

TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

BÀI GIẢNG
HỆ ĐIỀU HÀNH

GIẢNG VIÊN THỰC HIỆN: ThS. Nguyễn Bá Quảng
ThS. Nguyễn Huy Thịnh
ThS. Bùi Hải Phong

Số tín chỉ: 03

Số tiết giảng: 45 tiết

Lý thuyết: 45 tiết

Hà Nội, 6/ 2017

MỤC LỤC

GIỚI THIỆU MÔN HỌC	4
Chương 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH.....	6
1.1. Các khái niệm cơ bản.....	6
1.2. Các chức năng của hệ điều hành.....	17
1.3. Các thành phần và cấu trúc của hệ điều hành	19
Chương 2: QUẢN LÝ TIỀN TRÌNH.....	31
2.1. Các khái niệm cơ bản.....	31
2.2. Tài nguyên Găng (Critical Resource) và đoạn Găng (Critical Section)	44
2.3. Hiện tượng bế tắc (Deadlock).....	78
Chương 3: LẬP LỊCH CHO CPU	88
3.1. Các khái niệm cơ bản.....	88
3.2. Các thuật toán lập lịch	88
3.3. Ngắt.....	95
Chương 4: QUẢN LÝ BỘ NHỚ TRONG	99
4.1. Các khái niệm cơ bản.....	99
4.2. Các kỹ thuật cấp phát bộ nhớ.....	102
4.3. Bộ nhớ ảo	113
Chương 5: QUẢN LÝ BỘ NHỚ NGOÀI	135
5.1. Các khái niệm cơ bản.....	135
5.2. Các phương pháp quản lý không gian nhớ tự do	135
5.3. Các phương pháp cấp phát không gian nhớ tự do	137
5.4. Lập lịch cho đĩa.....	141
5.5. Hệ file.....	141
Chương 6: QUẢN LÝ THIẾT BỊ.....	143
6.1. Nguyên tắc tổ chức và quản lý thiết bị	143
6.2. Các kỹ thuật áp dụng trong quản lý thiết bị.....	143
Chương 7: BẢO VỆ VÀ AN TOÀN HỆ THỐNG	149
7.1. Bảo vệ hệ thống	149
7.2. An toàn hệ thống.....	154
7.3. Virus máy tính	156
Chương 8: HỆ ĐIỀU HÀNH ĐA XỬ LÝ	160
8.1. Tổng quan về hệ điều hành đa xử lý.....	160

8.2. Hệ điều hành đa xử lý tập trung.....	162
8.3. Hệ phân tán	163

TaiLieu.vn

GIỚI THIỆU MÔN HỌC

I. Giới thiệu chung

Hệ điều hành (Operating system) là thành phần quan trọng và không thể thiếu đối với các hệ thống máy tính. Nó đóng vai trò trung gian giữa người sử dụng hệ thống máy tính và phần cứng máy tính. Hệ điều hành cung cấp môi trường để người dùng có thể sử dụng hệ thống máy tính một cách hiệu quả và dễ dàng.

Kiến thức liên quan tới hệ điều hành có thể chia thành ba dạng chính: thứ nhất đó là các kiến thức và kỹ năng về việc cài đặt, sử dụng, khai thác, đánh giá hệ điều hành một cách hiệu quả. Các kiến thức này cần thiết cho người sử dụng cũng như các kỹ sư, chuyên gia trong việc vận hành, quản lý hệ thống máy tính nói chung. Thứ hai, hệ điều hành được xem xét dưới góc độ thiết kế và xây dựng. Đây là các kiến thức cần thiết cho các chuyên gia về hệ thống cũng như những người tham gia thiết kế, xây dựng hệ điều hành. Thứ ba, là các nguyên lý, các khái niệm về hệ điều hành như là một phần quan trọng của hệ thống máy tính.

Trong chương trình giảng dạy ở bậc đại học, môn học “*Hệ điều hành*” là môn học bắt buộc và đóng vai trò vô cùng quan trọng đối với sinh viên chuyên ngành Công nghệ thông tin.

Để giảng dạy và học tập tốt môn học, tài liệu giảng dạy là yêu cầu tiên quyết. Từ yêu cầu thực tế đó, Khoa Công nghệ thông tin tổ chức biên soạn tài liệu dùng chung “*Hệ điều hành*”. Tài liệu được xây dựng phục vụ cho mục đích giảng dạy, học tập, nghiên cứu, tham khảo cho giảng viên, sinh viên khoa Công nghệ thông tin, trường Đại học Kiến Trúc Hà Nội.

II. Mục tiêu môn học

Môn học cung cấp cho sinh viên chuyên ngành Công nghệ thông tin các kiến thức cần thiết về hệ điều hành; giúp sinh viên có lý thuyết và kỹ năng thực hành cần có về hệ điều hành; áp dụng các kiến thức của môn học để quản lý, khai thác, vận hành hệ thống máy tính có hiệu quả cao trong thực tế.

Kết thúc môn học sinh viên có kiến thức vững vàng về nguyên lý hệ điều hành; áp dụng các kiến thức về hệ điều hành để khai thác, vận hành tốt các hệ thống máy tính.

Tài liệu đề cập tới các nguyên lý cơ bản nhất của hệ điều hành: quản lý tiến trình; lập lịch CPU; quản lý bộ nhớ; quản lý thiết bị; các vấn đề liên quan tới bảo vệ, bảo mật hệ điều hành và một số chủ đề nâng cao về hệ điều hành.

III. Phương pháp nghiên cứu

Bài giảng cho môn học “*Hệ điều hành*” được biên soạn dựa trên các cơ sở như sau:

- Nghiên cứu, tham khảo các giáo trình, tài liệu liên quan đến bài giảng.
- Tổng hợp kiến thức cần thiết liên quan tới bài giảng.
- Xây dựng các ứng dụng, đưa ra các ví dụ liên quan tới hệ điều hành của các hệ thống máy tính trong thực tế để minh họa cho lý thuyết môn học.

Bài giảng được biên soạn cho Sinh viên chuyên ngành Công nghệ Thông tin học kỳ 4 Khoa Công nghệ thông tin - Đại học Kiến Trúc Hà Nội sau khi đã hoàn thành các môn học: Tin học đại cương, Nhập môn công nghệ thông tin, Cấu trúc dữ liệu và giải thuật.

Để học tốt môn học, sinh viên cần chú ý:

Kết hợp học lý thuyết và thực hành. Học tới đâu thực hành tới đó, đặc biệt chú trọng vào kỹ năng lập trình giải quyết các bài toán trong học phần.

Có ý thức tự giác học tập, tham gia đầy đủ các buổi học trên lớp.

Chương 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

Nếu không có phần mềm, máy tính chỉ là một thiết bị điện tử thông thường. Với sự hỗ trợ của phần mềm, máy tính có thể lưu trữ, xử lý thông tin và người sử dụng có thể gọi lại được thông tin này. Phần mềm máy tính có thể chia thành nhiều loại: chương trình hệ thống, quản lý sự hoạt động của chính máy tính. Chương trình ứng dụng, giải quyết các vấn đề liên quan đến việc sử dụng và khai thác máy tính của người sử dụng. Hệ điều hành thuộc nhóm các chương trình hệ thống và nó là một chương trình hệ thống quan trọng nhất đối với máy tính và cả người sử dụng. Hệ điều hành điều khiển tất cả các tài nguyên của máy tính và cung cấp một môi trường thuận lợi để các chương trình ứng dụng do người sử dụng viết ra có thể chạy được trên máy tính. Trong chương này chúng ta xem xét vai trò của hệ điều hành trong trường hợp này.

Một máy tính hiện đại có thể bao gồm: một hoặc nhiều processor, bộ nhớ chính, clocks, đĩa, giao diện mạng, và các thiết bị vào/ra khác. Tất cả nó tạo thành một hệ thống phức tạp. Để viết các chương trình để theo dõi tất cả các thành phần của máy tính và sử dụng chúng một cách hiệu quả, người lập trình phải biết processor thực hiện chương trình như thế nào, bộ nhớ lưu trữ thông tin như thế nào, các thiết bị đĩa làm việc (ghi/đọc) như thế nào, lỗi nào có thể xảy ra khi đọc một block đĩa, ... đây là những công việc rất khó khăn và quá khó đối với người lập trình. Nhưng rất may cho cả người lập trình ứng dụng và người sử dụng là những công việc trên đã được hệ điều hành hỗ trợ nên họ không cần quan tâm đến nữa. Chương này cho chúng ta một cái nhìn tổng quan về những gì liên quan đến việc thiết kế cài đặt cũng như chức năng của hệ điều hành để hệ điều hành đạt được mục tiêu: Giúp người sử dụng khai thác máy tính dễ dàng và chương trình của người sử dụng có thể chạy được trên máy tính.

1.1. Các khái niệm cơ bản

1.1.1. Khái niệm và phân loại hệ điều hành

a. Khái niệm hệ điều hành.

Khó có một khái niệm hay định nghĩa chính xác về hệ điều hành, vì hệ điều hành là một bộ phận được nhiều đối tượng khai thác nhất, họ có thể là người sử dụng thông thường, có thể là lập trình viên, có thể là người quản lý hệ thống và tùy theo mức độ khai thác hệ điều hành mà họ có thể đưa ra những khái niệm khác nhau về nó. Ở đây ta xem xét 3 khái niệm về hệ điều hành dựa trên quan điểm của người khai thác hệ thống máy tính:

Khái niệm 1: Hệ điều hành là một hệ thống mô hình hoá, mô phỏng hoạt động của máy tính, của người sử dụng và của lập trình viên, hoạt động trong chế độ đối thoại nhằm tạo môi trường khai thác thuận lợi hệ thống máy tính và quản lý tối ưu tài nguyên của hệ thống.

Khái niệm 2: Hệ điều hành là hệ thống chương trình với các chức năng giám sát, điều khiển việc thực hiện các chương trình của người sử dụng, quản lý và phân chia tài nguyên cho nhiều chương trình người sử dụng đồng thời sao cho việc khai thác chức năng của hệ thống máy tính của người sử dụng là thuận lợi và hiệu quả nhất.

Khái niệm 3: Hệ điều hành là một chương trình đóng vai trò như là giao diện giữa người sử dụng và phần cứng máy tính, nó điều khiển việc thực hiện của tất cả các loại chương trình. Khái niệm này rất gần với các hệ điều hành đang sử dụng trên các máy tính hiện nay.

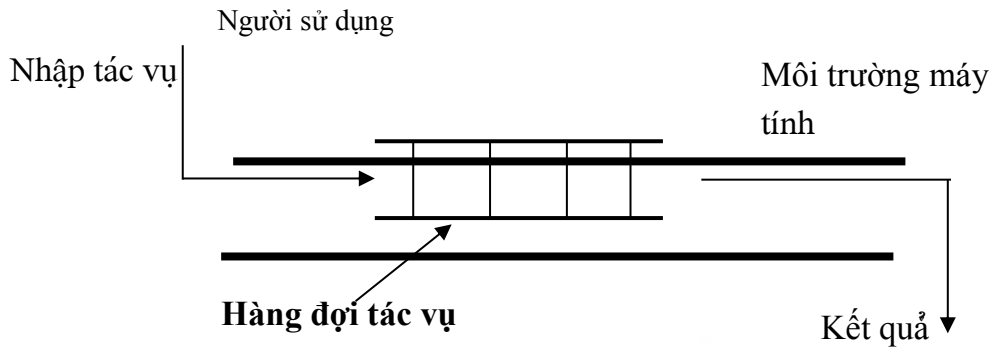
Từ các khái niệm trên chúng ta có thể thấy rằng: Hệ điều hành ra đời, tồn tại và phát triển là để giải quyết vấn đề sử dụng máy tính của người sử dụng, nhằm giúp người sử dụng khai thác hết các chức năng của phần cứng máy tính mà cụ thể là giúp người sử dụng thực hiện được các chương trình của họ trên máy tính.

b. Phân loại hệ điều hành.

Có nhiều cách khác nhau để phân loại hệ điều hành, ở đây chúng tôi dựa vào cách mà hệ điều hành thực hiện các công việc, các tác vụ, các tiến trình của người sử dụng để phân loại hệ điều hành.

Hệ điều hành xử lý theo lô đơn giản.

Hệ điều hành loại này thực hiện các tác vụ lần lượt theo những chỉ thị đã được xác định trước. Khi một tác vụ chấm dứt thì hệ thống sẽ tự động thực hiện tác vụ tiếp theo mà không cần sự can thiệp từ bên ngoài, do đó hệ thống đạt tốc độ thực hiện cao. Để thực hiện được điều này hệ điều hành phải có bộ phận giám sát thường trực để giám sát việc thực hiện của các tác vụ trong hệ thống, bộ phận này thường trú trong bộ nhớ chính.



Hình 1.1. Hoạt động của hệ điều hành lô đơn giản

Trong hệ điều hành này khi hệ thống cần thực hiện một tác vụ thì nó phải lưu chương trình và dữ liệu của các tác vụ vào hàng đợi các công việc, sau đó sẽ thực hiện lần lượt từng bộ chương trình và dữ liệu của tác vụ tương ứng trong hàng đợi và cho ra lần lượt các kết quả. Hình 1.1. ở trên minh họa cho sự hoạt động của hệ thống theo lô đa chương.

Với cách tổ chức *hàng đợi tác vụ*, thì hệ thống không thể thay đổi chương trình và dữ liệu của các tác vụ ngay cả khi chúng còn nằm trong hàng đợi, đây là một hạn chế. Mặt khác trong quá trình thực hiện tác vụ nếu tác vụ chuyển sang truy xuất trên thiết bị vào/ra thì processor rơi vào trạng thái chờ điều này gây lãng phí thời gian xử lý của processor.

Hệ điều hành xử lý theo lô đa chương

Một trong những hạn chế của hệ điều hành xử lý theo lô đơn giản là lãng phí thời gian xử lý của processor khi tác vụ hiện tại truy xuất đến thiết bị vào/ra. Hệ điều hành xử lý theo lô đa chương sẽ khắc phục hạn chế này.

Hệ điều hành loại này có khả năng thực hiện nhiều tác vụ, nhiều chương trình đồng thời. Khi cần thực hiện nhiều tác vụ đồng thời hệ điều hành sẽ nạp một phần code và data của các tác vụ vào bộ nhớ (các phần còn lại sẽ được nạp sau tại thời điểm thích hợp) và tất cả đều ở trạng thái sẵn sàng thực hiện, sau đó hệ điều hành bắt đầu thực hiện một tác vụ nào đó, nhưng khi tác vụ đang thực hiện cần truy xuất thiết bị vào/ra thì processor sẽ được chuyển sang thực hiện các tác vụ khác, và cứ như thế hệ điều hành tổ chức chuyển hướng processor để thực hiện hết các phần tác vụ trong bộ nhớ cũng như các tác vụ mà hệ thống yêu cầu.

Hệ điều hành loại này mang lại hai ưu điểm đó là tiết kiệm được bộ nhớ, vì không nạp hết code và data của các tác vụ vào bộ nhớ, và hạn chế thời gian rỗi của

processor. Tuy nhiên nó phải chi phí cao cho việc lập lịch processor, tức là khi có được processor hệ điều hành phải xem xét nên chuyển nó cho tác vụ nào trong số các tác vụ đang ở trạng thái sẵn sàng. Ngoài ra hệ điều hành còn phải giải quyết việc chia sẻ bộ nhớ chính cho các tác vụ khác nhau. Hệ điều hành MS_DOS là hệ điều hành đơn nhiệm, đa chương.

Hệ điều hành chia sẻ thời gian

Khái niệm chia sẻ thời gian ra đời đã đánh dấu một bước phát triển mới của hệ điều hành trong việc điều khiển các hệ thống đa người dùng. Chia sẻ thời gian ở đây chính là chia sẻ thời gian xử lý của processor cho các tác vụ, các tiến trình đang ở trong trạng thái sẵn sàng thực hiện.

Nguyên tắc của hệ điều hành chia sẻ thời gian tương tự như trong hệ điều hành xử lý theo lô đa chương nhưng việc chuyển processor từ tác vụ, tiến trình này sang tác vụ, tiến trình khác không phụ thuộc vào việc tác vụ, tiến trình hiện tại có truy xuất đến thiết bị vào/ra hay không mà chỉ phụ thuộc vào sự điều phối processor của hệ điều hành. Công việc điều phối processor của hệ điều hành rất phức tạp phụ thuộc vào nhiều yếu tố khác nhau, chúng ta sẽ đề cập đến vấn đề này trong chương sau của tài liệu này.

Trong hệ điều hành này thời gian chuyển đổi processor giữa các tác vụ là rất nhỏ nên ta có cảm giác các tác vụ thực hiện song song với nhau. Với hệ điều hành này người sử dụng có thể yêu cầu hệ điều hành thực hiện nhiều chương trình, tiến trình, tác vụ đồng thời với nhau.

Hệ điều hành chia sẻ thời gian là mở rộng logic của hệ điều hành đa chương và nó thường được gọi là hệ điều hành đa nhiệm (Multitasking). Hệ điều hành Windows 9x/NT là các hệ điều hành đa nhiệm.

Hệ điều hành đa vi xử lý

Là các hệ điều hành dùng để điều khiển sự hoạt động của các hệ thống máy tính có nhiều vi xử lý. Các hệ điều hành đa vi xử lý (multiprocessor) gồm có 2 loại:

Đa xử lý đối xứng (SMP: symmetric): Trong hệ thống này vi xử lý nào cũng có thể chạy một loại tiểu trình bất kỳ, các vi xử lý giao tiếp với nhau thông qua một bộ nhớ dùng chung. Hệ SMP cung cấp một cơ chế chịu lỗi và khả năng cân bằng tải tối ưu hơn, vì các tiểu trình của hệ điều hành có thể chạy trên bất kỳ vi xử lý nào nên nguy cơ xảy ra tình trạng tắc nghẽn ở CPU giảm đi đáng kể. Vấn đề đồng bộ giữa các vi xử lý được đặt lên hàng đầu khi thiết kế hệ điều hành cho hệ thống SMP. Hệ điều hành Windows NT, hệ điều hành Windows 2000 là các hệ điều hành đa xử lý đối xứng.

Đa xử lý bất đối xứng (ASMP: asymmetric): Trong hệ thống này hệ điều

hành dành ra một hoặc hai vi xử lý để sử dụng riêng, các vi xử lý còn lại dùng để điều khiển các chương trình của người sử dụng. Hệ ASMP đơn giản hơn nhiều so với hệ SMP, nhưng trong hệ này nếu có một vi xử lý trong các vi xử lý dành riêng cho hệ điều hành bị hỏng thì hệ thống có thể ngừng hoạt động.

Hệ điều hành xử lý thời gian thực

Hệ điều hành này khắc phục nhược điểm của hệ điều hành xử lý theo lô, tức là nó có khả năng cho kết quả tức thời, chính xác sau mỗi tác vụ.

Trong hệ điều hành này các tác vụ cần thực hiện không được đưa vào hàng đợi mà được xử lý tức thời và trả lại ngay kết quả hoặc thông báo lỗi cho người sử dụng có yêu cầu. Hệ điều hành này hoạt động đòi hỏi sự phối hợp cao giữa phần mềm và phần cứng.

Hệ điều hành mạng

Là các hệ điều hành dùng để điều khiển sự hoạt động của mạng máy tính. Ngoài các chức năng cơ bản của một hệ điều hành, các hệ điều hành mạng còn phải thực hiện việc chia sẻ và bảo vệ tài nguyên của mạng. Hệ điều hành Windows 9x/NT, Windows 200, Linux, là các hệ điều hành mạng máy tính.

Tóm lại: Qua sự phân loại hệ điều hành ở trên ta có thể thấy được quá trình phát triển (evolution) của hệ điều hành. Để khắc phục hạn chế về lãng phí thời gian xử lý của processor trong hệ điều hành theo lô thì hệ điều hành theo lô đa chương ra đời. Để khai thác tối đa thời gian xử lý của processor và tiết kiệm hơn nữa không gian bộ nhớ chính hệ điều hành chia sẻ thời gian ra đời. Chia sẻ thời gian xử lý của processor kết hợp với chia sẻ không gian bộ nhớ chính đã giúp cho hệ điều hành có thể đưa vào bộ nhớ chính nhiều chương trình, tiến trình hơn và các chương trình, tiến trình này có thể hoạt động đồng thời với nhau, nhờ đó mà hiệu suất của hệ thống tăng lên, và cũng từ đây khái niệm hệ điều hành đa chương ra đời. Hệ điều hành đa xử lý và hệ điều hành mạng được phát triển dựa trên hệ điều hành đa nhiệm. Hệ điều hành thời gian thực ra đời là để khắc phục hạn chế của hệ điều hành theo lô và điều khiển các hệ thống thời gian thực. Từ đây chúng ta rút ra một điều rằng: *các hệ điều hành ra đời sau luôn tìm cách khắc phục các hạn chế của hệ điều hành trước đó và phát triển nhiều hơn nữa để đáp ứng yêu cầu ngày càng cao của của người sử dụng và chương trình người sử dụng, cũng như khai thác tối đa các chức năng của phần cứng máy tính để nâng cao hiệu suất của hệ thống. Nhưng chức năng của hệ điều hành càng cao thì chi phí cho nó cũng tăng theo và cấu trúc của hệ điều hành cũng sẽ phức tạp hơn.*

1.1.2. Lịch sử phát triển của hệ điều hành

Thế hệ 1 (1945 - 1955):

Vào những năm 1950 máy tính dùng ống chân không ra đời. Ở thế hệ này mỗi máy tính được một nhóm người thực hiện, bao gồm việc thiết kế, xây dựng chương trình, thao tác, quản lý,

Ở thế hệ này người lập trình phải dùng ngôn ngữ máy tuyệt đối để lập trình. Khái niệm ngôn ngữ lập trình và hệ điều hành chưa được biết đến trong khoảng thời gian này.

Thế hệ 2 (1955 - 1965):

Máy tính dùng bán dẫn ra đời, và được sản xuất để cung cấp cho khách hàng. Bộ phận sử dụng máy tính được phân chia rõ ràng: người thiết kế, người xây dựng, người vận hành, người lập trình, và người bảo trì. Ngôn ngữ lập trình Assembly và Fortran ra đời trong thời kỳ này. Với các máy tính thế hệ này để thực hiện một thao tác, lập trình viên dùng Assembly hoặc Fortran để viết chương trình trên phiếu đục lỗ sau đó đưa phiếu vào máy, máy thực hiện cho kết quả ở máy in.

Hệ thống xử lý theo lô cũng ra đời trong thời kỳ này. Theo đó, các thao tác cần thực hiện trên máy tính được ghi trước trên băng từ, hệ thống sẽ đọc băng từ, thực hiện lần lượt và cho kết quả ở băng từ xuất. Hệ thống xử lý theo lô hoạt động dưới sự điều khiển của một chương trình đặc biệt, chương trình này là hệ điều hành sau này.

Thế hệ 3 (1965 - 1980)

Máy IBM 360 được sản xuất hàng loạt để tung ra thị trường. Các thiết bị ngoại vi xuất hiện ngày càng nhiều, do đó các thao tác điều khiển máy tính và thiết bị ngoại vi ngày càng phức tạp hơn. Trước tình hình này nhu cầu cần có một hệ điều hành sử dụng chung trên tất cả các máy tính của nhà sản xuất và người sử dụng trở nên bức thiết hơn. Và hệ điều hành đã ra đời trong thời kỳ này.

Hệ điều hành ra đời nhằm điều phối, kiểm soát hoạt động của hệ thống và giải quyết các yêu cầu tranh chấp thiết bị. Hệ điều hành đầu tiên được viết bằng ngôn ngữ Assembly. Hệ điều hành xuất hiện khái niệm đa chương, khái niệm chia sẻ thời gian và kỹ thuật Spool. Trong giai đoạn này cũng xuất hiện các hệ điều hành Multics và Unix.

Thế hệ 4 (từ 1980)

Máy tính cá nhân ra đời. Hệ điều hành MS_DOS ra đời gắn liền với máy tính IBM_PC. Hệ điều hành mạng và hệ điều hành phân tán ra đời trong thời kỳ này.

Trên đây chúng tôi không có ý định trình bày chi tiết, đầy đủ về lịch sử hình

thành của hệ điều hành, mà chúng tôi chỉ muốn mượn các mốc thời gian về sự ra đời của các thế hệ máy tính để chỉ cho bạn thấy quá trình hình thành của hệ điều hành gắn liền với quá trình hình thành máy tính. Mục tiêu của chúng tôi trong mục này là muốn nhấn mạnh với các bạn mấy điểm sau đây:

Các ngôn ngữ lập trình, đặc biệt là các ngôn ngữ lập trình cấp thấp, ra đời trước các hệ điều hành. Đa số các hệ điều hành đều được xây dựng từ ngôn ngữ lập trình cấp thấp trừ hệ điều hành Unix, nó được xây dựng từ C, một ngôn ngữ lập trình cấp cao.

Nếu không có hệ điều hành thì việc khai thác và sử dụng máy tính sẽ khó khăn và phức tạp rất nhiều và không phải bất kỳ ai cũng có thể sử dụng máy tính được.

Sự ra đời và phát triển của hệ điều hành gắn liền với sự phát triển của máy tính, và ngược lại sự phát triển của máy tính kéo theo sự phát triển của hệ điều hành. Hệ điều hành thực sự phát triển khi máy tính PC xuất hiện trên thị trường.

Ngoài ra chúng tôi cũng muốn giới thiệu một số khái niệm như: hệ thống xử lý theo lô, hệ thống đa chương, hệ thống chia sẻ thời gian, kỹ thuật Spool, ..., mà sự xuất hiện của những khái niệm này đánh dấu một bước phát triển mới của hệ điều hành. Chúng ta sẽ làm rõ các khái niệm trên trong các chương sau của tài liệu này.

1.1.3. Các khái niệm cơ bản của hệ điều hành

Tiến trình (Process) và tiểu trình (Thread)

Tiến trình là một bộ phận của chương trình đang thực hiện. Tiến trình là đơn vị làm việc cơ bản của hệ thống, trong hệ thống có thể tồn tại nhiều tiến trình cùng hoạt động, trong đó có cả tiến trình của hệ điều hành và tiến trình của chương trình người sử dụng. Các tiến trình này có thể hoạt động đồng thời với nhau.

Để một tiến trình đi vào trạng thái hoạt động thì hệ thống phải cung cấp đầy đủ tài nguyên cho tiến trình. Hệ thống cũng phải duy trì đủ tài nguyên cho tiến trình trong suốt quá trình hoạt động của tiến trình.

Ở đây cần phân biệt sự khác nhau giữa tiến trình và chương trình, chương trình là một tập tin thụ động nằm trên đĩa, tiến trình là trạng thái động của chương trình.

Các hệ điều hành hiện đại sử dụng mô hình đa tiểu trình, trong một tiến trình có thể có nhiều tiểu trình. Tiểu trình cũng là đơn vị xử lý cơ bản trong hệ thống, nó cũng xử lý tuần tự đoạn code của nó, nó cũng sở hữu một con trỏ lệnh, một tập các thanh ghi và một vùng nhớ stack riêng và các tiểu trình cũng chia sẻ thời gian xử lý

của processor như các tiến trình.

Các tiểu trình trong một tiến trình chia sẻ một không gian địa chỉ chung, điều này có nghĩa các tiểu trình có thể chia sẻ các biến toàn cục của tiến trình, có thể truy xuất đến stack của tiểu trình khác trong cùng tiến trình. Như vậy với mô hình tiểu trình, trong hệ thống có thể tồn tại nhiều dòng xử lý cùng chia sẻ một không gian địa chỉ bộ nhớ, các dòng xử lý này hoạt động song song với nhau.

Bộ xử lý lệnh (Shell)

Shell là một bộ phận hay một tiến trình đặc biệt của hệ điều hành, nó có nhiệm vụ nhận lệnh của người sử dụng, phân tích lệnh và phát sinh tiến trình mới để thực hiện yêu cầu của lệnh, tiến trình mới này được gọi là tiến trình đáp ứng yêu cầu.

Shell nhận lệnh thông qua cơ chế dòng lệnh, đó chính là nơi giao tiếp giữa người sử dụng và hệ điều hành, mỗi hệ điều hành khác nhau có cơ chế dòng lệnh khác nhau, với MS_DOS đó là con trỏ lệnh và dấu nhắc hệ điều hành (C:\>_), với Windows 9x đó là nút Start\Run. Tập tin Command.Com chính là Shell của MS_DOS.

Trong môi trường hệ điều hành đơn nhiệm, ví dụ như MS_DOS, khi tiến trình đáp ứng yêu cầu hoạt động thì Shell sẽ chuyển sang trạng thái chờ, để chờ cho đến khi tiến trình đáp ứng yêu cầu kết thúc thì Shell trở lại trạng thái sẵn sàng nhận lệnh mới.

Trong môi trường hệ điều hành đa nhiệm, ví dụ như Windows 9x, sau khi phát sinh tiến trình đáp ứng yêu cầu và đưa nó vào trạng thái hoạt động thì Shell sẽ chuyển sang trạng thái sẵn sàng nhận lệnh mới, nhờ vậy Shell có khả năng khởi tạo nhiều tiến trình đáp ứng yêu cầu để nó hoạt động song song với nhau, hay chính xác hơn trong môi trường hệ điều hành đa nhiệm người sử dụng có thể khởi tạo nhiều chương trình để nó hoạt động đồng thời với nhau.

Chú ý: Hầu hết các ngôn ngữ lập trình đều hỗ trợ các công cụ để người sử dụng hay người lập trình có thể gọi shell ngay trong các ứng dụng của họ. Khi một ứng dụng cần gọi thực hiện một chương trình nào đó thì:

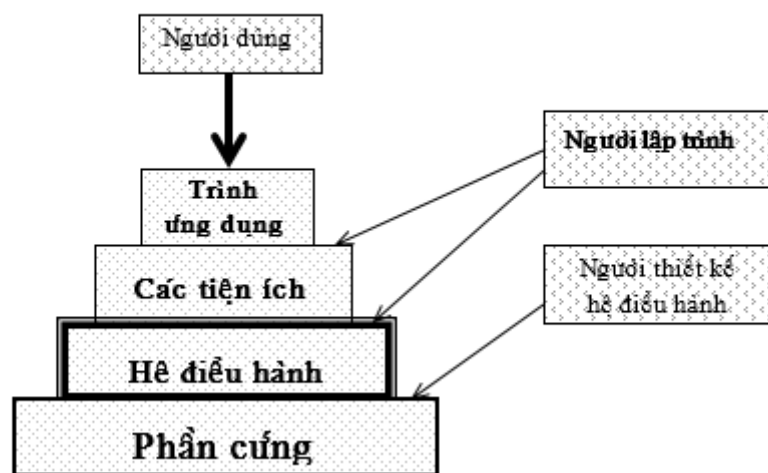
- Trong Assembly, các ứng dụng gọi hàm 4Bh/21h của MS_DOS.
- Trong Pascal, các ứng dụng gọi thủ tục Exec.
- Trong Visual Basic, các ứng dụng gọi hàm/ thủ tục Shell. Ví dụ dòng lệnh sau: Shell "C:\Windows\notepad.exe" có thể gọi thực hiện chương trình Notepad của Windows.
- Trong Windows 9x/ Windows NT, các ứng dụng gọi hàm ShellExecute.

Sự phân lớp hệ thống (System Layering)

Như đã biết, hệ điều hành là một hệ thống các chương trình bao quanh máy tính

thực (vật lý) nhằm tạo ra một máy tính mở rộng (logic) đơn giản và dễ sử dụng hơn. Theo đó, khi khai thác máy tính người sử dụng chỉ cần tác động vào lớp vỏ bọc bên ngoài của máy tính, mọi sự giao tiếp giữa lớp vỏ bọc này với các chi tiết phần cứng bên trong đều do hệ điều hành thực hiện.

Mỗi người sử dụng khác nhau yêu cầu khai thác hệ điều hành ở những mức độ khác nhau. Người sử dụng thông thường chỉ cần một môi trường thuận lợi để họ thực hiện các ứng dụng, các lập trình viên cần có một môi trường lập trình tốt để họ có thể triển khai các ứng dụng, các chuyên viên lập trình hệ thống cần hệ điều hành cung cấp cho họ các công cụ để họ can thiệp sâu hơn vào hệ thống phần cứng máy tính, ... Để đáp ứng yêu cầu của nhiều đối tượng người sử dụng khác nhau hệ điều



Hình 1.2. Sự phân lớp hệ thống

hành thực hiện phân lớp các chương trình bao quanh máy tính. Các hệ thống như vậy được gọi là hệ thống phân lớp. Hình vẽ 1.2. ở trên minh họa cho một hệ thống phân lớp.

Ta có thể hình dung một hệ thống phân lớp được tổ chức như sau:

- Trong cùng là hệ điều hành.
- Tiếp theo là các ngôn ngữ lập trình
- ...
- **Ngoài cùng là các chương trình ứng dụng .**

Người sử dụng tác động vào lớp trong cùng sẽ gặp nhiều khó khăn hơn khi tác

động vào lớp ngoài cùng.

Tài nguyên hệ thống (System Resources)

Tài nguyên hệ thống là những tồn tại về mặt vật lý tại một thời điểm nhất định hoặc tại mọi thời điểm, và nó có khả năng tác động đến hiệu suất của hệ thống. Một cách tổng quát có thể chia tài nguyên của hệ thống thành hai loại cơ bản:

- Tài nguyên không gian: là các không gian lưu trữ của hệ thống như đĩa, bộ nhớ chính, quan trọng nhất là không gian bộ nhớ chính, nơi lưu trữ các chương trình đang được CPU thực hiện.
- Tài nguyên thời gian: chính là thời gian thực hiện lệnh của processor và thời gian truy xuất dữ liệu trên bộ nhớ.

Sau đây là một vài tài nguyên hệ thống:

Bộ nhớ: Đặc trưng cơ bản của bộ nhớ là thời gian truy cập trực tiếp, thời gian truy cập tuần tự, và dung lượng nhớ. Bộ nhớ được gọi là thực hiện nếu processor có thể thực hiện một câu lệnh trong nó, loại bộ nhớ này có thời gian truy cập trực tiếp và tuần tự là như nhau. Bộ nhớ trong (RAM) của PC là bộ nhớ thực hiện và nó được quản lý bởi hệ thống.

Khi sử dụng bộ nhớ ta cần phân biệt 2 khái niệm: bộ nhớ và truy cập tới bộ nhớ. Bộ nhớ chỉ vùng vật lý chứa dữ liệu, truy cập bộ nhớ là quá trình tìm đến dữ liệu trên bộ nhớ. Có thể xem đây là 2 loại tài nguyên khác nhau vì chúng tồn tại độc lập với nhau.

Processor: Là tài nguyên quan trọng nhất của hệ thống, nó được truy cập ở mức câu lệnh và chỉ có nó mới làm cho câu lệnh thực hiện hay chỉ có Processor mới đưa tiến trình vào trạng thái hoạt động. Trong thực tế khi xem xét về processor người ta chỉ chú ý đến thời gian xử lý của processor.

Tài nguyên ảo/ tài nguyên logic (Virtual Resources): Là loại tài nguyên cung cấp cho chương trình người sử dụng dưới dạng đã được biến đổi, nó chỉ xuất hiện khi hệ thống cần tới nó hoặc khi hệ thống tạo ra nó và nó sẽ tự động mất đi khi hệ thống kết thúc hay chính xác hơn là khi tiến trình gắn với nó đã kết thúc. Tài nguyên ảo có thể là: Đĩa ảo trong môi trường MS_DOS. Điều khiển in trong môi trường mạng của Windows 9x/NT. Nội dung thư mục Spool trong Windows 9x.

Trên khía cạnh cấp phát tài nguyên cho các tiến trình đang hoạt động đồng thời thì tài nguyên hệ thống được chia thành 2 loại:

Tài nguyên phân chia được: là những tài nguyên mà tại một thời điểm nó có thể cấp phát cho nhiều tiến trình khác nhau, các tiến trình song song có thể đồng thời sử dụng các tài nguyên này. Bộ nhớ chính và Processor là 2 tài nguyên phân chia

được diễn hình nhất, bởi tại một thời điểm có thể có nhiều tiến trình cùng chia nhau sử dụng không gian lưu trữ của bộ nhớ chính và có thể có nhiều tiến trình thay nhau sử dụng thời gian xử lý của processor.

Tài nguyên không phân chia được: là những tài nguyên mà tại một thời điểm nó chỉ có thể cấp phát cho một tiến trình duy nhất. Máy in là một tài nguyên không phân chia được điển hình nhất.

Vấn đề đặt ra đối với hệ điều hành là phải biến các tài nguyên không phân chia được thành những tài nguyên phân chia được, theo một cách nào đó, để cấp phát cho các tiến trình khi nó có yêu cầu, đặc biệt là các tiến trình hoạt động đồng thời với nhau. Các hệ điều hành đa nhiệm đã cài đặt thành công mục tiêu này. Như chúng ta đã thấy trong môi trường Windows 9x/ NT có thể có nhiều tiến trình/ nhiều người sử dụng khác nhau đồng thời sử dụng một máy in.

Ngoài ra hệ điều hành còn phải giải quyết vấn đề tranh chấp tài nguyên giữa các tiến trình đồng thời khi yêu cầu phục vụ của các tiến trình này vượt quá khả năng cấp phát của một tài nguyên kể cả đó là tài nguyên phân chia được.

Lời gọi hệ thống (System Calls)

Để tạo môi trường giao tiếp giữa chương trình của người sử dụng và hệ điều hành, hệ điều hành đưa ra các lời gọi hệ thống. Chương trình của người sử dụng dùng các lời gọi hệ thống để liên lạc với hệ điều hành và yêu cầu các dịch vụ từ hệ điều hành.

Mỗi lời gọi hệ thống tương ứng với một thủ tục trong thư viện của hệ điều hành, do đó chương trình của người sử dụng có thể gọi thủ tục để thực hiện một lời gọi hệ thống. Lời gọi hệ thống còn được thiết dưới dạng các câu lệnh trong các ngôn ngữ lập trình cấp thấp. Lệnh gọi ngắt trong hợp ngữ (Int), và thủ tục gọi hàm API trong windows được xem là một lời gọi hệ thống.

Lời gọi hệ thống có thể được chia thành các loại: quản lý tiến trình, thao tác trên tập tin, thao tác trên thiết bị vào/ ra, thông tin liên tiến trình, ...

Sau đây là một số lời gọi hệ thống của hệ điều hành MS_DOS:

- S = Load_and_exec(processname): tạo tiến trình con và thực hiện nó.
- Fd = Open(filename, mode): mở file để đọc hoặc/và ghi.
- N = Write(Fd, buffer, nbyte): ghi dữ liệu từ đệm vào file.
- Addr = alloc_memory(nbyte): cấp phát một khối nhớ
- Keep_pro(mem_size, status): kết thúc và thường trú chương trình.

Chú ý: Cần phải phân biệt sự khác nhau giữa Shell và System Call. Shell tạo môi trường giao tiếp giữa người sử dụng và hệ điều hành, System Call tạo môi trường

giao tiếp giữa chương trình người sử dụng và hệ điều hành.

1.2. Các chức năng của hệ điều hành

Một hệ thống máy tính gồm 3 thành phần chính: phần cứng, hệ điều hành và các chương trình ứng dụng và người sử dụng. Trong đó hệ điều hành là một bộ phận quan trọng và không thể thiếu của hệ thống máy tính, nhờ có hệ điều hành mà người sử dụng có thể đối thoại và khai thác được các chức năng của phần cứng máy tính.

Có thể nói hệ điều hành là một hệ thống các chương trình đóng vai trò trung gian giữa người sử dụng và phần cứng máy tính. Mục tiêu chính của nó là cung cấp một môi trường thuận lợi để người sử dụng dễ dàng thực hiện các chương trình ứng dụng của họ trên máy tính và khai thác triệt để các chức năng của phần cứng máy tính.

Để đạt được mục tiêu trên hệ điều hành phải thực hiện 2 chức năng chính sau đây:

Giả lập một máy tính mở rộng: Máy tính là một thiết bị vi điện tử, nó được cấu thành từ các bộ phận như: Processor, Memory, I/O Device, Bus, ... , do đó để đối thoại hoặc khai thác máy tính người sử dụng phải hiểu được cơ chế hoạt động của các bộ phận này và phải tác động trực tiếp vào nó, tất nhiên là bằng những con số 0,1 (ngôn ngữ máy). Điều này là quá khó đối với người sử dụng. Để đơn giản cho người sử dụng hệ điều hành phải che đậy các chi tiết phần cứng máy tính bởi một máy tính mở rộng, máy tính mở rộng này có đầy đủ các chức năng của một máy tính thực nhưng đơn giản và dễ sử dụng hơn. Theo đó khi cần tác động vào máy tính thực người sử dụng chỉ cần tác động vào máy tính mở rộng, mọi sự chuyển đổi thông tin điều khiển từ máy tính mở rộng sang máy tính thực hoặc ngược lại đều do hệ điều hành thực hiện. Mục đích của chức năng này là: *Giúp người sử dụng khai thác các chức năng của phần cứng máy tính dễ dàng và hiệu quả hơn.*

Quản lý tài nguyên của hệ thống: Tài nguyên hệ thống có thể là: processor, memory, I/O device, printer, file, ..., đây là những tài nguyên mà hệ điều hành dùng để cấp phát cho các tiến trình, chương trình trong quá trình điều khiển sự hoạt động của hệ thống. Khi người sử dụng cần thực hiện một chương trình hay khi một chương trình cần nạp thêm một tiến trình mới vào bộ nhớ thì hệ điều hành phải cấp phát không gian nhớ cho chương trình, tiến trình đó để chương trình, tiến trình đó nạp được vào bộ nhớ và hoạt động được. Trong môi trường hệ điều hành đa nhiệm có thể có nhiều chương trình, tiến trình đồng thời cần được nạp vào bộ nhớ, nhưng không gian lưu trữ của bộ nhớ có giới hạn, do đó hệ điều hành phải tổ chức cấp phát bộ nhớ sao cho hợp lý để đảm bảo tất cả các chương trình, tiến trình khi cần đều được nạp vào bộ nhớ để hoạt động. Ngoài ra hệ điều hành còn phải tổ chức bảo vệ các không gian nhớ

đã cấp cho các chương trình, tiến trình để tránh sự truy cập bất hợp lệ và sự tranh chấp bộ nhớ giữa các chương trình, tiến trình, đặc biệt là các tiến trình đồng thời hoạt động trên hệ thống. Đây là một trong những nhiệm vụ quan trọng của hệ điều hành.

Trong quá trình hoạt động của hệ thống, đặc biệt là các hệ thống đa người dùng, đa chương trình, đa tiến trình, còn xuất hiện một hiện tượng khác, đó là nhiều chương trình, tiến trình đồng thời sử dụng một không gian nhớ hay một tập tin (dữ liệu, chương trình) nào đó. Trong trường hợp này hệ điều hành phải tổ chức việc chia sẻ và giám sát việc truy xuất đồng thời trên các tài nguyên nói trên sao cho việc sử dụng tài nguyên có hiệu quả nhưng tránh được sự mất mát dữ liệu và làm hỏng các tập tin.

Trên đây là hai dẫn chứng điển hình để chúng ta thấy vai trò của hệ điều hành trong việc quản lý tài nguyên hệ thống, sau này chúng ta sẽ thấy việc cấp phát, chia sẻ, bảo vệ tài nguyên của hệ điều hành là một trong những công việc khó khăn và phức tạp nhất. Hệ điều hành đã chi phí nhiều cho công việc nói trên để đạt được mục tiêu: *Trong mọi trường hợp tất cả các chương trình, tiến trình nếu cần được cấp phát tài nguyên để hoạt động thì sớm hay muộn nó đều được cấp phát và được đưa vào trạng thái hoạt động.*

Trên đây là hai chức năng tổng quát của một hệ điều hành, đó cũng được xem như là các mục tiêu mà các nhà thiết kế, cài đặt hệ điều hành phải hướng tới. Các hệ điều hành hiện nay có các chức năng cụ thể sau đây:

- Hệ điều hành cho phép thực hiện nhiều chương trình đồng thời trong môi trường đa tác vụ - **Multitasking Environment**. Hệ điều hành multitasking bao gồm: Windows NT, Windows 2000, Linux và OS/2. Trong hệ thống multitasking hệ điều hành phải xác định khi nào thì một ứng dụng được chạy và mỗi ứng dụng được chạy trong khoảng thời gian bao lâu thì phải dừng lại để cho các ứng dụng khác được chạy.

- Hệ điều hành tự nạp nó vào bộ nhớ - **It loads itself into memory**: Quá trình nạp hệ điều hành vào bộ nhớ được gọi là quá trình **Bootting**. Chỉ khi nào hệ điều hành đã được nạp vào bộ nhớ thì nó mới cho phép người sử dụng giao tiếp với phần cứng. Trong các hệ thống có nhiều ứng dụng đồng thời hoạt động trên bộ nhớ thì hệ điều hành phải chịu trách nhiệm chia sẻ không gian bộ nhớ RAM và bộ nhớ cache cho các ứng dụng này.

- **Hệ điều hành và API: Application Programming Interface**: API là một tập các hàm/thủ tục được xây dựng sẵn bên trong hệ thống, nó có thể thực hiện được nhiều chức năng khác nhau như shutdown hệ thống, đảo ngược hiệu ứng màn hình, khởi động các ứng dụng, ... Hệ điều hành giúp cho chương trình của người sử dụng giao tiếp với API hay thực hiện một lời gọi đến các hàm/thủ tục của API.

Nạp dữ liệu cần thiết vào bộ nhớ - **It loads the required data into memory**: Dữ liệu do người sử dụng cung cấp được đưa vào bộ nhớ để xử lý. Khi nạp dữ liệu vào bộ nhớ hệ điều hành phải lưu lại địa chỉ của bộ nhớ nơi mà dữ liệu được lưu ở đó. Hệ điều hành phải luôn theo dõi bản đồ cấp phát bộ nhớ, nơi dữ liệu và chương trình được lưu trữ ở đó. Khi một chương trình cần đọc dữ liệu, hệ điều hành sẽ đến các địa chỉ bộ nhớ nơi đang lưu trữ dữ liệu mà chương trình cần đọc để đọc lại nó.

- Hệ điều hành biên dịch các chỉ thị chương trình - **It interprets program instructions**: Hệ điều hành phải đọc và giải mã các thao tác cần được thực hiện, nó được viết trong chương trình của người sử dụng. Hệ điều hành cũng chịu trách nhiệm sinh ra thông báo lỗi khi hệ thống gặp lỗi trong khi đang hoạt động.

Hệ điều hành quản lý tài nguyên - **It manages resources**: Nó đảm bảo việc sử dụng thích hợp tất cả các tài nguyên của hệ thống như là: bộ nhớ, đĩa cứng, máy in, ...

1.3. Các thành phần và cấu trúc của hệ điều hành

Hệ điều hành là một hệ thống chương trình lớn, thực hiện nhiều nhiệm vụ khác nhau, do đó các nhà thiết kế thường chia hệ điều hành thành nhiều thành phần, mỗi thành phần đảm nhận một nhóm các nhiệm vụ nào đó, các nhiệm vụ này có liên quan với nhau. Cách phân chia nhiệm vụ cho mỗi thành phần, cách kết nối các thành phần lại với nhau để nó thực hiện được một nhiệm vụ lớn hơn khi cần và cách gọi các thành phần này khi cần nó thực hiện một nhiệm vụ nào đó, ... , tất cả các phương thức trên tạo nên cấu trúc của hệ điều hành.

1.3.1. Các thành phần của hệ điều hành

a. Thành phần quản lý tiến trình

Hệ điều hành phải có nhiệm vụ tạo lập tiến trình và đưa nó vào danh sách quản lý tiến trình của hệ thống. Khi tiến trình kết thúc hệ điều hành phải loại bỏ tiến trình ra khỏi danh sách quản lý tiến trình của hệ thống.

Hệ điều hành phải cung cấp đầy đủ tài nguyên để tiến trình đi vào hoạt động và phải đảm bảo đủ tài nguyên để duy trì sự hoạt động của tiến trình cho đến khi tiến trình kết thúc. Khi tiến trình kết thúc hệ điều hành phải thu hồi những tài nguyên mà hệ điều hành đã cấp cho tiến trình.

Trong quá trình hoạt động nếu vì một lý do nào đó tiến trình không thể tiếp tục hoạt động được thì hệ điều hành phải tạm dừng tiến trình, thu hồi tài nguyên mà tiến

trình đang chiếm giữ, sau đó nếu điều kiện thuận lợi thì hệ điều hành phải tái kích hoạt tiến trình để tiến trình tiếp tục hoạt động cho đến khi kết thúc.

Trong các hệ thống có nhiều tiến trình hoạt động song song hệ điều hành phải giải quyết vấn đề tranh chấp tài nguyên giữa các tiến trình, điều phối processor cho các tiến trình, giúp các tiến trình trao đổi thông tin và hoạt động đồng bộ với nhau, đảm bảo nguyên tắc tất cả các tiến trình đã được khởi tạo phải được thực hiện và kết thúc được.

Tóm lại, bộ phận quản lý tiến trình của hệ điều hành phải thực hiện những nhiệm vụ sau đây:

- Tạo lập, hủy bỏ tiến trình.
- Tạm dừng, tái kích hoạt tiến trình.
- Tạo cơ chế thông tin liên lạc giữa các tiến trình.
- Tạo cơ chế đồng bộ hóa giữa các tiến trình.

b. Thành phần quản lý bộ nhớ chính

Bộ nhớ chính là một trong những tài nguyên quan trọng của hệ thống, đây là thiết bị lưu trữ duy nhất mà CPU có thể truy xuất trực tiếp được.

Các chương trình của người sử dụng muốn thực hiện được bởi CPU thì trước hết nó phải được hệ điều hành nạp vào bộ nhớ chính, chuyển đổi các địa chỉ sử dụng trong chương trình thành những địa chỉ mà CPU có thể truy xuất được.

Khi chương trình, tiến trình có yêu cầu được nạp vào bộ nhớ thì hệ điều hành phải cấp phát không gian nhớ cho nó. Khi chương trình, tiến trình kết thúc thì hệ điều hành phải thu hồi lại không gian nhớ đã cấp phát cho chương trình, tiến trình trước đó.

Trong các hệ thống đa chương hay đa tiến trình, trong bộ nhớ tồn tại nhiều chương trình/ nhiều tiến trình, hệ điều hành phải thực hiện nhiệm vụ bảo vệ các vùng nhớ đã cấp phát cho các chương trình/ tiến trình, tránh sự vi phạm trên các vùng nhớ của nhau.

Tóm lại, bộ phận quản lý bộ nhớ chính của hệ điều hành thực hiện những nhiệm vụ sau:

- Cấp phát, thu hồi vùng nhớ.
- Ghi nhận trạng thái bộ nhớ chính.
- Bảo vệ bộ nhớ.
- Quyết định tiến trình nào được nạp vào bộ nhớ.

c. Thành phần quản lý xuất/ nhập