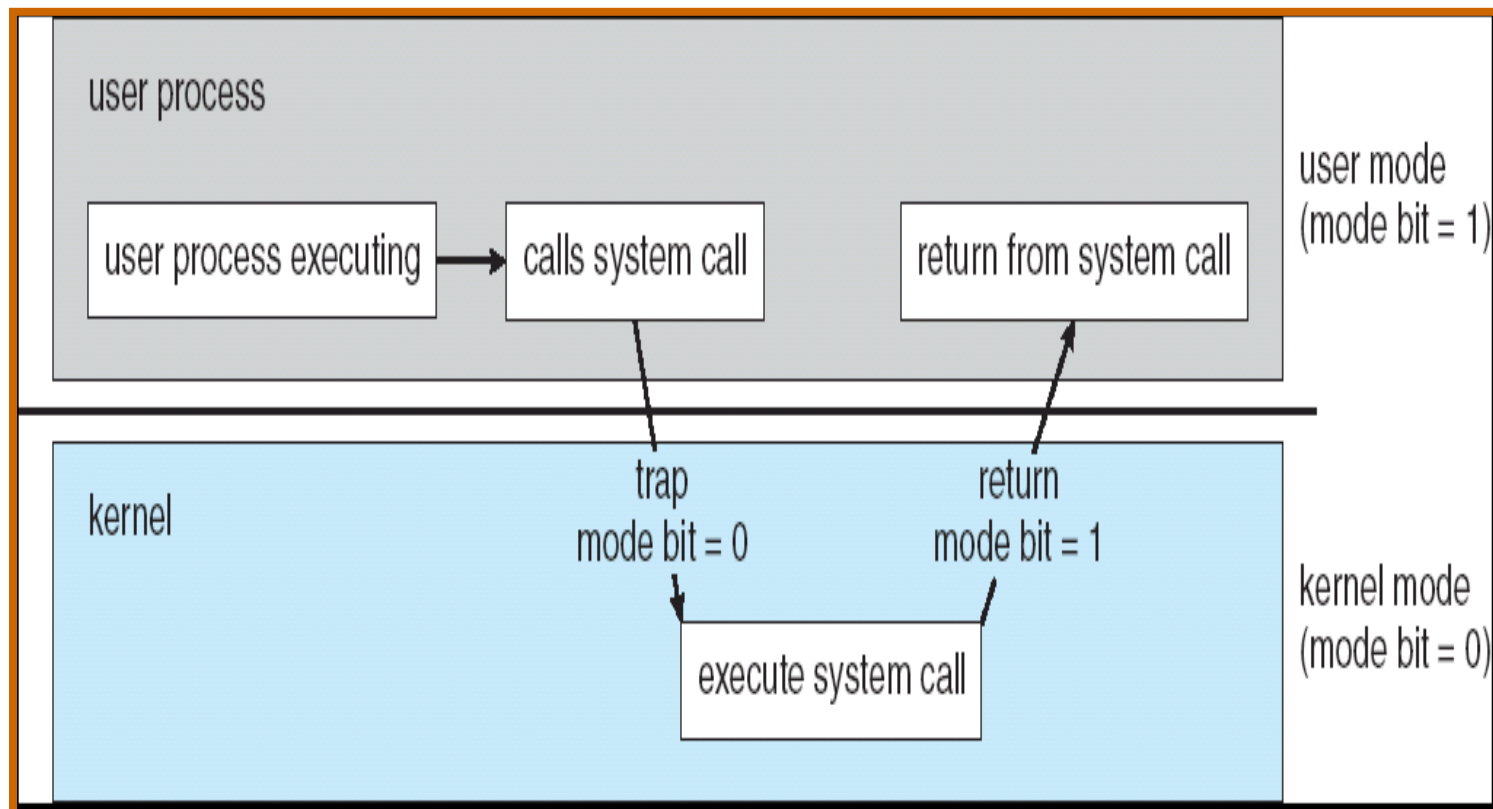
The CPU (4)

- **Điều hành chế độ kép (Dual-mode)** cho phép HĐH bảo vệ chính nó và các thành phần khác của hệ thống
 - **User mode** và **kernel mode**
 - **Mode bit** được cung cấp bởi phần cứng
 - Cung cấp khả năng phân biệt khi hệ thống đang chạy từ chế độ user sang chế độ kernel
 - Vài lệnh được thiết kế đặc quyền (**privileged**) chỉ có thể thực thi ở chế độ kernel
 - Các lời gọi hệ thống (System call) đổi sang chế độ kernel để thực hiện, sau đó quay trở lại chế độ user để thực hiện tiếp công việc ngay sau lời gọi hệ thống

Sơ lược phần cứng máy tính

The CPU (5)

- Chuyển đổi từ chế độ User sang chế độ Kernel

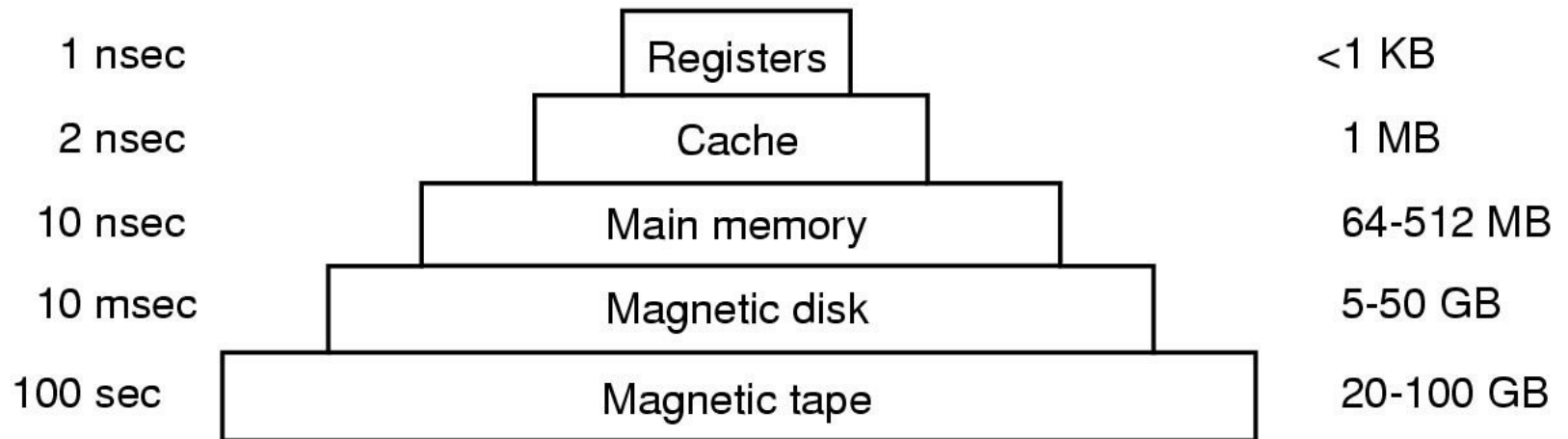


Sơ lược phần cứng máy tính

Memory (1)

Typical access time

Typical capacity



Sơ lược phần cứng máy tính

Memory (2)

- Các hệ thống lưu trữ được phân loại theo:
 - Tốc độ (Speed)
 - Giá thành (Cost)
 - Kích thước (Size)
 - Tính ổn định (Volatility)

Sơ lược phần cứng máy tính

Memory (3)

Các thanh ghi:

- Kích thước nhỏ, nhưng tốc độ nhanh
- Dung lượng:
 - 32*32 bit trên CPU 32-bit
 - 64*64 bit trên CPU 64-bit

Sơ lược phần cứng máy tính

Memory (4)

Caching – copy thông tin vào hệ thống lưu trữ nhanh hơn; bộ nhớ chính có thể được nhìn thấy ngay lần lưu trữ sau cùng đến thiết bị nhớ phụ.

Sơ lược phần cứng máy tính

Memory (5)

Caching

- Là tính năng quan trọng và được thực hiện ở nhiều lớp trong máy tính (trong phần cứng, hệ điều hành, phần mềm)
- Thông tin sử dụng được sao chép một cách tạm thời từ bộ nhớ chậm hơn sang bộ nhớ nhanh hơn
- Bộ nhớ nhanh hơn (Cache) kiểm tra trước để xác định xem có thông tin hay không
 - Nếu có, thông tin sẽ được dùng trực tiếp từ bộ nhớ cache
 - Nếu không, dữ liệu được copy đến bộ nhớ cache và được dùng ở đó
- Kích thước bộ nhớ Cache phải nhỏ hơn kho lưu trữ
 - Quản lý Cache là vấn đề quan trọng khi thiết kế
 - Kích thước Cache và chính sách thay thế

Sơ lược phần cứng máy tính

Memory (6)

Bộ nhớ chính (Main Memory)

- RAM (Random Access Memory)
- Vấn đề:
 - Làm thế nào để bảo vệ chương trình này từ chương trình khác và từ nhân của hệ điều hành
 - Làm thế nào để điều khiển tái cấp phát
- Giải pháp: CPU được trang bị 2 thanh ghi đặc biệt
- Thanh ghi Base (Base Register) và thanh ghi Limit (Limit Register)
- Thành phần Quản lý Bộ nhớ (MMU-Memory Management Unit) chuyển đổi địa chỉ ảo (Virtual Address) thành địa chỉ vật lý (Physical Address)
- Chuyển đổi ngữ cảnh; chuyển từ chương trình này sang chương trình khác

Sơ lược phần cứng máy tính

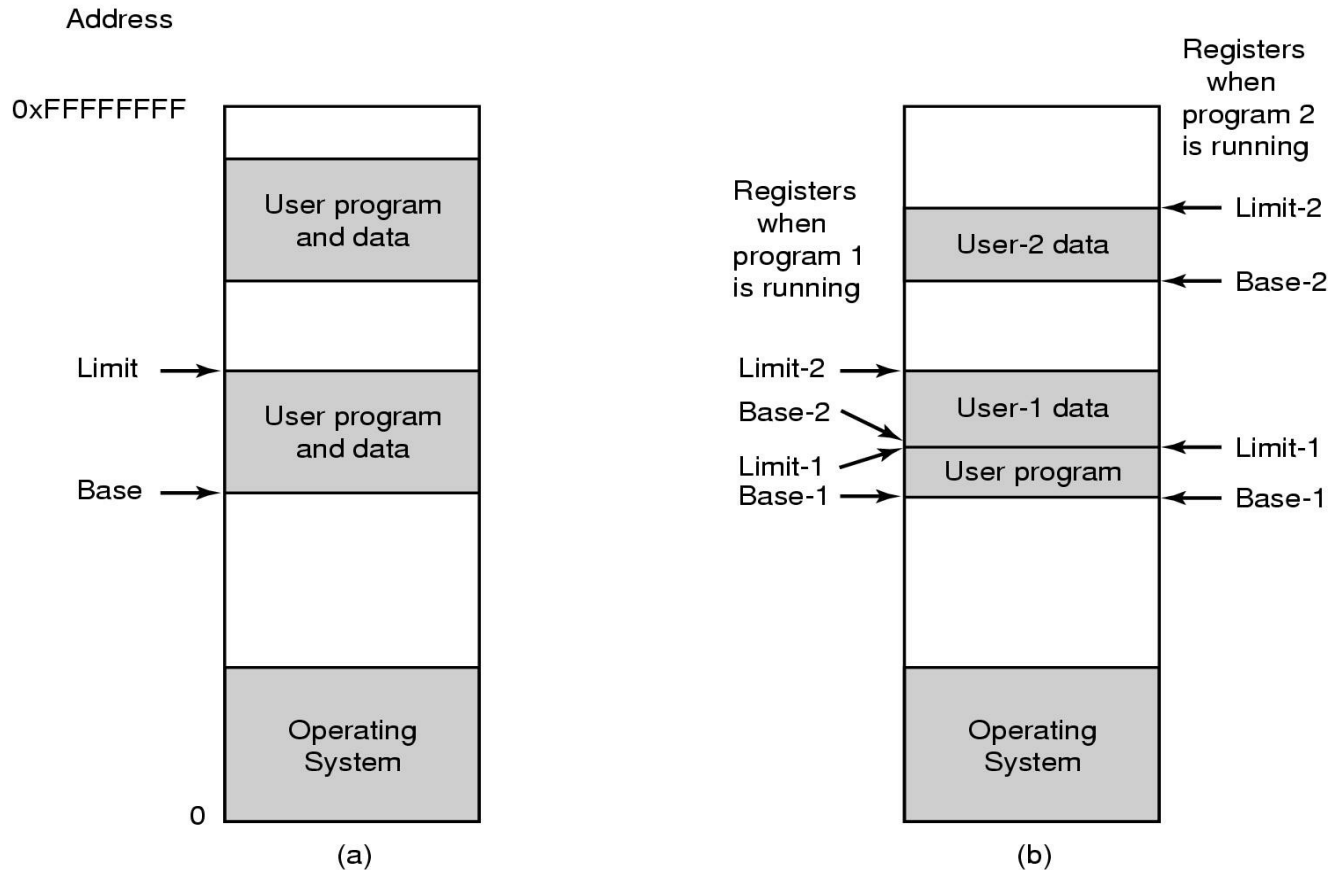
Memory (7)

Tái cấp phát và Bảo vệ

- Không thể được sử dụng nơi mà chương trình đã được nạp vào bộ nhớ
 - địa chỉ cấp phát cho các biến, đoạn code không được thay đổi
 - phải giữ cho chương trình nằm ngoài phân vùng của các tiến trình khác
- Sử dụng các giá trị base và limit
 - địa chỉ cấp phát được cộng với giá trị base để ánh xạ thành địa chỉ vật lý
 - địa chỉ cấp phát mà lớn hơn giá trị limit tức là lỗi

Sơ lược phần cứng máy tính

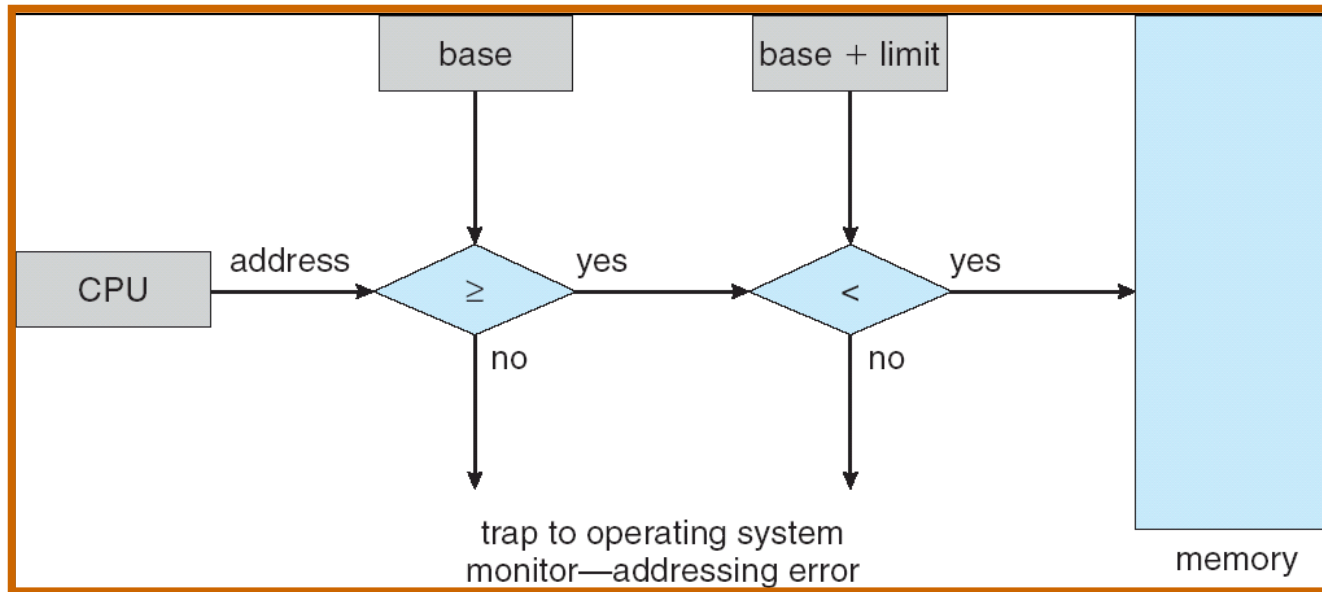
Memory (8)



Một cặp base-limit và hai cặp base-limit

Sơ lược phần cứng máy tính

Memory (9)



Sơ lược phần cứng máy tính

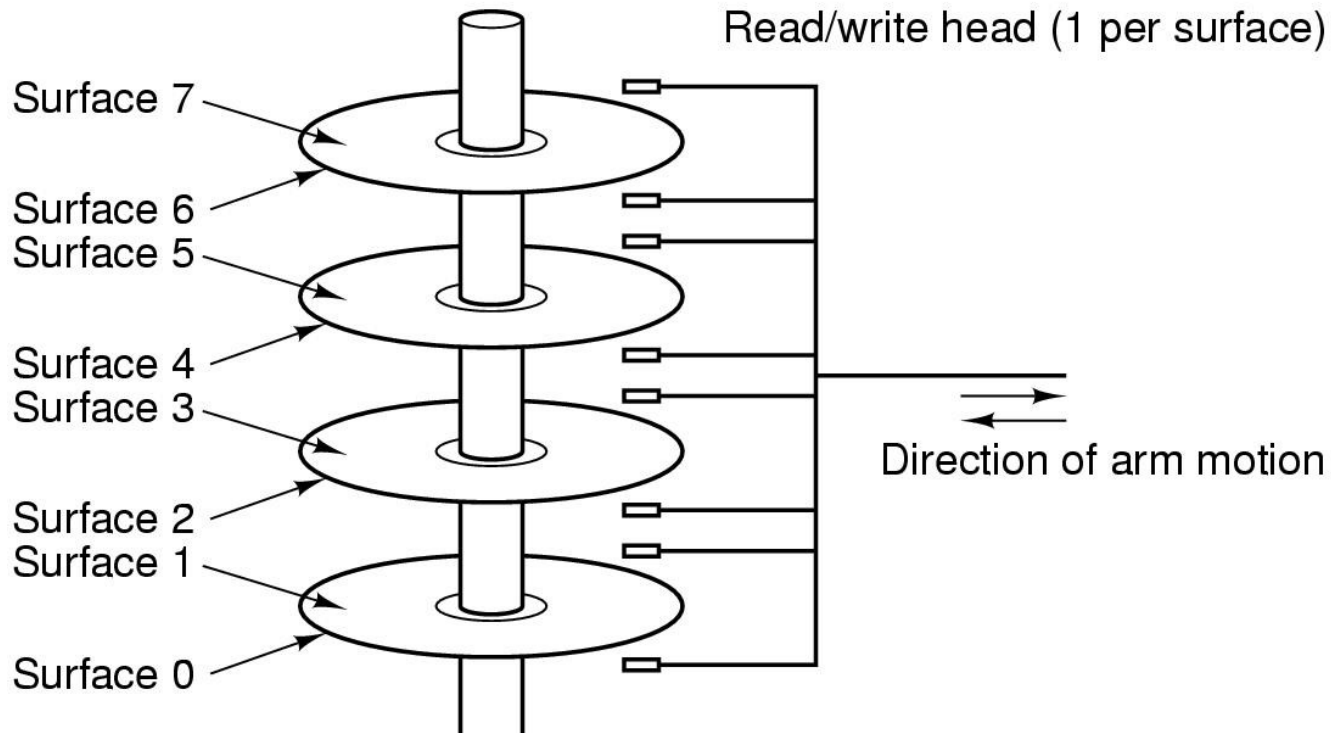
Memory (10)

Các bộ nhớ khác trong máy tính

- ROM (Read Only Memory)
- EEPROM (Electrically Erasable ROM):
 - BIOS Basic Input Output System
- CMOS:
 - Real time clock
 - Configuration Information

Sơ lược phần cứng máy tính

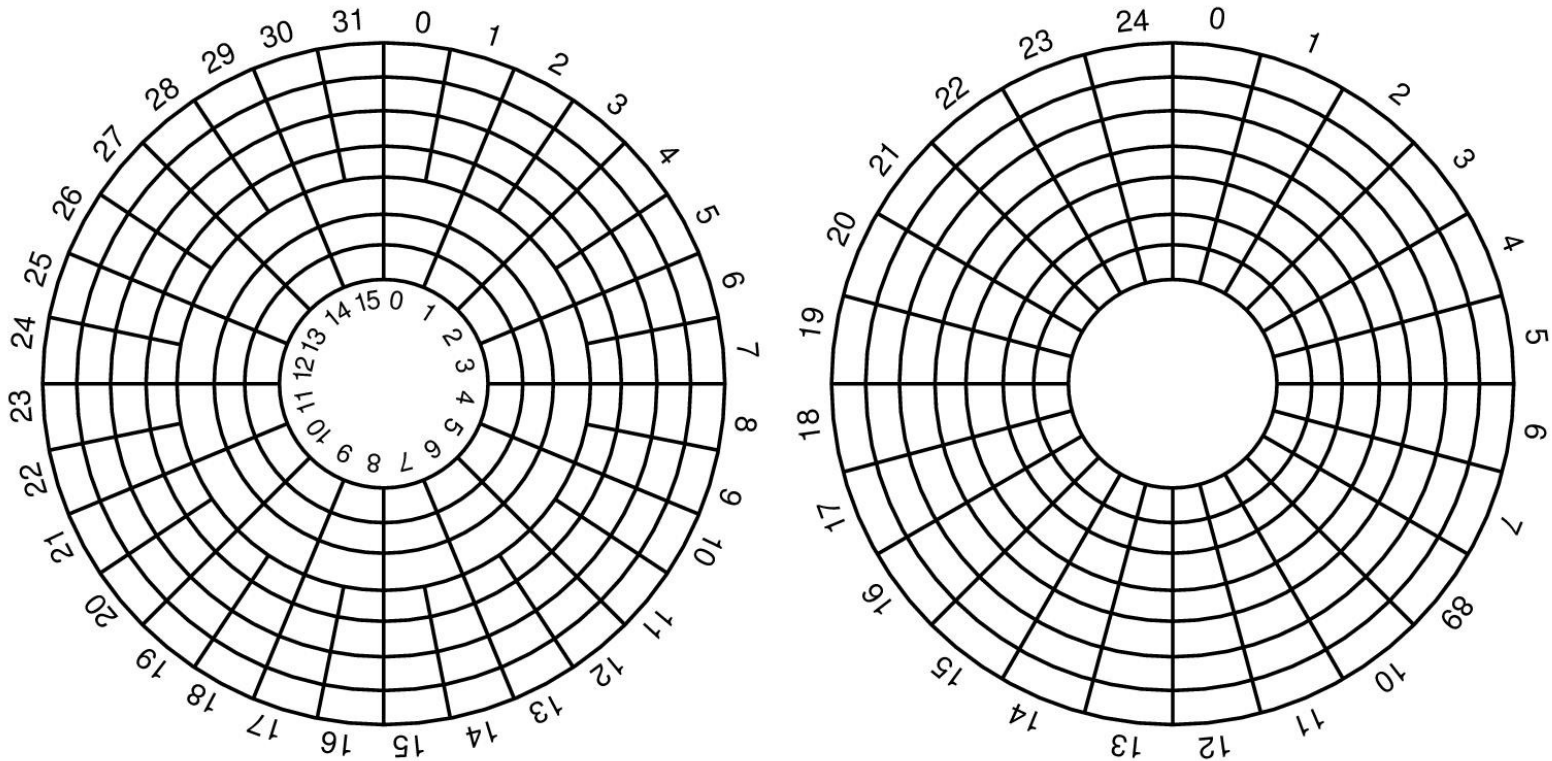
Memory (11)



Cấu trúc của ổ đĩa

Sơ lược phần cứng máy tính

Memory (12)



Đĩa (Disk): Cylinder, Track, sector

Sơ lược phần cứng máy tính

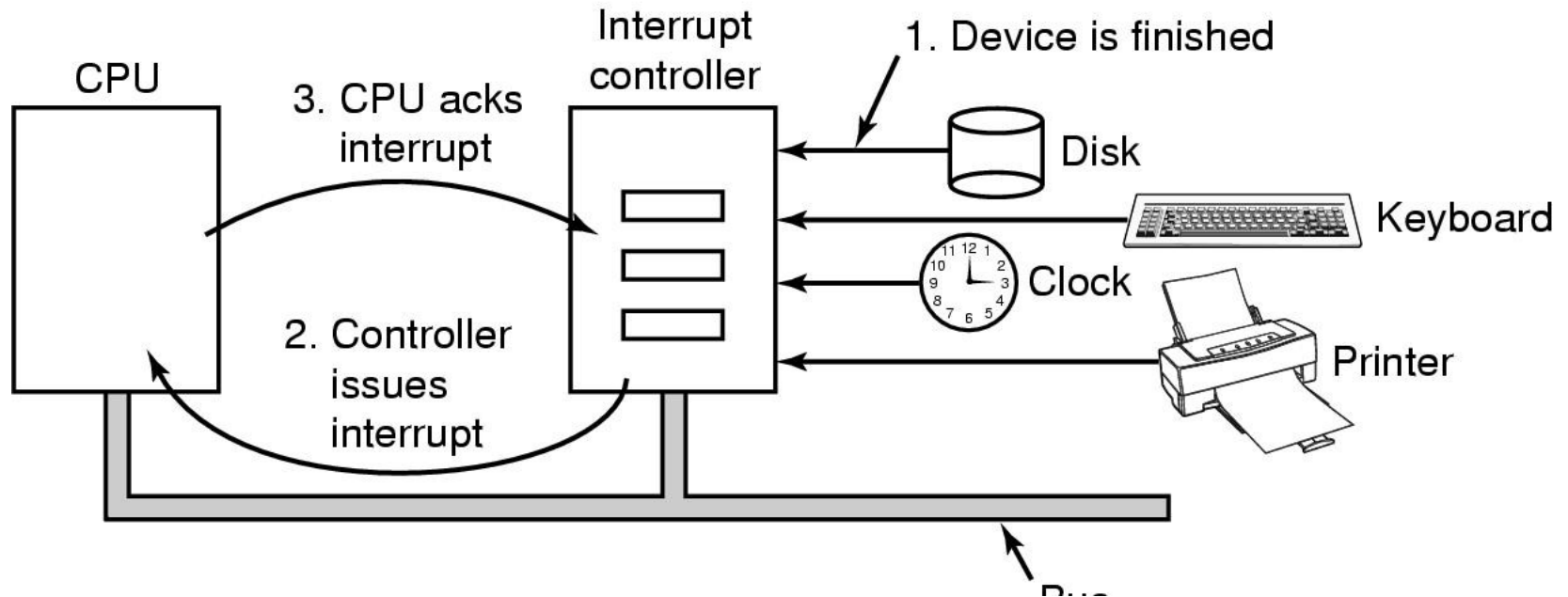
I/O Devices (1)

Điều khiển hệ thống máy tính

- Các thiết bị Vào/Ra (I/O devices) và CPU có thể thực hiện một cách đồng thời.
- Mỗi bộ điều khiển thiết bị (device controller) phụ trách một loại thiết bị cụ thể.
- Mỗi bộ điều khiển thiết bị có một bộ đệm trong (local buffer).
- CPU chuyển dữ liệu từ/đến bộ nhớ chính đến/từ các bộ đệm trong.
- Vào/Ra là từ thiết bị đến bộ đệm của bộ điều khiển thiết bị.
- Bộ điều khiển thiết bị báo cho CPU biết rằng dữ liệu đã được hoàn thành bằng cách tạo ra một ngắt (*interrupt*).

Sơ lược phần cứng máy tính

I/O Devices (2)

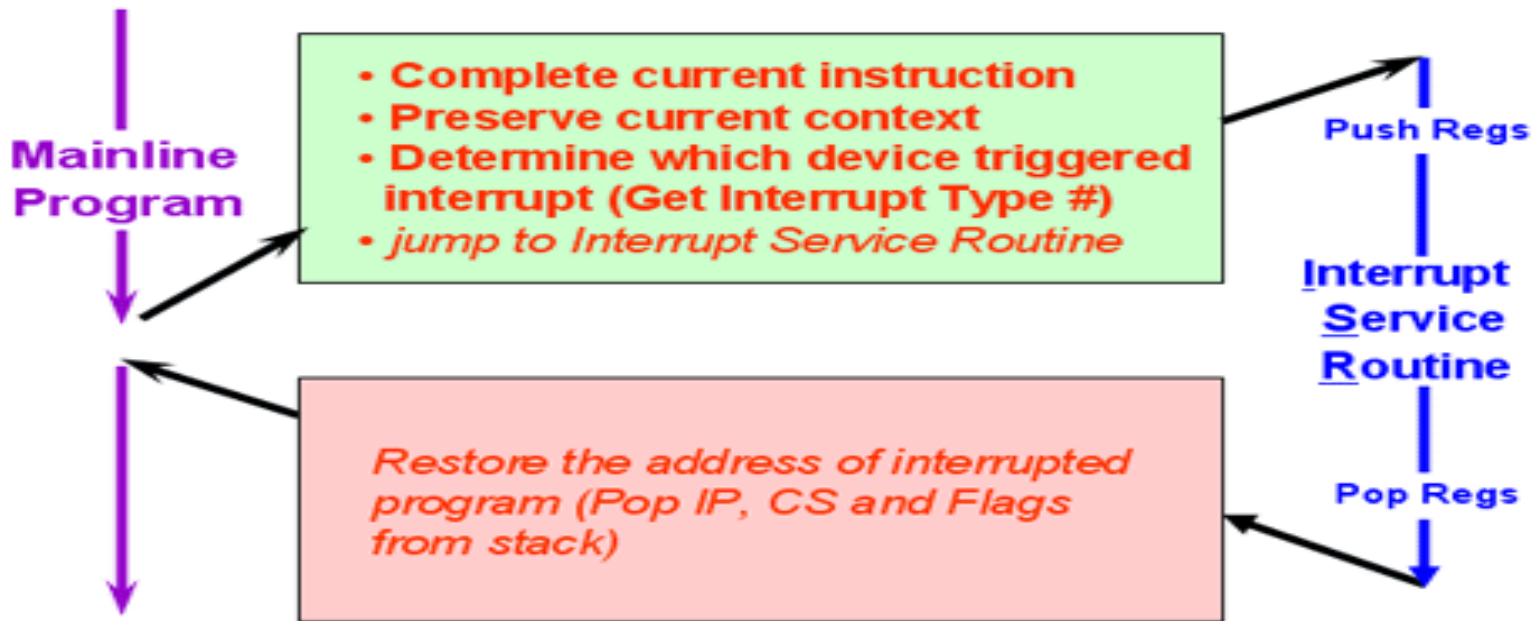


Làm thế nào để ngắt xảy ra. Các kết nối giữa thiết bị và bộ điều khiển ngắt thực sự thông qua đường ngắt trên bus hơn là các đường điện chuyên dụng

Sơ lược phần cứng máy tính

I/O Devices (3)

Ví dụ về xử lý Ngắt trên PC



Sơ lược phần cứng máy tính

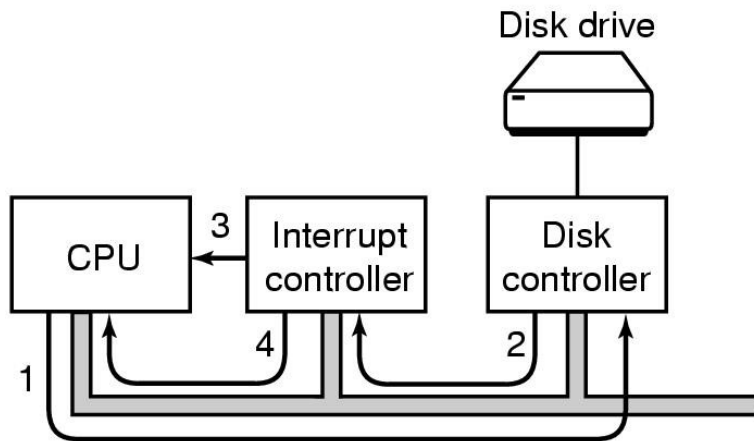
I/O Devices (3)

Các chức năng chung của Ngắt

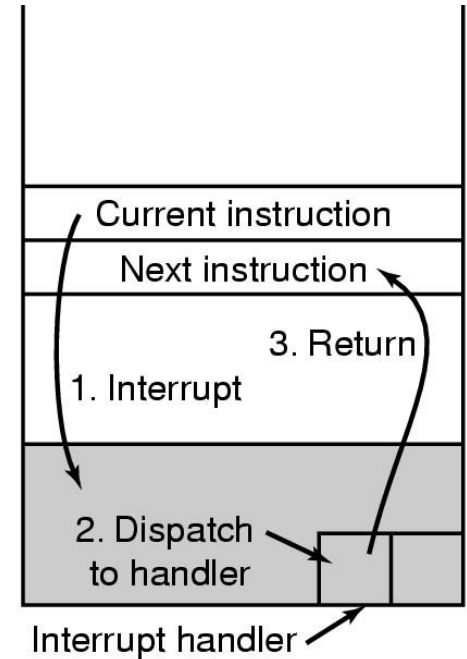
- Ngắt truyền điều khiển đến chương trình phục vụ ngắt (**interrupt service routine**) thông qua bảng vector ngắt, là nơi chứa các địa chỉ của các chương trình phục vụ ngắt.
- Kiến trúc ngắt phải lưu lại tất cả các địa chỉ của lệnh ngắt.
- Các Ngắt đến sẽ bị vô hiệu hóa trong khi một ngắt khác đang được xử lý để ngăn chặn thất lạc ngắt.
- Trap là một ngắt mềm được sinh ra hoặc do lỗi hay do một yêu cầu của người dùng.
- Hệ điều hành cũng là một bộ điều khiển ngắt.

Sơ lược phần cứng máy tính

I/O Devices (4)



(a)

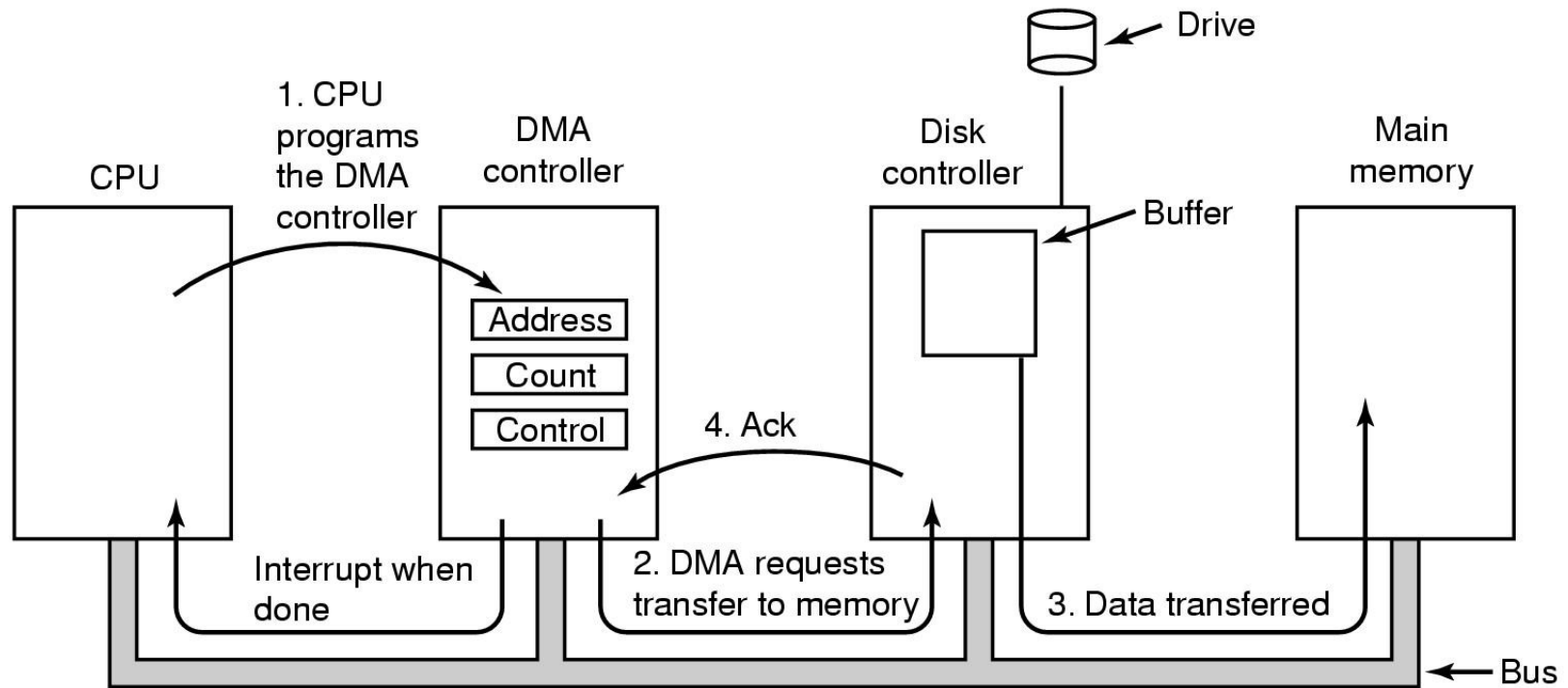


(b)

- (a) Các bước khởi tạo thiết bị Vào/Ra và bắt được ngắt
- (b) Làm thế nào để CPU bị ngắt

Sơ lược phần cứng máy tính

I/O Devices (6)



Thao tác của bộ chuyển đổi DMA
DMA (Direct Memory Access)

Sơ lược phần cứng máy tính

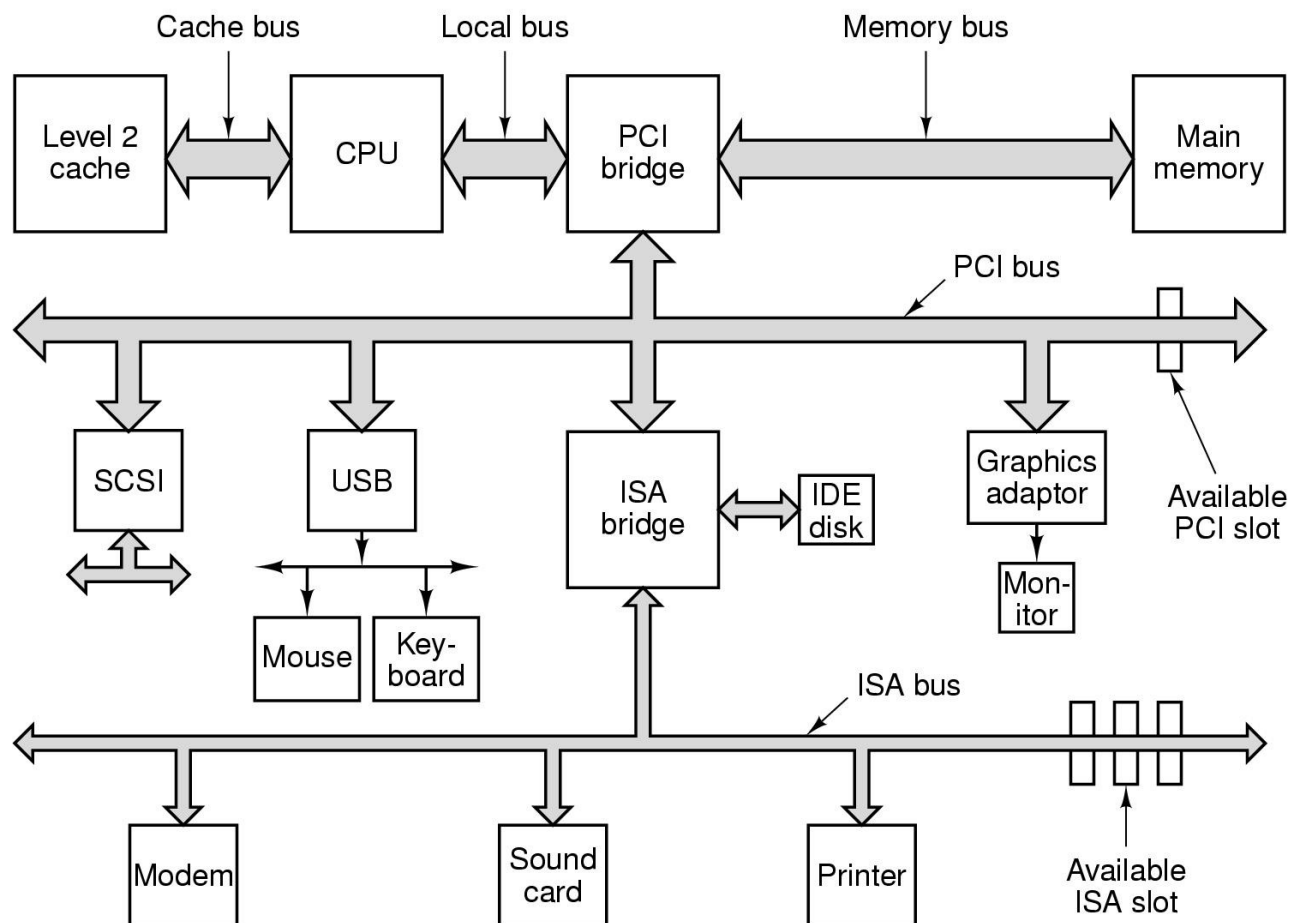
I/O Devices (7)

DMA

- Được dùng cho các thiết bị Vào/Ra tốc độ cao có thể truyền thông tin nhanh gần bằng với tốc độ của bộ nhớ.
- Bộ điều khiển thiết bị truyền các khối dữ liệu từ bộ đệm của thiết bị trực tiếp vào bộ nhớ chính mà không cần CPU.
- Chỉ một ngắt được tạo ra trên một khối, hơn là một ngắt trên một byte.

Sơ lược phần cứng máy tính

Cấu trúc bus của hệ thống Pentium



Sơ lược phần cứng máy tính

BUS

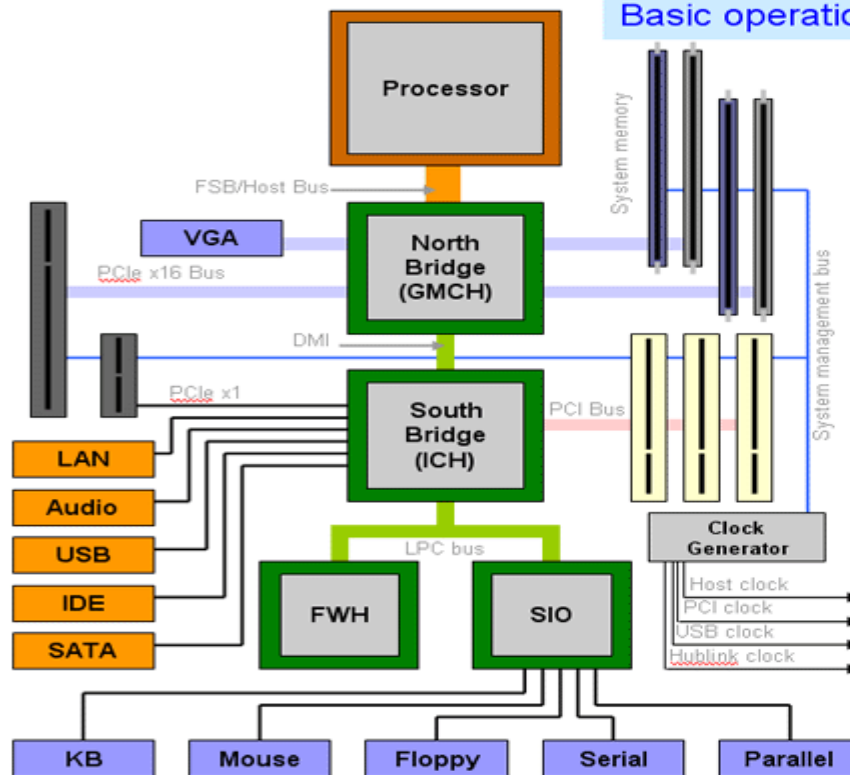
- Cache BUS
- Local BUS
- System BUS
- ISA Industry Standard Architecture
- PCI Peripheral Component Interconnect
- USB Universal Serial BUS
- SCSI Small Computer System Interface
- IDE Integrated Drive Electronic

Sơ lược phần cứng máy tính

Personal Computer (PC)

Chipset: North Bridge, South Bridge, & Firmware Hub

Basic operational design is called its architecture



- North Bridge
 - HOST-PCI Bridge
 - DRAM Controller
 - AGP Interface
- South Bridge (ICH)
 - PCI-PCI Bridge
 - 8254 Timer; 8259 Intr. Ctlr; 8237 DMAC; RTC
 - EIDE Interface
 - USB Ctlr
- Super I/O
 - Kybd/Mouse Ctlr
 - Floppy Ctlr
 - Serial & IR port
 - Parallel ports

1.3 Các khái niệm về Hệ điều hành

- Các thành phần của Hệ điều hành
- Các lời gọi hệ thống (System calls)
- Cấu trúc của Hệ điều hành

Các thành phần của HĐH

- Quản lý Tiến trình (Process Management)
- Quản lý Bộ nhớ (Main Memory Management)
- Quản lý File (File management)
- Quản lý Vào/Ra (I/O Management)
- Quản lý lưu trữ (Secondary Storage Management)
- Bảo vệ và An toàn (Protection and Security)
- Hệ vỏ (Shell)

Các khái niệm của HĐH

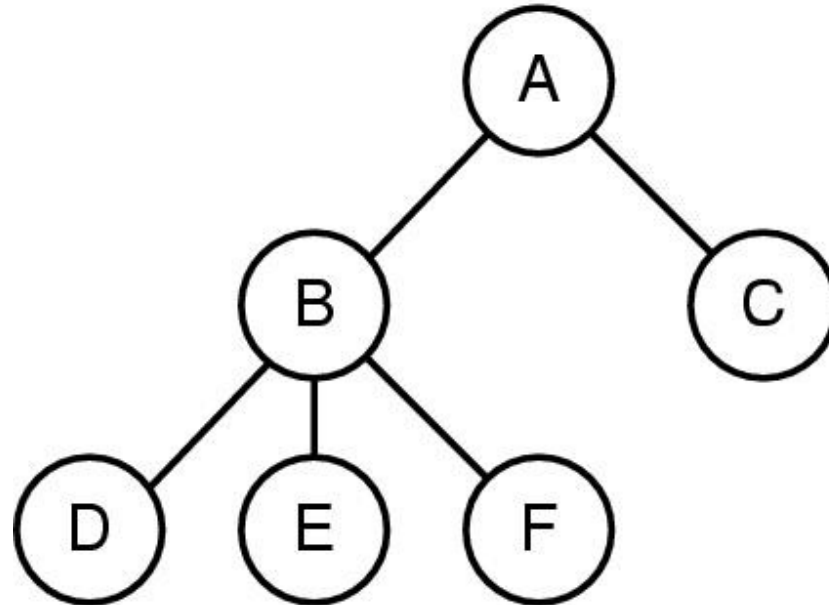
Các thành phần của HĐH (1)

- **Quản lý Tiến trình**

- Tiến trình là một chương trình đang thực hiện
- Một tiến trình cần một tập các tài nguyên, gồm thời gian CPU, bộ nhớ, các file, và các thiết bị Vào/Ra để hoàn thành tác vụ của nó.
- Các tác vụ của thành phần quản lý Tiến trình của Hệ điều hành:
 - Tạo lập và xóa Tiến trình.
 - Tạm dừng và tái kích hoạt Tiến trình
 - Cung cấp các cơ chế để:
 - Đồng bộ giữa các Tiến trình
 - Giao tiếp liên Tiến trình (Interprocess communication)
 - Ngăn chặn và phòng tránh bế tắc (deadlock)

Các khái niệm của HĐH

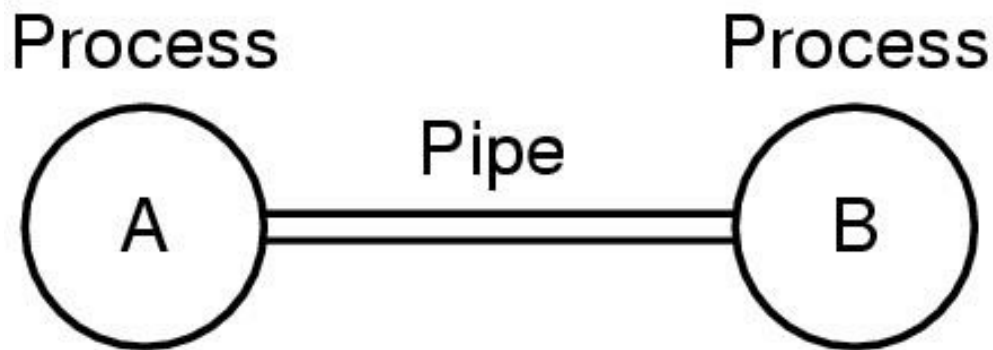
Các thành phần của HĐH (2)



- Một cây Tiến trình
 - A tạo ra 2 tiến trình con B và C
 - B tạo ra 2 tiến trình con D, E, và F

Các khái niệm của HĐH

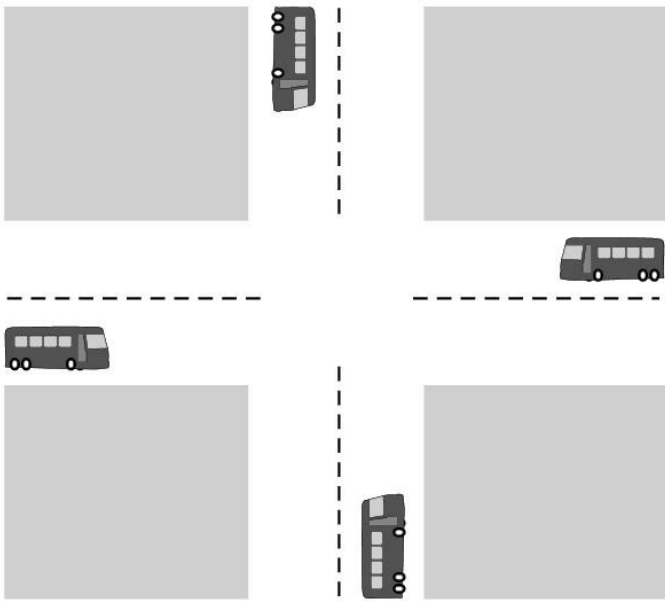
Các thành phần của HĐH (3)



Hai tiến trình kết nối bởi một đường ống (pipe)

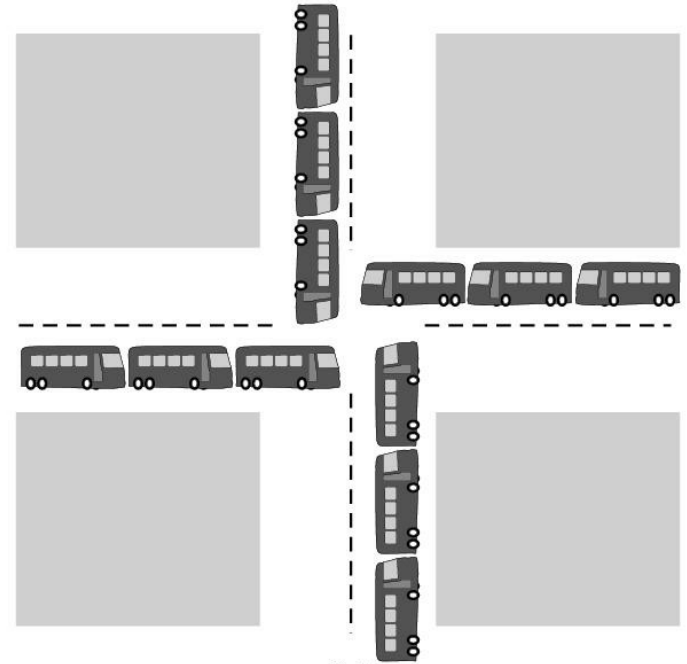
Các khái niệm của HĐH

Các thành phần của HĐH (4)



(a)

(a) Gần deadlock.



(b)

(b) deadlock.

Các khái niệm của HĐH

Các thành phần của HĐH (5)

Vấn đề bế tắc

- Một tập các Tiến trình bị chặn lại chiếm giữ một tài nguyên và đang chờ cấp phát tài nguyên đã bị chiếm giữ bởi một tiến trình khác trong tập các tiến trình bị chặn đó.
- Ví dụ
 - Hệ thống có 2 ổ đĩa.
 - P_1 và P_2 là 2 tiến trình, mỗi tiến trình chiếm giữ một ổ đĩa và mỗi tiến trình lại cần ổ đĩa kia.

Các khái niệm của HĐH

Các thành phần của HĐH (6)

- **Quản lý bộ nhớ chính:**

- Động cơ:

- Tăng hiệu năng hệ thống
 - Tối ưu hóa việc sử dụng bộ nhớ

- Tác vụ của thành phần quản lý bộ nhớ chính:

- Theo dõi những phần bộ nhớ nào hiện đang được sử dụng và bởi ai.
 - Quyết định tiến trình nào được nạp vào bộ nhớ khi có vùng nhớ giải phóng.
 - Cấp phát và tái cấp phát vùng nhớ khi cần

Các khái niệm của HĐH

Các thành phần của HĐH (7)

- **Quản lý File**

- Hệ thống File

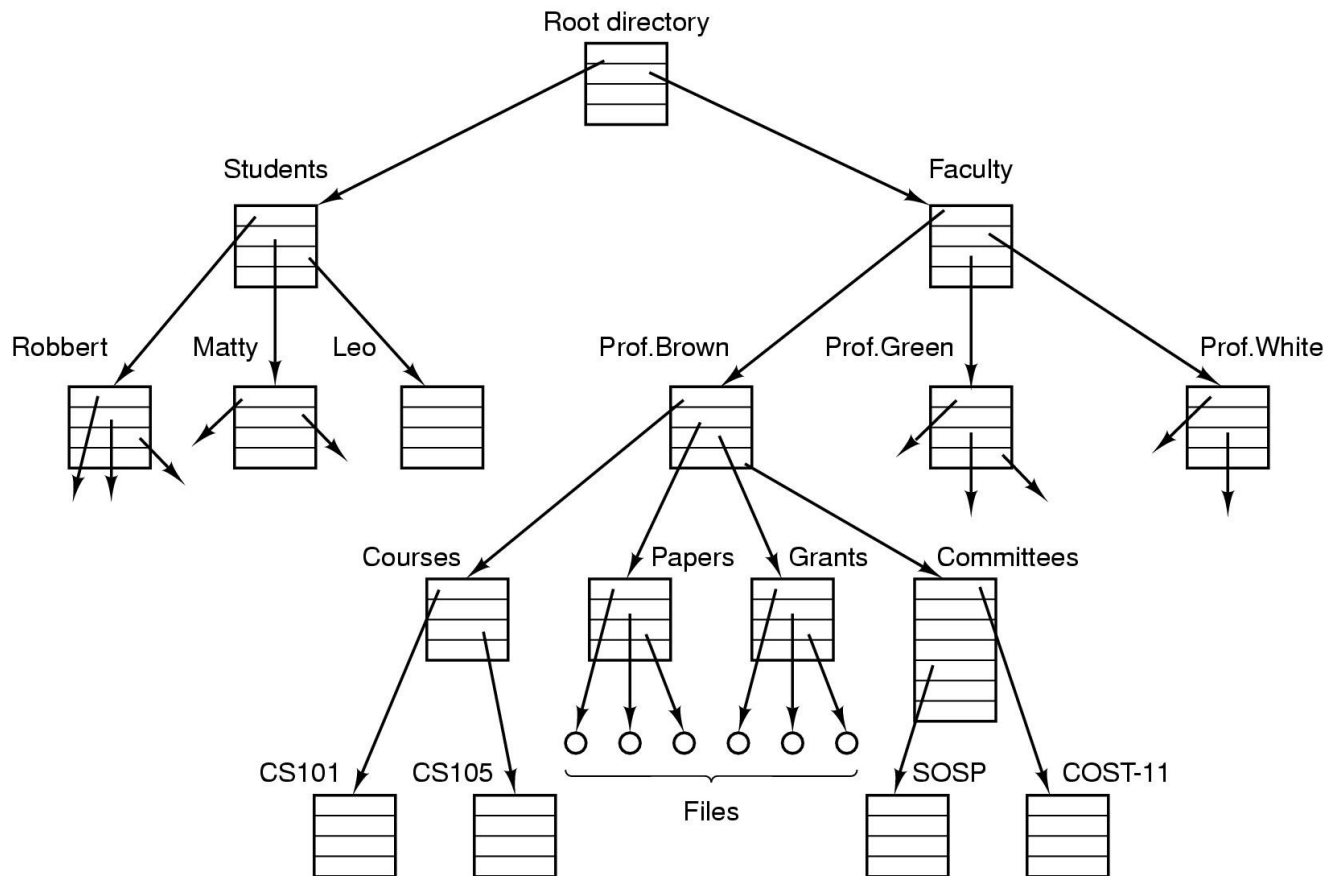
- File
 - Directory

- Các tác vụ của thành phần quản lý file của HĐH:

- Create và delete File/Dicrectory
 - Các thao tác: rename, copy, move, new,...
 - Ảnh xạ các files vào secondary storage.
 - Backup File lên (nonvolatile) storage media

Các khái niệm của HĐH

Các thành phần của HĐH (8)



Hệ thống File dành cho một khoa của trường

Các khái niệm của HĐH

Các thành phần của HĐH (9)

- **Quản lý Vào/Ra**

- Ẩn các chi tiết phức tạp của thiết bị Vào/Ra, chỉ cung cấp cho người dùng giao diện tiện lợi
- Tác vụ của thành phần Quản lý Vào/Ra của HĐH:
 - Quản lý vùng nhớ chính cấp cho các thiết bị dùng để caching, buffering, và spooling
 - Duy trì và cung cấp giao diện chung cho trình điều khiển thiết bị (device driver)
 - Trình điều khiển thiết bị cho thiết bị phần cứng cụ thể.

Các khái niệm của HĐH

Các thành phần của HĐH (10)

- **Quản lý thiết bị lưu trữ:**
 - Khi bộ nhớ chính (main memory) chỉ tạm thời và quá nhỏ để chứa tất cả dữ liệu và chương trình, hệ thống máy tính cần phải cung cấp bộ nhớ phụ để back up bộ nhớ chính.
 - Các thiết bị nhớ phụ phổ biến: Đĩa từ (Magnetic disk), đĩa quang (Optical disk), ...
 - Tác vụ của thành phần quản lý bộ nhớ phụ của HĐH:
 - Quản lý không gian trống
 - Cấp phát lưu trữ
 - Lập lịch Đĩa (Disk scheduling)

Các khái niệm của HĐH

Các thành phần của HĐH (11)

Bảo vệ và An toàn

- **Bảo vệ (Protection)** – bất kỳ cơ chế dành cho việc điều khiển truy cập của các Tiến trình và các User đến các tài nguyên đều được định nghĩa bởi Hệ điều hành.
- **An toàn (Security)** – bảo vệ hệ thống chống lại các tấn công từ bên trong và bên ngoài
 - Trong phạm vi rộng lớn, bao gồm từ chối dịch vụ, sâu, virus, đánh cắp danh tính, trộm dịch vụ,...

Các khái niệm của HĐH

Các thành phần của HĐH (12)

Bảo vệ và An toàn

- Hệ thống phân biệt trong số các User một cách chung chung, để chỉ rõ hệ thống cần biết được Ai có thể thực hiện và làm gì
 - Định danh của User (**user IDs**, security IDs) gồm Tên (name) và số liên kết (associated number), cho mỗi user
 - User ID sau đó được liên kết với tất cả các file, các tiến trình của User đó để kiểm soát truy cập
 - Định danh nhóm (**group ID**) cho phép tập các User được định nghĩa và quản lý các điều khiển, sau đó cũng liên kết với Tiến trình, file
 - Tăng quyền ưu tiên (**Privilege escalation**) cho phép User đổi thành ID hữu dụng hơn với nhiều quyền hơn

Các khái niệm của HĐH

Các dịch vụ của Hệ điều hành

- **Tập các dịch vụ của Hệ điều hành cung cấp các chức năng hữu ích cho người dùng:**
 - Giao diện người dùng (User interface) - Hầu hết các hệ điều hành đều có giao diện người dùng (UI)
 - Khác nhau giữa dòng lệnh (Command-Line -CLI), với giao diện đồ họa (Graphics User Interface -GUI)
 - Một chương trình thực thi - Hệ thống cần phải tải chương trình vào bộ nhớ và chạy thực hiện chương trình, kết thúc xử lý, ngay cả khi bình thường hoặc bất thường (chỉ ra lỗi)
 - Điều khiển truy cập đến thiết bị Vào/Ra
 - Thao tác với hệ thống File

Các khái niệm của HĐH

Các dịch vụ của Hệ điều hành

- Các giao tiếp (Communications) – Các tiến trình có thể trao đổi thông tin, trên một máy hay trên các máy thông qua mạng
- Lỗi kiểm tra – Hệ điều hành cần phải liên tục nhận được thông báo nếu có lỗi
-

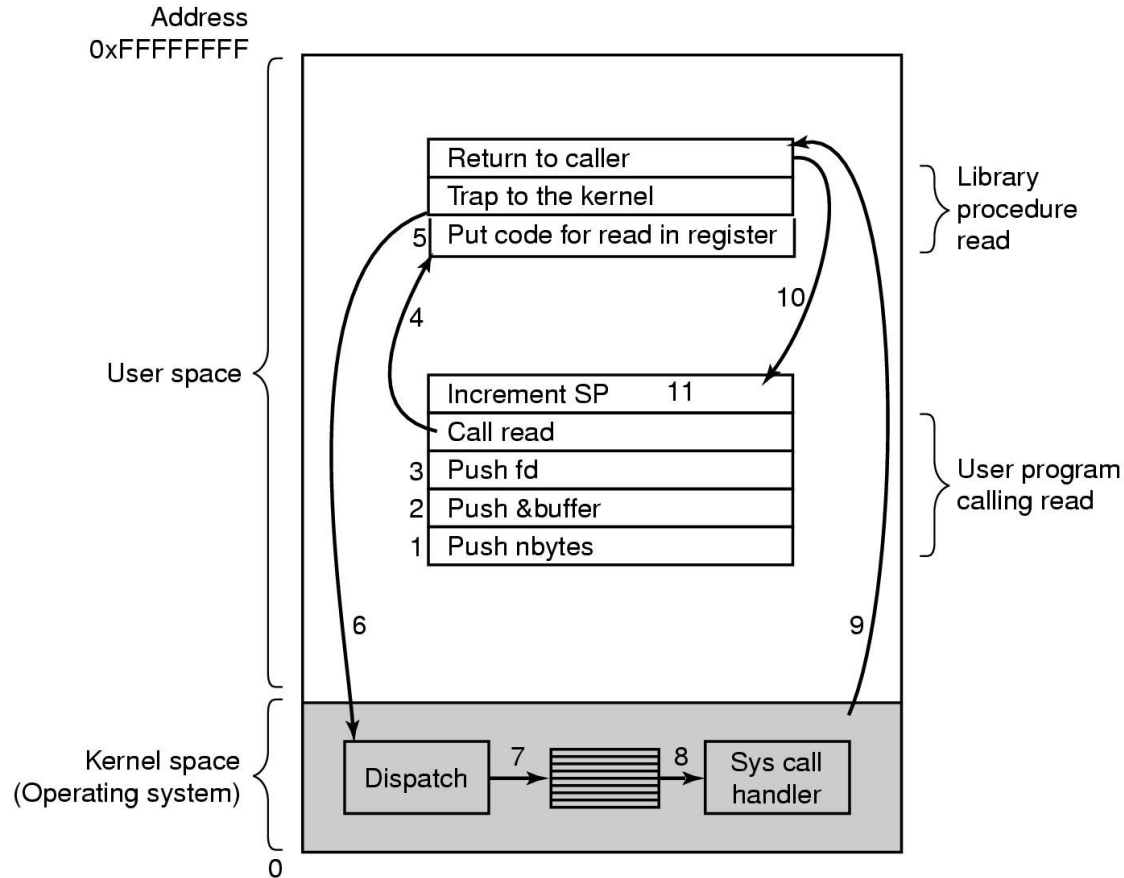
Các lời gọi hệ thống

System calls

- Thực hiện các lời gọi hệ thống
- Các lời gọi hệ thống chính của POSIX
- Các ví dụ

System Calls

Thực hiện các System Call (1)

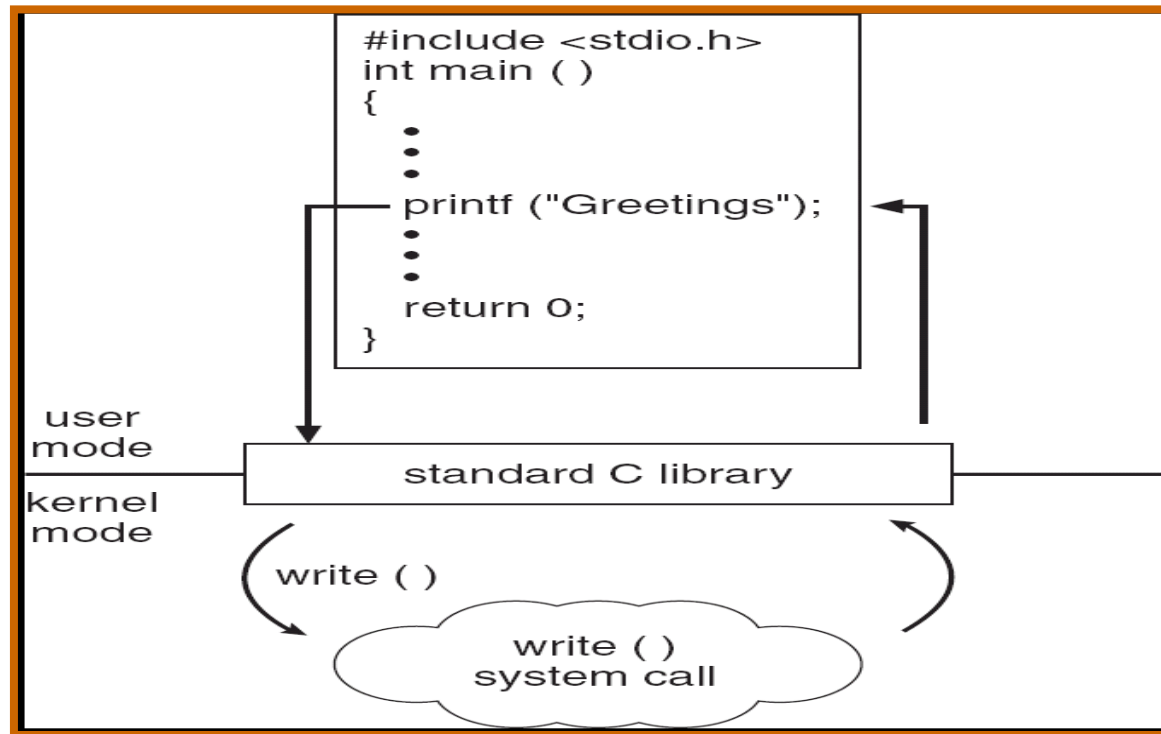


11 bước thực hiện lời gọi hệ thống
read (fd, buffer, nbytes)

System Calls

Thực hiện các System Call (2)

- Chương trình C gọi hàm thư viện printf(), hàm thư viện gọi lời gọi hệ thống write()



System Calls

Lời gọi hệ thống của POSIX (1)

Lời gọi hệ thống của thành phần Quản lý Process
POSIX (Portable Operating System Interface)

Process management	
Call	Description
<code>pid = fork();</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status
File management	
Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information
Directory and file system management	
Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system
Miscellaneous	
Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

System Calls

Lời gọi hệ thống của POSIX (2)

- Lời gọi hệ thống của thành phần Quản lý File

Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

System Calls

Lời gọi hệ thống của POSIX (3)

Lời gọi hệ thống của thành phần Quản lý Directory

Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

System Calls

Lời gọi hệ thống của POSIX (4)

Một số lời gọi hệ thống của Miscellaneous Tasks

Process management	
Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status
File management	
Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information
Directory and file system management	
Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system
Miscellaneous	
Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

System Calls

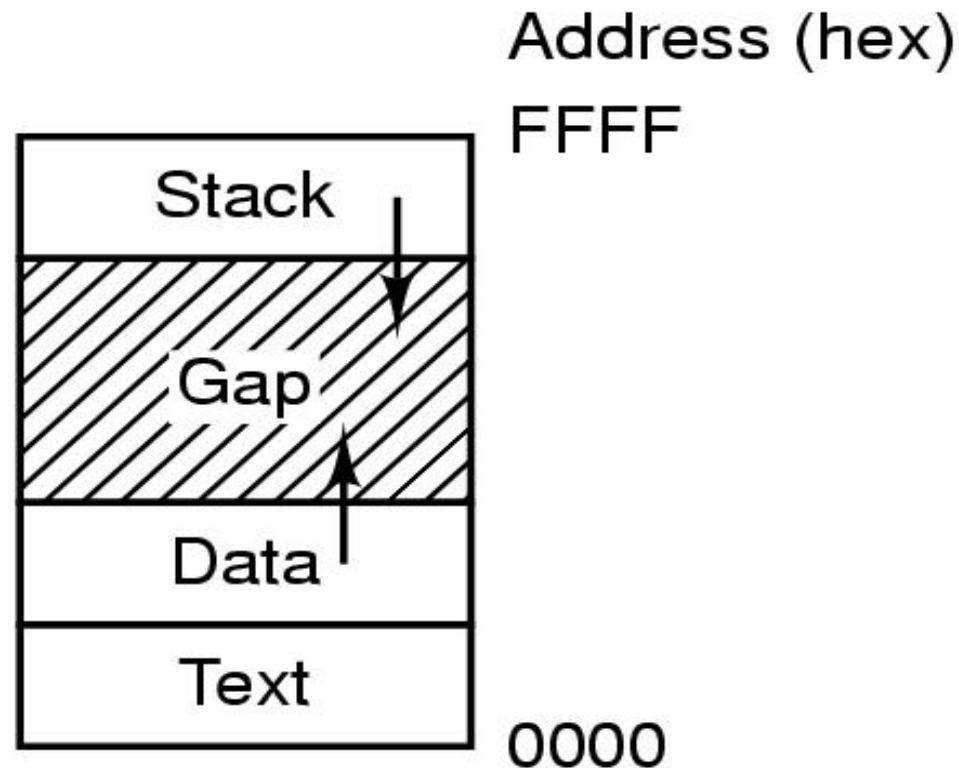
Ví dụ (1)

- Một đoạn chương trình hệ vỏ:

```
while (TRUE) {                                /* repeat forever */
    type_prompt( );                            /* display prompt */
    read_command (command, parameters)        /* input from terminal
    */
    if (fork() != 0) {                         /* fork off child process */
        /* Parent code */
        waitpid( -1, &status, 0);             /* wait for child to exit */
    } else {
        /* Child code */
        execve (command, parameters, 0);      /* execute command */
    }
}
```

System Calls

Ví dụ (2)



- Tiến trình có 3 đoạn: text, data, stack

System Calls

Ví dụ (3)

/usr/ast		/usr/jim	
16	mail	31	bin
81	games	70	memo
40	test	59	f.c.
		38	prog1

(a)

/usr/ast		/usr/jim	
16	mail	31	bin
81	games	70	memo
40	test	59	f.c.
70	note	38	prog1

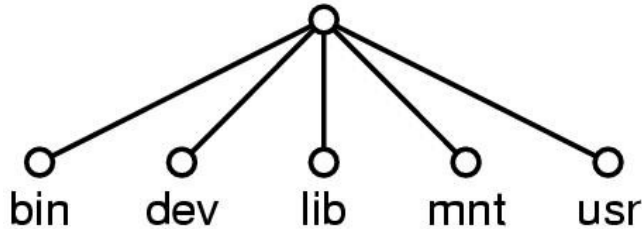
(b)

(a) Hai thư mục trước khi liên kết
/usr/jim/memo đến thư mục ast

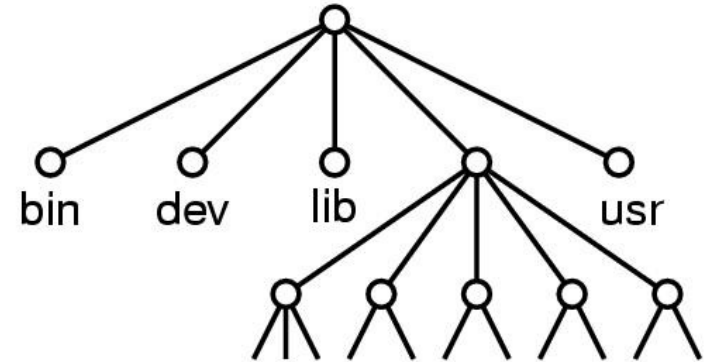
(b) Cùng thư mục sau khi liên kết

System Calls

Ví dụ (4)



(a)



(b)

(a) Hệ thống File trước khi mount

(b) Hệ thống File sau khi mount

System Calls

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

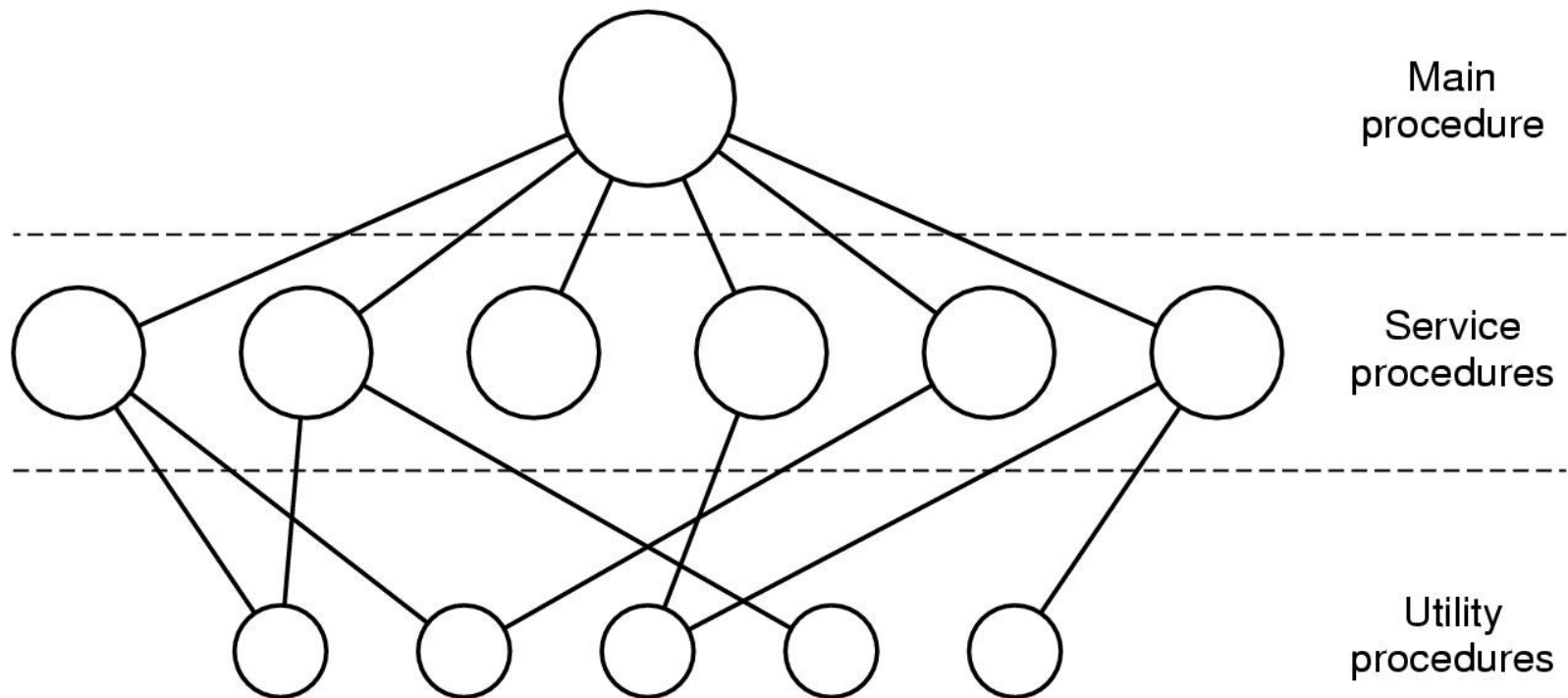
Một số lời gọi của Win32 API

Cấu trúc của Hệ điều hành

- Hệ thống nguyên khối - Monolithic system
- Hệ thống phân lớp - Layered System
- Máy ảo - Virtual Machine
- Mô hình Client-Server - Client-server model
- Microkernel

Cấu trúc của Hệ điều hành

Hệ thống nguyên khối (1)



Mô hình cấu trúc đơn giản của hệ thống nguyên khối

Cấu trúc của Hệ điều hành

Hệ thống nguyên khối (2)

Cấu trúc của Hệ điều hành:

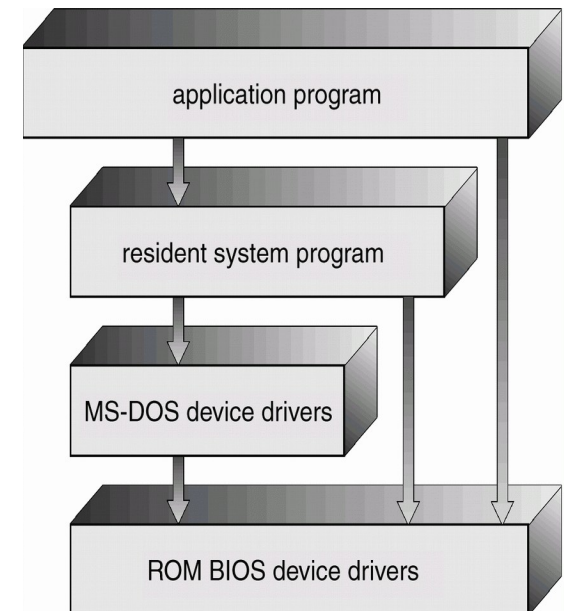
- Một chương trình chính có thể gọi yêu cầu đến thủ tục phục vụ (service procedure).
- Một tập thủ tục phục vụ thực hiện các lời gọi hệ thống.
- Một tập các thủ tục tiện ích (utility procedure) trợ giúp cho các thủ tục phục vụ.

Cấu trúc của Hệ điều hành

Hệ thống nguyên khối (3): Ví dụ

- Monolithic

- MS-DOS – được viết để cung cấp các chức năng ít tốn bộ nhớ nhất:
 - không chia thành các module;
 - Mặc dù MS-DOS có một vài cấu trúc, nhưng giao diện và các chức năng ở mức thấp thì không phân chia tốt.



Cấu trúc của Hệ điều hành

Hệ thống phân lớp (1)

- Bao gồm nhiều lớp
- Mỗi lớp được định nghĩa rõ các hàm của lớp đó
- Lớp trên có thể gọi đến các hàm của lớp dưới gần nó
- Ích lợi:
 - Dễ dàng mở rộng
 - Dễ dàng gỡ lỗi từ lớp thấp lên lớp cao

Cấu trúc của Hệ điều hành

Hệ thống phân lớp (2): Ví dụ



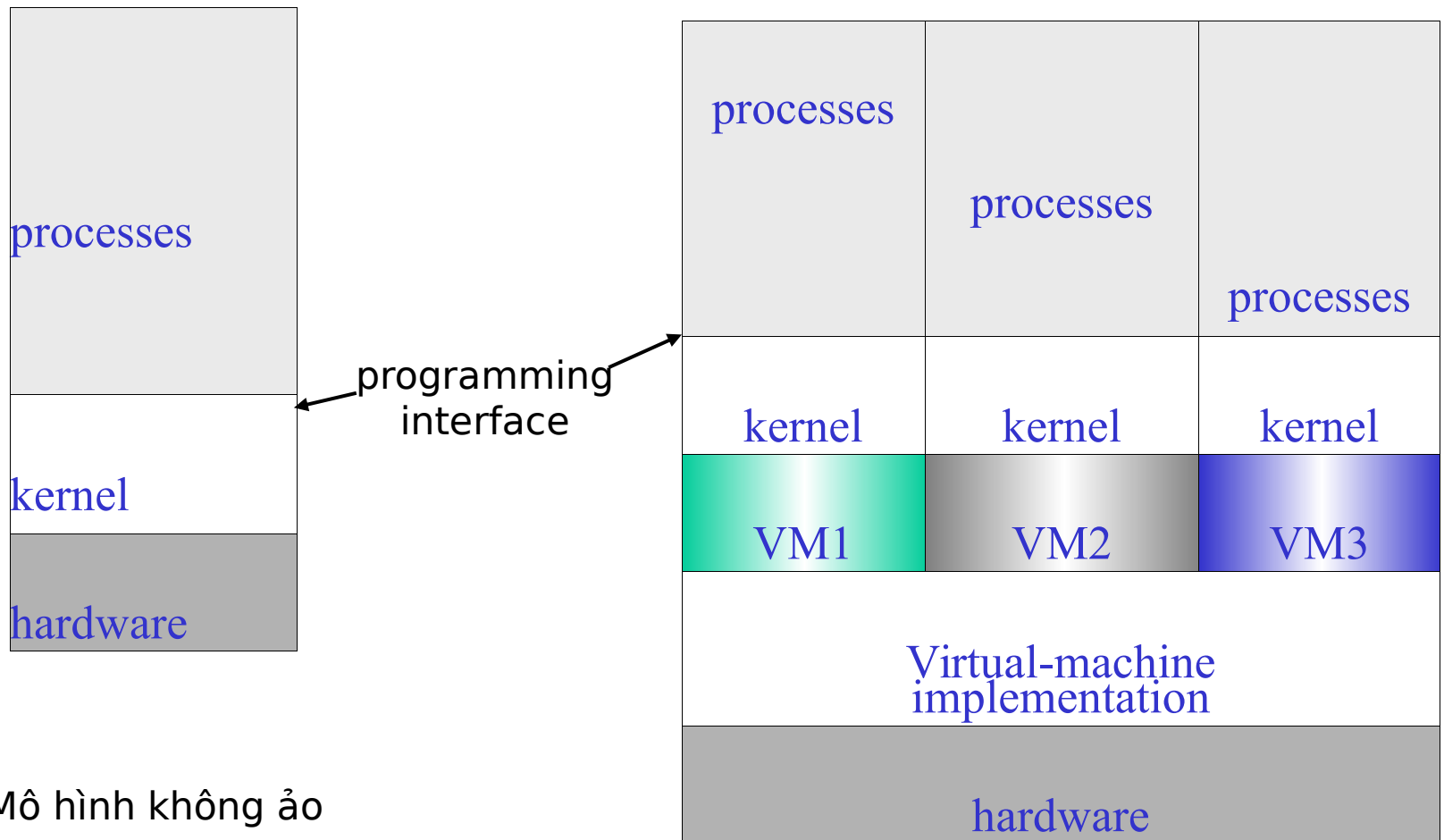
Cấu trúc của Hệ điều hành

Hệ thống phân lớp (3): Ví dụ

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Cấu trúc của hệ điều hành THE

Cấu trúc của Hệ điều hành Máy ảo (1)



Cấu trúc của Hệ điều hành

Máy ảo (2)

- Một máy ảo có cách tiếp cận phân lớp để dàn xếp một cách logic kết quả của nó. Nó xử lý phần cứng và hạt nhân của hệ điều hành như thể chúng là tất cả phần cứng
- Một máy ảo cung cấp một giao diện giống hệt với phần cứng
- Hệ điều hành tạo ảo giác về đa tiến trình, mỗi một tiến trình thực hiện trên bộ vi xử lý riêng và bộ nhớ ảo riêng của nó

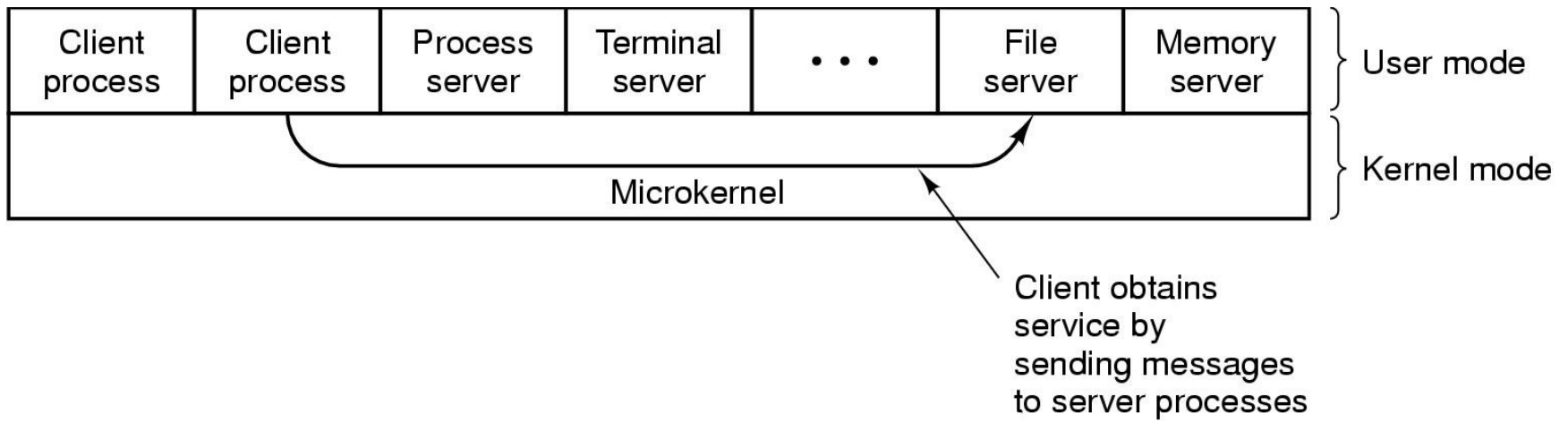
Cấu trúc của Hệ điều hành

Máy ảo (3)

- Các tài nguyên của máy tính vật lý được chia sẻ để tạo ra các máy ảo
 - Bộ lập lịch CPU có thể tạo ra giao diện mà ở đó các user có thể sở hữu bộ xử lý riêng
 - Spooling và hệ thống file cung cấp đầu đọc ảo và máy in ảo

Cấu trúc của Hệ điều hành

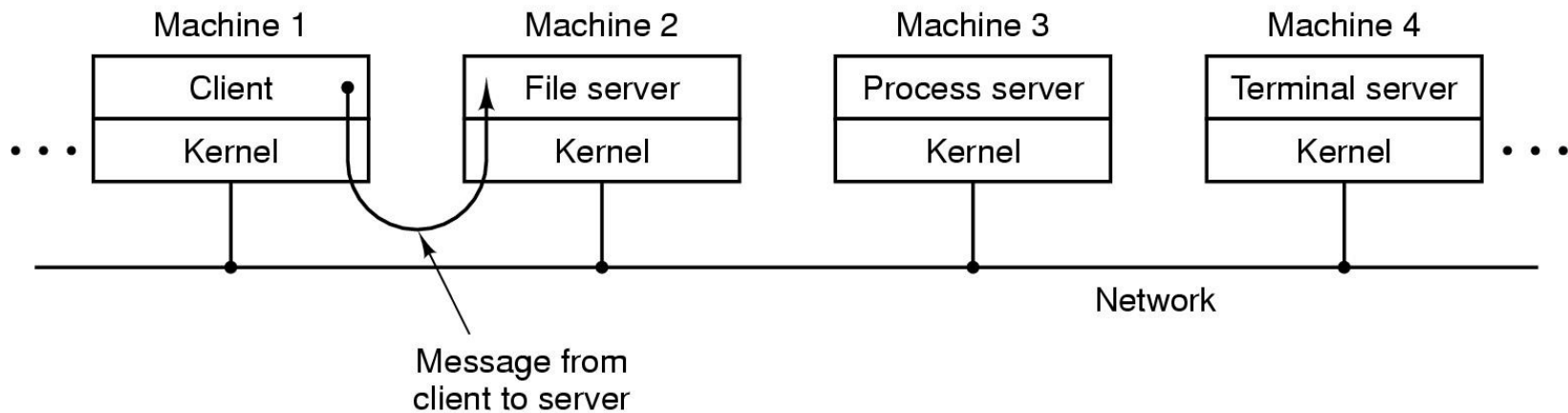
Mô hình Client-server (1)



Mô hình client-server

Cấu trúc của Hệ điều hành

Mô hình Client-server (2)



Mô hình client-server trong hệ phân tán

Cấu trúc của Hệ điều hành

Microkernel: Ví dụ

Microkernel

OS design: The operating system is divided into *microkernel* (CMU Mach OS - 1980)

- Moves as much from the kernel into “*user*” space
- kernel → microkernel
- Communication takes place between user modules using message passing.

