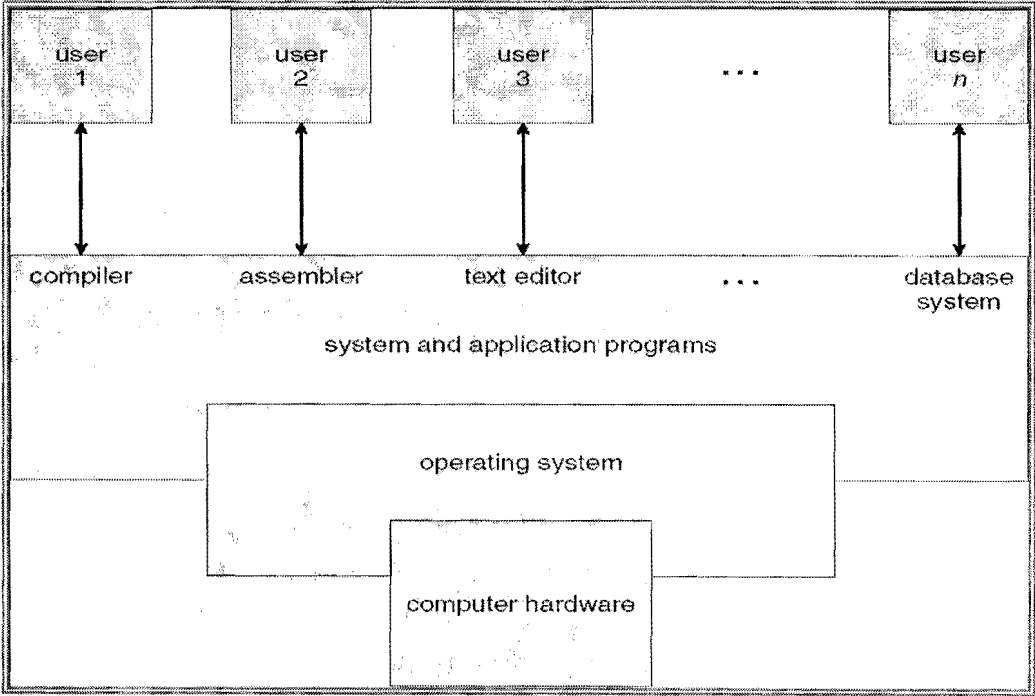


CHƯƠNG 1. TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Khái niệm Hệ điều hành

Một hệ điều hành là một thành phần quan trọng của mọi hệ thống máy tính. Một hệ thống máy tính có thể được chia thành bốn thành phần: phần cứng, hệ điều hành, các chương trình ứng dụng và người sử dụng. Quan hệ giữa các thành phần theo sơ đồ phân lớp.

- **Phần cứng** (Hardware) bao gồm bộ xử lý trung tâm (CPU), bộ nhớ, thiết bị xuất/nhập,...cung cấp tài nguyên cơ bản cho hệ thống.
- **Các chương trình ứng dụng** (application programs) bao gồm trình biên dịch (compiler), trình soạn thảo văn bản (text editor), hệ cơ sở dữ liệu (database system), trình duyệt Web,... mà trong đó các tài nguyên được sử dụng để giải quyết yêu cầu của người dùng.
- **Người sử dụng** (user): Có nhiều loại người sử dụng máy tính khác nhau, thực hiện những yêu cầu khác nhau, do đó sẽ có nhiều ứng dụng khác nhau.
- **Hệ điều hành** (operating system) hay còn gọi là chương trình hệ thống, điều khiển và phối hợp việc sử dụng phần cứng giữa những chương trình ứng dụng khác nhau cho những người dùng khác nhau. Hệ điều hành có thể hiểu theo phía người sử dụng hoặc từ phía người làm hệ thống.



Hình 1.1. Sơ đồ phân lớp của hệ thống máy tính

**Định nghĩa hệ điều hành (Operating System):**

Hệ điều hành là một chương trình quản lý phần cứng máy tính. Nó cung cấp nền tảng cho các chương trình ứng dụng và đóng vai trò trung gian giao tiếp giữa người dùng máy tính và phần cứng của máy tính đó

Mục tiêu của OS là làm cho người sử dụng:

- + Thực thi dễ dàng các ứng dụng của mình
- + Thao tác điều khiển máy tính trở nên thuận tiện.
- + Khai thác phần cứng máy tính một cách có hiệu quả

Thảo luận kĩ càng hơn về khái niệm này ta thấy từ quan điểm của người làm hệ thống, hệ điều hành là **chương trình gần gũi với phần cứng**. Chúng ta có thể thấy một hệ điều hành như bộ cấp phát tài nguyên. Hệ thống máy tính có nhiều tài nguyên - phần cứng và phần mềm - mà có thể được yêu cầu để giải quyết một vấn đề: thời gian CPU, không gian bộ nhớ, không gian lưu trữ tập tin, các thiết bị xuất/nhập,... Hệ điều hành hoạt động như bộ quản lý tài nguyên. Đương đầu với một lượng lớn các yêu cầu có thể xung đột về tài nguyên, hệ điều hành phải quyết định cách cấp phát tài nguyên tới những chương trình cụ thể và người dùng để có thể điều hành hệ thống máy tính hữu hiệu và công bằng.

Một cách nhìn khác của hệ điều hành nhấn mạnh sự cần thiết để điều khiển các thiết bị xuất/nhập khác nhau và chương trình ứng dụng. Một hệ điều hành là một chương trình điều khiển. Chương trình điều khiển quản lý sự thực thi của các chương trình người dùng để ngăn chặn lỗi và việc sử dụng không hợp lý máy tính. Nó đặc biệt quan tâm với những thao tác và điều khiển các thiết bị nhập/xuất.

Mục tiêu cơ bản của hệ thống máy tính là thực thi chương trình người dùng và giải quyết vấn đề người dùng dễ dàng hơn. Hướng đến mục tiêu này, phần cứng máy tính được phát triển. Tuy nhiên, chỉ đơn thuần là phần cứng thì không dễ sử dụng và phát triển các chương trình ứng dụng. Các chương trình khác nhau này đòi hỏi những thao tác chung nào đó, chẳng hạn như điều khiển thiết bị xuất/nhập, hệ điều hành sinh ra để giải quyết những vấn đề đó.

Cũng khó xác định phần gì thuộc về hệ điều hành, phần gì không. Một quan điểm đơn giản là một chương trình chạy liên tục trên máy tính (thường gọi là nhân kernel) thuộc về hệ điều hành, những chương trình còn lại thuộc về chương trình ứng dụng.

**Xác định mục tiêu** của hệ điều hành dễ hơn xác định hệ điều hành là gì. Mục đích chính của hệ điều hành là dễ dàng sử dụng. Vì sự tồn tại của hệ điều hành hỗ trợ nhiều cho máy tính trong việc đáp ứng các ứng dụng của người dùng. Quan niệm này đặc biệt rõ ràng hơn khi nhìn hệ điều hành trên các máy tính cá nhân.

Mục tiêu thứ hai của hệ điều hành là điều hành hữu hiệu hệ thống máy tính. Mục tiêu này đặc biệt quan trọng cho các hệ thống lớn, được chia sẻ, nhiều người dùng. Những hệ thống đặc biệt này khá đắt, khai thác hiệu quả nhất các hệ thống này luôn là điều mong muốn. Tuy nhiên, hai mục tiêu tiện dụng và hữu hiệu đôi khi mâu thuẫn nhau. Trong quá khứ, xem xét tính hữu hiệu thường quan trọng hơn tính tiện dụng. Do đó, lý thuyết hệ điều hành tập trung nhiều vào việc tối ưu hoá sử dụng tài nguyên tính toán. Hệ điều hành cũng phát triển dần theo thời gian. Thí dụ, UNIX bắt

đầu với bàn phím và máy in như giao diện của nó giới hạn tính tiện dụng đối với người dùng. Qua thời gian, phần cứng thay đổi và UNIX được gắn vào phần cứng mới với giao diện thân thiện với người dùng hơn. Nhiều giao diện người dùng đồ hoạ GUIs (graphical user interfaces) được bổ sung cho phép tiện dụng hơn với người dùng trong khi vẫn quan tâm tính hiệu quả.

Thiết kế hệ điều hành là một công việc phức tạp. Người thiết kế gặp phải nhiều sự thoả hiệp trong thiết kế và cài đặt. Việc nhiều người tham gia mang đến hệ điều hành những cải tiến mới và điều đó cho phép liên tục nâng cấp những phiên bản tốt hơn.

Hệ điều hành và kiến trúc máy tính có mối quan hệ khăng khít nhau. Để dễ dàng sử dụng phần cứng, hệ điều hành được phát triển. Sau đó, các người dùng hệ điều hành đề nghị những chuyển đổi trong thiết kế phần cứng để đơn giản chúng. Nhìn lại lịch sử ngắn ngủi này, chú trọng cách giải quyết những vấn đề về hệ điều hành để giới thiệu những đặc điểm phần cứng.

Chúng ta có thể thảo luận thêm về khái niệm hệ điều hành về phía người sử dụng.

- Hầu hết những người dùng máy tính ngồi trước máy tính cá nhân gồm có màn hình, bàn phím, chuột và bộ xử lý hệ thống (system unit). Một hệ thống như thế được thiết kế cho một người dùng độc quyền sử dụng tài nguyên của nó để tối ưu hoá công việc mà người dùng đang thực hiện. Trong trường hợp này, hệ điều hành được thiết kế để dành cho việc sử dụng với sự quan tâm về năng lực nhưng không quan tâm tới việc sử dụng tài nguyên. Năng lực thực hiện là quan trọng với người dùng nhưng không là vấn đề quan trọng nếu hầu hết hệ thống đang rảnh, chờ tốc độ xuất/nhập chậm từ phía người dùng.
- Trường hợp khác người dùng ngồi tại thiết bị đầu cuối (terminal) được nối kết tới máy tính lớn (mainframe) hay máy tính tầm trung (minicomputer). Những người khác đang truy xuất cùng máy tính thông qua các thiết bị đầu cuối khác. Những người dùng này chia sẻ các tài nguyên và có thể trao đổi thông tin. Hệ điều hành được thiết kế để tối ưu hoá việc sử dụng tài nguyên-để đảm bảo rằng tất cả thời gian sẵn sàng thực hiện của CPU, bộ nhớ và thiết bị xuất nhập được sử dụng hữu hiệu và cá nhân người dùng sử dụng độc quyền tài nguyên hơn là chia sẻ công bằng.
- Trường hợp tiếp theo người dùng khác ngồi tại trạm làm việc, được nối kết tới mạng của các trạm làm việc khác và máy chủ. Những người dùng này có tài nguyên tận hiến là trạm làm việc của mình nhưng họ cũng chia sẻ các tài nguyên trên mạng và các máy chủ- tập tin, tính toán và các máy phục vụ in. Do đó, hệ điều hành của họ được thiết kế để thoả hiệp giữa khả năng sử dụng cá nhân và việc tận dụng tài nguyên.
- Đối với trường hợp của máy tính cầm tay, các thiết bị này được sử dụng chỉ bởi cá nhân người dùng. Một vài máy tính này được nối mạng hoặc nối trực tiếp bằng dây hay thông qua các modem không dây. Do sự giới hạn về năng lượng (điện) và giao diện, chúng thực hiện tương đối ít các thao tác ở xa. Hệ điều hành được thiết kế chủ yếu cho việc sử dụng cá nhân nhưng năng lực thực hiện trên thời gian sống của pin cũng là yếu tố quan trọng.
- Một số máy tính người dùng nó mà không để ý đến sự hiện diện của nó. Thí dụ, các máy tính được nhúng vào các thiết bị gia đình và xe ô tô có thể có một bảng số và các đèn hiển thị trạng thái mở, tắt nhưng hầu hết chúng và các hệ điều hành được thiết kế để chạy mà không cần giao tiếp.

## II. Phân loại hệ điều hành theo cấu trúc phần cứng và nguyên lý điều khiển

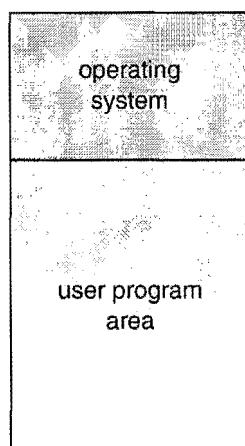
### 2.1. Hệ thống mainframe

Những hệ thống máy tính mainframe là những máy tính đầu tiên được dùng để xử lý ứng dụng thương mại và khoa học. Trong phần này, chúng ta lần theo sự phát triển của hệ thống mainframe từ các hệ thống lô (batch systems), ở đó máy tính chỉ chạy một và chỉ một ứng dụng, tới các hệ chia sẻ thời gian (time-shared systems), mà cho phép người dùng tương tác với hệ thống máy tính.

#### a) Hệ thống điều khiển theo lô đơn giản

Những máy tính thời kỳ đầu là những máy cực lớn chạy từ một thiết bị cuối (console). Những thiết bị nhập thường là những bộ đọc thẻ và các ổ đĩa băng từ. Các thiết bị xuất thông thường là những máy in dòng (line printers), các ổ đĩa từ và các phiếu đục lỗ. Người sử dụng không giao tiếp trực tiếp với các hệ thống máy tính. Thay vào đó, người dùng chuẩn bị công việc: chương trình, dữ liệu và các thông tin điều khiển của công việc và gửi nó đến người điều hành máy tính. Công việc này thường được thực hiện trên các phiếu đục lỗ.

Hệ điều hành trong các máy tính thời kỳ đầu này tương đối đơn giản. Tác vụ chính là chuyển điều khiển tự động từ một công việc này sang công việc khác, mỗi thời điểm chỉ có một công việc được giải quyết, công việc này xong mới chuyển sang công việc khác. Hệ điều hành luôn được thường trú trong bộ nhớ.



**Hình 1.2. Sơ đồ bộ nhớ trong của máy tính thực hiện theo lô đơn giản**

Để tăng tốc việc xử lý, người điều hành tạo một lô (batch) các công việc cùng với yêu cầu và chạy chúng thông qua máy tính như một chương trình (chương trình lô). Do đó, các lập trình viên sẽ đưa chương trình của họ cho người điều hành. Người điều hành sẽ sắp xếp các chương trình và yêu cầu thành lô và khi thực hiện sẽ chạy chương trình lô này.

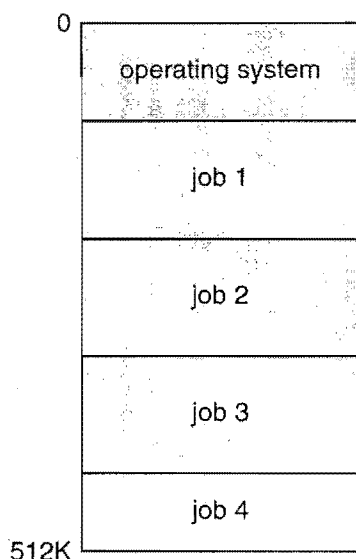
Trong môi trường thực thi này, CPU luôn rỗi (chờ) vì tốc độ của các thiết bị truyền dữ liệu luôn chậm hơn tốc độ của các thiết bị xử lý. Chẳng hạn với hệ thống Pentium 4, bộ xử lý có thể chạy với xung nhịp 3.2 Ghz nhưng đường truyền hệ thống của nó chỉ chạy được với xung nhịp 800 Mhz. Dĩ nhiên theo thời gian, sự tiến bộ trong công nghệ dẫn đến sự ra đời những thiết bị nhập/xuất nhanh hơn. Tuy nhiên, tốc độ CPU tăng với một tỷ lệ lớn hơn vì thế vấn đề không những không được giải quyết mà

còn trầm trọng hơn.

### b) Hệ điều khiển theo lô đa chương

Thông thường, một ứng dụng giữ CPU và luôn tạo cho nó thời gian rỗi vì sự không đồng bộ tốc độ với hệ thống truyền giữ liệu. Đa chương (MultiProgram) gia tăng khả năng sử dụng CPU bằng cách tổ chức các công việc để CPU luôn có một công việc để thực thi, giảm thời gian rỗi.

Ý tưởng của kỹ thuật đa chương có thể minh hoạ như sau: Hệ điều hành giữ nhiều công việc trong bộ nhớ tại một thời điểm. Tập hợp các công việc này là tập con của các công việc được giữ trong vùng công việc, bởi vì số lượng các công việc có thể được giữ cùng lúc trong bộ nhớ thường nhỏ hơn số công việc có thể có trong vùng đệm. Hệ điều hành sẽ lấy và bắt đầu thực thi một trong các công việc có trong bộ nhớ. Trong hệ thống đơn chương, CPU sẽ chờ ở trạng thái rỗi. Trong hệ thống đa chương, hệ điều hành sẽ chuyển sang thực thi công việc khác. Cuối cùng, công việc đầu tiên kết thúc việc chờ và nhận CPU trở lại. Chỉ cần ít nhất một công việc cần thực thi, CPU sẽ không bao giờ ở trạng thái rỗi.



**Hình 1.2.** Sơ đồ đơn giản của bộ nhớ của hệ thống đa chương

Hệ điều hành đa chương tương đối phức tạp. Tất cả công việc đưa vào hệ thống được giữ trong vùng công việc. Vùng này chứa tất cả chương trình ở bên ngoài chờ được cấp phát bộ nhớ chính. Nếu nhiều công việc sẵn sàng được mang vào bộ nhớ và nếu không đủ không gian cho tất cả thì hệ điều hành phải chọn một trong chúng. Khi hệ điều hành chọn một công việc từ vùng công việc, nó nạp công việc đó vào bộ nhớ để thực thi. Có nhiều chương trình trong bộ nhớ tại cùng thời điểm yêu cầu phải có sự quản lý bộ nhớ. Ngoài ra, nếu nhiều công việc sẵn sàng chạy cùng thời điểm, hệ thống phải chọn một trong chúng. Thực hiện quyết định này là định thời CPU. Cuối cùng, nhiều công việc chạy đồng hành mà các hoạt động của chúng có thể ảnh hưởng tới một công việc khác thì được hệ điều hành điều phối bao gồm các việc lập lịch công việc (định thời biểu), lưu trữ đĩa, quản lý bộ nhớ...

### c) Hệ thống chia sẻ thời gian.

Hệ thống lô đa chương cung cấp một môi trường nơi mà nhiều tài nguyên khác nhau (chẳng hạn như CPU, bộ nhớ, các thiết bị ngoại vi) được sử dụng hiệu quả. Tuy nhiên, nó không cung cấp giao tiếp tương tác người dùng với hệ thống máy tính. Chia sẻ thời gian (timesharing hay đa nhiệm) là sự mở rộng logic của đa chương. CPU thực thi nhiều công việc bằng cách chuyển đổi qua lại giữa chúng, nhưng những chuyển đổi xảy ra quá thường xuyên để người dùng có thể giao tiếp với mỗi chương trình trong khi chạy.

Một hệ thống máy tính tương tác (interactive computer) cung cấp giao tiếp trực tuyến giữa người dùng và hệ thống. Người dùng cho những chỉ thị tới hệ điều hành hay trực tiếp tới một chương trình, sử dụng bàn phím hay chuột và chờ nhận kết quả tức thì, do đó thời gian đáp ứng nên ngắn. Một hệ thống chia sẻ thời gian cũng cho phép nhiều người dùng chia sẻ máy tính cùng một thời điểm (MultiUser). Vì mỗi hoạt động hay lệnh trong hệ chia thời được phục vụ ngắn, chỉ một ít thời gian CPU được dành cho mỗi người dùng. Khi hệ thống nhanh chóng chuyển từ một người dùng này sang người dùng kế, mỗi người dùng được cho cảm giác rằng toàn bộ hệ thống máy tính được tận hiến cho mình, nhưng thật sự máy tính đó đang được chia sẻ cho nhiều người dùng.

Một hệ điều hành chia sẻ thời gian sử dụng lịch biểu CPU và đa chương để cung cấp mỗi người dùng với một phần nhỏ của máy tính chia sẻ thời gian. Mỗi người dùng có ít nhất một chương trình riêng trong bộ nhớ. Một chương trình được nạp vào trong bộ nhớ và thực thi thường được gọi là một tiến trình.

Xuất/nhập có thể được giao tiếp; nghĩa là dữ liệu xuất hiện thị trên màn hình cho người dùng và dữ liệu nhập từ bàn phím, chuột hay thiết bị khác. Vì giao tiếp xuất/nhập chủ yếu chạy ở "tốc độ người dùng", nó có thể mất một khoảng thời gian dài để hoàn thành. Thí dụ, dữ liệu nhập có thể bị giới hạn bởi tốc độ nhập của người dùng; 7 ký tự trên giây là nhanh đối với người dùng, nhưng quá chậm so với máy tính. Thay vì để CPU rảnh khi người dùng nhập liệu, hệ điều hành sẽ nhanh chóng chuyển CPU tới một chương trình khác.

Hệ điều hành chia sẻ thời gian phức tạp hơn nhiều so với hệ điều hành đa chương. Trong cả hai dạng, nhiều công việc được giữ cùng lúc trong bộ nhớ vì thế hệ thống phải có cơ chế quản lý bộ nhớ và bảo vệ. Để đạt được thời gian đáp ứng hợp lý, các công việc có thể được hoán vị vào ra bộ nhớ chính. Một phương pháp chung để đạt mục tiêu này là bộ nhớ ảo, là kỹ thuật cho phép việc thực thi của một công việc có thể không hoàn toàn ở trong bộ nhớ chính. Ưu điểm chính của cơ chế bộ nhớ ảo là các chương trình có thể lớn hơn bộ nhớ vật lý. Ngoài ra, nó trừu tượng hoá bộ nhớ chính thành mảng lưu trữ lớn và đồng nhất, chia bộ nhớ logic như được thấy bởi người dùng từ bộ nhớ vật lý. Sự sắp xếp này giải phóng lập trình viên quan tâm đến giới hạn lưu trữ của bộ nhớ.

Các hệ chia sẻ thời gian cũng phải cung cấp một hệ thống tập tin. Hệ thống tập tin định vị trên một tập hợp đĩa; do đó chức năng quản lý đĩa phải được cung cấp. Hệ chia sẻ thời gian cũng cung cấp cơ chế cho việc thực thi đồng hành, yêu cầu cơ chế lập biểu CPU phức tạp. Để đảm bảo thứ tự thực thi, hệ thống phải cung cấp các cơ chế cho việc đồng bộ hoá và giao tiếp công việc, và có thể đảm bảo rằng các công việc không bị deadlock (tắc nghẽn), chờ đợi công việc khác mãi mãi.

Ý tưởng chia sẻ thời gian được giới thiệu từ những năm 1960, nhưng vì hệ chia

sẽ thời gian là phức tạp và rất đắt để xây dựng, chúng không phổ biến cho tới những năm 1970. Mặc dù xử lý theo lô vẫn được thực hiện nhưng hầu hết hệ thống ngày nay là chia sẻ thời gian. Do đó, đa chương và chia sẻ thời gian là những chủ đề trung tâm của hệ điều hành hiện đại và chúng là chủ đề trọng tâm của giáo trình này.

## 2.2. Hệ thống để bàn (Desktop system)

Máy tính cá nhân (PC) xuất hiện vào những năm 1970. Trong suốt thập niên đầu, CPU trong PC thiếu các đặc điểm cần thiết để bảo vệ hệ điều hành từ chương trình ứng dụng. Do đó, các hệ điều hành PC không phải là đa người dùng hoặc đa nhiệm.

Tuy nhiên, các mục tiêu của hệ điều hành này thay đổi theo thời gian; thay vì tối ưu hoá việc sử dụng CPU và thiết bị ngoại vi, các hệ thống chọn lựa tối ưu hoá sự tiện dụng và đáp ứng người dùng. Các hệ thống này gồm các PC chạy các hệ điều hành Microsoft Windows và Apple Macintosh. Hệ điều hành MS-DOS từ Microsoft được thay thế bằng nhiều ấn bản của Microsoft Windows và IBM đã nâng cấp MS-DOS thành hệ đa nhiệm OS/2. Hệ điều hành Apple Macintosh được gắn nhiều phần cứng hiện đại hơn và ngày nay chứa nhiều đặc điểm mới như bộ nhớ ảo và đa nhiệm. Với sự phát hành MacOS X, lõi của hệ điều hành ngày nay dựa trên Mach và FreeBSD UNIX cho sự mở rộng năng lực nhưng nó vẫn giữ lại giao diện đồ hoạ người dùng GUI. LINUX, một hệ điều hành tương tự như UNIX dùng cho máy PC trở nên phổ biến gần đây.

Hệ điều hành cho các máy tính này có những thuận lợi trong nhiều cách từ sự phát triển của hệ điều hành cho mainframes. Máy vi tính (microcomputer) lập tức có thể được chấp nhận một số công nghệ được phát triển cho hệ điều hành lớn hơn. Thêm vào đó, chi phí phần cứng cho máy vi tính đủ thấp để các cá nhân có thể một mình sử dụng máy tính. Do đó, một số nguyên lý được thực hiện trong hệ điều hành cho mainframes có thể không hợp lý cho các hệ thống nhỏ hơn.

Những nguyên lý khác vẫn được áp dụng. Thí dụ, trước hết bảo vệ hệ thống tập tin không cần thiết trên máy cá nhân. Tuy nhiên, hiện nay các máy tính này thường được nối vào các máy tính khác qua mạng cục bộ hay Internet. Khi những máy tính khác và người dùng khác có thể truy xuất các tập tin này trên một PC, bảo vệ tập tin một lần nữa cũng trở thành một đặc điểm cần thiết của hệ điều hành. Sự thiếu bảo vệ tạo điều kiện dễ dàng cho những chương trình nguy hiểm phá huỷ dữ liệu trên những hệ thống như MS-DOS và hệ điều hành Macintosh. Các chương trình này có thể tự nhân bản và phát tán nhanh chóng bằng cơ chế worm hay virus và làm tê liệt mạng của các công ty hay thậm chí mạng toàn cầu. Đặc điểm được cải tiến như bộ nhớ bảo vệ và quyền tập tin là chưa đủ để bảo vệ một hệ thống từ sự tấn công bên ngoài.

## 2.3. Hệ thống đa bộ xử lý

Hầu hết các hệ thống ngày nay là các hệ thống đơn bộ xử lý, nghĩa là chỉ có một CPU chính. Tuy nhiên, các hệ thống đa bộ xử lý (hay còn gọi là hệ song song hay hệ kết nối chặt) vẫn được chú trọng phát triển. Các hệ thống như thế có nhiều hơn một bộ xử lý, chia sẻ bus máy tính, đồng hồ xung nhịp, đôi khi còn là bộ nhớ hay thiết bị ngoại vi.

Hệ thống đa xử lý có ba ưu điểm chính:

- **Thông lượng được gia tăng:** bằng cách tăng số lượng bộ xử lý, chúng

ta hy vọng thực hiện nhiều công việc hơn với thời gian ít hơn. Tỉ lệ giữa sự tăng tốc với  $N$  bộ xử lý không là  $N$ ; đúng hơn nó nhỏ hơn  $N$ . Trong khi nhiều bộ xử lý cộng tác trên một công việc, một lượng chi phí phải chịu trong việc giữ các thành phần làm việc phù hợp. Chi phí này cộng với chi phí cạnh tranh tài nguyên được chia sẻ, làm giảm kết quả được mong đợi từ những bộ xử lý bổ sung. Tương tự như một nhóm gồm  $N$  lập trình viên làm việc với nhau không dẫn đến kết quả công việc đang đạt được tăng  $N$  lần.

○ **Tính kinh tế của việc mở rộng:** hệ thống đa xử lý có thể tiết kiệm nhiều chi phí hơn hệ thống đơn bộ xử lý, bởi vì chúng có thể chia sẻ ngoại vi, thiết bị lưu trữ và điện. Nếu nhiều chương trình điều hành trên cùng tập hợp dữ liệu thì lưu trữ dữ liệu đó trên một đĩa và tất cả bộ xử lý chia sẻ chúng sẽ rẻ hơn là có nhiều máy tính với đĩa cục bộ và nhiều bản sao dữ liệu.

○ **Khả năng tin cậy được gia tăng:** nếu các chức năng được phân bổ hợp lý giữa các bộ xử lý thì lỗi trên một bộ xử lý sẽ không dừng hệ thống, chỉ năng lực bị giảm. Nếu chúng ta có 10 bộ xử lý và có 1 bộ xử lý bị sự cố thì mỗi bộ xử lý trong 9 bộ xử lý còn lại phải chia sẻ của công việc của bộ xử lý bị lỗi. Do đó, toàn bộ hệ thống chỉ giảm 10% năng lực hơn là dừng hoạt động. Các hệ thống được thiết kế như thế được gọi là hệ thống có khả năng chịu lỗi (fault tolerant). Việc điều hành vẫn tiếp tục trong sự hiện diện của lỗi yêu cầu một cơ chế cho phép lỗi được phát hiện, chuẩn đoán và sửa lỗi nếu có thể. Hệ thống Tandem sử dụng sự nhân đôi phần cứng và phần mềm để đảm bảo sự điều hành vẫn tiếp tục mặc dù có lỗi xảy ra. Hệ thống này chứa hai bộ xử lý, mỗi bộ xử lý có bộ nhớ cục bộ riêng. Các bộ xử lý được nối kết bởi một bus. Một bộ xử lý chính và bộ xử lý kia là dự phòng. Cả hai bản sao được giữ ở mỗi bộ xử lý: một là chính và một là dự phòng. Tại các điểm kiểm tra (checkpoints) trong việc thực thi của hệ thống, thông tin trạng thái của mỗi công việc-gồm một bản sao hình ảnh bộ nhớ-được chép từ máy chính tới máy dự phòng. Nếu một lỗi được phát hiện, bản sao dự phòng được kích hoạt và được khởi động lại từ điểm kiểm tra mới nhất. Giải pháp này đắt vì nó bao gồm việc nhân đôi phần cứng.

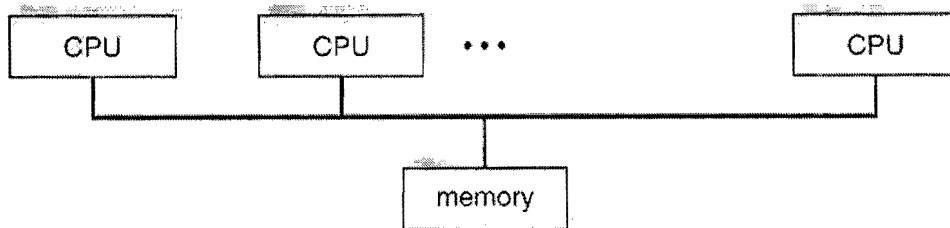
Các hệ thống đa xử lý thông dụng nhất hiện nay sử dụng **đa xử lý đối xứng** (symmetric multiprocessing-SMP). Trong hệ thống này mỗi bộ xử lý chạy bản sao của hệ điều hành và những bản sao này giao tiếp với các bản sao khác khi cần.

Vài hệ thống sử dụng **đa xử lý bất đối xứng** (asymmetric multiprocessing). Trong hệ thống này mỗi bộ xử lý được gán một công việc xác định. Một bộ xử lý chủ điều khiển hệ thống; những bộ xử lý còn lại hoặc chờ bộ xử lý chủ ra chỉ thị hoặc có những tác vụ được định nghĩa trước. Cơ chế này định nghĩa mối quan hệ chủ-tớ. Bộ xử lý chính lập thời biểu và cấp phát công việc tới các bộ xử lý tớ.

Đa xử lý đối xứng có nghĩa tất cả bộ xử lý là ngang hàng; không có mối quan hệ chủ-tớ tồn tại giữa các bộ xử lý. Hình 1.3. minh họa một kiến trúc đa xử lý đối xứng điển hình. Một thí dụ của đa xử lý đối xứng là ấn bản của Encore của UNIX cho máy tính Multimax. Máy tính này có thể được cấu hình như nó đang thực hiện nhiều bộ xử lý, tất cả bộ xử lý đều chạy bản sao của UNIX. Ưu điểm của mô hình này là nhiều quá trình có thể chạy cùng một lúc -  $N$  quá trình có thể chạy nếu có  $N$  CPU- không gây ra sự giảm sút to lớn về năng lực. Tuy nhiên, chúng ta phải điều khiển cẩn thận xuất/nhập để đảm bảo rằng dữ liệu dẫn tới bộ xử lý tương ứng. Vì các CPU là riêng rẽ, một CPU có thể đang rảnh trong khi CPU khác quá tải dẫn đến việc sử dụng không hữu hiệu tài



nguyên của hệ thống. Sự không hiệu quả này có thể tránh được nếu các bộ xử lý chia sẻ các cấu trúc dữ liệu. Một hệ thống đa xử lý của dạng này sẽ cho phép các quá trình và tài nguyên – như bộ nhớ - được chia sẻ tự động giữa các quá trình khác nhau và có thể làm giảm sự khác biệt giữa các bộ xử lý. Hầu như tất cả hệ điều hành hiện đại - gồm Windows NT, Solaris, Digital UNIX, OS/2 và LINUX - hiện nay cung cấp sự hỗ trợ đa xử lý đối xứng.



**Hình 1.3.** Kiến trúc đa xử lý đối xứng

Sự khác biệt giữa đa xử lý đối xứng và bất đối xứng có thể là do phần cứng hoặc phần mềm. Phần cứng đặc biệt có thể khác nhau trên nhiều bộ xử lý, hoặc phần mềm có thể được viết để cho phép chỉ một chủ và nhiều tớ. Thí dụ, SunOS ấn bản 4 cung cấp đa xử lý không đối xứng, ngược lại, ấn bản 5 (Solaris 2) là đối xứng trên cùng phần cứng.

Khi các bộ vi xử lý trở nên rẻ hơn và mạnh hơn các chức năng bổ sung của hệ điều hành là chuyển tới bộ xử lý tớ. Thí dụ, tương đối dễ để thêm bộ vi xử lý với bộ nhớ riêng để quản lý hệ thống đĩa. Bộ vi xử lý có thể nhận một chuỗi các yêu cầu từ bộ nhớ chính và cài đặt hàng đợi đĩa riêng và giải thuật định thời. Sự sắp xếp này làm giảm chi phí định thời đĩa của CPU. PC chứa một bộ vi xử lý trong bàn phím để chuyển những phím nóng thành mã để gởi tới CPU. Thực tế, việc sử dụng các bộ vi xử lý trở nên quá phổ biến đến nỗi mà đa xử lý không còn được xem xét.

## 2.4. Hệ phân tán

Một mạng, trong thuật ngữ đơn giản nhất, là một đường dẫn truyền thông giữa hai hay nhiều hệ thống. Hệ phân tán phụ thuộc vào mạng với những khả năng của nó. Bằng cách cho phép truyền thông, hệ phân tán có thể chia sẻ các tác vụ tính toán và cung cấp nhiều chức năng tới người dùng.

Các mạng với sự đa dạng về giao thức được dùng, khoảng cách giữa các nút và phương tiện truyền. TCP/IP là giao thức mạng phổ biến nhất mặc dù ATM và các giao thức khác được sử dụng rộng rãi. Tương tự, hệ điều hành hỗ trợ sự đa dạng về giao thức. Hầu hết các hệ điều hành hỗ trợ TCP/IP, gồm Windows và UNIX. Một số hệ điều hành khác hỗ trợ các giao thức riêng phù hợp với yêu cầu của chúng. Đối với một hệ điều hành, một giao thức mạng chỉ cần một thiết bị giao diện – thí dụ: một card mạng-với một trình điều khiển thiết bị để quản lý nó và một phần mềm để đóng gói dữ liệu trong giao thức giao tiếp để gởi nó và mở gói để nhận nó.

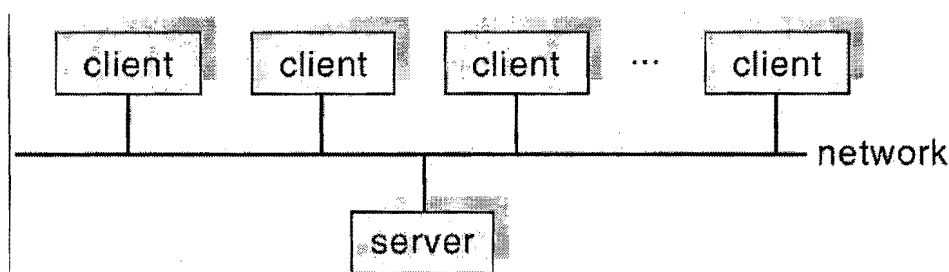
Mạng thường dựa trên các khoảng cách giữa các nút. Một mạng cục bộ (LAN-Local Area Network) tồn tại trong phạm vi một phòng, một tầng, hay một toà nhà. Một mạng diện rộng (Wide-Area Network) thường tồn tại giữa các toà nhà, các thành phố, các quốc gia. Một công ty toàn cầu có thể có một mạng diện rộng để nối kết tới các văn phòng của nó. Các mạng này có thể chạy với một hay nhiều giao thức. Sự tiến bộ liên tục của công nghệ mới hình thành nhiều dạng mạng khác nhau. Thí dụ, mạng đô

thị (MAN-Metropolitan Area Network) cũng liên kết các toà nhà trong cùng một thành phố. Các thiết bị Bluetooth giao tiếp qua một khoảng cách ngắn khoảng vài bộ (feet), chủ yếu tạo ra một mạng phạm vi nhỏ (small-area network).

Các phương tiện truyền thông làm các mạng tương đối khác nhau. Chúng gồm cáp đồng, cáp quang, truyền không dây giữa vệ tinh, các đĩa vi sóng (microwave dishes) và sóng radio. Khi các thiết bị tính toán được nối kết tới các điện thoại di động, chúng tạo ra một mạng. Thậm chí rất nhiều giao tiếp hồng ngoại dây ngắn có thể được dùng cho mạng. Tại cấp độ cơ bản, bất cứ khi nào các máy tính giao tiếp chúng sử dụng hay tạo ra một mạng. Các mạng này cũng rất khác nhau về năng lực và khả năng tin cậy.

### Hệ thống khách hàng-máy phục vụ (client-server)

Một PC có thể trở nên nhanh hơn, mạnh hơn, rẻ hơn nếu người thiết kế thay đổi rất xa từ kiến trúc hệ thống tập trung. Các thiết bị đầu cuối được nối kết tới các hệ tập trung hiện nay đang được thay thế bởi các PC. Tương ứng, chức năng giao diện người dùng được dùng quản lý trực tiếp bởi các hệ tập trung đang được quản lý tăng dần bởi các PC. Do đó, các hệ tập trung ngày nay hoạt động như hệ máy phục vụ để thoả mãn các yêu cầu phát sinh bởi hệ thống khách hàng. Đặc điểm chung của hệ khách hàng-máy phục vụ được mô tả trong hình 1.4.



**Hình 1.4.** Hệ thống máy khách hàng – đại lý phục vụ

Các hệ máy chủ có thể được phân loại rộng rãi như máy phục vụ tính toán và máy phục vụ tập tin

- **Hệ máy phục vụ tính toán** (Compute-server systems): cung cấp giao diện mà khách hàng có thể gửi các yêu cầu để thực hiện hoạt động. Chúng thực thi hoạt động và gửi kết quả cho khách hàng.

- **Hệ máy phục vụ tập tin** (File-server systems ): cung cấp một giao diện hệ thống tập tin nơi khách hàng có thể tạo, cập nhật, đọc và xoá tập tin.

### Hệ thống ngang hàng (peer-to-peer)

Sự phát triển của mạng máy tính-đặc biệt là Internet và Word Wide Web (WWW)-có ảnh hưởng sâu sắc đến sự phát triển gần đây của hệ điều hành. Khi PC được giới thiệu vào những năm 1970, chúng được thiết kế cho việc sử dụng “cá nhân” và thường được xem như là các máy tính đơn lẻ (standalone computer). Với việc bắt đầu sử dụng Internet phổ biến và rộng rãi vào những năm 1980 với e-mail, ftp, gopher, nhiều PC được nối vào mạng máy tính. Với sự giới thiệu dịch vụ Web vào giữa những năm 1990, nối kết mạng trở thành một thành phần quan trọng của một hệ thống mạng

máy tính.

Gần như tất cả máy tính PC hiện đại và các trạm làm việc có thể chạy trình duyệt Web để truy xuất tài liệu siêu văn bản trên Web. Các hệ điều hành (như Windows, OS/2, MacOS và UNIX) hiện nay cũng chứa phần mềm hệ thống (như TCP/IP và PPP) cho phép một máy tính truy xuất Internet qua một mạng cục bộ hay nối kết qua đường điện thoại. Nhiều hệ điều hành chứa trình duyệt Web cũng như khách hàng và máy phục vụ e-mail, đăng nhập từ xa, và ftp.

Tương phản với các hệ thống kết nối chặt được thảo luận trong phần 2.3, mạng máy tính được dùng trong các ứng dụng này gồm tập hợp các bộ xử lý không chia sẻ bộ nhớ hay đồng hồ. Thay vào đó, mỗi bộ xử lý có bộ nhớ cục bộ riêng. Bộ xử lý giao tiếp với bộ xử lý khác thông qua các đường truyền thông như các bus tốc độ cao hay các đường điện thoại. Các hệ thống này thường được xem như các hệ thống kết nối lỏng (hay hệ thống phân tán).

Vài hệ điều hành thực hiện khái niệm mạng hơn là chú trọng cung cấp nối kết mạng. Một hệ điều hành mạng là một hệ điều hành cung cấp các đặc tính như chia sẻ tập tin qua mạng, nó chứa một cơ chế giao tiếp cho phép các quá trình khác nhau trên các máy khác nhau trao đổi thông điệp. Một máy tính chạy một hệ điều hành mạng hoạt động tự trị từ tất cả máy tính khác trên mạng, mặc dù nó nhận thức sự hiện diện của mạng và có thể giao tiếp với các máy tính được nối mạng khác. Một hệ điều hành phân tán là một môi trường ít tự trị hơn: Các hệ điều hành phân tán giao tiếp đủ gần để cung cấp một hình ảnh mà chỉ một hệ điều hành đơn lẻ điều khiển mạng.

## 2.5. Hệ thống bó (Clustered Systems)

Tương tự các hệ song song, hệ thống bó tập hợp nhiều CPU với nhau để thực hiện công việc tính toán. Tuy nhiên, hệ thống bó khác hệ thống song song ở điểm chúng được hợp thành từ hai hay nhiều hệ thống đơn được kết hợp với nhau.

Định nghĩa của thuật ngữ bó (clustered) là không cụ thể. Định nghĩa thông thường có thể chấp nhận là các máy tính nhóm chia sẻ việc lưu trữ và được liên kết gần qua LAN. Bó thường được thực hiện để cung cấp khả năng an toàn cao. Một lớp phần mềm bó chạy trên các nút bó (cluster nodes). Mỗi nút có thể kiểm soát một hay nhiều hơn một nút (qua mạng LAN). Nếu máy bị kiểm soát gặp sự cố, máy kiểm soát có thể lấy quyền sở hữu việc lưu trữ của nó và khởi động lại (các) ứng dụng mà chúng đang chạy trên máy bị sự cố. Máy bị sự cố vẫn chưa hoạt động nhưng người dùng và khách hàng của ứng dụng chỉ thấy một sự gián đoạn ngắn của dịch vụ.

Trong bó bất đối xứng (asymmetric clustering), một máy ở trong chế độ dự phòng nóng (hot standby) trong khi các máy khác đang chạy các ứng dụng. Máy dự phòng không là gì cả ngoại trừ theo dõi server hoạt động. Nếu server đó bị lỗi, máy chủ dự phòng nóng trở thành server hoạt động. Trong chế độ đối xứng (symmetric mode), hai hay nhiều máy chủ đang chạy ứng dụng và chúng đang kiểm soát lẫn nhau. Chế độ này chú trọng tính hiệu quả khi nó sử dụng tất cả phần cứng sẵn có. Nó thực hiện yêu cầu nhiều hơn một ứng dụng sẵn dùng để chạy.

Các hình thức khác của bó gồm các bó song song (parallel clusters) và bó qua một WAN. Các bó song song cho phép nhiều máy chủ truy xuất cùng dữ liệu trên thiết bị lưu trữ được chia sẻ. Vì hầu hết các hệ điều hành hỗ trợ nghèo nàn việc truy xuất dữ liệu đồng thời bởi nhiều máy chủ, các bó song song thường được thực hiện bởi các ấn

bản phần mềm đặc biệt và sự phát hành của các ứng dụng đặc biệt. Thí dụ, Oracle Parallel Server là một ấn bản cơ sở dữ liệu của Oracle, và lớp phần mềm ghi vết việc truy xuất tới đĩa được chia sẻ. Mỗi máy có truy xuất đầy đủ tới dữ liệu trong cơ sở dữ liệu.

Mặc dù có nhiều cải tiến trong tính toán phân tán, hầu hết các hệ thống không cung cấp các hệ thống tập tin phân tán mục đích chung (general-purpose distributed file systems). Do đó, hầu hết các bó không cho phép truy xuất được chia sẻ tới dữ liệu trên đĩa. Cho mục đích này, các hệ thống tập tin phân tán phải cung cấp điều khiển truy xuất và khoá các tập tin để đảm bảo không có các thao tác xung đột xảy ra. Loại dịch vụ này thường được gọi là bộ quản lý khoá phân tán (distributed lock manager-DLM).

Công nghệ bó đang nhanh chóng thay đổi. Những định hướng bó gồm các bó toàn cục, trong đó các máy có thể định vị bất cứ nơi nào trên thế giới (hay bất cứ nơi nào một WAN đạt tới). Các dự án như thế vẫn là chủ đề cho nghiên cứu và phát triển.

Việc sử dụng hệ thống bó và các đặc tính nên mở rộng như mạng vùng lưu trữ (storage-area networks-SAN) chiếm ưu thế. SAN cho phép gắn dễ dàng nhiều máy chủ tới nhiều đơn vị lưu trữ. Các bó hiện tại thường bị giới hạn tới hai hay bốn máy chủ do sự phức tạp của nối kết các máy chủ tới thiết bị lưu trữ được chia sẻ.

## 2.6. Hệ thống thời gian thực

Một dạng khác của hệ điều hành có mục đích đặc biệt là hệ thời gian thực (real-time system). Hệ thời gian thực được dùng khi các yêu cầu thời gian khắt khe được đặt trên thao tác của một bộ xử lý hay dòng dữ liệu; do đó, nó thường được dùng như một thiết bị điều khiển trong một ứng dụng tận hiến. Các bộ cảm biến mang dữ liệu tới máy tính.

Máy tính phải phân tích dữ liệu và có thể thích ứng các điều khiển để hiệu chỉnh các dữ liệu nhập cảm biến. Các hệ thống điều khiển các thí nghiệm khoa học, hệ thống ảnh hoá y tế, hệ thống điều khiển công nghệ và các hệ thống hiển thị,... Các hệ thống phun dầu động cơ ô tô, các bộ điều khiển dụng cụ trong nhà, hệ thống vũ khí cũng là các hệ thống thời gian thực.

Một hệ thống thời gian thực có sự ràng buộc cố định, rõ ràng. Xử lý phải được thực hiện trong phạm vi các ràng buộc được định nghĩa hay hệ thống sẽ thất bại. Một hệ thời gian thực thực hiện đúng chức năng chỉ nếu nó trả về kết quả đúng trong thời gian ràng buộc. Tương phản với yêu cầu này trong hệ chia thời, ở đó nó mong muốn (nhưng không bắt buộc) đáp ứng nhanh, hay đối với hệ thống bó, nó không có ràng buộc thời gian gì cả.

Hệ thời gian thực có hai dạng: cứng và mềm. Hệ thời gian thực cứng đảm bảo rằng các tác vụ tới hạn được hoàn thành đúng giờ. Mục tiêu này đòi hỏi tất cả trì hoãn trong hệ thống bị giới hạn, từ việc lấy lại dữ liệu được lưu trữ thời gian hệ điều hành hoàn thành bất cứ yêu cầu cho nó. Các ràng buộc thời gian như thế ra lệnh các phương tiện sẵn có trong hệ thời gian thực cứng. Thiết bị lưu trữ phụ của bất cứ thứ hạng nào thường bị giới hạn hay bị mất với dữ liệu đang được lưu trong bộ nhớ lưu trữ ngắn hạn (short-term memory) hay trong bộ nhớ chỉ đọc (ROM). Hầu hết các hệ điều hành hiện đại không cung cấp đặc điểm này vì chúng có khuynh hướng tách rời người dùng từ phần cứng và sự tách rời này dẫn đến lượng thời gian không xác định mà thao tác sẽ

mất. Thí dụ, bộ nhớ ảo hầu như chưa bao giờ thấy trong hệ thời gian thực. Do đó, những hệ thời gian thực cứng xung đột với thao tác của hệ chia thời và hai hệ này không thể đan xen nhau. Vì không có hệ điều hành đa mục đích đã có hỗ trợ chức năng thời gian thực cứng; chúng ta không tập trung với loại hệ thống này trong chương này.

Một loại thời gian thực ít hạn chế hơn là hệ thời gian thực mềm, ở đó tác vụ thời gian thực tới hạn có độ ưu tiên hơn các tác vụ khác và duy trì độ ưu tiên đó cho đến khi chúng hoàn thành. Như trong hệ thời gian thực cứng, sự trì hoãn nhân (kernel) của hệ điều hành trì hoãn yêu cầu được giới hạn. Một tác vụ thời gian thực không thể giữ việc chờ không xác định đối với nhân để thực thi. Thời gian thực mềm là mục tiêu có thể đạt được và có thể được đan xen với các loại hệ thống khác. Tuy nhiên, hệ thời gian thực mềm có những tiện ích giới hạn hơn hệ thời gian thực cứng. Vì không hỗ trợ tốt cho thời điểm tới hạn, nên hệ thời gian thực mềm dễ gây rủi ro khi dùng cho việc kiểm soát công nghệ và tự động hoá.

Tuy nhiên, chúng có ích trong nhiều lĩnh vực như đa phương tiện, thực tế ảo, dự án khoa học tiên tiến-như khám phá trong lòng đại dương và khám phá hành tinh. Những hệ thống này cần những đặc điểm hệ điều hành tiên tiến mà không được hỗ trợ bởi hệ thời gian thực cứng. Vì việc sử dụng chức năng thời gian thực mềm được mở rộng nên chúng ta đang tìm cách đưa chúng vào trong hầu hết các hệ điều hành hiện tại, gồm các bản chính thức của UNIX.

## 2.7. Hệ thống cầm tay

Hệ cầm tay gồm các máy hỗ trợ cá nhân dùng kỹ thuật số (personal digital assistants-PDAs) như Palm hay điện thoại di động (cellular telephone) với nối kết tới mạng như Internet. Những người phát triển hệ cầm tay và ứng dụng gặp phải nhiều thử thách, nhất là sự giới hạn về kích thước của thiết bị. Thí dụ, một PDA điển hình cao khoảng 5 inches và rộng khoảng 3 inches và trọng lượng của nó ít hơn 0.5 pound.

Do sự giới hạn về kích thước này, hầu hết các thiết bị cầm tay có bộ nhớ nhỏ gồm các bộ xử lý thấp và màn hình hiển thị nhỏ. Bây giờ chúng ta sẽ xem xét mỗi sự giới hạn này.

Nhiều thiết bị cầm tay có dung lượng bộ nhớ 512KB và 8 MB (ngược lại, các máy PC hay trạm làm việc có hàng trăm MB bộ nhớ). Do đó, hệ điều hành và các ứng dụng phải quản lý bộ nhớ hiệu quả. Điều này gồm trả về tất cả bộ nhớ được cấp phát tới bộ quản lý bộ nhớ một khi bộ nhớ không còn được dùng nữa. Hiện nay, nhiều thiết bị cầm tay không dùng kỹ thuật bộ nhớ ảo do đó buộc người phát triển chương trình làm việc trong phạm vi giới hạn của bộ nhớ vật lý.

Vấn đề thứ hai quan tâm đến người phát triển các thiết bị cầm tay là tốc độ của bộ xử lý được dùng trong thiết bị. Các bộ xử lý đối với hầu hết các thiết bị cầm tay thường chạy với tốc độ chỉ bằng một phần tốc độ của một bộ xử lý trong máy PC. Các bộ xử lý nhanh hơn yêu cầu điện năng nhiều hơn. Để chứa một bộ xử lý nhanh hơn bên trong thiết bị cầm tay nên yêu cầu nhiều pin hơn và phải được nạp lại thường xuyên. Để tối thiểu hoá kích thước của các thiết bị cầm tay đòi hỏi bộ xử lý nhỏ hơn, chậm hơn tiêu thụ ít điện năng hơn. Do đó, hệ điều hành và các ứng dụng phải được thiết kế không đòi hỏi sử dụng nhiều bộ xử lý.

Vấn đề cuối cùng gây khó khăn cho người thiết kế chương trình cho các thiết bị

cầm tay là màn hình hiển thị nhỏ. Trong khi một màn hình cho máy tính ở nhà kích thước có thể 21 inches, màn hình cho thiết bị cầm tay thường có diện tích không quá 3 inches. Những tác vụ quen thuộc như đọc e-mail hay hiển thị các trang web, phải được cô đọng vào màn hình nhỏ hơn. Một phương pháp để hiển thị nội dung các trang web là cắt xén web (web clipping), ở đó chỉ một tập hợp nhỏ trang web được phân phát và hiển thị trên thiết bị cầm tay.

Một số thiết bị cầm tay có thể dùng công nghệ không dây như Bluetooth, cho phép truy xuất từ xa tới e-mail và trình duyệt web. Các điện thoại di động với nối kết Internet thuộc loại này. Tuy nhiên, nhiều PDA hiện tại không cung cấp truy xuất không dây. Để tải dữ liệu xuống các thiết bị này, trước tiên người dùng tải dữ liệu xuống PC hay trạm và sau đó tải dữ liệu xuống PDA. Một số PDA cho phép dữ liệu chép trực tiếp từ một thiết bị này tới thiết bị khác dùng liên kết hồng ngoại. Nhìn chung, các giới hạn trong chức năng của PDA được cân bằng bởi những tiện dụng và linh động của chúng. Việc sử dụng chúng tiếp tục mở rộng khi các nối kết mạng trở nên sẵn dùng và các chọn lựa khác như máy ảnh và MP3 players, mở rộng tiện ích của chúng.

## 2.8. Ghi nhớ

Hệ điều hành được phát triển hơn 45 năm qua với hai mục đích chính. Mục đích thứ nhất, hệ điều hành cố gắng lập thời biểu các hoạt động tính toán để đảm bảo năng lực thực hiện của hệ thống là tốt. Mục đích thứ hai, nó cung cấp một môi trường tiện dụng để phát triển và thực thi chương trình. Ban đầu, hệ thống máy tính được dùng từ một màn hình và bàn phím (thiết bị đầu cuối). Các phần mềm như bộ hợp ngữ (assembler), bộ nạp (loader), bộ liên kết (linkers) và các trình biên dịch (compiler) cải tiến sự tiện dụng của việc lập trình hệ thống nhưng cũng yêu cầu thời gian thiết lập đáng kể. Để giảm thời gian thiết lập, các phương tiện thuê người điều hành và các công việc tương tự được bỏ.

Các hệ thống xử lý theo lô cho phép sắp xếp công việc tự động bởi hệ điều hành và cải tiến rất nhiều việc tận dụng toàn bộ máy tính. Máy tính không còn phải chờ các thao tác của người dùng. Tuy nhiên, việc tận dụng CPU vẫn còn thấp vì tốc độ của thiết bị xuất nhập thấp hơn nhiều so với tốc độ của CPU. Thao tác ngoại vi (off-line operation) của các thiết bị chậm cung cấp một phương tiện sử dụng nhiều hệ thống bộ đọc tới băng từ (reader-to-tape) và băng từ tới máy in (tape-to-printer) cho một CPU. Để cải tiến toàn bộ năng lực thực hiện của hệ thống máy tính, người phát triển giới thiệu khái niệm đa chương để mà nhiều công việc có thể được giữ cùng lúc trong bộ nhớ tại cùng một thời điểm. CPU được chuyển qua lại giữa chúng để gia tăng việc tận dụng CPU và giảm toàn bộ thời gian được yêu cầu để thực thi các công việc.

Đa chương cũng cho phép chia sẻ thời gian. Hệ điều hành chia sẻ thời gian cho phép nhiều người dùng (từ một tới vài trăm) sử dụng hệ thống máy tính giao tiếp tại cùng một thời điểm.

PC là máy vi tính; chúng xem như nhỏ hơn và rẻ hơn hệ thống mainframe. Các hệ điều hành cho các máy tính này lợi hơn việc phát triển hệ điều hành cho máy tính mainframe trong nhiều cách. Tuy nhiên, vì mỗi cá nhân là người dùng duy nhất sử dụng máy tính nên việc tận dụng CPU không còn là mối quan tâm chủ yếu. Do đó, một vài quyết định thiết kế được thực hiện cho hệ điều hành cho máy mainframe có thể không phù hợp cho cả hệ thống nhỏ và lớn, hiện nay khi các PCs có thể được nối kết

tới các máy tính khác và người dùng thông qua mạng và Web.

Các hệ song song có nhiều hơn một CPU trong giao tiếp gần; các CPU chia sẻ bus máy tính và đôi khi chia sẻ bộ nhớ và các thiết bị ngoại vi. Những hệ thống như thế có thể cung cấp thông lượng và khả năng tin cậy tăng. Các hệ thống phân tán cho phép chia sẻ tài nguyên trên những máy chủ được phân tán về mặt địa lý. Các hệ thống được nhóm cho phép nhiều máy thực hiện việc tính toán trên dữ liệu được chứa trên thiết bị lưu trữ chia sẻ và để việc tính toán tiếp tục trong trường hợp lỗi của tập hợp con các thành viên nhóm.

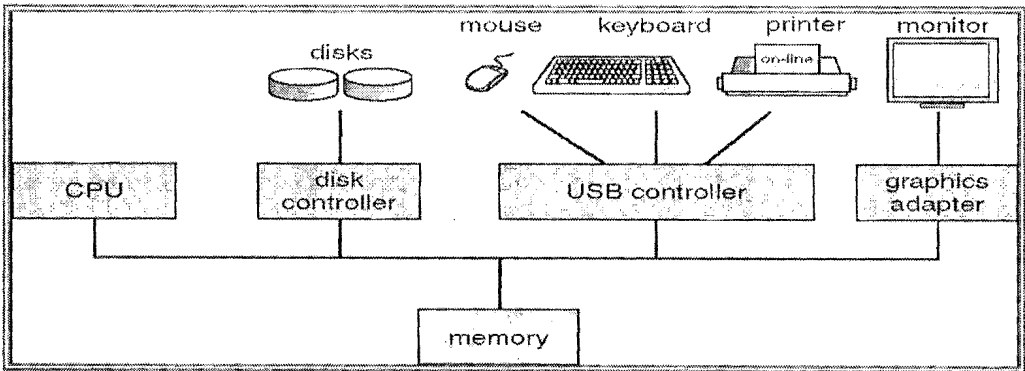
Một hệ thời gian thực cứng thường được dùng như một thiết bị điều khiển trong một ứng dụng tận hiến. Một hệ điều hành thời gian thực cứng có ràng buộc hoàn toàn xác định và thời gian cố định. Xử lý phải được thực hiện trong các ràng buộc được xác định hoặc hệ thống sẽ bị lỗi. Các hệ thống thời gian thực mềm có ràng buộc thời gian ít nghiêm khắc hơn và không hỗ trợ thời biểu tới hạn.

Gần đây sự tác động của Internet và World Wide Web khuyến khích sự phát triển của các hệ điều hành hiện đại. Các hệ điều hành này chứa các trình duyệt Web, mạng và phần mềm truyền thông như là các đặc điểm tích hợp.

Chúng ta đã thể hiện tiến trình logic của sự phát triển hệ điều hành, được định hướng bởi sự bao gồm các đặc điểm trong phần cứng CPU được yêu cầu cho chức năng tiên tiến. Xu hướng này có thể được thấy ngày nay trong cuộc cách mạng của PC, với phần cứng ngày một rẻ hơn và đang được cải tiến đủ để cho phép cải tiến các đặc điểm.

3. Tổ chức hoạt động của hệ thống máy tính

Một thiết bị nhập/xuất (I/O) thường bao gồm một phần cơ và phần điện tử. Thành phần điện tử được gọi là bộ điều khiển (I/O device) hay bộ điều hợp (adapter), trên máy vi tính bộ điều hợp thường có dạng thẻ (card) ghép nối qua một khe cắm hoặc được gắn lên bảng mạch chính. Thành phần cơ chính là bản thân thiết bị đó. Hệ điều hành thường làm việc (điều khiển) bộ điều khiển chứ không làm việc trực tiếp với thành phần cơ. Tức là hệ điều hành thường không trực tiếp điều khiển thiết bị nhập/xuất mà chỉ gửi yêu cầu đến bộ điều khiển, còn bộ điều khiển sẽ trực tiếp điều khiển các thiết bị nhập/xuất theo yêu cầu của hệ điều hành.



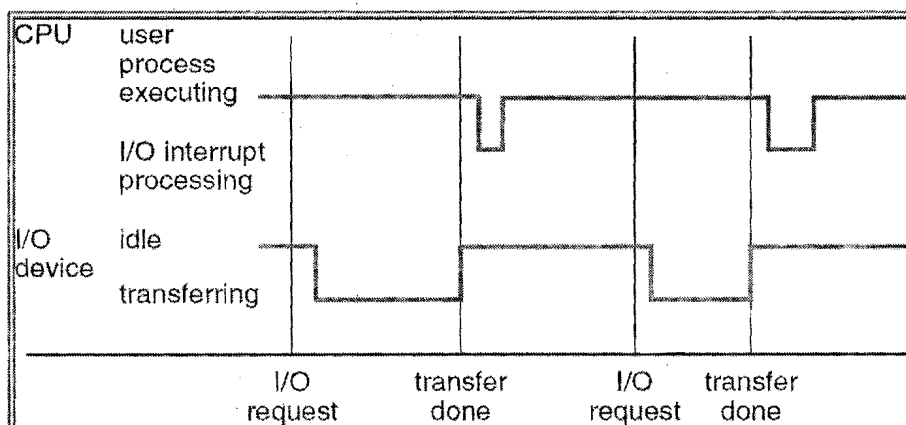
Hình 1.5. Sơ đồ cấu trúc phần cứng máy tính

Hầu hết các máy tính cỡ nhỏ đều dùng đường truyền dữ liệu (bus) để liên lạc giữa bộ xử lý và các bộ điều khiển. Mô hình khái quát thường kết nối một hoặc nhiều CPU, bộ điều khiển bằng đường truyền chung cùng truy nhập đến bộ nhớ được chia sẻ. Sự hoạt động đồng thời của CPU và các thiết bị nhập xuất dẫn đến cạnh tranh sử dụng bộ nhớ. Sơ đồ được mô tả ở hình 1.5.

### 3.1. Các hoạt động của hệ thống

Chúng ta xem xét quá trình khởi động máy tính, trong tất cả các máy tính đều có một chương trình khởi động gọi là **bootstrap program** được nạp tại thời điểm bật máy, chương trình này thường được lưu trữ trong một bộ nhớ gắn liền với cấu trúc máy. Chương trình này sẽ kiểm tra và khởi động tất cả các bộ phận phần cứng của hệ thống. Sau khi hệ thống phần cứng đã được khởi động, bootstrap sẽ nạp hạt nhân của hệ điều hành, trao quyền điều khiển và chấm dứt hoạt động. Chính cách tổ chức này làm cho một máy tính có thể chạy được với nhiều hệ điều hành khác nhau.

Máy tính được thiết kế để các thiết bị I/O và CPU có thể hoạt động đồng thời, cách thức tổ chức này sẽ đảm bảo không lệ thuộc lẫn nhau về mặt thời gian, vì hệ thống I/O là phần cứng có tốc độ hoạt động rất chậm. Bộ xử lý trung tâm và các bộ điều khiển I/O quan hệ với nhau qua các ngắt (interrupts). Khi cần truyền dữ liệu, CPU gửi yêu cầu đến bộ điều khiển I/O (I/O request), bộ điều khiển I/O nhận được sẽ có hoạt động đáp ứng, sau khi hoàn thành nó sẽ trả lại một ngắt báo hoàn thành (transfer done), khi đó CPU sẽ dừng quá trình xử lý ứng dụng để đáp ứng ngắt. Sơ đồ quan hệ được mô tả trong hình 1.6.



Hình 1.6. Sơ đồ xử lý ngắt I/O

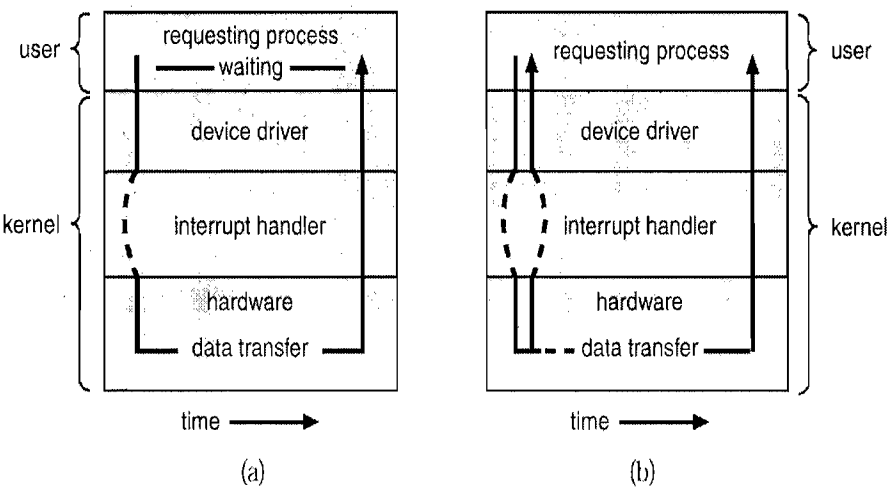
#### Hai phương pháp điều khiển I/O

- Điều khiển đồng bộ: Sau khi I/O bắt đầu, điều khiển được chuyển tới chương trình của người sử dụng chỉ khi I/O hoàn thành. CPU sẽ trong trạng thái chờ cho đến khi có một ngắt tiếp. Phương pháp này được dùng trong trường hợp chương trình ứng dụng thực sự cần dữ liệu để tiếp tục hoạt động, chẳng hạn yêu cầu mở tập tin văn bản trong một hệ xử lý văn bản (word process)
- Điều khiển không đồng bộ: Sau khi I/O bắt đầu, điều khiển được chuyển tới chương trình của người sử dụng không chờ I/O hoàn thành. Phương pháp này được dùng trong các ứng dụng mà dữ liệu nó yêu cầu chỉ là dữ liệu bổ



sung, chẳng hạn khi ta dùng trình duyệt web (browse), dữ liệu sẽ được tải từ mạng xuống dần dần, tải đến đâu, trình duyệt web hiển thị đến đó.

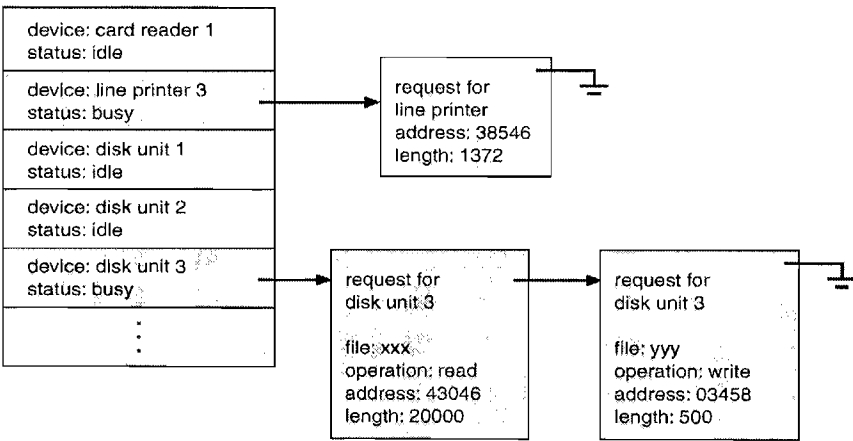
Chúng ta có thuật ngữ *System call* để chỉ một yêu cầu của chương trình ứng dụng chuyển tới OS đòi hỏi đáp ứng I/O. Trong hình 1.7. là sơ đồ mô tả hai cách điều khiển đó.



Hình 1.7. (a) Điều khiển đồng bộ, (b) Điều khiển không đồng bộ

**Bảng trạng thái các thiết bị (Device-Status Table)**

*Device-status table* chứa các entry của mỗi thiết bị I/O ghi kiểu, địa chỉ và trạng thái hoạt động hiện thời của các thiết bị. Bảng trạng thái thiết bị là dữ liệu hệ điều hành tạo ra và quản lý, thường mỗi entry sẽ dẫn đến một cấu trúc dữ liệu danh sách móc nối tuyến tính, mỗi nút của danh sách này ghi rõ các hoạt động sẽ được thực hiện trên thiết bị đó. Sơ đồ mô phỏng cấu trúc bảng trạng thái thiết bị xem trong hình 1.8.



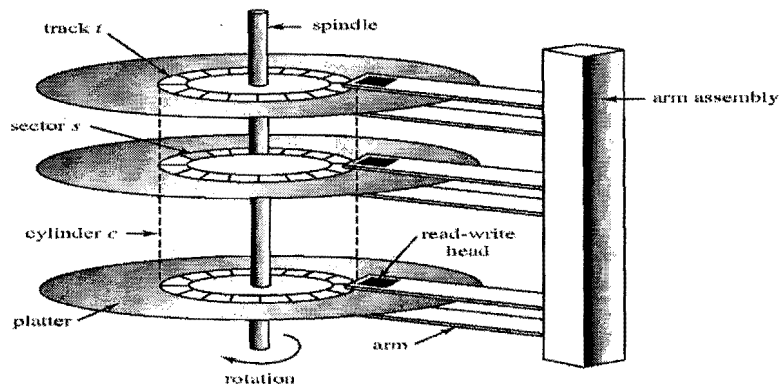
Hình 1.8. Sơ đồ mô tả cấu trúc của bảng các trạng thái thiết bị

**3.2. Cấu trúc lưu trữ**

Trong cấu trúc của máy tính có rất nhiều loại bộ nhớ, được sử dụng với nhiệm vụ khác nhau. Bộ nhớ chính – Bộ nhớ chủ – Bộ nhớ trong (Main memory) là thiết bị lưu trữ lớn mà CPU có thể truy xuất trực tiếp, nhiệm vụ của bộ lưu trữ này là chứa tất cả các dữ liệu liên quan đến hoạt động hiện thời của máy tính, bao gồm các mã chương trình, dữ liệu xử lý, dữ liệu quản lý của hệ điều hành, ... Bộ nhớ chính chỉ lưu trữ dữ liệu tạm thời, khi máy tính dừng hoạt động thì dữ liệu sẽ bị xóa. Ngoài ra bên trong của máy tính còn có nhiều loại bộ nhớ khác nữa đó là thanh ghi, cache, ROM BIOS.

Bộ nhớ thứ cấp (Secondary storage) là tên gọi của tất cả các thiết bị lưu trữ mở rộng cung cấp khả năng lưu trữ lớn và lâu dài, chúng ta còn gọi là bộ nhớ ngoài. Bộ nhớ ngoài là thiết bị lưu trữ mà CPU không truy xuất trực tiếp được, truy xuất dữ liệu trên thiết bị lưu trữ ngoài phải thông qua bộ điều khiển thiết bị. Các thiết bị lưu trữ ngoài có đặc điểm là tốc độ truy xuất chậm nhưng lưu trữ với dung lượng lớn và độ ổn định thông tin rất cao. Các thiết bị lưu trữ ngoài phổ biến hiện nay là ổ đĩa cứng (hard disk), ổ đĩa quang (CD ROM, DVD ROM, CD W/R, DVD W/R), ổ đĩa thẻ cứng (flash disk, USB flash, electronics disk), băng từ (tape). Các thiết bị lưu trữ ngoài có nguyên lý lưu trữ rất khác nhau. Hình 1.9 mô tả sơ đồ cấu trúc của ổ đĩa cứng:

- Đĩa từ (magnetic disk) – kim loại cứng hoặc phủ các vật liệu ghi từ tính.
- Bề mặt đĩa được chia ra về phương diện logic thành các track và mỗi track được chia nhỏ hơn thành các sector
- *Disk controller* điều khiển sự tương tác logic giữa thiết bị và máy tính.

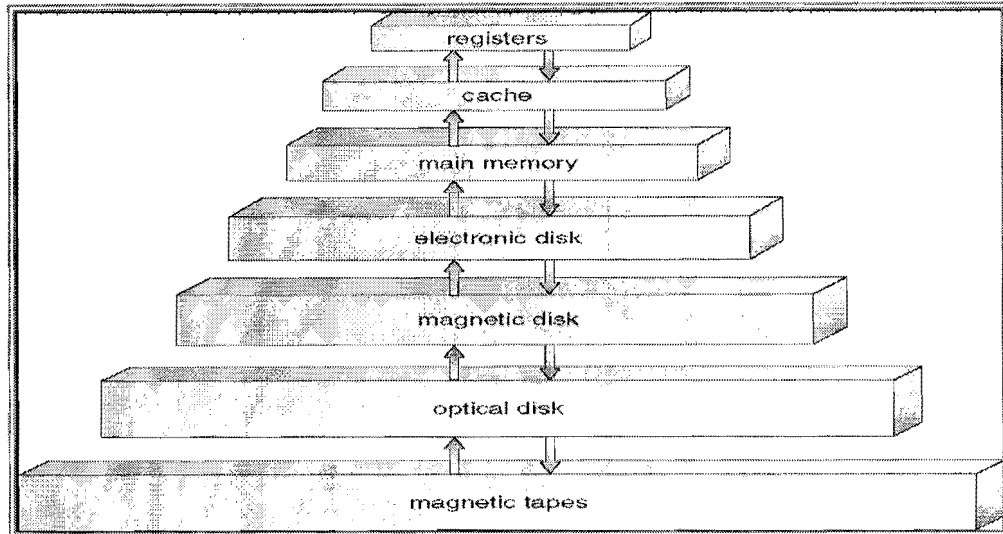


**Hình 1.9.** Sơ đồ tổng quát cấu trúc của ổ đĩa cứng

### Phân bậc bộ lưu trữ (Storage Hierarchy)

Hệ thống lưu trữ được đánh giá theo tiêu chí.

- Tốc độ (Speed) : Đối với bộ nhớ ngoài tốc độ được tính bằng số bit mà máy tính truy xuất được trong một giây, đơn vị là bps (bit per second). Đối với bộ nhớ trong có thể dùng tần số làm việc của bộ quản lý bộ nhớ.



**Hình 1.10. Lược đồ phân bậc bộ lưu trữ**

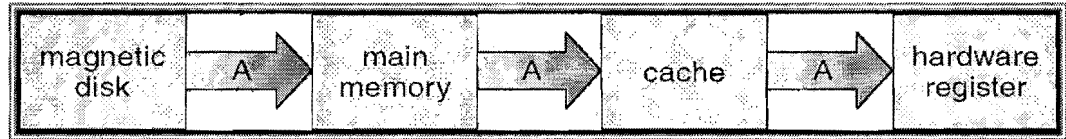
- **Giá thành (Cost)** : Giá thành trên một đơn vị lưu trữ, đây là một chỉ số tương đối. Trước kia người ta tính giá thành bằng Mb, hiện tại đơn vị so sánh là Gb thậm chí Tb.
- **Độ ổn định thông tin (Volatility)** : Bộ nhớ trong có độ ổn định thông tin thấp nhất, nó chỉ lưu trữ thông tin khi máy tính đang hoạt động. Đĩa quang học có độ ổn định thông tin cao nhất.

Dựa trên ba tiêu chí đã nêu, chúng ta có thể đánh giá phân bậc theo lược đồ như ở hình 1.10. Các bộ nhớ ở bậc trên có tốc độ cao hơn bộ nhớ bậc dưới, nhưng giá thành đắt hơn và như vậy thường được sử dụng với dung lượng lưu trữ nhỏ. Độ ổn định thông tin của các bộ nhớ ngoài cũng cao hơn bộ nhớ trong.

### Caching

Caching là nguyên lý rất quan trọng được dùng với nhiều mức độ trong máy tính (hardware, operating system, software), nói một cách ngắn gọn đó là *dữ liệu cần sử dụng được sao chép từ bộ lưu trữ tốc độ chậm vào bộ lưu trữ tốc độ nhanh, quá trình caching dừng lại ở bộ nhớ cache*.

Chúng ta biết rằng dữ liệu được xử lý tại bộ xử lý trung tâm (CPU), nơi lưu trữ chủ yếu của chúng tại bộ nhớ trong. Bộ xử lý hoạt động với tốc độ rất cao nhưng bộ nhớ trong hoạt động với tốc độ thấp hơn và tốc độ đường truyền dữ liệu lại còn thấp hơn nữa. Mỗi khi thực thi hệ thống tiêu tốn rất nhiều thời gian vào việc tìm kiếm và vận chuyển dữ liệu. Để tăng tốc độ xử lý của hệ thống người ta đã tạo ra bộ nhớ cache có tốc độ cao hơn bộ nhớ trong, nằm gần CPU hơn và dung lượng nhỏ hơn để giảm thời gian tìm kiếm. Hệ điều hành phải dự đoán dữ liệu sắp được sử dụng và sao chép nó vào bộ nhớ cache. Khi cần dùng dữ liệu nào đó, đầu tiên bộ nhớ cache được kiểm tra. Nếu có, dữ liệu được lấy trực tiếp từ cache (ta gọi là cache hit). Nếu không có, dữ liệu được sao chép vào cache và được sử dụng (ta gọi là cache miss). Cache là bộ lưu trữ nhỏ nhất được caching. Quản trị cache là bài toán thiết kế quan trọng và rất khó khăn, hiệu suất của caching phụ thuộc nhiều vào khả năng dự báo caching của hệ điều hành. Hình 1.11 mô tả quá trình caching của dữ liệu A từ ổ đĩa và kết thúc tại bộ nhớ cache.



Hình 1.11. Sự di chuyển của giá trị A từ đĩa tới thanh ghi

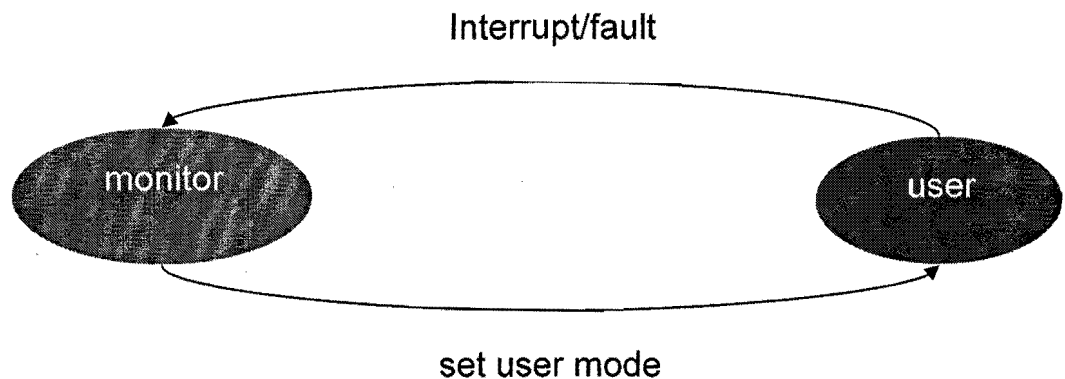
### 3.3. Bảo vệ ở mức phần cứng (Hardware Protection)

#### 3.3.1. Dual-Mode Operation

Hệ thống chia sẻ tài nguyên phải đảm bảo ranh giới hoạt động của các chỉ thị của hệ điều hành và chỉ thị của các chương trình ứng dụng. Điều đó dẫn đến cấu trúc phần cứng của hệ thống phải hỗ trợ hai kiểu hoạt động:

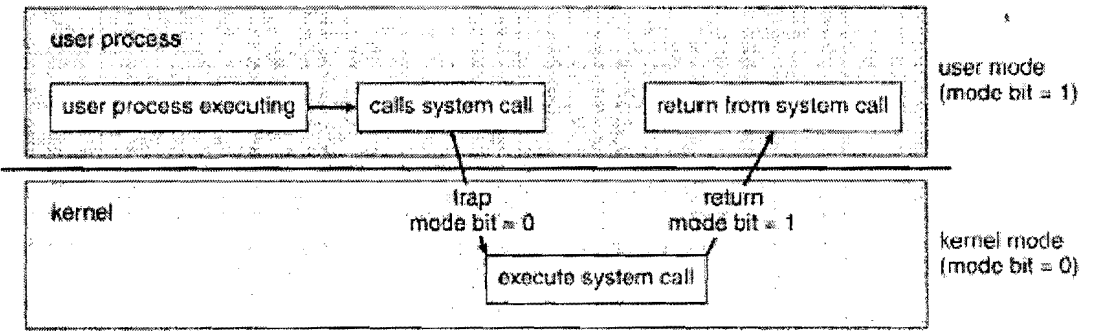
1. *User mode* là kiểu hoạt động của các chương trình lớp ứng dụng.
2. *Monitor mode* (*kernel mode* hoặc *system mode*) là kiểu hoạt động của các chương trình hệ điều hành.

Để đảm bảo hệ thống không bị lỗi do các chương trình ứng dụng gây ra hay nói cách khác đảm bảo hệ thống luôn chịu sự kiểm soát của hệ điều hành các hoạt động của hệ thống được thiết kế sao cho kiểu hoạt động user mode phải được kiểm soát bởi kernel mode cho phép và sẽ chuyển ngay lập tức về kernel mode nếu như hệ thống có một ngắt hoặc chương trình ứng dụng có lỗi. Mối quan hệ giữa hai kiểu hoạt động được mô tả như hình 1.12.



Hình 1.12. Quan hệ giữa user mode và kernel mode

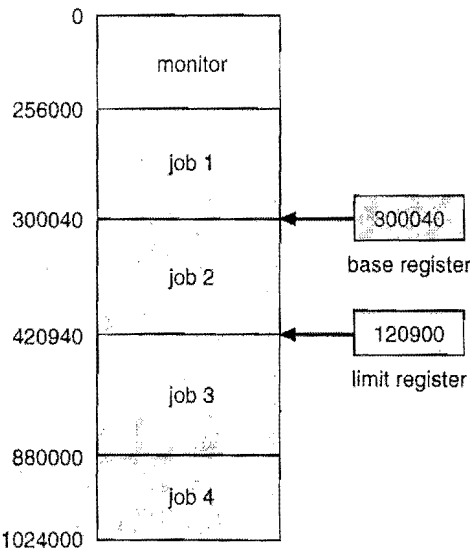
Để điều khiển quan hệ, một bit mode xác định kiểu được bổ xung vào phần cứng để chỉ định kiểu hiện thời: monitor (0) hoặc user (1). Khi một ngắt hoặc lỗi xuất hiện trong kiểu user mode, phần cứng sẽ chuyển hệ thống tới monitor mode (xem hình 1.13). Khi hệ thống hoàn thành việc khởi động (boot time), lúc đó phần cứng máy tính hoạt động trong chế độ kernel mode (bit sẽ được đặt về giá trị 0). Khi một chương trình ứng dụng được khởi động sẽ đưa hệ thống vào chế độ user mode (bit sẽ được chuyển về giá trị 1), hệ thống luôn đặt bit mode về 1 trước khi chuyển điều khiển cho chương trình ứng dụng.



Hình 1.13. Sơ đồ dịch chuyển user mode và kernel mode

3.3.2. Memory Protection

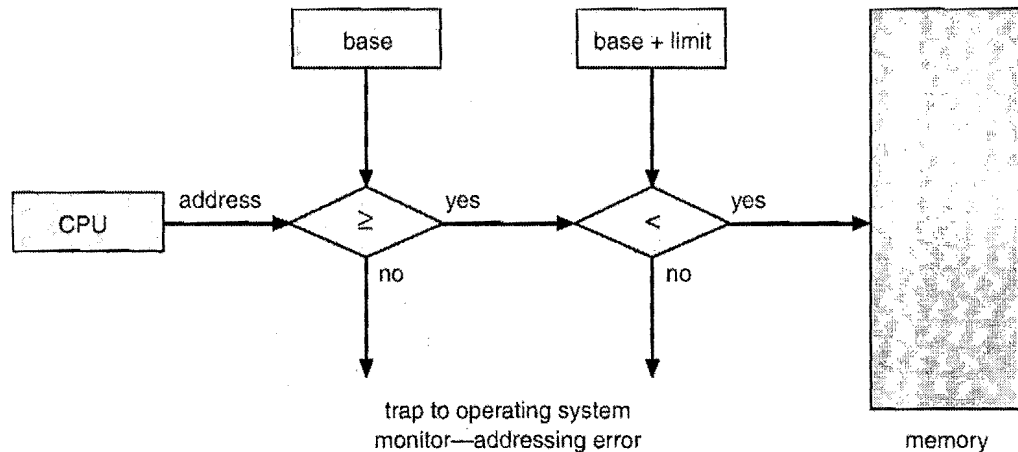
Hệ thống hoạt động một cách chính xác một phần phụ thuộc vào các dữ liệu điều khiển được lưu trữ trong các vùng nhớ xác định, trong đó quan trọng nhất là các vector ngắt (nơi chứa địa chỉ của các chương trình con phục ngắt), như vậy chúng ta phải bảo vệ dữ liệu này chống lại sự thay đổi các dữ liệu này do các tác động bên ngoài, mà chủ yếu là các tác động từ chương trình ứng dụng. Một kỹ thuật thường được sử dụng là điều khiển phạm vi truy xuất của các chỉ thị của chương trình ứng dụng trong vùng nhớ được quy định cho ứng dụng đó, không cho phép nó truy xuất ra ngoài vùng được quy định. Để làm điều đó, phần cứng máy tính thường được bổ sung một mô đun kiểm tra địa chỉ truy xuất, khi phát hiện địa chỉ truy xuất không hợp lệ nó sẽ phát ra một ngắt lỗi và hệ thống lập tức chuyển về chế độ kernel mode.



Hình 1.14. Sơ đồ bộ nhớ liên tục

Ví dụ, trong mô hình quản lý bộ nhớ theo kiểu liên tục (sơ đồ như trong hình 1.14). Mỗi ứng dụng được sử dụng một bộ nhớ liên khối, được xác định bởi địa chỉ bắt đầu và độ dài vùng nhớ. Hệ thống sẽ sử dụng hai thanh ghi chứa hai giá trị này (base register và limit register) để giới hạn truy xuất. Sơ đồ hoạt động được mô tả trong hình 1.15. Chúng ta lưu ý rằng hoạt động giới hạn địa chỉ chỉ áp dụng cho các chỉ thị

của chương trình ứng dụng, các chỉ thị từ hệ điều hành không bị áp dụng giới hạn này (ta gọi các chỉ thị đó là chỉ thị đặc quyền - *privileged instruction*).



**Hình 1.15.** Lược đồ kiểm tra giới hạn truy xuất

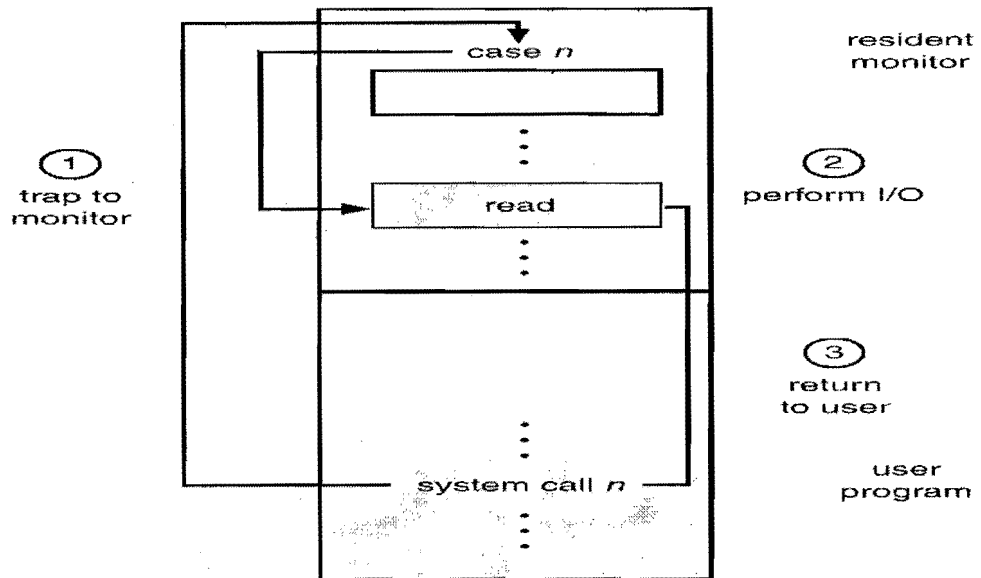
### 3.3.3. CPU Protection

Có một vấn đề cần xem xét, đó là trong trường hợp chương trình ứng dụng hoạt động rơi vào vòng lặp không kết thúc hoặc gọi tới một dịch vụ không bao giờ trả lại quyền điều khiển cho hệ điều hành và như vậy hệ thống của chúng ta rơi vào trạng thái mất điều khiển. Để duy trì quyền điều khiển của hệ điều hành với hệ thống, cấu trúc phần cứng của máy tính cần có một *Timer* - đồng hồ ngắt giờ. Đồng hồ ngắt bắt đầu với một con số và sẽ giảm đều đặn theo một chu kì nhất định. Khi đồng hồ ngắt giờ giảm đến giá trị 0 thì một ngắt xảy ra (*load timer*), ngắt này trả lại quyền điều khiển cho hệ điều hành. *Timer* thông thường được sử dụng cho những hệ điều hành chia sẻ thời gian và cũng còn được sử dụng cho các máy tính thời gian thực. *Load-timer* là một *privileged instruction*.

### 3.3.4. I/O Protection

Một chương trình ứng dụng có thể phá vỡ hoạt động bình thường của hệ thống với một chỉ thị I/O, bởi vì nó có thể truy xuất vị trí vùng nhớ như chính bản thân hệ điều hành. Phải chắc chắn một chương trình của user không bao giờ giành được điều khiển của máy tính trong kernel mode (chẳng hạn nó không bao giờ có thể thay đổi địa chỉ trong vector ngắt). Tất cả các chỉ thị I/O là chỉ thị đặc quyền và chỉ được phát ra trong chế độ kernel mode. Các chương trình ứng dụng muốn sử dụng dịch vụ I/O nó có thể gửi một yêu cầu hệ thống (*system call*) tới hệ điều hành và hệ điều hành sẽ đáp ứng I/O nếu có thể.

Lược đồ đáp ứng I/O của một chương trình ứng dụng được mô tả trong hình 1.16. Chương trình ứng dụng hoạt động trong chế độ user mode khi cần dịch vụ I/O nó sẽ gửi một *system call* tới hệ điều hành, lúc này hệ thống sẽ chuyển về kernel mode. Hệ điều hành xem xét yêu cầu và đáp ứng I/O, một chỉ thị I/O sẽ được phát ra trong chế độ kernel mode, sau khi hoàn thành yêu cầu I/O hệ điều hành sẽ trả hệ thống trở lại chế độ user mode, quyền điều khiển được giao lại cho chương trình ứng dụng.



**Hình 1.16.** Lược đồ đáp ứng I/O của chương trình ứng dụng

### 3.3.5. Các lời gọi hệ thống (System Calls)

System calls cung cấp giao tiếp giữa một chương trình đang chạy và hệ điều hành. Lời gọi hệ thống có khả năng như là một chỉ thị của hợp ngữ (assembly language) và thường được chứa trong một danh sách các dịch vụ của hợp ngữ. Một số hệ thống cho phép sử dụng system call từ trong ngôn ngữ bậc cao, trong trường hợp đó nó được xây dựng thành các hàm và thủ tục trong thư viện của ngôn ngữ. Một vài ngôn ngữ như C, C++ hoặc Perl đã xây dựng tập các lời gọi hệ thống thay cho hợp ngữ khi lập trình hệ thống. Chẳng hạn, system call của UNIX có thể dùng trực tiếp từ ngôn ngữ C hoặc C++. System call của hệ điều hành Microsoft Windows nằm trong tập giao tiếp Win32 API (Win32 application programmer interface).

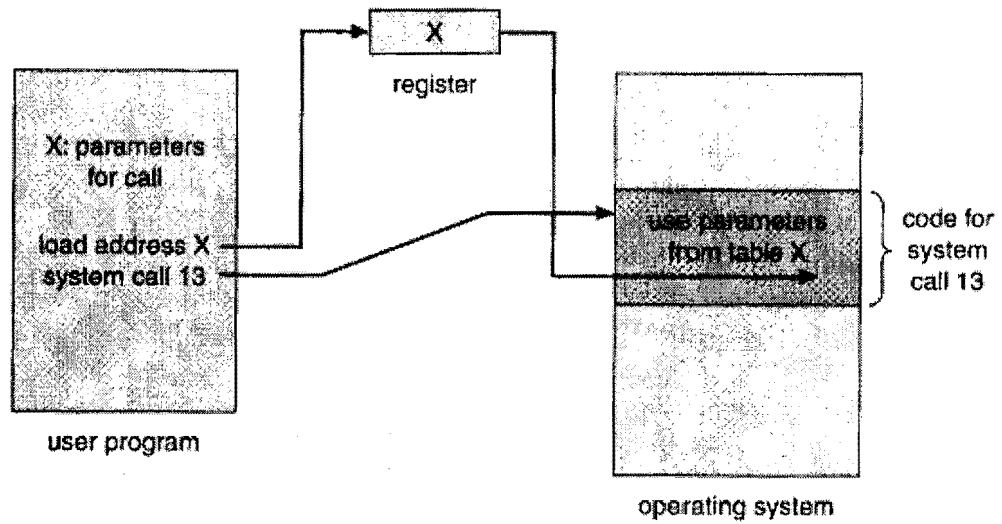
Có ba phương pháp để truyền thông số giữa chương trình ứng dụng đang chạy và hệ điều hành:

- Truyền thông số qua các thanh ghi.
- Lưu trữ trong bảng trong memory và địa chỉ bảng được chuyển qua thanh ghi (xem hình 1.17).
- *Push* các thông số vào một *stack* bởi chương trình và *Pop* bởi hệ điều hành.

Chúng ta phân loại các System Calls thành các nhóm theo chức năng:

- Điều khiển tiến trình (Process control)
- Quản trị tệp (File management)
- Quản trị thiết bị (Device management)
- Duy trì thông tin (Information maintenance)

- Truyền tin (Communication)



**Hình 1.17.** Sơ đồ phương pháp chuyển địa chỉ bằng qua thanh ghi