



Giáo trình hệ điều hành

Biên tập bởi:

Giảng viên . Lê Khắc Nhiên Ân

Giáo trình hệ điều hành

Biên tập bởi:

Giảng viên . Lê Khắc Nhiên Ân

Các tác giả:

Giảng viên . Trần Hạnh Nhi

Phiên bản trực tuyến:

<http://voer.edu.vn/c/a039fa79>

MỤC LỤC

1. Tổng quan về hệ điều hành
2. Phân loại hệ điều hành
3. Cấu trúc hệ điều hành
4. Lịch sử phát triển hệ điều hành
5. Bài tập tự giải
6. Các mô hình xử lý đồng hành
7. Khái niệm tiến trình(Process) và mô hình đa tiến trình(multiprocess)
8. Khái niệm tiểu trình(thread) và mô hình đa tiểu trình(multithread)
9. Tóm tắt và bài tập
10. Quản lý tiến trình
11. Điều phối tiến trình
12. Quản lý tiến trình-Tóm tắt
13. Liên lạc giữa các tiến trình và vấn đề đồng bộ hóa
14. Cơ chế thông tin liên lạc
15. Nhu cầu đồng bộ hóa(synchronisation)
16. Các giải pháp đồng bộ hóa
17. Các giải pháp "Sleep and wakeup"
18. Các vấn đề cổ điển của đồng bộ hoá
19. Tắc nghẽn (Deadlock)
20. Quản lý bộ nhớ
21. Không gian địa chỉ và không gian vật lý
22. Cấp phát liên tục
23. Cấp phát không liên tục
24. Quản lý bộ nhớ-Tóm tắt
25. Bộ nhớ ảo
26. Thay thế trang
27. Cấp phát khung trang
28. Bộ nhớ ảo-Tóm tắt
29. Hệ thống quản lý tập tin
30. Mô hình tổ chức và quản lý các tập tin
31. Các phương pháp cài đặt hệ thống quản lý tập tin
32. Bảng phân phối vùng nhớ
33. Tập tin chia sẻ

34. Quản lý đĩa
 35. Độ an toàn của hệ thống tập tin
 36. Giới thiệu một số hệ thống tập tin
 37. Windows95
 38. Windows NT
 39. Unix
 40. Hệ thống quản lý nhập-xuất
 41. Phần cứng nhập-xuất
 42. Phần mềm nhập xuất
 43. Giới thiệu một số hệ thống I-O
 44. Hệ thống I-O chuẩn (terminals)
 45. Cài đặt đồng hồ
 46. Bảo vệ an toàn hệ thống
 47. Miền bảo vệ (Domain of Protection)
 48. Ma trận quyền truy xuất (Access matrix)
 49. Cài đặt ma trận quyền truy xuất
 50. An toàn hệ thống (Security)
 51. Hệ điều hành windows NT
 52. Hệ điều hành Linux -Giới thiệu
 53. Tổ chức hệ thống
- Tham gia đóng góp

Tổng quan về hệ điều hành

Bài học này cung cấp cho chúng ta một cái nhìn tổng quát về những nguyên lý cơ bản của hệ điều hành. Chúng ta bắt đầu với việc xem xét mục tiêu và các chức năng của hệ điều hành, sau đó khảo sát các dạng khác nhau của chúng cũng như xem xét quá trình phát triển qua từng giai đoạn. Các phần này được trình bày thông qua các nội dung như sau:

[Khái niệm về hệ điều hành](#)

[Phân loại hệ điều hành](#)

[Cấu trúc của hệ điều hành](#)

[Lịch sử phát triển của hệ điều hành](#)

Bài học này giúp chúng ta hiểu được hệ điều hành là gì, có cấu trúc ra sao. Hệ điều hành được phân loại theo những tiêu chuẩn nào. Quá trình phát triển của hệ điều hành phụ thuộc vào những yếu tố nào.

Bài học này đòi hỏi những kiến thức về : kiến trúc máy tính.

KHÁI NIỆM VỀ HỆ ĐIỀU HÀNH

Hệ điều hành là một *chương trình* hay một *hệ chương trình* hoạt động giữa người sử dụng (user) và phần cứng của máy tính. Mục tiêu của hệ điều hành là cung cấp một môi trường để người sử dụng có thể thi hành các chương trình. Nó làm cho máy tính dễ sử dụng hơn, thuận lợi hơn và hiệu quả hơn.

Hệ điều hành là một phần quan trọng của hầu hết các hệ thống máy tính. Một hệ thống máy tính thường được chia làm bốn phần chính : phần cứng, hệ điều hành, các chương trình ứng dụng và người sử dụng.

Phần cứng bao gồm CPU, bộ nhớ, các thiết bị nhập xuất, đây là những tài nguyên của máy tính. *Chương trình ứng dụng* như các chương trình dịch, hệ thống cơ sở dữ liệu, các trò chơi, và các chương trình thương mại. Các chương trình này sử dụng tài nguyên của máy tính để giải quyết các yêu cầu của người sử dụng. *Hệ điều hành* điều khiển và phối hợp việc sử dụng phần cứng cho những ứng dụng khác nhau của nhiều người sử dụng khác nhau. Hệ điều hành cung cấp một môi trường mà các chương trình có thể làm việc hữu hiệu trên đó.

Hệ điều hành có thể được coi như là bộ phân phối tài nguyên của máy tính. Nhiều tài nguyên của máy tính như thời gian sử dụng CPU, vùng bộ nhớ, vùng lưu trữ tập tin, thiết bị nhập xuất v.v... được các chương trình yêu cầu để giải quyết vấn đề. Hệ điều hành hoạt động như một bộ quản lý các tài nguyên và phân phối chúng cho các chương trình và người sử dụng khi cần thiết. Do có rất nhiều yêu cầu, hệ điều hành phải giải quyết vấn đề tranh chấp và phải quyết định *cấp phát tài nguyên* cho những yêu cầu theo thứ tự nào để hoạt động của máy tính là hiệu quả nhất. Một hệ điều hành cũng có thể được coi như là một chương trình kiểm soát việc sử dụng máy tính, đặc biệt là các thiết bị nhập xuất.

Tuy nhiên, nhìn chung chưa có định nghĩa nào là hoàn hảo về hệ điều hành. Hệ điều hành tồn tại để giải quyết các vấn đề sử dụng hệ thống máy tính. Mục tiêu cơ bản của nó là giúp cho việc thi hành các chương trình dễ dàng hơn. Mục tiêu thứ hai là hỗ trợ cho các thao tác trên hệ thống máy tính hiệu quả hơn. Mục tiêu này đặc biệt quan trọng trong những hệ thống nhiều người dùng và trong những hệ thống lớn (phần cứng + quy mô sử dụng). Tuy nhiên hai mục tiêu này cũng có phần tương phản vì vậy lý thuyết về hệ điều hành tập trung vào việc tối ưu hóa việc sử dụng tài nguyên của máy tính.

Phân loại hệ điều hành

Hệ thống xử lý theo lô

Bộ giám sát thường trực :

Khi một công việc chấm dứt, hệ thống sẽ thực hiện công việc kế tiếp mà không cần sự can thiệp của người lập trình, do đó thời gian thực hiện sẽ mau hơn. Một chương trình, còn gọi là bộ giám sát thường trực được thiết kế để giám sát việc thực hiện dãy các công việc một cách tự động, chương trình này luôn luôn thường trú trong bộ nhớ chính.

Hệ điều hành theo lô thực hiện các công việc lần lượt theo những chỉ thị định trước.

CPU và thao tác nhập xuất :

CPU thường hay nhàn rỗi do tốc độ làm việc của các thiết bị nhập xuất (thường là thiết bị cơ) chậm hơn rất nhiều lần so với các thiết bị điện tử. Cho dù là một CPU chậm nhất, nó cũng nhanh hơn rất nhiều lần so với thiết bị nhập xuất. Do đó phải có các phương pháp để đồng bộ hóa việc hoạt động của CPU và thao tác nhập xuất.

Xử lý off_line :

Xử lý off_line là thay vì CPU phải đọc trực tiếp từ thiết bị nhập và xuất ra thiết bị xuất, hệ thống dùng một *bộ lưu trữ trung gian*. CPU chỉ thao tác với bộ phận này. Việc đọc hay xuất đều đến và từ bộ lưu trữ trung gian.

Spooling :

Spool (simultaneous peripheral operation on-line) là đồng bộ hóa các thao tác bên ngoài on-line. Cơ chế này cho phép xử lý của CPU là on-line, sử dụng đĩa để lưu các dữ liệu nhập cũng như xuất.

Hệ thống xử lý theo lô đa chương

Khi có nhiều công việc cùng truy xuất lên thiết bị, vấn đề lập lịch cho các công việc là cần thiết. Khía cạnh quan trọng nhất trong việc lập lịch là khả năng đa chương. *Đa chương (multiprogram)* gia tăng khai thác CPU bằng cách tổ chức các công việc sao cho CPU luôn luôn phải trong tình trạng làm việc .

Ý tưởng như sau : hệ điều hành lưu giữ một phần của các công việc ở nơi lưu trữ trong bộ nhớ . CPU sẽ lần lượt thực hiện các phần công việc này. Khi đang thực hiện, nếu có yêu cầu truy xuất thiết bị thì CPU không nghỉ mà thực hiện tiếp công việc thứ hai...

Với hệ đa chương hệ điều hành ra quyết định cho người sử dụng vì vậy, *hệ điều hành đa chương* rất tinh vi. Hệ phải xử lý các vấn đề lập lịch cho công việc, lập lịch cho bộ nhớ và cho cả CPU nữa.

Hệ thống chia sẻ thời gian

Hệ thống chia sẻ thời gian là một mở rộng logic của hệ đa chương. Hệ thống này còn được gọi là *hệ thống đa nhiệm* (multitasking). Nhiều công việc cùng được thực hiện thông qua cơ chế chuyển đổi của CPU như hệ đa chương nhưng thời gian mỗi lần chuyển đổi diễn ra rất nhanh.

Hệ thống chia sẻ được phát triển để cung cấp việc sử dụng bên trong của một máy tính có giá trị hơn. *Hệ điều hành chia sẻ* thời gian dùng lập lịch CPU và đa chương để cung cấp cho mỗi người sử dụng một phần nhỏ trong máy tính chia sẻ. Một chương trình khi thi hành được gọi là một tiến trình. Trong quá trình thi hành của một tiến trình, nó phải thực hiện các thao tác nhập xuất và trong khoảng thời gian đó CPU sẽ thi hành một tiến trình khác. Hệ điều hành chia sẻ cho phép nhiều người sử dụng chia sẻ máy tính một cách đồng bộ do thời gian chuyển đổi nhanh nên họ có cảm giác là các tiến trình đang được thi hành cùng lúc.

Hệ điều hành chia sẻ phức tạp hơn hệ điều hành đa chương. Nó phải có các chức năng : quản trị và bảo vệ bộ nhớ, sử dụng bộ nhớ ảo. Nó cũng cung cấp hệ thống tập tin truy xuất on-line...

Hệ điều hành chia sẻ là kiểu của các hệ điều hành hiện đại ngày nay.

Hệ thống song song

Ngoài các hệ thống chỉ có một bộ xử lý còn có các hệ thống có nhiều bộ xử lý cùng chia sẻ hệ thống đường truyền dữ liệu, đồng hồ, bộ nhớ và các thiết bị ngoại vi. Các bộ xử lý này liên lạc bên trong với nhau .

Có nhiều nguyên nhân xây dựng dạng hệ thống này. Với sự gia tăng số lượng bộ xử lý, công việc được thực hiện nhanh chóng hơn, Nhưng không phải theo đúng tỉ lệ thời gian, nghĩa là có n bộ xử lý không có nghĩa là sẽ thực hiện nhanh hơn n lần.

Hệ thống với máy nhiều bộ xử lý sẽ tối ưu hơn hệ thống có nhiều máy có một bộ xử lý vì các bộ xử lý chia sẻ các thiết bị ngoại vi, hệ thống lưu trữ, nguồn ... và rất thuận tiện cho nhiều chương trình cùng làm việc trên cùng một tập hợp dữ liệu.

Một lý do nữa là độ tin cậy. Các chức năng được xử lý trên nhiều bộ xử lý và sự hỏng hóc của một bộ xử lý sẽ không ảnh hưởng đến toàn bộ hệ thống.

Hệ thống đa xử lý thông thường sử dụng cách *đa xử lý đối xứng*, trong cách này mỗi bộ xử lý chạy với một bản sao của hệ điều hành, những bản sao này liên lạc với nhau khi cần thiết. Một số hệ thống sử dụng đa xử lý bất đối xứng, trong đó mỗi bộ xử lý được giao một công việc riêng biệt.. Một bộ xử lý chính kiểm soát toàn bộ hệ thống, các bộ xử lý khác thực hiện theo lệnh của bộ xử lý chính hoặc theo những chỉ thị đã được định nghĩa trước. Mô hình này theo dạng quan hệ chủ tớ. Bộ xử lý chính sẽ lập lịch cho các bộ xử lý khác.

Một ví dụ về hệ thống xử lý đối xứng là version Encore của UNIX cho máy tính Multimax. Hệ thống này có hàng tá bộ xử lý. Ưu điểm của nó là nhiều tiến trình có thể thực hiện cùng lúc. Một hệ thống đa xử lý cho phép nhiều công việc và tài nguyên được chia sẻ tự động trong những bộ xử lý khác nhau.

Hệ thống đa xử lý không đồng bộ thường xuất hiện trong những hệ thống lớn, trong đó hầu hết thời gian hoạt động đều dành cho xử lý nhập xuất.

Hệ thống phân tán

Hệ thống này cũng tương tự như hệ thống chia sẻ thời gian nhưng các bộ xử lý không chia sẻ bộ nhớ và đồng hồ, thay vào đó mỗi bộ xử lý có bộ nhớ cục bộ riêng. Các bộ xử lý thông tin với nhau thông qua các đường truyền thông như những bus tốc độ cao hay đường dây điện thoại.

Các bộ xử lý trong hệ phân tán thường khác nhau về kích thước và chức năng. Nó có thể bao gồm máy vi tính, trạm làm việc, máy mini, và những hệ thống máy lớn. Các bộ xử lý thường được tham khảo với nhiều tên khác nhau như site, node, computer v.v.... tùy thuộc vào trạng thái làm việc của chúng.

Các nguyên nhân phải xây dựng hệ thống phân tán là:

Chia sẻ tài nguyên : Một người sử dụng A có thể sử dụng máy in laser của người sử dụng B và người sử dụng B có thể truy xuất những tập tin của A. Tổng quát, chia sẻ tài nguyên trong hệ thống phân tán cung cấp một cơ chế để chia sẻ tập tin ở vị trí xa, xử lý thông tin trong một cơ sở dữ liệu phân tán, in ấn tại một vị trí xa, sử dụng những thiết bị ở xa để thực hiện các thao tác.

Tăng tốc độ tính toán : Một thao tác tính toán được chia làm nhiều phần nhỏ cùng thực hiện một lúc. Hệ thống phân tán cho phép phân chia việc tính toán trên nhiều vị trí khác nhau để tính toán song song.

An toàn : Nếu một vị trí trong hệ thống phân tán bị hỏng, các vị trí khác vẫn tiếp tục làm việc.

Thông tin liên lạc với nhau : Có nhiều lúc , chương trình cần chuyển đổi dữ liệu từ vị trí này sang vị trí khác. Ví dụ trong hệ thống Windows, thường có sự chia sẻ và chuyển dữ liệu giữa các cửa sổ. Khi các vị trí được nối kết với nhau trong một hệ thống mạng, việc trao đổi dữ liệu diễn ra rất dễ. Người sử dụng có thể chuyển tập tin hay các E_mail cho nhau từ cùng vị trí hay những vị trí khác.

Hệ thống xử lý thời gian thực

Hệ thống xử lý thời gian thực được sử dụng khi có những đòi hỏi khắt khe về thời gian trên các thao tác của bộ xử lý hoặc dòng dữ liệu, nó thường được dùng điều khiển các thiết bị trong các ứng dụng tận hiến (dedicated). Máy tính phân tích dữ liệu và có thể chỉnh các điều khiển giải quyết cho dữ liệu nhập.

Một hệ điều hành xử lý thời gian thực phải được định nghĩa tốt, thời gian xử lý nhanh. Hệ thống phải cho kết quả chính xác trong khoảng thời gian bị thúc ép nhanh nhất. Có hai hệ thống xử lý thời gian thực là hệ thống thời gian thực cứng và hệ thống thời gian thực mềm..

Hệ thống thời gian thực cứng là công việc được hoàn tất đúng lúc. Lúc đó dữ liệu thường được lưu trong bộ nhớ ngắn hạn hay trong ROM. Việc xử lý theo thời gian thực sẽ xung đột với tất cả hệ thống liệt kê ở trên.

Dạng thứ hai là hệ thống thời gian thực mềm, mỗi công việc có một độ ưu tiên riêng và sẽ được thi hành theo độ ưu tiên đó. Có một số lĩnh vực áp dụng hữu hiệu phương pháp này là multimedia hay thực tại ảo.

Cấu trúc hệ điều hành

Các thành phần của hệ thống

Quản lý tiến trình

Một chương trình không thực hiện được gì cả nếu như nó không được CPU thi hành. Một *tiến trình* là một chương trình đang được thi hành, nhưng ý nghĩa của nó còn rộng hơn. Một công việc theo lô là một tiến trình. Một chương trình người dùng chia xẻ thời gian là một tiến trình, một công việc của hệ thống như soopling xuất ra máy in cũng là một tiến trình.

Một tiến trình phải sử dụng tài nguyên như thời gian sử dụng CPU, bộ nhớ, tập tin, các thiết bị nhập xuất để hoàn tất công việc của nó. Các tài nguyên này được cung cấp khi tiến trình được tạo hay trong quá trình thi hành. Khi tiến trình được tạo, nó sử dụng rất nhiều tài nguyên vật lý và luận lý. cũng như một số khởi tạo dữ liệu nhập. Ví dụ , khảo sát tiến trình hiển thị trạng thái của tập tin lên màn hình. Đầu vào của tiến trình là tên tập tin, và tiến trình sẽ thực hiện những chỉ thị thích hợp, thực hiện lời gọi hệ thống để nhận được những thông tin mong muốn và hiển thị nó lên màn hình. Khi tiến trình kết thúc, hệ điều hành sẽ tái tạo lại các tài nguyên có thể được dùng lại..

Một tiến trình là hoạt động (active) hoàn toàn-ngược lại với một tập tin trên đĩa là thụ động (passive)-với một bộ đếm chương trình cho biết lệnh kế tiếp được thi hành. Việc thi hành được thực hiện theo cơ chế tuần tự , CPU sẽ thi hành từ lệnh đầu đến lệnh cuối.

Một tiến trình được coi là một đơn vị làm việc của hệ thống. Một hệ thống có thể có nhiều tiến trình cùng lúc , trong đó một số tiến trình là của hệ điều hành, một số tiến trình là của người sử dụng. các tiến trình này có thể diễn ra đồng thời.

Vai trò của hệ điều hành trong việc quản lý tiến trình là :

- Tạo và hủy các tiến trình của người sử dụng và của hệ thống.
- Ngưng và thực hiện lại một tiến trình.
- Cung cấp cơ chế đồng bộ tiến trình.
- Cung cấp cách thông tin giữa các tiến trình.
- Cung cấp cơ chế kiểm soát deadlock(khái niệm này sẽ được trình bày trong chương II).

Quản lý bộ nhớ chính :

Trong hệ thống máy tính hiện đại, *bộ nhớ chính* là trung tâm của các thao tác, xử lý. Bộ nhớ chính có thể xem như một mảng kiểu byte hay kiểu word. Mỗi phần tử đều có địa chỉ. Đó là nơi lưu dữ liệu được CPU truy xuất một cách nhanh chóng so với các thiết bị nhập/xuất. CPU đọc những chỉ thị từ bộ nhớ chính. Các thiết bị nhập/xuất cài đặt cơ chế DMA(xem chương IV) cũng đọc và ghi dữ liệu trong bộ nhớ chính. Thông thường bộ nhớ chính chứa các thiết bị mà CPU có thể định vị trực tiếp. Ví dụ CPU truy xuất dữ liệu từ đĩa, những dữ liệu này được chuyển vào bộ nhớ qua lời gọi hệ thống nhập/xuất.

Một chương trình muốn thi hành trước hết phải được ánh xạ thành địa chỉ tuyệt đối và nạp vào bộ nhớ chính. Khi chương trình thi hành, hệ thống truy xuất các chỉ thị và dữ liệu của chương trình trong bộ nhớ chính. Ngay cả khi tiến trình kết thúc, dữ liệu vẫn còn trong bộ nhớ cho đến khi một tiến trình khác được ghi chồng lên.

Để tối ưu hóa quá trình hoạt động của CPU và tốc độ của máy tính, một số tiến trình được lưu giữ trong bộ nhớ. Có rất nhiều kế hoạch quản trị bộ nhớ do có nhiều ứng dụng bộ nhớ khác nhau và hiệu quả của các thuật toán phụ thuộc vào tùy tình huống cụ thể. Lựa chọn một thuật toán cho một hệ thống được mô tả trước phụ thuộc vào nhiều yếu tố, đặc biệt là phần cứng của hệ thống.

Hệ điều hành có những vai trò như sau trong việc quản lý bộ nhớ chính :

- Lưu giữ thông tin về các vị trí trong bộ nhớ đã được sử dụng và ai sử dụng.
- Quyết định tiến trình nào được nạp vào bộ nhớ chính, khi bộ nhớ đã có thể dùng được.
- Cấp phát và thu hồi bộ nhớ khi cần thiết.

Quản lý bộ nhớ phụ :

Mục tiêu chính của hệ thống máy tính là thi hành chương trình. Những chương trình với dữ liệu truy xuất của chúng phải được đặt trong bộ nhớ chính trong suốt quá trình thi hành. Nhưng bộ nhớ chính quá nhỏ để có thể lưu giữ mọi dữ liệu và chương trình, ngoài ra dữ liệu sẽ mất khi không còn được cung cấp năng lượng. Hệ thống máy tính ngày nay cung cấp *hệ thống lưu trữ phụ*. Đa số các máy tính đều dùng đĩa để lưu trữ cả chương trình và dữ liệu. Hầu như tất cả chương trình : chương trình dịch, hợp ngữ, thủ tục, trình soạn thảo, định dạng... đều được lưu trữ trên đĩa cho tới khi nó được thực hiện, nạp vào trong bộ nhớ chính và cũng sử dụng đĩa để chứa dữ liệu và kết quả xử lý. Vì vậy một bộ quản lý hệ thống đĩa rất quan trọng cho hệ thống máy tính.

Vai trò của hệ điều hành trong việc quản lý đĩa :

- Quản lý vùng trống trên đĩa.

-Định vị lưu trữ.

-Lập lịch cho đĩa.

Vì hệ thống đĩa được sử dụng thường xuyên, nên nó phải được dùng hiệu quả. Tốc độ của toàn bộ hệ thống tùy thuộc rất nhiều vào tốc độ truy xuất đĩa.

Quản lý hệ thống nhập xuất :

Một trong những mục tiêu của hệ điều hành là *che giấu* những đặc thù của các thiết bị phần cứng đối với người sử dụng thay vào đó là một lớp thân thiện hơn, người sử dụng dễ thao tác hơn. Một hệ thống nhập/xuất bao gồm :

-Hệ thống buffer caching.

-Giao tiếp điều khiển thiết bị (device drivers) tổng quát.

-Bộ điều khiển cho các thiết bị phần cứng.

Chỉ có device driver mới hiểu đến cấu trúc đặc thù của thiết bị mà nó mô tả.

Quản lý hệ thống tập tin :

Hệ thống quản lý tập tin là thành phần rõ ràng nhất trong hệ điều hành. Máy tính có thể lưu trữ thông tin trong nhiều dạng thiết bị vật lý khác nhau : băng từ, đĩa từ, , đĩa quang, ... Mỗi dạng có những đặc thù riêng về mặt tổ chức vật lý. Mỗi thiết bị có một bộ kiểm soát như bộ điều khiển đĩa (disk driver) và có những tính chất riêng. Những tính chất này là tốc độ, khả năng lưu trữ, tốc độ truyền dữ liệu và cách truy xuất.

Để cho việc sử dụng hệ thống máy tính thuận tiện, hệ điều hành cung cấp một cái nhìn logic đồng nhất về hệ thống lưu trữ thông tin. Hệ điều hành định nghĩa một đơn vị lưu trữ logic là tập tin. Hệ điều hành tạo một ánh xạ từ tập tin đến vùng thông tin trên đĩa và truy xuất những tập tin này thông qua thiết bị lưu trữ.

Một tập tin là một tập hợp những thông tin do người tạo ra nó xác định. Thông thường một tập tin đại diện cho một chương trình và dữ liệu. Dữ liệu của tập tin có thể là số, là ký tự, hay ký số. Tập tin thường có dạng tự do, như tập tin văn bản, nhị phân...(là tập tin chứa dãy các bit). (Xem bài VIII)

Vai trò của hệ điều hành trong việc quản lý tập tin :

-Tạo và xoá một tập tin.

-Tạo và xoá một thư mục.

-Hỗ trợ các thao tác trên tập tin và thư mục.

-Ánh xạ tập tin trên hệ thống lưu trữ phụ.

-Backup tập tin trên các thiết bị lưu trữ.

Hệ thống bảo vệ :

Trong một hệ thống nhiều người sử dụng và cho phép nhiều tiến trình diễn ra đồng thời, các tiến trình phải được bảo vệ đối với những hoạt động khác. Do đó, hệ thống cung cấp cơ chế để đảm bảo rằng tập tin, bộ nhớ, CPU, và những tài nguyên khác chỉ được truy xuất bởi những tiến trình có quyền. Ví dụ, bộ nhớ đảm bảo rằng tiến trình chỉ được thi hành trong phạm vi địa chỉ của nó. Bộ thời gian đảm bảo rằng không có tiến trình nào độc chiếm CPU. Cuối cùng các thiết bị ngoại vi cũng được bảo vệ.

Hệ thống bảo vệ là một cơ chế kiểm soát quá trình truy xuất của chương trình, tiến trình, hoặc người sử dụng với tài nguyên của hệ thống. Cơ chế này cũng cung cấp cách thức để mô tả lại mức độ kiểm soát.

Hệ thống bảo vệ cũng làm tăng độ an toàn khi kiểm tra lỗi trong giao tiếp giữa những hệ thống nhỏ bên trong.

Hệ thống cơ chế dòng lệnh :

Một trong những phần quan trọng của chương trình hệ thống trong một hệ điều hành là cơ chế *dòng lệnh*, đó là giao tiếp giữa người sử dụng và hệ điều hành. Một số hệ điều hành đặt cơ chế dòng lệnh bên trong hạt nhân, số khác như MS-DOS và UNIX thì xem hệ điều hành như là một chương trình đặt biệt, được thi hành khi các công việc bắt đầu hoặc khi người sử dụng login lần đầu tiên.

Các lệnh đưa vào hệ điều hành thông qua *bộ điều khiển lệnh*. Trong các hệ thống chia sẻ thời gian một chương trình có thể đọc và thông dịch các lệnh điều khiển được thực hiện một cách tự động. Chương trình này thường được gọi là bộ thông dịch điều khiển card, cơ chế dòng lệnh hoặc Shell. Chức năng của nó rất đơn giản đó là lấy lệnh kế tiếp và thi hành.

Mỗi hệ điều hành sẽ có những giao tiếp khác nhau, dạng đơn giản theo cơ chế dòng lệnh, dạng thân thiện với người sử dụng như giao diện của Macintosh có các biểu tượng, cửa sổ thao tác dùng chuột.

Các lệnh có quan hệ với việc tạo và quản lý các tiến trình, kiểm soát nhập xuất, quản lý bộ lưu trữ phụ, quản lý bộ nhớ chính, truy xuất hệ thống tập tin và cơ chế bảo vệ.

Các dịch vụ của hệ điều hành

Hệ điều hành cung cấp một môi trường để thi hành các chương trình, bằng cách cung cấp các dịch vụ cho chương trình và cho người sử dụng. Các dịch vụ này trên mỗi hệ thống là khác nhau nhưng cũng có những lớp chung. Các dịch vụ này giúp cho các lập trình viên thuận tiện hơn và việc lập trình dễ dàng hơn.

-Thi hành chương trình: hệ thống phải có khả năng nạp chương trình vào bộ nhớ và thi hành nó. Chương trình phải chấm dứt thi hành theo cách thông thường hay bất thường (có lỗi).

-Thao tác nhập xuất : Một chương trình thi hành có thể yêu cầu nhập xuất. Nhập xuất này có thể là tập tin hay thiết bị. Đối với thiết bị có một hàm đặc biệt được thi hành. Để tăng hiệu quả, người sử dụng không truy xuất trực tiếp các thiết bị nhập xuất mà thông qua cách thức do hệ điều hành cung cấp.

-Thao tác trên hệ thống tập tin .

-Thông tin : có nhiều tình huống một tiến trình cần trao đổi thông tin với một tiến trình khác. Có hai cách thực hiện: Một là thực hiện thay thế tiến trình trên cùng máy tính, hai là thay thế tiến trình trên hệ thống khác trong hệ thống mạng. Thông tin có thể được cài đặt qua chia sẻ bộ nhớ, hoặc bằng kỹ thuật chuyển thông điệp. Việc chuyển thông tin được thực hiện bởi hệ điều hành.

-Phát hiện lỗi: hệ điều hành phải có khả năng báo lỗi. Lỗi xảy ra có thể do CPU, bộ nhớ, trong thiết bị nhập xuất, ... hay trong các chương trình. Đối với mỗi dạng lỗi, hệ điều hành sẽ có cách giải quyết tương ứng.

Lời gọi hệ thống

Lời gọi hệ thống cung cấp một giao tiếp giữa tiến trình và hệ điều hành. Lời gọi này cũng như các lệnh hợp ngữ.

Một số hệ thống cho phép lời gọi hệ thống được thực hiện từ cấp lập trình ngôn ngữ cấp cao, như các hàm và lời gọi hàm. Nó có thể phát sinh lời gọi từ các thủ tục hay gọi trực tiếp trong dòng.

Để hiểu quá trình hoạt động của lời gọi hệ thống chúng ta cùng khảo sát một chương trình nhỏ dùng để đọc dữ liệu từ một tập tin chép qua tập tin khác. Dữ liệu nhập đầu tiên của của chương trình là tên của hai tập tin : tập tin nhập và tập tin xuất. Tên này được mô tả bằng nhiều cách tùy thuộc vào thiết kế hệ điều hành như : chương trình yêu

cầu người sử dụng cho biết tên của hai tập tin, họ cũng có thể cung cấp bằng cách lựa chọn với chuột. Khi có tên của hai tập tin, chương trình mở tập tin nhập và tạo tập tin xuất. Mỗi thao tác này được thực hiện bởi những lời gọi hệ thống khác. Cũng có những trường hợp phát sinh lỗi : Khi chương trình mở tập tin nhập, có thể xảy ra trường hợp không có tập tin có tên như mô tả hoặc tập tin bị cấm truy cập. Trong trường hợp này chương trình phải xuất thông điệp lên màn hình. Nếu tập tin nhập tồn tại, phải tạo tập tin mới. Hệ thống phải kiểm tra tiếp xem đã có tập tin xuất tồn tại không và sẽ có những lời gọi hệ thống tương ứng để giải quyết hoặc là hủy tiến trình, hai là xóa tập tin đã tồn tại và tạo tập tin mới. Sau khi đã thiết lập xong tập tin, hệ thống tiếp tục tạo vòng lặp đọc dữ liệu từ tập tin nhận và ghi lên tập tin xuất. Mỗi bước đều có kiểm tra lỗi. Sau khi chép xong, chương trình sẽ đóng hai tập tin lại (dùng một lời gọi hệ thống khác), xuất thông báo lên màn hình (dùng lời gọi hệ thống) cuối cùng chấm dứt chương trình (lời gọi hệ thống cuối cùng).

Trong các ngôn ngữ lập trình cấp cao, người sử dụng không cần quan tâm đến chi tiết mà chỉ cần thông qua các hàm hay các lệnh để thực hiện. Lời gọi hệ thống có thể diễn ra theo một cách khác. Kiểu và khối lượng thông tin tùy thuộc vào hệ thống và lúc gọi.

Có ba phương pháp được sử dụng để chuyển tham số cho hệ điều hành. Cách đơn giản nhất là chuyển tham số vào thanh ghi. Nếu có nhiều tham số, nó sẽ được lưu trữ trong khối hoặc bảng trong bộ nhớ. Cách cuối cùng là dùng cơ chế stack.

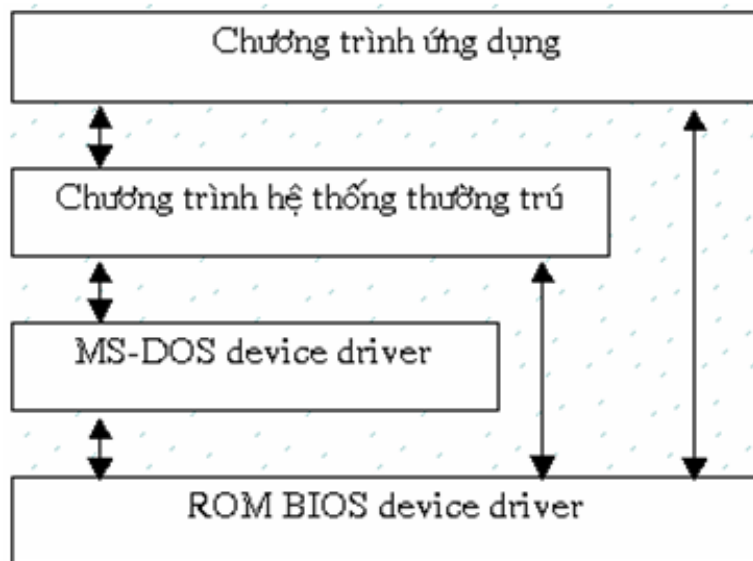
Lời gọi hệ thống có thể được chia thành các loại : kiểm soát tiến trình, thao tác tập tin, thao tác thiết bị, thông tin.

Cấu trúc hệ thống

Cấu trúc đơn giản

Cấu trúc này trong một số hệ thống thương mại và không có cấu trúc được định nghĩa tốt. Thông thường hệ điều hành bắt đầu là một hệ thống nhỏ, đơn giản và có giới hạn.

MS-DOS là một hệ điều hành có cấu trúc đơn giản, nó cung cấp những chức năng cần thiết nhất trong một không gian nhỏ nhất do sự giới hạn của phần cứng mà nó chạy trên đó và không chia thành những đơn thể rõ rệt.



Hình 1.2 Cấu trúc của MS-DOS

Mặc dù MS-DOS có cấu trúc nhưng giữa giao diện và chức năng không có sự phân chia rõ rệt. Các chương trình ứng dụng có thể truy xuất trực tiếp các thủ tục nhập xuất cơ bản và ghi trực tiếp lên màn hình hay bộ điều khiển đĩa.

Một hệ điều hành cũng có cấu trúc đơn giản là UNIX với những version đầu tiên. Cấu trúc của nó chỉ bao gồm hai phần : hạt nhân và các chương trình hệ thống. Hạt nhân được chia thành một chuỗi giao tiếp và device driver(bộ điều khiển thiết bị, xem bài XI).

Người sử dụng		
Shell và lệnh		
Biên dịch và thông dịch		
Thư viện hệ thống		
<i>Giao tiếp lời gọi hệ thống với hạt nhân</i>		
Tín hiệu kiểm soát hệ thống nhập xuất tuần tự của bộ điều khiển terminal	Hệ thống tập tin chuyển đổi giữa hệ thống nhập xuất khối và bộ điều khiển đĩa	Lập lịch CPU, thay thế phân trang , yêu cầu phân trang trong bộ nhớ ảo
<i>Giao tiếp giữa hạt nhân và hạt nhân</i>		
Bộ kiểm soát terminal	Bộ kiểm soát đĩa	Bộ kiểm soát bộ nhớ

Hình 1.3 Cấu trúc của UNIX

Những gì dưới lời gọi hệ thống và trên phần cứng là *hạt nhân*. Hạt nhân cung cấp hệ thống tập tin, lập lịch CPU, quản trị bộ nhớ và những chức năng hệ điều hành khác thông qua lời gọi hệ thống. Tóm lại là toàn bộ chức năng của hệ thống được kết hợp trong một lớp. Những chương trình hệ thống dùng những lời gọi hệ thống được hỗ trợ bởi hạt nhân để cung cấp những chức năng hữu ích như biên dịch và thao tác tập tin. Lời gọi hệ thống định nghĩa một giao tiếp lập trình cho UNIX, đó là tập hợp những chương trình hệ thống thông thường trong đó có định nghĩa giao tiếp với người sử dụng.

Cấu trúc theo lớp

Những version mới của UNIX được thiết kế để sử dụng phần cứng phức tạp hơn, do đó hệ điều hành được chia thành nhiều phần nhỏ hơn.

Bằng cách sử dụng kỹ thuật *topdown*, những chức năng và đặc tính của hệ thống được chia làm nhiều thành phần nhỏ. Che dấu thông tin, không cho chương trình của người sử dụng có thể cài đặt những hàm truy xuất cấp thấp, thay vào đó là những lớp giao tiếp bên trong.

Hệ điều hành được chia thành nhiều lớp. Lớp dưới cùng là phần cứng, lớp trên cùng là giao tiếp với người sử dụng. Lớp hệ điều hành được cài đặt thành những đối tượng trừu tượng. Thông thường một lớp của hệ điều hành bao gồm một số cấu trúc dữ liệu và các hàm có thể được gọi bởi lớp ở trên và bản thân nó gọi những chức năng của lớp bên dưới. Mỗi lớp cài đặt chỉ sử dụng những thao tác do lớp dưới cung cấp. Một lớp cũng không cần biết hệ điều hành được cài đặt như thế nào, nó chỉ cần biết những thao tác này làm gì thôi.

Cấu trúc lớp này lần đầu tiên được thiết kế và áp dụng cho hệ điều hành THE (Technische Hogeschool Eindhoven). Hệ thống này được chia thành sáu lớp như hình sau:

Lớp 5	Chương trình của người sử dụng
Lớp 4	Tạo buffer cho thiết bị nhập xuất
Lớp 3	Device driver thao tác màn hình
Lớp 2	Quản lý bộ nhớ
Lớp 1	Lập lịch CPU
Lớp 0	Phần cứng

Hình 1.4 Cấu trúc của hệ điều hành THE

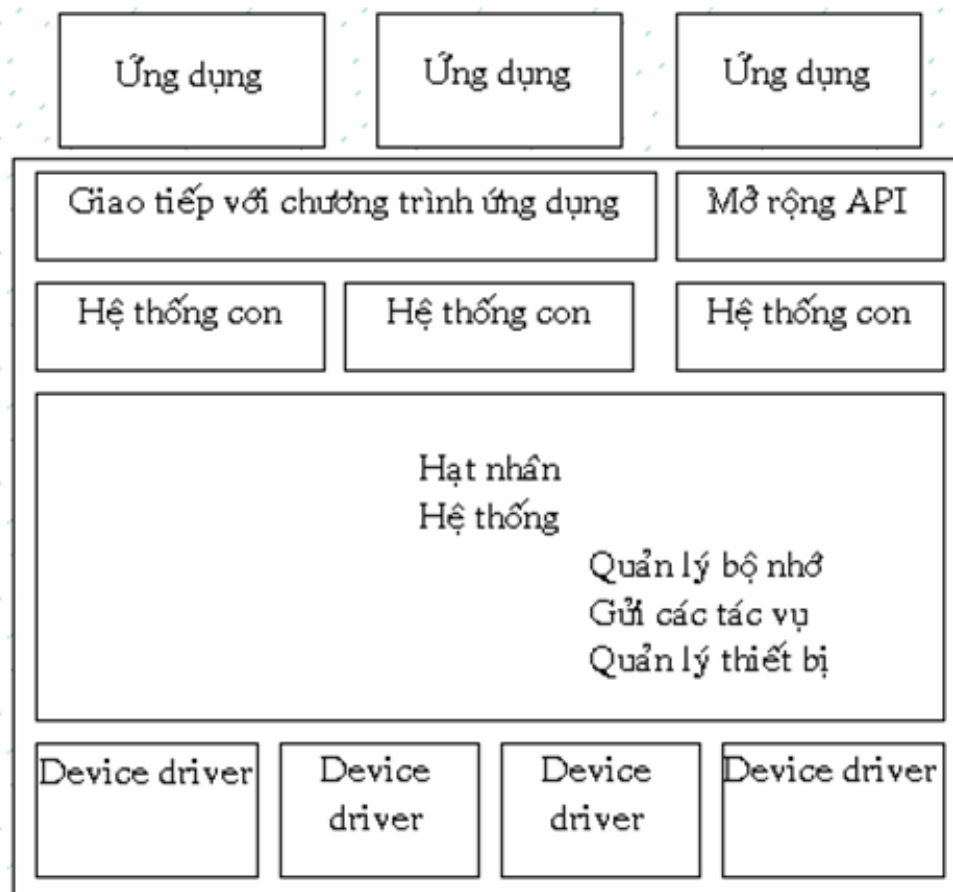
Lớp dưới cùng là phần cứng, lớp kế tiếp cài đặt lập lịch CPU, lớp tiếp theo cài đặt quản lý bộ nhớ. Bộ nhớ ở đây là bộ nhớ ảo. Lớp tiếp nữa chứa device driver cho các thao tác

với màn hình. Lớp kế là tổ chức buffer cho việc nhập xuất thiết bị. Cuối cùng là chương trình của người sử dụng.

Các ví dụ khác như cấu trúc lớp của hệ điều hành VENUS và OS/2

Lớp 6	Chương trình của người sử dụng
Lớp 5	Device driver và bộ lập lịch
Lớp 4	Bộ nhớ ảo
Lớp 3	Kênh nhập xuất
Lớp 2	Lập lịch CPU
Lớp 1	Thông dịch các chỉ thị
Lớp 0	Phần cứng

Hình 1.5 Cấu trúc lớp của VENUS



Hình 1.6 Cấu trúc lớp của OS/2

Máy ảo

Thông thường, một hệ thống máy tính bao gồm nhiều lớp. Phần cứng ở lớp thấp nhất. Hạt nhân ở lớp kế dùng các chỉ thị của phần cứng để tạo một tập hợp các lời gọi hệ thống. Các chương trình hệ thống có thể sử dụng hoặc là các lời gọi hệ thống hoặc là các chỉ thị của phần cứng. Vì vậy nó xem phần cứng và lời gọi hệ thống như cùng lớp.

Một số hệ thống có tổ chức sao cho các chương trình ứng dụng có thể gọi dễ dàng các chương trình hệ thống. Mặc dù chương trình hệ thống ở lớp cao hơn các phần khác nhưng chương trình ứng dụng có thể xem mọi phần dưới nó là một phần của máy. Lớp ứng dụng này sử dụng một khái niệm là máy ảo. Ví dụ hệ điều hành máy ảo của IBM.

Bằng cách sử dụng lập lịch cho CPU và kỹ thuật bộ nhớ ảo, một hệ điều hành có thể tạo nhiều tiến trình phức ảo, mỗi cái sẽ thực hiện trên một bộ xử lý và bộ nhớ riêng. Những tiến trình này có những đặc điểm riêng như lời gọi hệ thống và hệ thống tập tin không được cung cấp phần cứng trực tiếp.

Tài nguyên của hệ thống được chia xẻ để tạo những máy ảo. Lập lịch CPU chia xẻ CPU cho các người sử dụng. Spooling và hệ thống tập tin được chia thành những card đọc ảo và máy in ảo. Một terminal cung cấp các chức năng tạo các thao tác màn hình ảo.

Vấn đề phức tạp nhất của máy ảo là hệ thống đĩa. Giả sử hệ thống chỉ có ba bộ điều khiển đĩa nhưng có tới bảy máy ảo. Như vậy không thể gán cho mỗi máy ảo một bộ điều khiển đĩa và giải pháp là xây dựng hệ thống đĩa ảo.

Mặc dù khái niệm máy ảo rất hữu ích nhưng khó cài đặt. Máy ảo phải thực hiện ở hai dạng: dạng giám sát (monitor) và dạng người sử dụng. Ngoài ra máy ảo còn phải giải quyết các vấn đề về vận chuyển dữ liệu và thời gian.