

Chương 1

Tổng quan về Hệ điều hành



Mục tiêu

- Cung cấp một cách nhìn tổng quan về các thành phần chính của hệ điều hành.
- Cung cấp sự tổng quát về tổ chức hệ thống máy tính cơ bản.

Nội dung:

- Hệ điều hành (Operating System) làm việc gì?
- Tổ chức của hệ thống máy tính
- Cấu trúc của HĐH
- Hoạt động của HĐH
- Sự quản lý tiến trình
- Sự quản lý bộ nhớ chính
- Sự quản lý bộ nhớ lưu trữ
- Protection và Security
- Các môi trường sử dụng máy tính

1.1. Hệ điều hành làm việc gì

- Hệ điều hành (Operating System): Là một chương trình hoạt động như một lớp trung gian giữa người sử dụng máy tính và phần cứng máy tính.
- Các mục đích của HĐH:
 - Thực hiện các chương trình của người sử dụng và giúp việc giải các bài toán của người sử dụng dễ dàng hơn.
 - Giúp cho việc sử dụng hệ thống máy tính thuận tiện hơn.
 - Sử dụng phần cứng máy tính theo một cách hiệu quả.

1.1. Hệ điều hành làm việc gì

❖ UNIX

- Một trong những hệ điều hành đa nhiệm, đa người dùng đầu tiên
- Được phát triển vào năm 1969 cho các máy tính siêu lớn và các máy chủ
- Phiên bản mới hơn cho các hệ thống máy tính để bàn bao gồm giao diện đồ họa (GUI)
- Sử dụng rộng rãi trong các trường đại học và các tổ chức nghiên cứu khoa học, các doanh nghiệp lớn
- Thường được sử dụng trên các máy cơ khí hoặc các ứng dụng hỗ trợ thiết kế trên máy tính (CAD)

1.1. Hệ điều hành làm việc gì

❖ Linux

- Phát triển đầu tiên vào năm 1991
- Miễn phí và có khả năng tùy chỉnh
- Đóng gói thành các định dạng được gọi là các bản phân phối
- Bao gồm hệ điều hành, các tiện ích và các thư viện khác nhau, và thậm chí là một số phần mềm ứng dụng
- Được sử dụng rộng rãi trên các siêu máy tính và máy chủ, máy tính cá nhân
- Rất phổ biến với các cơ quan, doanh nghiệp

1.1. Hệ điều hành làm việc gì

❖ Các hệ điều hành cho thiết bị cầm tay (Handheld Operating Systems)

- Sử dụng trên các PDA và Smartphone
- Tùy chọn cho mỗi hệ thống khác nhau tùy thuộc vào loại thiết bị cầm tay
- Các hệ điều hành dành cho thiết bị cầm tay phổ biến: Symbian, Windows Mobile, Palm, iOS, Android và Blackberry



1.1. Hệ điều hành làm việc gì

❖ Các hệ điều hành nhúng (Embedded Operating Systems)

- Quản lý và điều khiển mọi hoạt động của các loại thiết bị cụ thể
- Khi thiết bị được khởi động, hệ điều hành nhúng được tải lên bộ nhớ
- Được thiết kế để gắn chặt và được chuyên biệt cao
- Chúng chỉ bao gồm các chức năng cụ thể cho các thiết bị cụ thể mà nó được phát triển

Example Embedded Operating Systems

What is an embedded operating system?

- Found on most mobile computers, PDAs, and other small devices
- Windows CE is scaled-down version of Windows
- Windows Mobile for Pocket PC
- Palm OS for Palm
- Embedded Linux
- Blackberry OS
- Symbian OS



1.1. Hệ điều hành làm việc gì

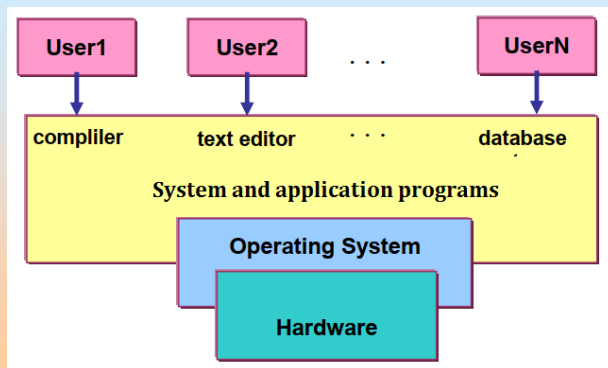
Khả năng và giới hạn của HĐH:

- ❖ Tên tập tin có thể dài đến 255 ký tự
 - Trên PC, không có thể bao gồm \/: * <>? | trong tên tập tin
 - Trên Mac, không thể sử dụng dấu hai chấm trong tên tập tin
- ❖ Có thể mở nhiều chương trình cùng lúc
- ❖ Có thể tùy chỉnh cho phù hợp với sở thích cá nhân
- ❖ Máy tính độc lập (Stand-alone computer) - có thể cài đặt chương trình hoặc tải về các tập tin từ Internet
- ❖ Máy tính mạng (Networked computer) - có thể không có đủ quyền để thực hiện một số tác vụ nhất định

Cấu trúc của hệ thống máy tính

1. Phần cứng (Hardware) – cung cấp các tài nguyên tính toán cơ bản (CPU, memory, I/O devices)
2. Hệ điều hành (Operating system) – điều khiển và sắp xếp việc sử dụng phần cứng trong các chương trình ứng dụng khác nhau đối với những người sử dụng khác nhau.
3. Các chương trình ứng dụng (Applications programs) – định cách sử dụng tài nguyên hệ thống để giải quyết các bài toán của người sử dụng (word processors, compilers, database systems, video games)
4. Đối tượng sử dụng (Users): people, machines, other computers

Bốn thành phần hệ thống máy tính



Các cách nhìn đối với hệ điều hành

- Là trình phân phối tài nguyên (Resource allocator) – quản lý và quyết định phân phối các tài nguyên (CPU, không gian bộ nhớ, các thiết bị vào/ra...) cho các yêu cầu nhằm đạt hiệu quả và công bằng.
- Là một chương trình điều khiển (Control program) – điều khiển sự thực hiện các chương trình của người sử dụng và sự hoạt động của các thiết bị vào/ra để ngăn các lỗi và sự sử dụng sai.
 - Không có định nghĩa hoàn toàn đúng về HĐH.
- **Kernel** (nhân) – là một chương trình chạy liên tục không ngừng trên máy tính (tất cả các chương trình khác là chương trình hệ thống hoặc chương trình ứng dụng).

Những gì dễ nhầm với hệ điều hành?

- Các lệnh/ứng dụng truy vấn trạng thái hệ thống: *ls* (UNIX), Task Manager (Windows). Đó là các công cụ và không chạy liên tục.
- Các trình điều khiển thiết bị ngoại vi (drivers): khiến cho HĐH có thể sử dụng phần cứng mới, chúng là OS extensions chứ không phải là bản thân OS. Cũng giống như browser plugin và browser.
- Các phần mềm có thể đơn phương truy nhập phần cứng:
 - VMWare là một virtual PC (không phải hardware). Có thể cài Windows “trên” nó.
 - Java VM là một *virtual machine* (không phải hardware), API (application program interface) của nó là một HĐH suy rộng. (Careful: JavaOS là một project riêng)

Tại sao phải nghiên cứu HĐH?

- Thiết kế HĐH là một nghiên cứu riêng (case study) rất tốt về thiết kế kỹ nghệ phần mềm.
- Càng hiểu rõ về hệ điều hành → viết ứng dụng tốt hơn. Càng hiểu rõ hơn về lỗi của nó → biết cách khắc phục chúng tốt hơn.
- HĐH sử dụng các thuật giải phức tạp, rất nhiều trong đó có thể được sử dụng lại trong các phần mềm khác, vd: phát hiện bế tắc (deadlock detection).
- HĐH cần có sự phát triển nhanh của phần cứng.

Sự khởi động máy tính

- **bootstrap program** (chương trình mồi) được nạp khi bật máy hoặc khi khởi động lại.
 - Thường được chứa trong ROM hoặc EPROM, thường được gọi là **firmware**.
 - Khởi tạo tất cả các khía cạnh của hệ thống.
 - Nạp nhân (kernel) của HĐH và bắt đầu sự thực hiện.

Sự khởi động máy tính

Chuyện gì xảy ra khi bật máy tính ?

- Quá trình bật máy tính và tải hệ điều hành được gọi là khởi động (*booting*) máy tính
- Chip ROM-BIOS làm thủ tục khởi động có thể:
 - Tải các lệnh lưu trữ trong ROM-BIOS vào bộ nhớ và thực thi các lệnh
 - Máy tính tiến hành kiểm tra các thiết bị bên trong và bên ngoài của nó. Quá trình này được gọi là quá trình tự kiểm tra POST (Power-On Self-Test)
 - Chương trình BIOS kiểm tra và đếm bộ nhớ, sau đó máy tính tìm kiếm và tải hệ điều hành vào bộ nhớ

Sự khởi động máy tính

Chuyện gì xảy ra khi bật máy tính ?

- Khi hệ điều hành được tải xong, Windows sẽ hiển thị màn hình Chào mừng
- Nếu máy tính được kết nối vào một mạng nội bộ hoặc được thiết lập cho nhiều người dùng, Windows sẽ hiển thị màn hình đăng nhập

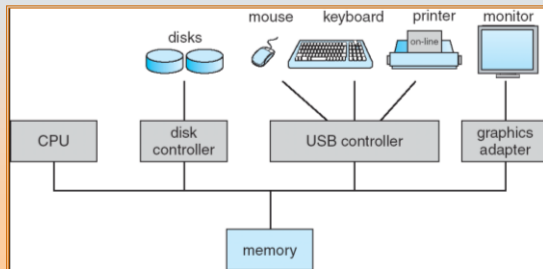


Phân loại OS theo nguyên lý điều khiển

- ❖ Hệ thống máy tính lớn (Mainframe Systems)
- ❖ Hệ thống để bàn (Desktop Systems)
- ❖ Hệ thống đa bộ xử lý (Multiprocessor Systems)
- ❖ Hệ thống phân tán (Distributed Systems)
- ❖ Hệ thống song song (Parallel Systems)
- ❖ Hệ thống bó (Clustered Systems)
- ❖ Hệ thống thời gian thực (Real – Time Systems)
- ❖ Hệ thống cầm tay (Handheld Systems)
- ❖ Môi trường tính toán (Computing Environments)

1.2. Tổ chức của hệ thống máy tính

- Một hoặc nhiều CPU, device controller (bộ điều khiển thiết bị) kết nối với nhau bằng bus chung cho phép truy nhập bộ nhớ chia sẻ.
- Sự thực hiện đồng thời của CPU và các thiết bị cạnh tranh các chu kỳ bộ nhớ.



Hoạt động của hệ thống máy tính

- Các thiết bị vào-ra và CPU có thể thực hiện đồng thời.
- Mỗi device controller phụ trách một loại thiết bị riêng và có một bộ nhớ đệm riêng (local buffer).
- CPU chuyển dữ liệu từ/đến bộ nhớ chính đến/từ các buffer.
- Vào-ra từ thiết bị đến local buffer của device controller.
- Device controller thông báo cho CPU biết nó đã hoàn tất công việc của nó bằng cách gây ra một *ngắt* (interrupt).

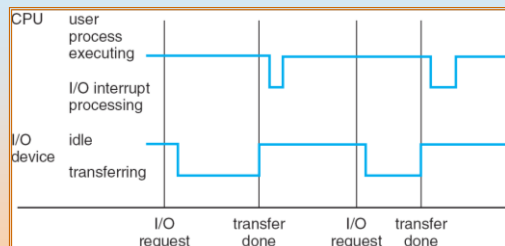
Các chức năng chính của ngắt

- HĐH được điều khiển bằng ngắt (*interrupt driven*), nghĩa là mã lệnh của nó chỉ được gọi đến khi ngắt xuất hiện.
- Thông thường ngắt chuyển điều khiển cho thường trình dịch vụ ngắt thông qua *vector ngắt* (*interrupt vector*), có chứa địa chỉ của tất cả các thường trình dịch 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* (*trap*) là phần mềm tạo ngắt gây ra bởi một lỗi hoặc yêu cầu của người sử dụng.

Xử lý ngắt

- HĐH 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.
- HĐH xác định loại ngắt nào đã xuất hiện và có những hành động thực hiện tương ứng:
 - *polling*
 - *vectored*
- Thường trình dịch vụ ngắt (*interrupt service routine*) chịu trách nhiệm thực hiện các ngắt, CPU được dành cho xử lý ngắt.
- Sau khi phục vụ ngắt, HĐH khôi phục lại ngữ cảnh trước ngắt

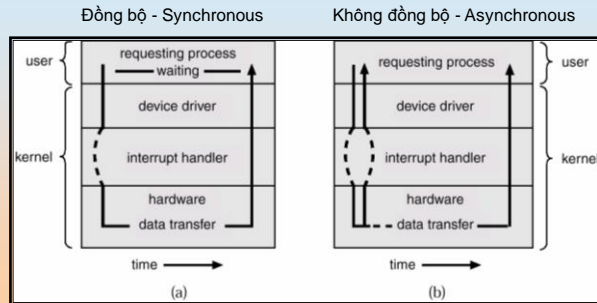
Interrupt Timeline



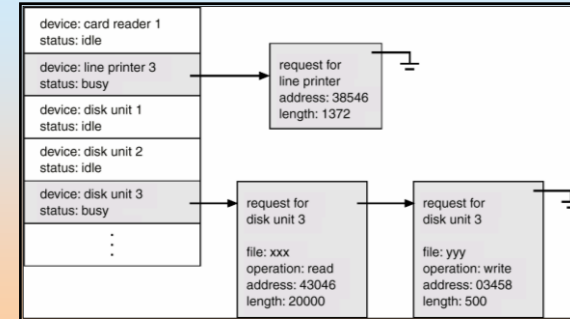
Cấu trúc vào-ra

- *Synchronous I/O*: Sau khi bắt đầu vào-ra, quyền kiểm soát chỉ quay lại chương trình của người sử dụng khi vào-ra đó kết thúc.
 - Chờ lệnh làm rồi CPU cho đến khi có lệnh ngắt kế tiếp.
 - Chờ theo vòng lặp kiểm tra CPU rồi → tranh chấp truy nhập bộ nhớ.
 - Tại một thời điểm, chỉ có 1 yêu cầu vào-ra được thực hiện, không có sự xử lý I/O đồng thời.
- *Asynchronous I/O*: Sau khi I/O bắt đầu, quyền kiểm soát quay lại chương trình của người sử dụng mà không cần chờ I/O kết thúc.
 - *System call* – gửi yêu cầu tới HĐH cho phép người sử dụng đợi I/O kết thúc (nếu muốn).
 - *Bảng trạng thái thiết bị* (*Device-status table*) chứa thông tin (entry) của mỗi thiết bị I/O cho biết trạng thái, địa chỉ và loại của thiết bị.
 - HĐH tra cứu vào bảng I/O device để xác định trạng thái thiết bị và sửa đổi thông tin để thêm thông tin ngắt.

Mô tả 2 phương pháp vào-ra



Bảng trạng thái thiết bị



Cấu trúc DMA (Direct Memory Access)

- Được sử dụng cho các thiết bị tốc độ cao (disk, communications network) để có thể tăng tốc độ trao đổi thông tin gần tới tốc độ bộ nhớ.
- Device driver chuyển toàn bộ các khối dữ liệu (block of data) từ bộ nhớ buffer trực tiếp tới bộ nhớ chính không qua sự can thiệp của CPU.
- Chỉ có 01 ngắt được sinh ra đối với mỗi block, tối ưu hơn là mỗi ngắt đối với 01 byte (hoặc word) đối với các thiết bị tốc độ chậm.

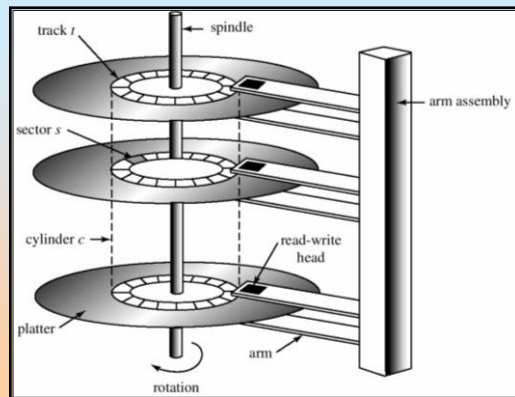
Cấu trúc hệ thống nhớ

- Bộ nhớ chính (Main memory: RAM) – phương tiện lưu trữ lớn duy nhất mà CPU có thể truy nhập trực tiếp.
 - Quá nhỏ để có thể lưu trữ ổn định tất cả các chương trình và dữ liệu cần thiết.
 - Có tính không ổn định \Rightarrow mất dữ liệu khi tắt nguồn.
- Bộ nhớ thứ cấp (Secondary storage) – là sự mở rộng của bộ nhớ chính, để cung cấp dung lượng bộ nhớ lớn và ổn định.

Vd: Đĩa từ (Magnetic disks: đĩa cứng, đĩa mềm)

 - Cấu tạo bằng kim loại cứng hoặc các miếng kính được bao bọc bởi vật liệu nhiễm từ.
 - Bề mặt đĩa được chia (vật lý) thành các *tracks*, mỗi track được chia thành các *sectors*.
 - Mạch điều khiển đĩa (*disk controller*) xác định sự tương tác vật lý giữa thiết bị và máy tính.

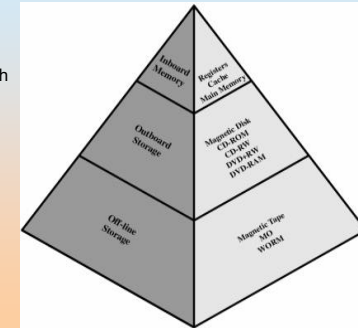
Cơ cấu đĩa có đầu từ chuyển động



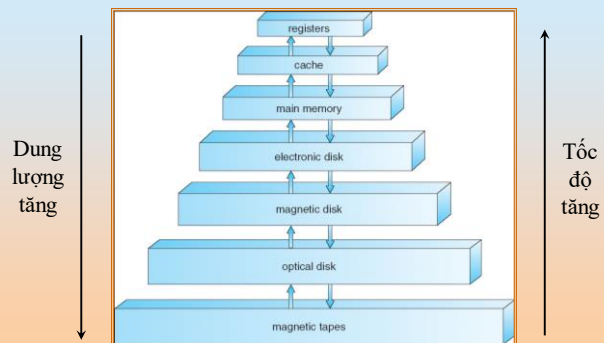
Sơ đồ phân cấp hệ thống nhớ

- Hệ thống lưu trữ được tổ chức dạng sơ đồ phân cấp dựa vào:

- Tốc độ
- Giá thành
- Tính không ổn định



Sơ đồ phân cấp thiết bị nhớ



Caching

- *Caching* – nguyên lý quan trọng, được thực hiện tại nhiều mức trong 1 máy tính (trong phần cứng, HĐH, phần mềm)
- Là kỹ thuật làm tăng tốc độ xử lý của hệ thống bằng cách:
 - thực hiện copy thông tin đang sử dụng 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.
- Bộ nhớ nhanh hơn (cache) được kiểm tra trước tiên xem thông tin có ở đó không:
 - Nếu có, thông tin được sử dụng trực tiếp từ cache (nhanh)
 - Nếu không, dữ liệu được copy vào cache rồi được sử dụng ở đó
- Yêu cầu: dữ liệu phải được lưu trữ đồng bộ trong nhiều mức hệ thống nhớ để đảm bảo tính nhất quán (*consistent*).
- Vì dung lượng cache có hạn, yêu cầu có sự quản lý cache (*cache management*) để tăng hiệu năng.

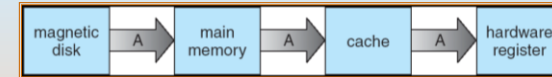
Sự thực thi tại nhiều mức bộ nhớ

- Sự di chuyển giữa các mức phân cấp bộ nhớ có thể là rõ ràng hoặc không.

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000,000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Sự di trú của số nguyên A từ đĩa tới thanh ghi

- Các môi trường đa nhiệm phải **cẩn thận** để sử dụng được giá trị mới nhất, dù nó được chứa ở đâu trong phân cấp bộ nhớ



- Môi trường đa bộ vi xử lý phải cung cấp tính gắn kết cache trong phần cứng để tất cả các CPU có được giá trị mới nhất trong cache của nó.
- Môi trường phân tán còn phức tạp hơn
 - Có thể tồn tại nhiều bản copy của dữ liệu

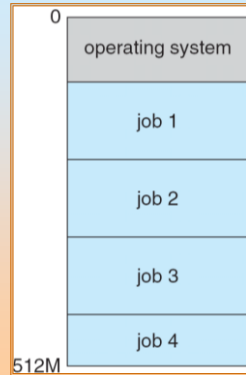
1.3. Cấu trúc hệ điều hành

- **Multiprogramming** (kỹ thuật đa chương trình) cần có để đạt hiệu quả
 - User đơn không thể giữ CPU và các thiết bị vào-ra hoạt động tại mọi thời điểm.
 - Multiprogramming tổ chức các công việc (job, gồm code và data) để CPU luôn có việc để thực hiện
 - Một tập con của tất cả các công việc trong hệ thống được giữ trong bộ nhớ
 - Một công việc được chọn và chạy thông qua **job scheduling**
 - Khi nó phải đợi (vd đợi vào-ra), HĐH chuyển sang một công việc khác.

Cấu trúc hệ điều hành (tiếp)

- **Timesharing (multitasking)** (kỹ thuật chia sẻ thời gian, đa nhiệm) là sự mở rộng logic mà trong đó CPU chuyển giữa các công việc rất thường xuyên để những user có thể tương tác với mỗi công việc trong khi nó đang chạy, tạo thành sự tính toán **tương tác**
 - **Thời gian đáp ứng** (Response time) nên < 1 giây
 - Mỗi user có ít nhất 1 chương trình đang thực hiện trong bộ nhớ ⇒ **process** (tiến trình)
 - Nếu một số công việc sẵn sàng chạy tại cùng thời điểm ⇒ **CPU scheduling** (lập lịch CPU)
 - Nếu các tiến trình không chứa vừa trong bộ nhớ, **swapping** (hoán đổi) chuyển chúng vào và ra để chạy
 - **Virtual memory** (bộ nhớ ảo) cho phép sự thực hiện các tiến trình không hoàn toàn trong bộ nhớ.

Bố trí bộ nhớ của HĐH đa chương trình

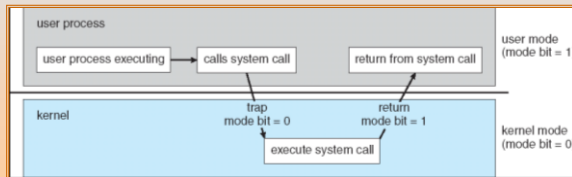


1.4. Hoạt động của hệ điều hành

- Ngắt do phần cứng
- Lỗi phần mềm hoặc yêu cầu tạo ra **exception** hay **trap**
 - Chia cho 0, yêu cầu dịch vụ của HĐH
- Các vấn đề tiến trình khác gồm: lặp vô hạn, các tiến trình thay đổi lẫn nhau hoặc thay đổi HĐH.
- Hoạt động **chế độ kép** (Dual-mode) cho phép HĐH bảo vệ chính nó và các thành phần hệ thống khác
 - **User mode** và **kernel mode**
 - **Mode bit** được cung cấp bởi phần cứng
 - ▶ Cung cấp khả năng phân biệt khi nào hệ thống chạy user code hay kernel code
 - ▶ Một số lệnh được thiết kế là **đặc quyền** (privileged), chỉ có thể thực hiện được trong kernel mode
 - ▶ **System call** thay đổi chế độ thành kernel mode, **return from call** thiết lập nó thành user mode.

Chuyển từ User sang Kernel Mode

- Định thời để ngăn lặp vô hạn / tiến trình lấy quá các tài nguyên
 - Thiết lập ngắt sau khoảng thời gian xác định
 - HĐH giảm bộ đếm
 - Khi bộ đếm bằng 0 thì sinh ra một ngắt
 - Thiết lập tiến trình lập lịch trước đó để giành lại được sự điều khiển hoặc chấm dứt chương trình vượt quá thời gian được cấp.



1.5. Sự quản lý tiến trình

- Tiến trình (process) là một chương trình đang được thực hiện. Nó là một đơn vị công việc trong hệ thống. Chương trình là một **thực thể bị động**, tiến trình là một **thực thể chủ động**.
- Tiến trình cần các tài nguyên để hoàn tất công việc:
 - CPU, bộ nhớ, các thiết bị vào-ra, các tệp (files)
 - Dữ liệu khởi tạo
- Sự chấm dứt tiến trình đòi hỏi sự giành lại bất kỳ tài nguyên nào có thể tái sử dụng.
- Tiến trình đơn luồng (thread) có một **program counter** xác định vị trí của lệnh kế tiếp để thực hiện
 - Tiến trình thực hiện các lệnh tuần tự, mỗi thời điểm một lệnh cho đến khi kết thúc.
- Tiến trình đa luồng: mỗi luồng có một program counter.
- Hệ thống thông thường có nhiều tiến trình, một số user, một số HĐH chạy đồng thời trên một hay nhiều CPU.
 - Đồng thời bằng sự đa nhiệm các CPU giữa các tiến trình / các luồng.

Các hoạt động quản lý tiến trình

HDH chịu trách nhiệm đối với các hoạt động sau trong quản lý tiến trình

- Tạo và xoá các tiến trình của cả user và hệ thống.
- Tạm ngừng và tiếp tục lại các tiến trình.
- Cung cấp các cơ chế cho:
 - ▶ sự đồng bộ hoá tiến trình
 - ▶ sự giao tiếp tiến trình
 - ▶ sự xử lý bế tắc (deadlock)

1.6. Sự quản lý bộ nhớ (chính)

- Tất cả dữ liệu ở trong bộ nhớ trước và sau xử lý
- Tất cả các lệnh ở trong bộ nhớ để thực hiện
- Sự quản lý bộ nhớ xác định cái gì được ở trong bộ nhớ khi
 - Tối ưu hóa sự sử dụng CPU và sự đáp ứng máy tính với các user
- Các hoạt động quản lý bộ nhớ
 - 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 (hoặc những phần của chúng) và dữ liệu nào được đưa vào và đưa ra khỏi bộ nhớ.
 - Phân phối và thu hồi bộ nhớ khi cần đến.

1.7. Sự quản lý bộ nhớ lưu trữ

- HDH cung cấp cái nhìn logic, giống nhau đối với lưu trữ thông tin
 - Trừu tượng hóa các thuộc tính vật lý thành đơn vị lưu trữ logic - **file**
 - Mỗi phương tiện được điều khiển bởi thiết bị (nghĩa là: disk drive, tape drive)
 - ▶ Các thuộc tính khác nhau gồm: tốc độ truy nhập, dung lượng, tốc độ truyền dữ liệu, phương pháp truy nhập (tuần tự hoặc ngẫu nhiên)
- A) Sự quản lý Hệ thống file
 - Các file thường được tổ chức trong các thư mục
 - Kiểm soát truy nhập trên hầu hết các hệ thống để xác định ai có thể truy nhập cái gì
 - Các hoạt động của HDH gồm:
 - ▶ Tạo và xóa các file và thư mục
 - ▶ Hỗ trợ từ gốc (primitive) đối với việc thao tác với các file và thư mục (read/write).
 - ▶ Ảnh xạ các file vào bộ nhớ thứ cấp.
 - ▶ Sao dự phòng (Backup) file trên các phương tiện lưu trữ ổn định.

Sự quản lý bộ nhớ lưu trữ (tiếp)

- B) Sự quản lý bộ nhớ lưu trữ lớn
 - Thường sử dụng các đĩa để chứa dữ liệu không chứa vừa trong bộ nhớ chính hoặc dữ liệu cần được giữ lâu dài.
 - Quản lý đúng cách đóng vai trò quan trọng trung tâm.
 - Toàn bộ tốc độ thực hiện của máy tính xoay quanh hệ thống con đĩa và các giải thuật của nó.
 - Các hoạt động của HDH
 - Quản lý các vùng nhớ tự do
 - Phân phối bộ nhớ
 - Lập lịch đĩa (Disk scheduling)
 - Một số bộ nhớ lưu trữ (storage) không cần phải nhanh
 - Bộ nhớ cấp ba gồm: bộ nhớ quang, băng từ
 - Vẫn cần được quản lý
 - Khác nhau giữa WORM (write-once, read-many-times) và RW (read-write)

C) Hệ thống con vào-ra (I/O Subsystem)

- Một trong những mục đích của HĐH là ẩn các tính chất khác thường của các thiết bị phần cứng không cho user thấy. Chức năng đó do hệ thống vào-ra đảm nhận.
- Hệ thống con vào-ra chịu trách nhiệm đối với:
 - Quản lý bộ nhớ của vào-ra gồm:
 - ▶ buffering (chứa dữ liệu tạm thời trong khi nó đang được truyền)
 - ▶ caching (chứa các phần của dữ liệu trong bộ nhớ nhanh hơn để tăng hiệu năng)
 - ▶ spooling (gửi chông output của 1 công việc với input của các công việc khác)
 - 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.

1.8. Protection và Security

- **Protection** – mọi cơ chế để kiểm soát sự truy nhập của các tiến trình hoặc user tới các tài nguyên được xác định bởi HĐH.
- **Security** – sự bảo vệ của hệ thống chống lại những sự tấn công từ bên trong và bên ngoài
 - Rất nhiều dạng, bao gồm denial-of-service, worms, viruses, identity theft, theft of service
- Các hệ thống đầu tiên thường phân biệt giữa các user để xác định ai có thể làm cái gì
 - User identifiers (**user IDs**, security IDs) gồm tên và số kèm theo, mỗi user một số.
 - User ID sau đó được liên kết với tất cả các file, các tiến trình của user đó để xác định kiểm soát truy nhập.
 - Group identifier (**group ID**) cho phép tập các user để được xác định và các kiểm soát được quản lý, sau đó cũng được liên kết với mỗi tiến trình, file.
 - **Privilege escalation** (sự leo thang đặc quyền) cho phép user thay đổi thành ID có nhiều quyền hơn.

1.9. Các môi trường sử dụng máy tính

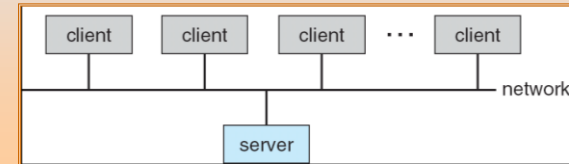
A) Máy tính truyền thống

- Môi trường văn phòng
 - ▶ Các PC được nối vào một mạng, các máy đầu cuối (terminal) được gắn vào mainframe hoặc minicomputers cung cấp xử lý theo lô và chia sẻ thời gian.
 - ▶ Ngày nay các cổng cho phép nối mạng và các hệ thống từ xa truy nhập tới cùng các tài nguyên.
- Các mạng gia đình
 - ▶ Đã thường là các hệ thống đơn, sử dụng các modem
 - ▶ Ngày nay được nối mạng, bảo vệ bằng firewall.

Các môi trường sử dụng máy tính (tiếp)

■ B) Client-Server Computing

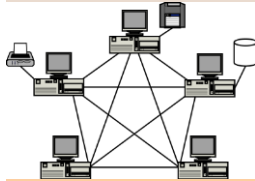
- Các máy đầu cuối cần được thay thế bởi các PC thông minh
- Giờ đây nhiều hệ thống là **server**, đáp ứng các yêu cầu từ các **client**.
 - ▶ **Compute-server** cung cấp một giao diện cho client yêu cầu các dịch vụ (database, DHCP, DNS,....)
 - ▶ **File-server** cung cấp các giao diện cho các client truy cập và lấy ra các file.



Các môi trường sử dụng máy tính (tiếp)

C) Peer-to-Peer Computing

- Một mô hình khác của hệ thống phân tán (distributed system)
- P2P không phân biệt các client và server
 - Tất cả các nút được coi là ngang nhau
 - Mỗi nút có thể hoạt động như client, server hoặc cả hai
 - Nút phải gia nhập mạng P2P



- » Đăng ký dịch vụ của nó với dịch vụ tìm kiếm trung tâm trên mạng
- » Truyền quảng bá (broadcast) yêu cầu dịch vụ và đáp ứng các yêu cầu thông qua *giao thức khám phá (discovery protocol)*

Các môi trường sử dụng máy tính (tiếp)

D) Web-Based Computing

- Web đã trở nên phổ biến khắp nơi
- Các PC đã là thiết bị phổ dụng
- Càng nhiều thiết bị có thể được nối mạng để cho phép truy nhập web.
- Loại thiết bị mới để quản lý truyền web trong các server tương tự nhau: **load balancers** (thiết bị cân bằng tải)
- Sự sử dụng các HĐH như Windows/Linux, mô hình Client-Server, đã tiến triển thành các hệ thống mạng lớn.



BÀI TẬP

1. Nêu sơ đồ các thành phần của một hệ thống máy tính ?
2. Hệ điều hành là gì ?
3. Các chức năng chính của hệ điều hành ?
4. Các đặc điểm của hệ thống song song ?
5. Các đặc điểm của hệ thống phân tán ?
6. Các đặc điểm của hệ thống xử lý thời gian thực ?
7. Trình bày về cơ chế DMA ?
8. Trình bày về cơ chế caching ?