

Chương 1: Kiến trúc hệ thống & Tổng quan về Hệ điều hành

Mục tiêu: Hệ thống lại kiến trúc hệ thống phần cứng. Tìm hiểu tổng quan về chức năng và các thành phần của Hệ điều hành



Nội dung

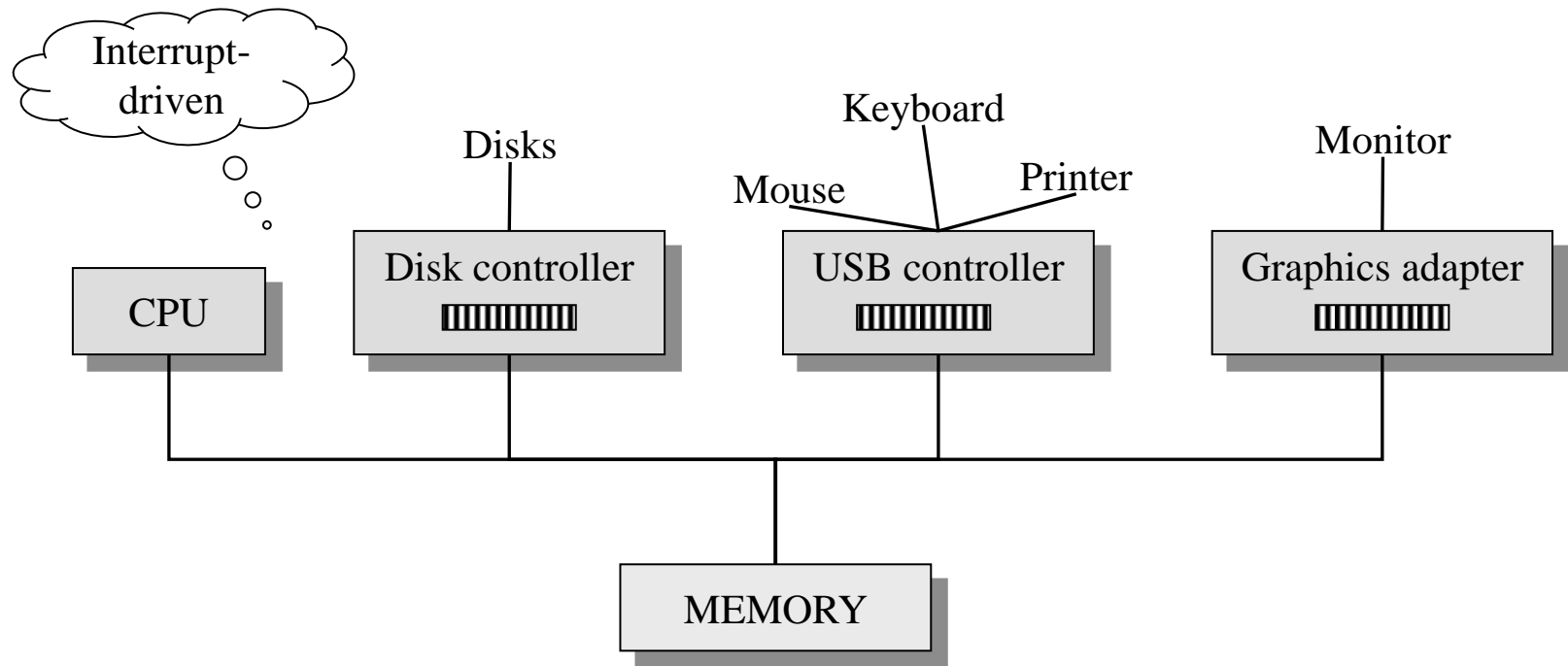
- Cấu trúc Hệ thống
- Tổng quan Hệ điều hành



1. Kiến trúc hệ thống

- Hoạt động của hệ thống máy tính
- Cơ chế vào-ra
- Phân cấp hệ thống nhớ
- Bảo vệ phần cứng (Hardware Protection)

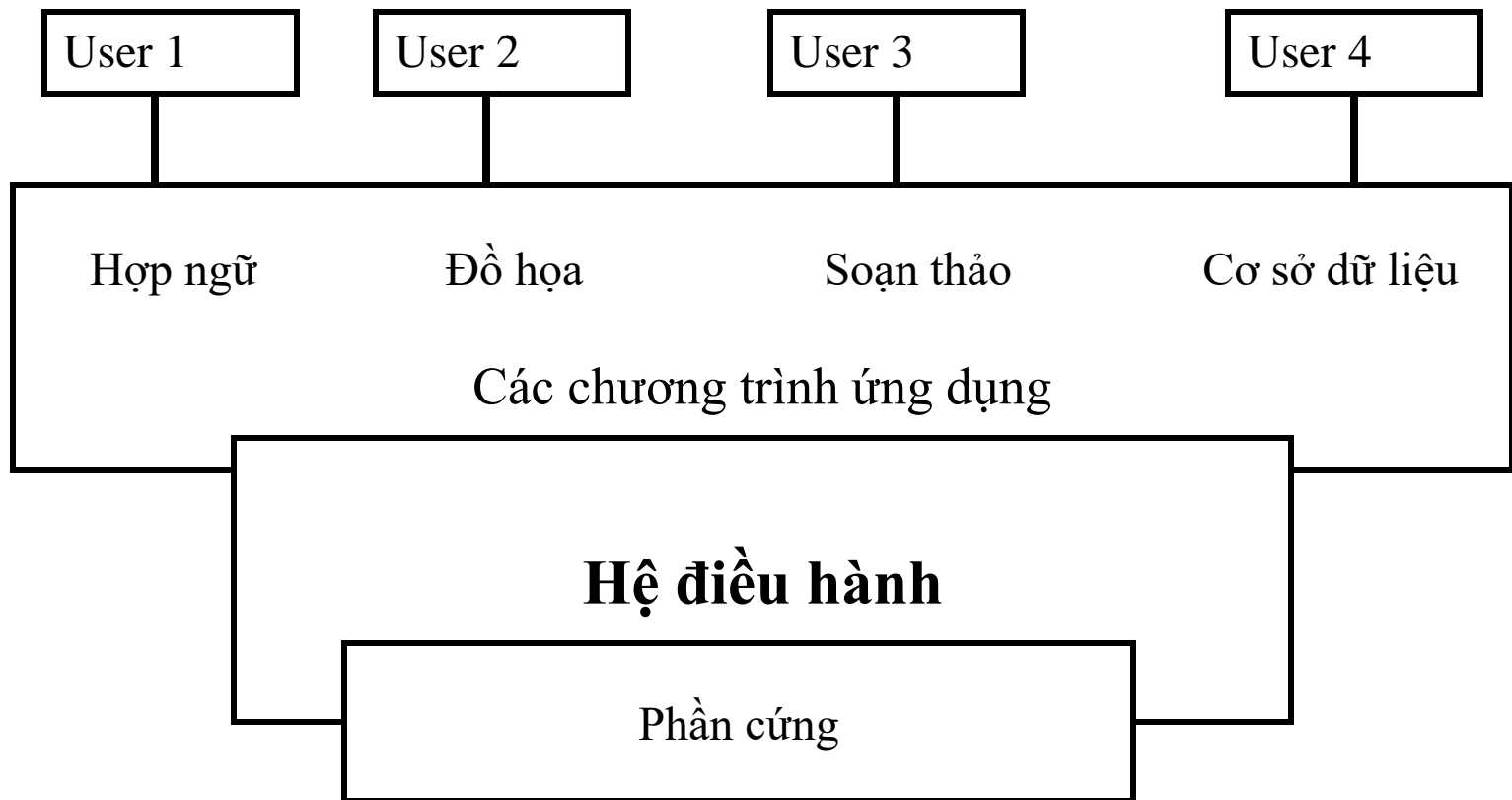
1.1. Hoạt động của hệ thống máy tính(1): Kiến trúc phần cứng



Kiến trúc Hệ thống máy tính hiện đại

▮▮▮▮ Đệm dữ liệu (local buffer)

1.1. Hoạt động của hệ thống máy tính(2): Mô hình trừu tượng



1.1. Hoạt động của hệ thống máy tính(3)

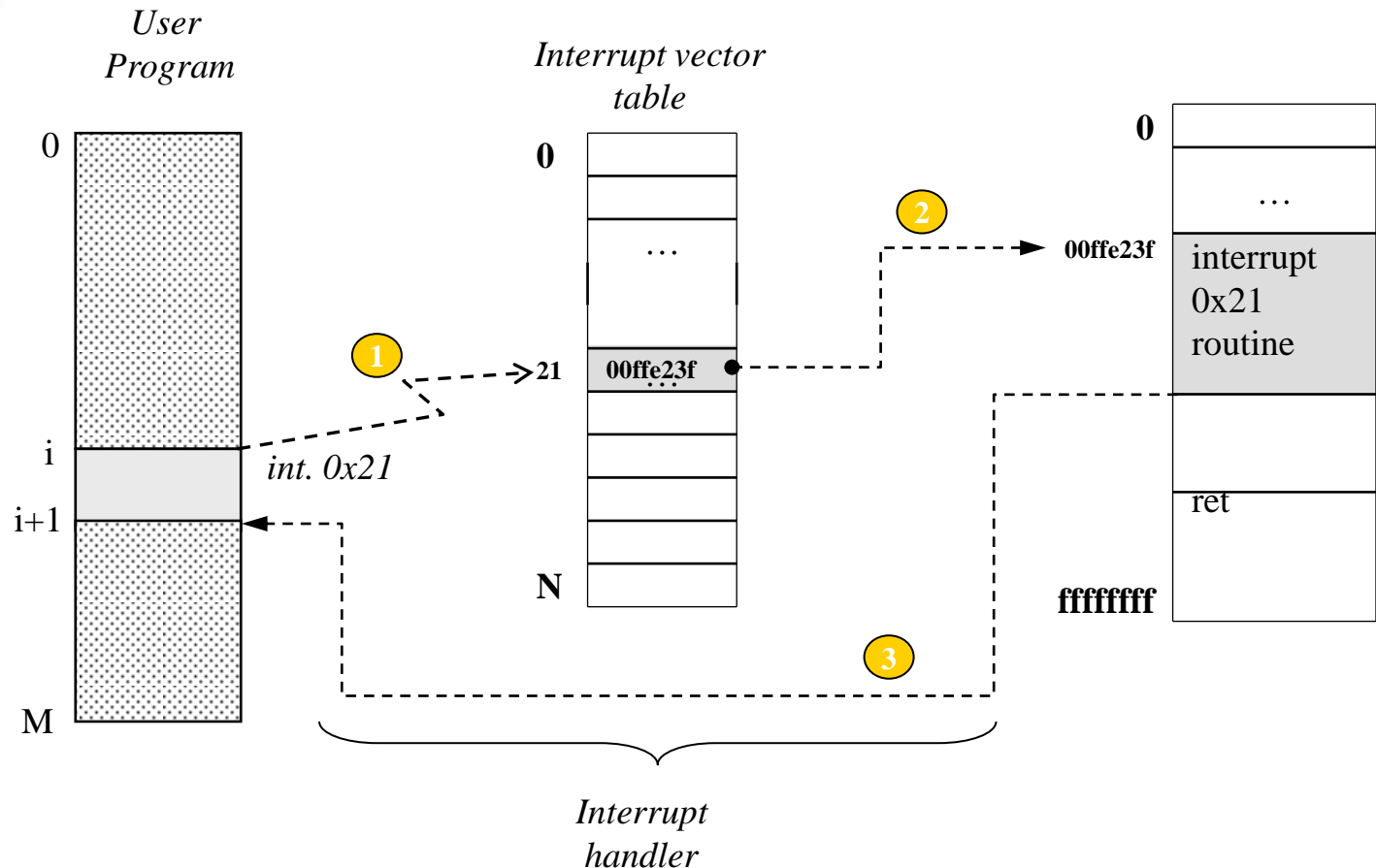
- Hệ thống hoạt động như một *interrupt* driven (quá trình điều khiển ngắt):
 - Mã lệnh của nó chỉ được gọi đến khi ngắt xuất hiện.
- Thông thường ngắt chuyển điều khiển cho chương trình phục vụ ngắt thông qua *vector ngắt* (*interrupt vector*).
 - Bảng vector ngắt chứa địa chỉ của tất cả các chương trình phục vụ ngắt (*interrupt service routine*).
- Kiến trúc ngắt (*Interrupt architecture*) phải lưu địa chỉ của lệnh bị ngắt.
- Các ngắt đến bị vô hiệu (*disabled*) trong khi một ngắt đang được thực hiện để tránh bị mất ngắt (*lost interrupt*).
- Một *bẫy lỗi* (*trap*) là ngắt mềm gây ra bởi một lỗi hoặc yêu cầu của người sử dụng.

1.1. Hoạt động của hệ thống máy tính(3)



- Hệ thống duy trì trạng thái của CPU bằng cách lưu giữ nội dung các thanh ghi, bộ đếm chương trình (program counter - PC) và địa chỉ của lệnh bị ngắt.
- Xác định loại ngắt nào đã xuất hiện và có những hành động tương ứng
- **Chương trình phục vụ ngắt** (interrupt service routine) chịu trách nhiệm thực hiện các ngắt, CPU được dành cho xử lý ngắt.
- Sau khi phục vụ ngắt, khôi phục lại ngữ cảnh trước ngắt

1.1. Hoạt động của hệ thống máy tính(5): Quá trình Xử lý ngắt





1.2. Các cơ chế I/O(1)

- Cơ chế Polling (xoay vòng):
 - CPU lần lượt hỏi (hoặc xoay vòng) các thiết bị I/O để xác định xem thiết bị có yêu cầu trao đổi tt
 - Chương trình điều khiển I/O được viết sẵn
 - Hoạt động: dữ liệu được trao đổi thông qua I/O Port
 - CPU ghi byte dữ liệu vào thanh ghi dữ liệu, thiết lập 1 bit của thanh ghi điều khiển để báo hiệu cho module điều khiển I/O
 - Module điều khiển I/O đọc byte dữ liệu, xóa bit thiết lập của thanh ghi điều khiển, CPU tiếp tục gửi byte kế tiếp
 - Module điều khiển I/O không gây ngắt khi xong việc -> CPU phải dùng cơ chế xoay vòng để kiểm tra trạng thái thiết bị

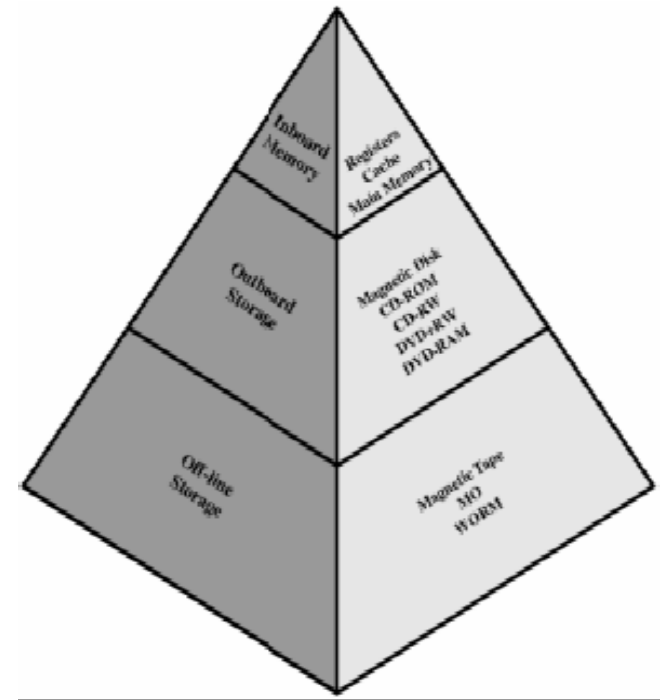


1.2. Các cơ chế I/O(2)

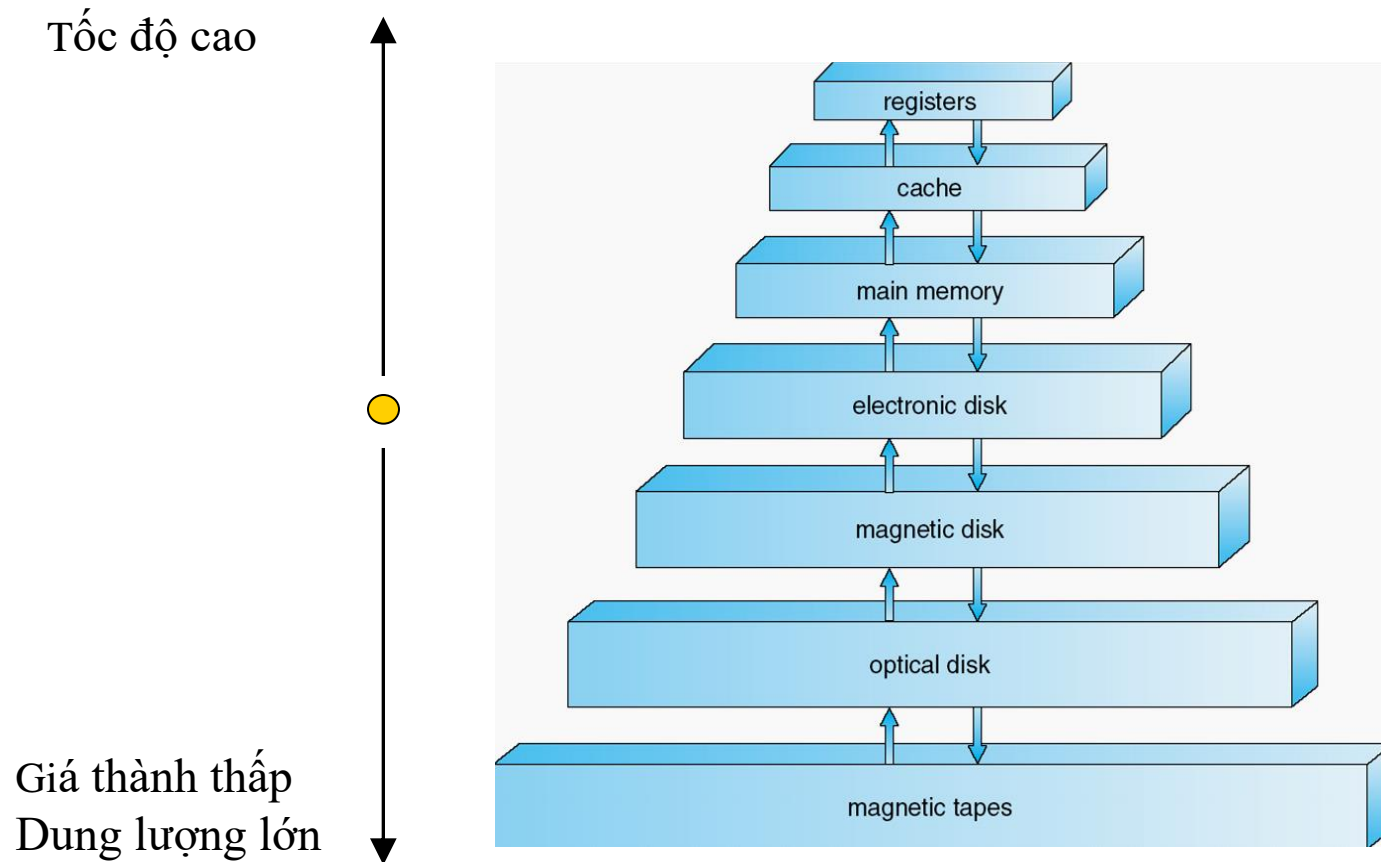
- Cơ chế I/O sử dụng ngắt
 - CPU không thực hiện Polling mà thiết bị I/O chủ động kích hoạt quá trình
 - Thiết bị I/O gây ngắt khi có yêu cầu thực hiện quá trình I/O
 - Trong khi thực hiện I/O, CPU có thể thực hiện các quá trình khác

1.3. Cấu trúc hệ thống nhớ(1)

- Hệ thống lưu trữ được tổ chức dạng sơ đồ phân cấp dựa vào:
 - Tốc độ
 - Giá thành
 - Tính không ổn định

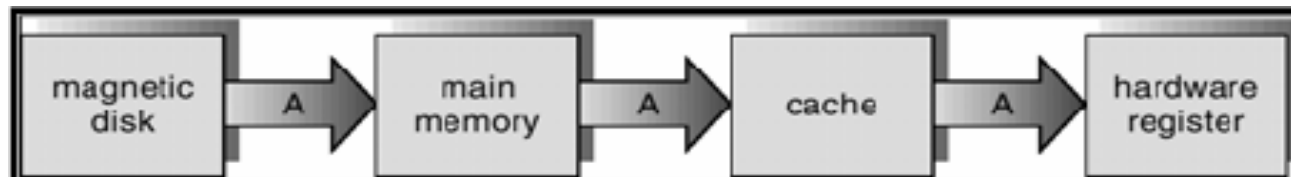


1.3. Cấu trúc hệ thống nhớ(2)



1.3. Cấu trúc hệ thống nhớ(3)

- *Caching* – kỹ thuật làm tăng tốc độ xử lý của hệ thống bằng cách(ví dụ google search):
 - Thực hiện copy thông tin tới thiết bị nhớ nhanh hơn để tăng tốc độ xử lý của hệ thống.
 - Giữ lại các dữ liệu mới được truy nhập trong thiết bị tốc độ cao đó.
 - Yêu cầu: dữ liệu phải được lưu trữ đồng bộ trong nhiều mức hệ thống nhớ để nhất quán (*consistent*).
 - Vì dung lượng cache có hạn, yêu cầu có sự quản lý cache (*cache management*) để tăng hiệu năng.



1.4. Các cơ chế bảo vệ phần cứng(1)

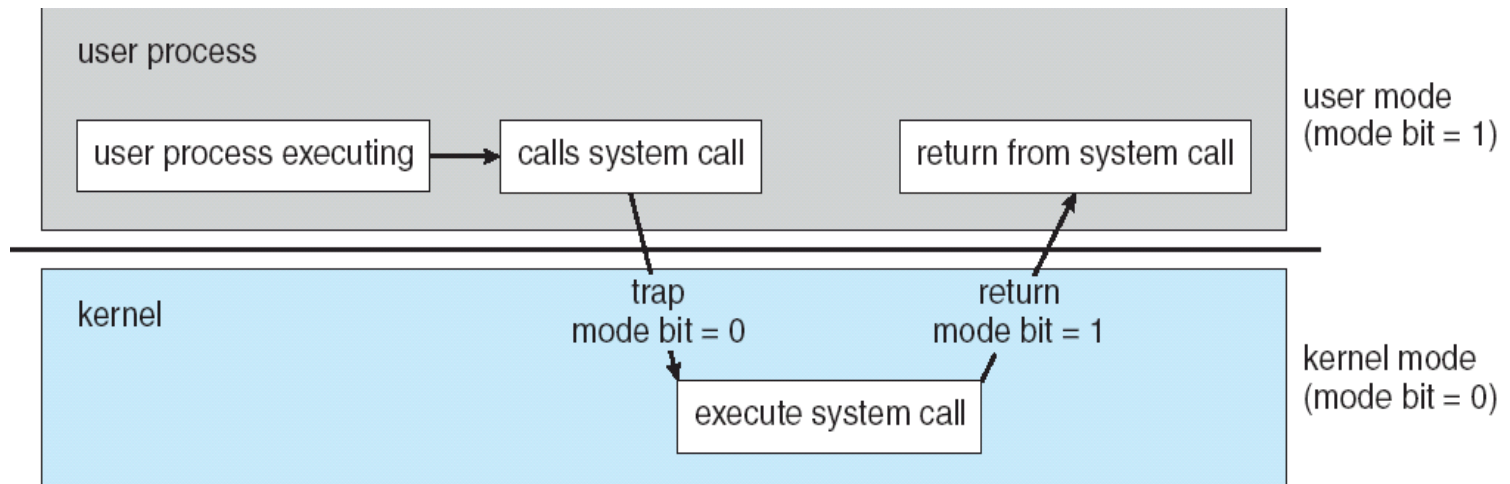
- Thực thi trong chế độ kép (Dual-Mode Operation)
- Bảo vệ vào-ra (I/O Protection)
- Bảo vệ bộ nhớ (Memory Protection)
- Bảo vệ CPU (CPU Protection)

1.4. Các cơ chế bảo vệ phần cứng(2): Dual-Mode

- Các tài nguyên hệ thống chia sẻ (Sharing system resources) yêu cầu HĐH bảo đảm rằng một chương trình bị lỗi không thể gây cho các chương trình khác thực hiện sai theo.
- HĐH phải cung cấp sự hỗ trợ phần cứng để phân biệt giữa ít nhất 2 chế độ thực hiện.
 - *User mode* – thực hiện với tư cách người sử dụng: chỉ được thực hiện các lệnh không đặc quyền
 - *Monitor mode* – thực hiện với tư cách HĐH: được thực hiện các lệnh đặc quyền.

1.4. Các cơ chế bảo vệ phần cứng(3): Dual-Mode

- Cần phần cứng hỗ trợ:
 - Phần cứng có thêm 1 bit gọi là *Mode bit* được đưa vào phần cứng máy tính để chỉ rõ chế độ hiện tại: monitor (0) hoặc user (1).
 - Monitor: có toàn quyền truy xuất tài nguyên hệ thống
 - User: có quyền hạn chế của user bình thường
- Khi một ngắt hoặc lỗi xuất hiện, phần cứng chuyển sang monitor mode.



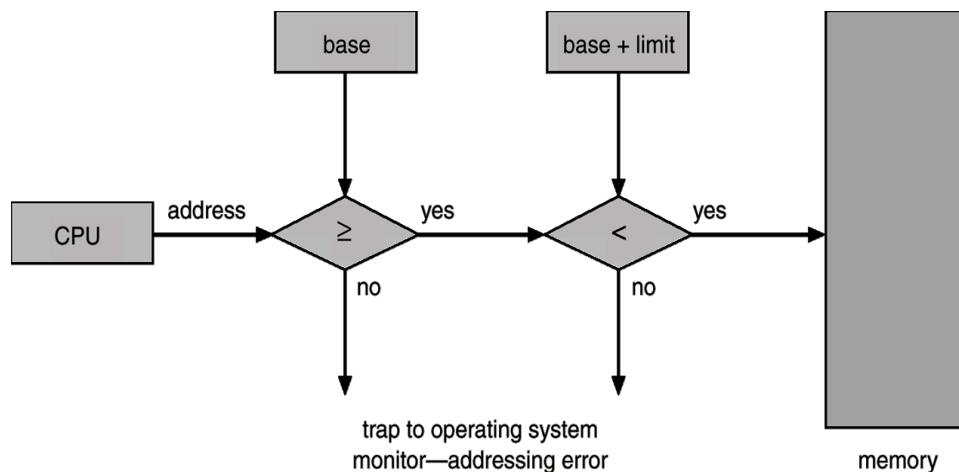
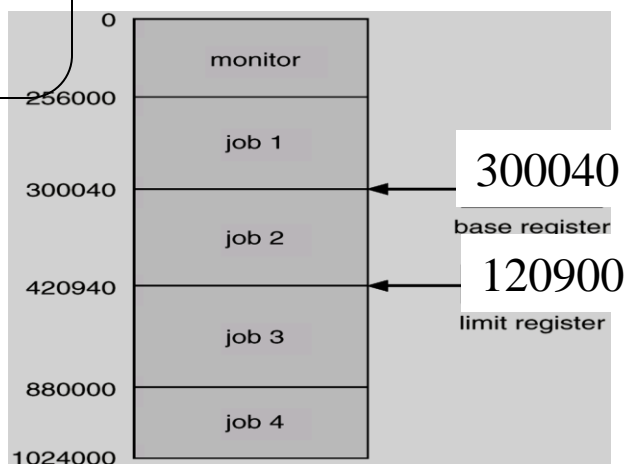
1.4. Các cơ chế bảo vệ phần cứng(4): I/O Protection

- Để tránh việc user thực hiện các vào-ra không hợp lệ phá vỡ hệ thống:
 - Tất cả các lệnh vào-ra là các lệnh đặc quyền
 - user không thể thực hiện trực tiếp vào-ra mà phải thông qua các **lời gọi hệ thống (system calls)**.
 - System call:
 - Là lời gọi(yêu cầu) **sử dụng các dịch vụ HĐH** cung cấp
 - Khi thực hiện -> Gây ngắt mềm, HĐH sẽ chuyển điều khiển cho chương trình phục vụ ngắt, chuyển mode = 0
 - HĐH kiểm tra các đối số, thực hiện yêu cầu và chuyển điều khiển đến lệnh tiếp theo, chuyển mode = 1

1.4. Các cơ chế bảo vệ phần cứng(5): Memmmory Protection

Mục đích:
tránh
vùng nhớ
dành cho
ngắt bị
thay đổi
bởi ctr
người
dùng

- Sử dụng 2 thanh ghi base & limit (tùy thuộc vào CPU)
- Khi 1 tiến trình thực thi trong chế độ monitor, HĐH không giới hạn sự truy cập vùng nhớ
- Để *giới hạn vùng nhớ truy cập* căn cứ vào giá trị của thanh ghi base & limit



1.4. Các cơ chế bảo vệ phần cứng(6): CPU Protection

- *Timer* – ngắt máy tính sau khoảng thời gian định sẵn để đảm bảo HĐH duy trì sự điều khiển, tránh chương trình của người sử dụng bị tắc trong vòng lặp vô hạn.
 - Timer giảm sau mỗi khoảng thời gian xác định (vd: 1/60 giây) hoặc biến đổi (vd: từ 1 ms đến 1 s).
 - Khi timer giảm đến giá trị 0, một ngắt xuất hiện.



2. Tổng quan Hệ điều hành

- Khái niệm
- Phân loại
- Các chức năng của hệ điều hành
- Các dịch vụ của HĐH
- Lời gọi HĐH (System Call)



2.1. Khái niệm(1)

- Khái niệm HĐH

- Hệ điều hành là một **bộ chương trình**(mã nguồn và dữ liệu-các tham số hệ thống) đã được cài đặt sẵn có chức năng:
 - **Điều phối, phân phối công việc** cho các thành phần chức năng trong máy tính
 - Thỏa mãn tối đa yêu cầu người dùng

- Giao tiếp với HĐH

- Cầu nối giao tiếp giữa user và hệ thống máy tính
- User giao tiếp với hệ điều hành -> phải có ngôn ngữ chung:
 - Giao tiếp dòng lệnh: thể hiện yêu cầu thông qua các lệnh
 - Giao tiếp biểu tượng



2.1. Khái niệm(2)

- Khi hoạt động, các chương trình ứng dụng & ctr hệ thống đều thực hiện theo các tiến trình
- **Tiến trình**: là một ctr đang xử lý(hoạt động, thực hiện), sở hữu một con trỏ lệnh, tập các thanh ghi và các biến
 - Để hoàn thành tác vụ, tiến trình cần tài nguyên: CPU, bộ nhớ, thiết bị I/O,...
- Phân biệt **tiến trình & chương trình**:
 - Chương trình là một **thực thể thụ động** chứa lệnh & dữ liệu để tiến hành một tác vụ(công việc). Khi thực hiện các lệnh, chương trình chuyển thành **tiến trình**.
 - **Tiến trình** là một **thực thể hoạt động**



2.2. Phân loại Hệ điều hành(1)

- Hệ điều hành đơn chương trình
 - Toàn bộ hệ thống máy tính phục vụ 1 chương trình từ lúc bắt đầu khi ctr được đưa vào bộ nhớ đến khi kết thúc chương trình
 - Khi 1 chương trình được đưa vào bộ nhớ và thực hiện => nó chiếm giữ mọi tài nguyên hệ thống nên không thể đưa ctr khác vào bộ nhớ
- Hệ điều hành đa chương trình
- Hệ điều hành thời gian thực
- Hệ song song
- Hệ phân tán



2.2. Phân loại Hệ điều hành(2)

- Hệ điều hành đa chương trình
 - Tại một thời điểm có nhiều ctr có mặt đồng thời trong bộ nhớ
 - Các ctr đều có nhu cầu được phân phối bộ nhớ và CPU
 - Gồm 2 loại:
 - Hệ điều hành hoạt động theo mẻ (batch - lo)
 - Hệ điều hành chia sẻ thời gian



2.2. Phân loại Hệ điều hành(3)

- Hệ điều hành đa chương trình(tiếp)
 - **Hệ điều hành hoạt động theo lô:** hướng tới mục tiêu cực đại số lượng bài toán được giải quyết trong một đơn vị thời gian
 - Có 2 loại: MFT và MVT
 - **MFT(Multiprogramming with Fixed number of Task)**
 - Quy định sẵn số tiến trình có mặt đồng thời trong bộ nhớ
 - Bộ nhớ được chia sẵn thành các phân vùng(Partition) mỗi phân vùng chỉ chứa một tiến trình
 - **MVT(Multiprogramming with Variable number of Task):** bộ nhớ không được phân chia sẵn, tiến trình được nạp liên tục khi còn đủ bộ nhớ, số lượng tiến trình đồng thời trong bộ nhớ luôn thay đổi tùy thuộc vào dung lượng



2.2. Phân loại Hệ điều hành(4)

- Hệ điều hành đa chương trình(tiếp)
 - Hệ điều hành phân chia thời gian:
 - Diễn hình của **hệ điều hành đa người dùng**
 - Tại một thời điểm có nhiều user làm việc với máy tính thông qua các trạm đầu cuối khác nhau
 - Hệ điều hành phân phối CPU lần lượt cho từng user
 - Mỗi ctr chiếm giữ CPU một khoảng thời gian như nhau gọi là **lượng tử thời gian(quantum)**
 - Bộ nhớ luôn chứa ctr của mọi user



2.2. Phân loại Hệ điều hành(5)

- Hệ điều hành thời gian thực
 - Đảm bảo giải quyết bài toán(tiến trình) không muộn hơn một thời điểm xác định
 - Mỗi tiến trình được gắn với một thời gian xác định phải hoàn thành gọi là **DeadTime**
 - Hoàn thiện bài toán muộn hơn không có ý nghĩa
 - Ví dụ: giải quyết bài toán trong các lò phản ứng hạt nhân; game thời gian thực



2.2. Phân loại Hệ điều hành(6)

- Hệ song song:
 - Cho các hệ thống có **nhiều CPU** chia sẻ đường truyền, bộ nhớ, đồng hồ, thiết bị I/O...
 - Các CPU làm việc phụ thuộc độc lập do phải chia sẻ tài nguyên
 - So với hệ thống 1 CPU:
 - Nhanh hơn
 - An toàn hơn: một CPU hỏng không ảnh hưởng tới toàn bộ hệ thống



2.2. Phân loại Hệ điều hành(7)

- Hệ phân tán:
 - Tập hợp các thiết bị tính riêng rẽ có thể giao tiếp với nhau
- Hệ phân tán vs. hệ song song
 - Hệ song song phối hợp nhiều bộ xử lý nhằm giải quyết một vấn đề cho trước một cách nhanh nhất
 - Trong hệ phân tán, mỗi bộ xử lý nói chung có chương trình làm việc riêng bán độc lập, vì lợi ích chung nên cần phối hợp hành động với nhau



2.3. Các chức năng hệ điều hành

- Điều phối hoạt động: Quản lý tiến trình (Process Management)
- Thành phần quản lý và phân phối tài nguyên:
 - Quản lý bộ nhớ chính (Main Memory Management)
 - Quản lý File (File Management)
 - Quản lý hệ thống vào-ra (I/O System Management)
 - Quản lý bộ nhớ thứ cấp (Secondary Storage Management)
- Hoạt động mạng (Networking)
- Hệ thống bảo vệ (Protection System)
- Hệ thống thông dịch lệnh (Command-Interpreter System)



2.3.1. Process Management

- Các chương trình đều thực hiện theo các tiến trình để hoàn thành công việc
- HĐH chịu trách nhiệm đối với việc quản lý tiến trình:
 - Tạo và xoá process của người sử dụng và của hệ thống.
 - Tạm ngừng và tiếp tục lại process.
 - Cung cấp cơ chế cho:
 - Sự đồng bộ hoá process
 - Sự giao tiếp process

2.3.2. Main Memory Management

- Bộ nhớ là một mảng lớn các từ nhớ hoặc byte có địa chỉ riêng.
 - Chứa dữ liệu có thể truy nhập nhanh chóng
 - Được chia sẻ bởi CPU và các thiết bị vào-ra.
- Main memory là một thiết bị lưu trữ tạm.
 - Mất nội dung bên trong khi hệ thống ngừng hoạt động.
- Hoạt động quản lý bộ nhớ của HĐH:
 - Lưu lại dấu vết của các phần bộ nhớ đang được sử dụng và được sử dụng bởi tiến trình nào.
 - Quyết định xem những tiến trình nào được nạp khi có bộ nhớ trống.
 - Phân phối và thu hồi bộ nhớ cho các tiến trình.



2.3.3. File Management

- Một file là một tập hợp các thông tin liên quan được tổ chức một cách logic, được lưu trữ trên thiết bị nhớ phụ.
- Thông thường, các file đại diện cho dữ liệu và các chương trình (cả source, data và object forms).
- HĐH chịu trách nhiệm đối với các hoạt động sau khi nối kết với file management:
 - Tạo và xoá file.
 - Tổ chức file (Tạo và xoá thư mục).
 - Hỗ trợ từ gốc (primitive) đối với việc thao tác với các file và thư mục (read/write).
 - Ánh xạ các file vào bộ nhớ thứ cấp.
 - Backup file trên các phương tiện lưu trữ ổn định.



2.3.4. I/O System Management

- Một trong những chức năng chính của hệ điều hành là quản lý tất cả những thiết bị I/O.
- HĐH phải **ra các chỉ thị điều khiển** thiết bị, kiểm soát các **ngắt và lỗi**.
- HĐH phải **cung cấp một cách giao tiếp đơn giản và tiện dụng giữa các thiết bị và phần còn lại của hệ thống** và giao tiếp này phải **độc lập** với thiết bị.
- Hệ thống vào-ra bao gồm:
 - Một thành phần quản lý bộ nhớ I/O
 - Một giao diện device-driver chung
 - Các driver cho các thiết bị phần cứng riêng biệt
- Chỉ device driver biết các tính chất đặc biệt của thiết bị mà nó điều khiển.

2.3.5. Secondary-Storage Management

- Dữ liệu trên bộ nhớ chính (*primary storage*):
 - Tạm thời, dung lượng nhỏ
 - Để lưu trữ lâu dài, dung lượng lớn -> sử dụng bộ nhớ thứ cấp (*secondary storage*) để back up bộ nhớ chính.
- Hầu hết các hệ thống máy tính hiện đại sử dụng các ổ đĩa như là các phương tiện lưu trữ trực tuyến cơ sở cho cả chương trình và dữ liệu.
- HĐH chịu trách nhiệm đối với các hoạt động sau khi nối kết với disk management:
 - Quản lý các vùng nhớ tự do
 - Phân phối bộ nhớ
 - Lập lịch ổ đĩa (Disk scheduling)

2.3.6. Nhân Hệ điều hành (Kernel)

- Hệ điều hành có dung lượng lớn => không thể đưa tất cả vào bộ nhớ
- Các ctr của hệ điều hành phân thành 2 loại:
 - Nhân HĐH: gồm các ctr luôn tồn tại trong bộ nhớ để điều khiển máy tính làm việc
 - Các ctr chỉ được nạp vào bộ nhớ khi cần thiết
- Theo chức năng Nhân HĐH thường bao gồm các nhóm ctr:
 - Module ctr tải(loader) có chức năng đưa một ctr vào bộ nhớ và cho phép ctr đã tải nhận điều khiển hay không
 - Module ctr điều phối chính: đảm nhận việc lựa chọn các bước làm việc của toàn bộ hệ thống
 - Module ctr lập lịch: chọn tiến trình làm việc tiếp theo
 - Các thông tin về hệ thống(các tham số hệ thống)



2.4. Các dịch vụ của hệ điều hành

- **Thực hiện chương trình** (Program execution) – khả năng của hệ thống để nạp một chương trình vào bộ nhớ và chạy nó.
- **Thực hiện vào-ra** (I/O operations) – vì chương trình của người sử dụng không thể thực hiện trực tiếp các hoạt động vào/ra, HĐH phải cung cấp một số phương pháp để thực hiện vào/ra.
- **Thao tác với hệ thống file** (File-system manipulation) – khả năng của chương trình để đọc, ghi, tạo, xoá các file.
- **Giao tiếp** (Communications) – trao đổi thông tin giữa các tiến trình đang thực hiện trên cùng 1 máy tính hoặc trên các hệ thống khác nhau được nối mạng. Việc thực hiện thông qua bộ nhớ chia sẻ (*shared memory*) hoặc chuyển thông điệp (*message passing*).
- **Phát hiện lỗi** (Error detection) – bảo đảm việc tính toán đúng đắn bằng cách phát hiện các lỗi trong CPU và bộ nhớ, trong các thiết bị vào-ra, hoặc trong chương trình của người sử dụng.

2.4.1. Các chức năng mở rộng của HĐH

- Các chức năng cộng thêm tồn tại không phải để giúp người sử dụng mà để đảm bảo cho các hoạt động của hệ thống hiệu quả hơn.
 - **Resource allocation** – phân phối tài nguyên cho nhiều người sử dụng hoặc cho nhiều công việc chạy cùng lúc.
 - **Accounting** – theo dõi và ghi lại loại tài nguyên và thời gian sử dụng (tài nguyên) của user nhằm mục đích thống kê.
 - **Protection** – đảm bảo sự kiểm soát đối với tất cả sự truy nhập tới các tài nguyên hệ thống.



2.5. System Calls(1)

- **System calls** cung cấp giao diện giữa một tiến trình và HĐH.
 - Mục đích là yêu cầu **sử dụng các dịch vụ HĐH**
 - Thường là các câu lệnh viết bằng ngôn ngữ assembly.
 - Các ngôn ngữ bậc cao (vd: C, C++) có thể thay thế ngôn ngữ assembly để cho phép việc lập trình hệ thống tạo trực tiếp các system call.
- Ba phương thức chung được sử dụng để *truyền tham số* giữa tiến trình và HĐH.
 - Truyền tham số trong các **thanh ghi**.
 - Chứa các tham số trong một **bảng tham số** trong bộ nhớ, và địa chỉ của bảng được truyền như một tham số trong một thanh ghi.
 - Đẩy (*push*, store) các tham số vào **stack** bằng chương trình, và lấy ra khỏi stack (*pop*) bởi HĐH.



2.3. System Calls(2): Phân loại

- Điều khiển tiến trình (Process control)
 - Kết thúc, bỏ dở (abort)
 - Nạp, thực hiện
 - Tạo, kết thúc tiến trình
 - Lấy/ thiết lập các thuộc tính của tiến trình chờ đợi
 - Đợi sự kiện, báo hiệu sự kiện
 - Phân phối và giải phóng bộ nhớ



2.3. System Calls(3): Phân loại

- Quản lý file (File management)
 - Tạo file, xóa file
 - Mở, đóng
 - Đọc, ghi, định vị
 - Lấy/ thiết lập thuộc tính file
- Quản lý thiết bị (Device management)
 - Yêu cầu thiết bị, giải phóng thiết bị
 - Đọc, ghi, định vị
 - Lấy/ thiết lập các thuộc tính thiết bị
 - Gắn kết (attach), tháo gỡ (detach) mức logic các thiết bị

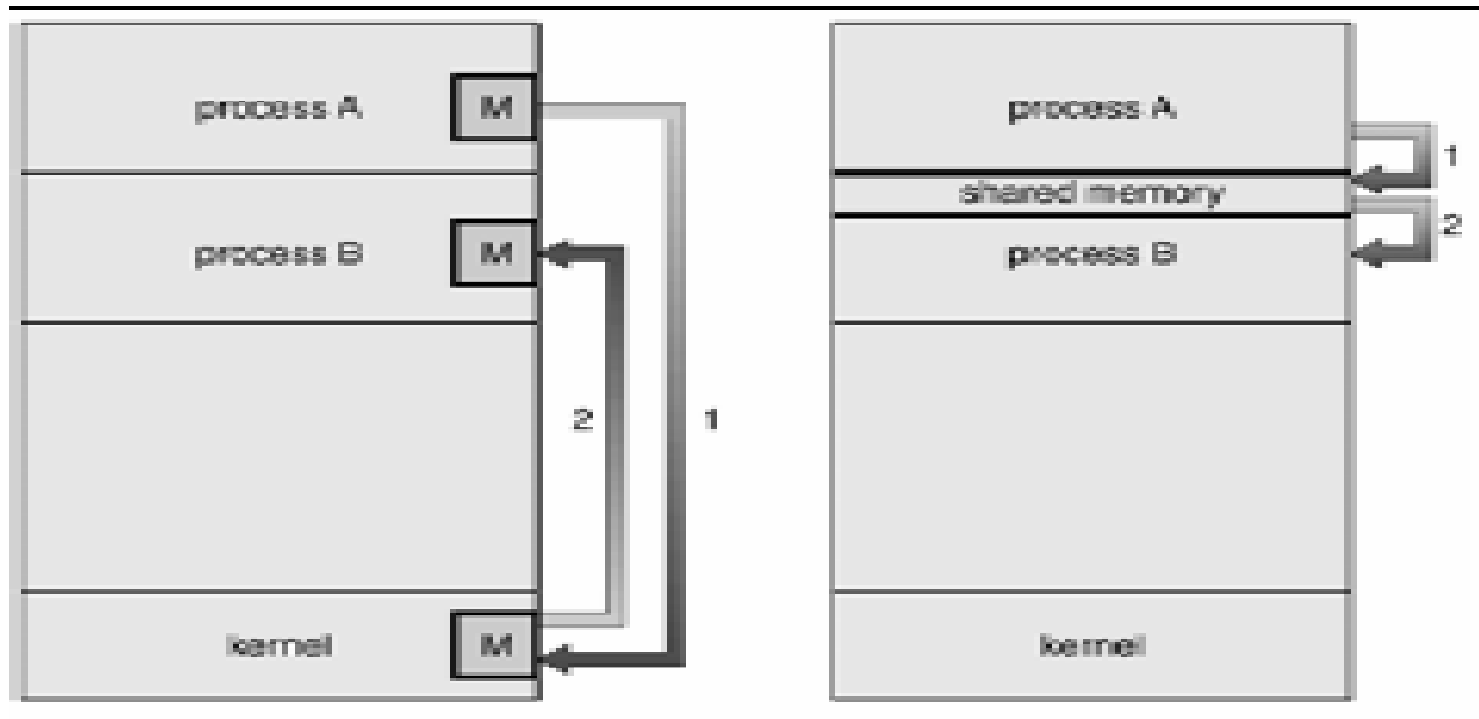


2.3. System Calls(4): Phân loại

- Duy trì thông tin (Information maintenance)
 - Lấy/ thiết lập giờ hoặc ngày
 - Lấy/ thiết lập dữ liệu hệ thống
 - Lấy/ thiết lập thuộc tính của tiến trình, file, thiết bị
- Giao tiếp (Communications)
 - Tạo, xóa kết nối giao tiếp
 - Gửi, nhận thông điệp
 - Truyền lại thông tin trạng thái
 - Gắn kết, tháo gỡ logic các thiết bị ở xa (remote device)

2.3. System Calls(5): Phương thức giao tiếp

- Sự giao tiếp có thể thực hiện bằng cách sử dụng phương thức: **message passing** hoặc **shared memory**.





Q & A

...