

Nội dung

Tổng quan

Quản lí Tiến trình

Quản lí Bộ nhớ

Quản lí Thiết bị

Hệ điều hành nhiều Processor

Tài liệu tham khảo

- [1]. Hồ Đức Phương, *Giáo trình Nguyên lý hệ điều hành*, NXB Giáo dục, 2011.
- [2]. Nguyễn Thanh Tùng, *Giáo trình Hệ điều hành*, ĐH Bách Khoa HN.
- [3]. Hà Quang Thụy, *Giáo trình nguyên lý các Hệ điều hành*, NXB KHKT, 2009.
- [4]. Trần Hồ Thủy Tiên, *Giáo trình Nguyên lý hệ điều hành*, Đại học Đà Nẵng, 2007.
- [5]. Milan Milenkovic, *Operating systems concept and design*, Tata Mcgraw Hill, 2001.
- [6]. Achyut Godbole, *Operating system*, Mc Graw Hill, 2010.
- [7]. Andrew S. Tanenbaum, *Modern Operating system*, Prentice Hall.

Tổng quan

- Tổng quan
- Định nghĩa OS
- Kiến trúc OS
- Phân loại OS
- Tính chất của OS
- Các thành phần cơ bản của OS
- Các nguyên tắc xây dựng OS



Tổng quan

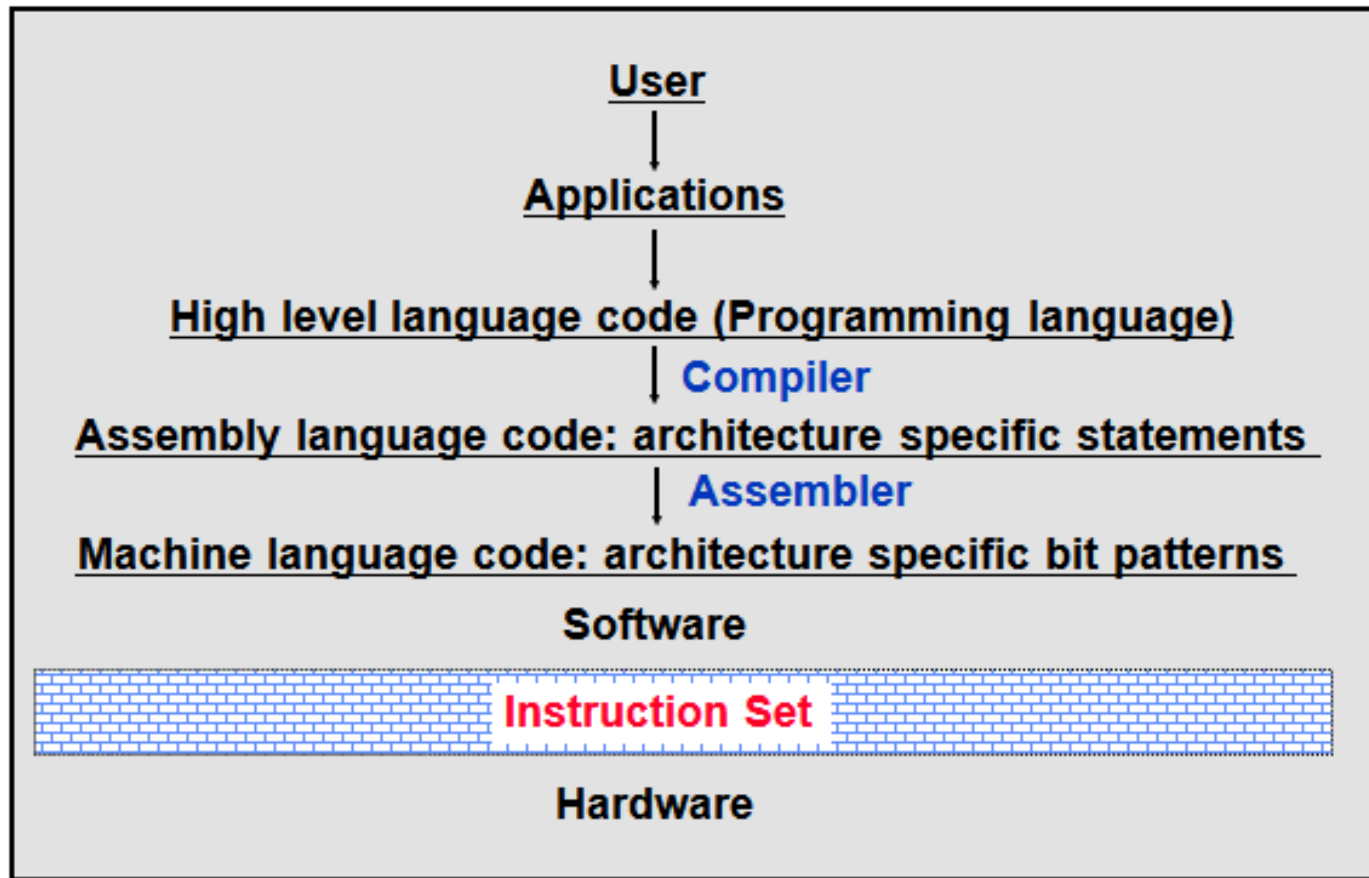
■ Tổng quan

- Máy tính (computer): thiết bị điện tử có khả năng thao tác (lưu trữ, xử lý) trên các thông tin (dữ liệu).
- Các thao tác: thực hiện theo một chương trình - một dãy các câu lệnh



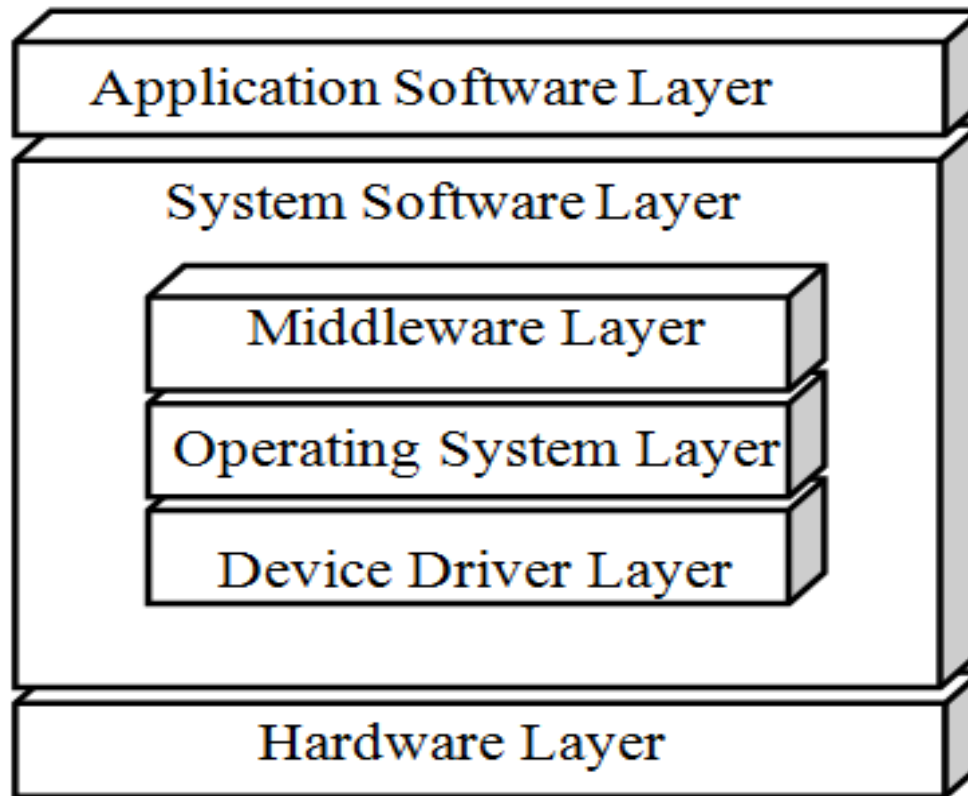
Tổng quan

- Giao tiếp người - máy



Tổng quan

- Giao tiếp người - máy



Tổng quan

- Chương trình ứng dụng: qui định cách sử dụng các tài nguyên hệ thống để giải quyết những vấn đề của người sử dụng
- Hệ điều hành: cung cấp môi trường làm việc (giao diện) cho người dùng, các phần mềm hoạt động trên hệ thống; quản lý các tài nguyên phần cứng và phần mềm đảm bảo cho hệ thống hoạt động một cách hiệu quả, tin cậy
- Phần cứng: CPU, bộ nhớ, thiết bị vào/ra..

Tổng quan

■ Giao diện

- Cơ chế Lệnh: người dùng phải tự nắm bắt trước các công việc mà hệ thống có thể làm được, qua đó chỉ thị cho hệ thống làm việc
- Cơ chế Menu:
 - Hệ thống giới thiệu cho người dùng các khả năng phục vụ của mình dưới dạng các bảng chọn, người dùng chỉ chờ cho hệ thống trình bày danh mục các công việc và lựa chọn công việc có thể yêu cầu
 - Các công việc được phân nhóm theo từng phạm trù để dễ tìm kiếm
 - Hệ thống mang tính chất tự đào tạo: càng làm việc càng hiểu sâu hơn

Tổng quan

- Quản lí tài nguyên:
 - Quản lý tiến trình (Process): cung cấp các cơ chế quản lý các phần mềm đang được thực hiện trong hệ thống
 - Quản lý bộ nhớ (Memory): không gian nhớ trong hệ thống được chia sẻ bởi tất cả các tiến trình → việc truy nhập và phân bổ các phần tử nhớ phải được quản lý chặt chẽ
 - Quản lý hệ thống vào/ra (I/O System): các thiết bị/vào ra được chia sẻ bởi nhiều tiến trình → phải được quản lý (quản lý việc truy nhập, truyền thông, file)

Tổng quan

- Tài nguyên hệ thống
 - Thời gian
 - Không gian
 - Thiết bị ngoại vi

Tổng quan

■ Thời gian

- Processor (P): một tài nguyên quan trọng của hệ thống, được truy nhập ở mức câu lệnh và chỉ có nó mới làm cho câu lệnh được thực hiện
- P được dùng cho nhiều tiến trình khác nhau (P_{logic}) do đó việc phân chia thời gian sử dụng P của mỗi tiến trình phải được tối ưu hoá, đặc biệt là khi chúng còn dùng chung tài nguyên khác: chương trình, dữ liệu, thiết bị vào ra...
- Thời gian: thời gian thực hiện một câu lệnh

Tổng quan

- Không gian
 - Không gian thực hiện lệnh - Bộ nhớ logic.
 - Không gian lưu trữ - Bộ nhớ vật lí
 - Phân cấp Hệ thống nhớ

Tổng quan

- Thiết bị ngoại vi
 - Số lượng nhiều
 - Chủng loại đa dạng
 - Tốc độ xử lý \ll tốc độ P

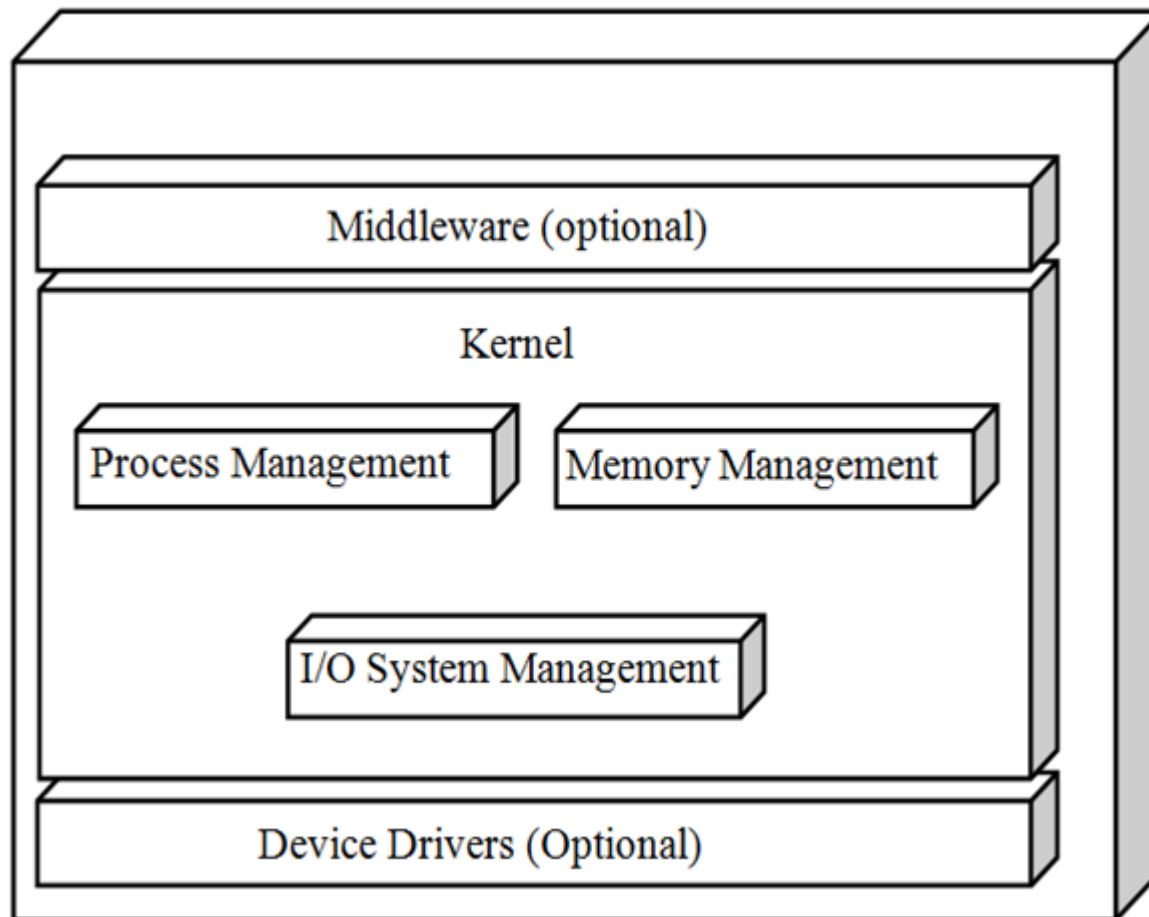
Tổng quan

■ Định nghĩa OS:

- Với User: OS là hệ thống chương trình tạo điều kiện để khai thác tài nguyên hệ thống tính toán một cách dễ dàng, thuận tiện
- Với Nhà quản lí: OS là tập các chương trình phục vụ quản lý chặt chẽ và sử dụng tối ưu các tài nguyên hệ thống
- Với cán bộ kỹ thuật: OS là hệ thống chương trình trang bị cho một máy tính cụ thể mức vật lý để tạo ra một máy logic mới với các tài nguyên và khả năng mới
- Với cán bộ lập trình hệ thống: OS là một hệ thống mô hình hoá mô phỏng các hoạt động của máy, của người dùng và của thao tác viên hoạt động trong chế độ đối thoại nhằm tạo môi trường khai thác thuận tiện và quản lý tối ưu các tài nguyên của hệ thống tính toán

Tổng quan

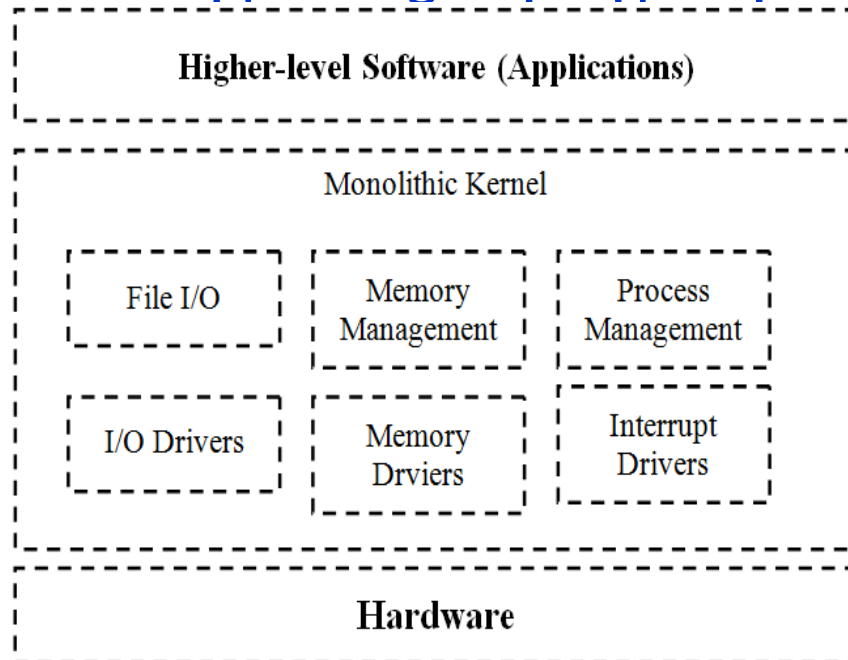
■ Mô hình kiến trúc OS



Tổng quan

■ Kiến trúc Monolithic

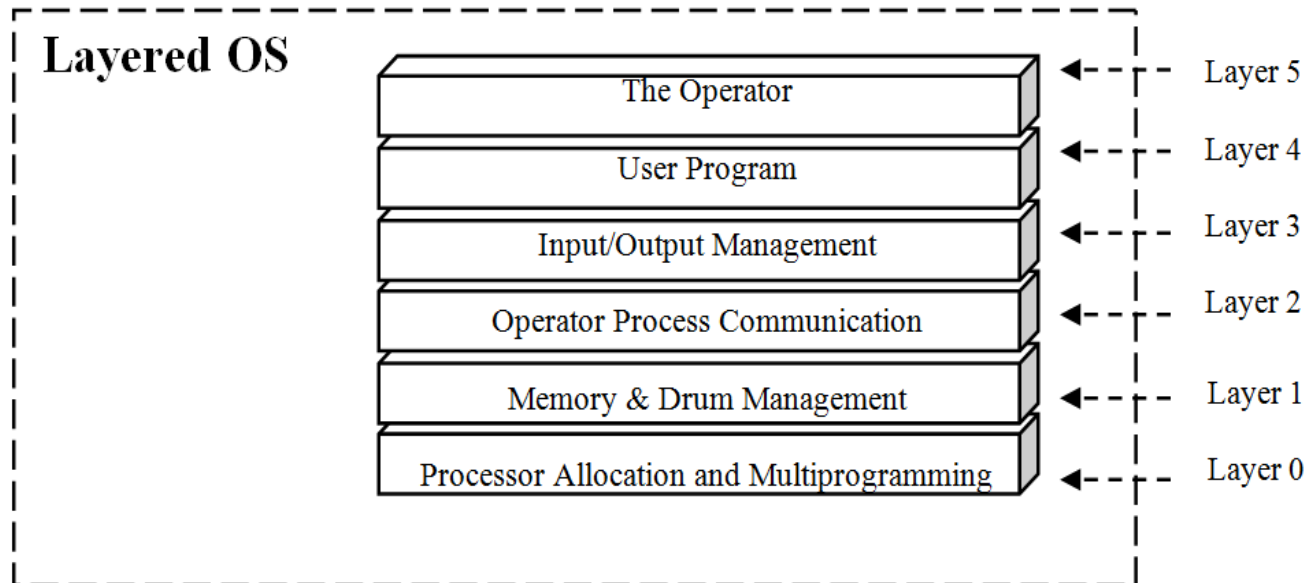
- Phần mềm trung gian (Middleware) và trình điều khiển thiết bị được tích hợp trong nhân của hệ điều hành
- Các chức năng được phân tích theo mô đun, các mô đun được tích hợp trong một tập thực thi duy nhất



Tổng quan

■ Kiến trúc Layer

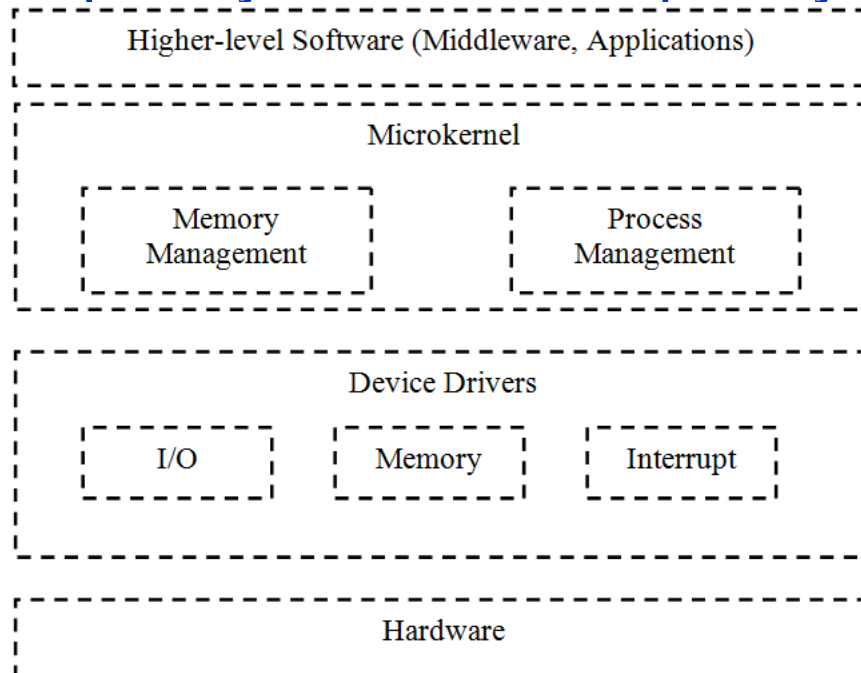
- Hệ điều hành được chia thành nhiều lớp, các lớp trên được đáp ứng bởi dịch vụ của các lớp thấp hơn
- Phần mềm trung gian và trình điều khiển thiết bị được tích hợp trong nhân của hệ điều hành theo một file duy nhất



Tổng quan

■ Kiến trúc Microkernel

- Phần mềm trung gian và trình điều khiển thiết bị được tách biệt khỏi hệ điều hành
- Nhân của hệ điều hành chỉ bao gồm các thành phần chính là quản lý tiến trình và quản lý bộ nhớ



Tổng quan

■ Phân loại OS:

- OS đơn nhiệm và OS đa nhiệm: dựa vào cách thức đưa chương trình vào bộ nhớ
 - OS đơn nhiệm: tại một thời điểm xác định, chỉ duy nhất một chương trình được thực thi
 - OS đa nhiệm: tại một thời điểm có nhiều chương trình được thực thi, các tài nguyên được chia sẻ cho các chương trình
- OS đơn chương và đa chương (MultiUsers)
 - OS đơn chương: tại một thời điểm xác định, OS chỉ cho phép một người sử dụng thao tác với hệ thống
 - OS đa chương: tại một thời điểm có thể phục vụ nhiều người sử dụng

Tổng quan

■ Phân loại OS:

- OS tập trung và phân tán
 - OS tập trung: trên hệ thống chỉ có một OS duy nhất cài đặt tại máy chủ. Các máy trạm được khởi động nhờ máy chủ và nó chỉ làm chức năng nhập/xuất dữ liệu. Mọi xử lý đều tập trung ở máy chủ
 - OS phân tán: trên mỗi máy tồn tại một OS, máy chủ chịu trách nhiệm cung ứng các dịch vụ để truy nhập đến các tài nguyên chung và điều hành toàn hệ thống, các phép xử lý có thể tiến hành ở máy trạm
- OS phân chia thời gian và OS thời gian thực
 - OS phân chia thời gian (Share time): một CPU luôn phiên phục vụ các tiến trình và 1 tiến trình có thể rơi vào trạng thái chờ đợi khi chưa được phân phối CPU
 - OS thời gian thực (Real time): một tiến trình khi đã xâm nhập vào hệ thống thì ở bất kỳ lúc nào đều được phân phối CPU

Tổng quan

■ Tính chất của OS:

- Độ tin cậy cao: mọi hoạt động thông báo của hệ điều hành chuẩn xác tuyệt đối, khi chắc chắn đúng thì máy mới cung cấp thông tin cho người dùng và mọi công việc bao giờ cũng được kiểm tra, đánh giá
- Độ an toàn: tổ chức cho dữ liệu và chương trình không bị xoá hoặc thay đổi ngoài ý muốn; chức năng bảo vệ thông tin được chia thành nhiều mức
- Hiệu quả: các tài nguyên phải được khai thác triệt để ngay cả khi điều kiện tài nguyên hạn chế song vẫn có thể giải quyết các yêu cầu phức tạp
- Tổng quát: kế thừa các phiên bản trước đây, thích nghi với những thay đổi có thể có trong tương lai
- Thuận tiện: dễ sử dụng, có nhiều mức hiệu quả khác nhau tùy kinh nghiệm và kiến thức người dùng

Tổng quan

■ Thành phần của OS:

- Ngôn ngữ làm việc và giao tiếp: tồn tại ba ngôn ngữ làm việc và giao tiếp
 - Ngôn ngữ máy (ngôn ngữ thực hiện): ngôn ngữ thực hiện duy nhất của hệ thống, mọi ngôn ngữ khác đều phải được ánh xạ sang ngôn ngữ thực hiện
 - Ngôn ngữ vận hành (hệ điều hành): thao tác viên giao tiếp với hệ thống
 - Ngôn ngữ thuật toán: người dùng giao tiếp với hệ thống: Pascal, C...
- Các Module chương trình của hệ thống:
 - Chương trình điều khiển: quản lý tài nguyên, tiến trình, dữ liệu và các chương trình thư ký, điều phối nhiệm vụ
 - Chương trình phục vụ: biên tập, biên dịch chương trình

Tổng quan

■ Nguyên tắc xây dựng OS:

- Module: xây dựng từ các module độc lập quan hệ với nhau thông qua dữ liệu vào/ra → tồn tại cơ chế liên kết các module độc lập thành hệ thống có tổ chức
- Nguyên tắc tương đối trong định vị: các module được viết theo địa chỉ tương đối kể từ đầu bộ nhớ, khi thực hiện chúng được định vị tại vùng nhớ cụ thể như vậy hệ thống sử dụng bộ nhớ linh hoạt hơn và hệ điều hành không phụ thuộc vào cấu hình bộ nhớ
- Macroprocessor: khi có một công việc cụ thể, hệ thống sẽ xây dựng các phiếu yêu cầu, liệt kê các bước phải thực hiện → xây dựng và thực hiện chương trình tương ứng
- Phủ chức năng: một công việc có thể được thực hiện bằng nhiều phương tiện khác nhau cho phép người dùng chọn giải pháp tối ưu với bài toán của mình

Tổng quan

- Nguyên tắc xây dựng OS:
 - Giá trị chuẩn (ngầm định): hệ thống chuẩn bị sẵn các bảng giá trị cho các tham số điều khiển → nếu trong các câu lệnh của người dùng còn thiếu những tham số giá trị thì hệ thống sẽ tự động lấy giá trị tương ứng ở bảng giá trị chuẩn ra để thực hiện
 - Bảo vệ dữ liệu và chương trình: dữ liệu và chương trình của người sử dụng phải được bảo vệ nhiều mức, bằng nhiều khoá

Quản lí Tiến trình

- Tổng quan
- Quản lí Tiến trình
- Quản lí Processor



Quản lí Tiến trình

Tiến trình (Process):

- Saltzer: Tiến trình là chương trình do một processor logic thực hiện
- Dijkstra: Tiến trình là những gì liên quan đến hệ thống tính toán xuất hiện khi thực hiện một chương trình
- Định nghĩa của Horning & Randell: Tiến trình như một quá trình chuyển từ trạng thái này sang trạng thái khác dưới tác động của hàm hành động và xuất phát từ trạng thái ban đầu nào đó
- Quan điểm của người dùng: Tiến trình là quá trình thực hiện chương trình: luật hướng dẫn Processor thực hiện

Quản lí Tiến trình

Tổ chức tiến trình: cấu trúc thông tin

- Biến trạng thái: trạng thái hiện tại của tiến trình
- Vùng bộ nhớ lưu trữ giá trị của các thanh ghi tiến trình sử dụng
- Thông tin về tài nguyên tiến trình đang sử dụng hoặc có quyền sử dụng

Phân loại tiến trình:

- Tuần tự: một tiến trình chỉ bắt đầu sau khi tiến trình kia kết thúc
- Song song: thời điểm bắt đầu của tiến trình này nằm giữa thời điểm bắt đầu và kết thúc của một tiến trình khác

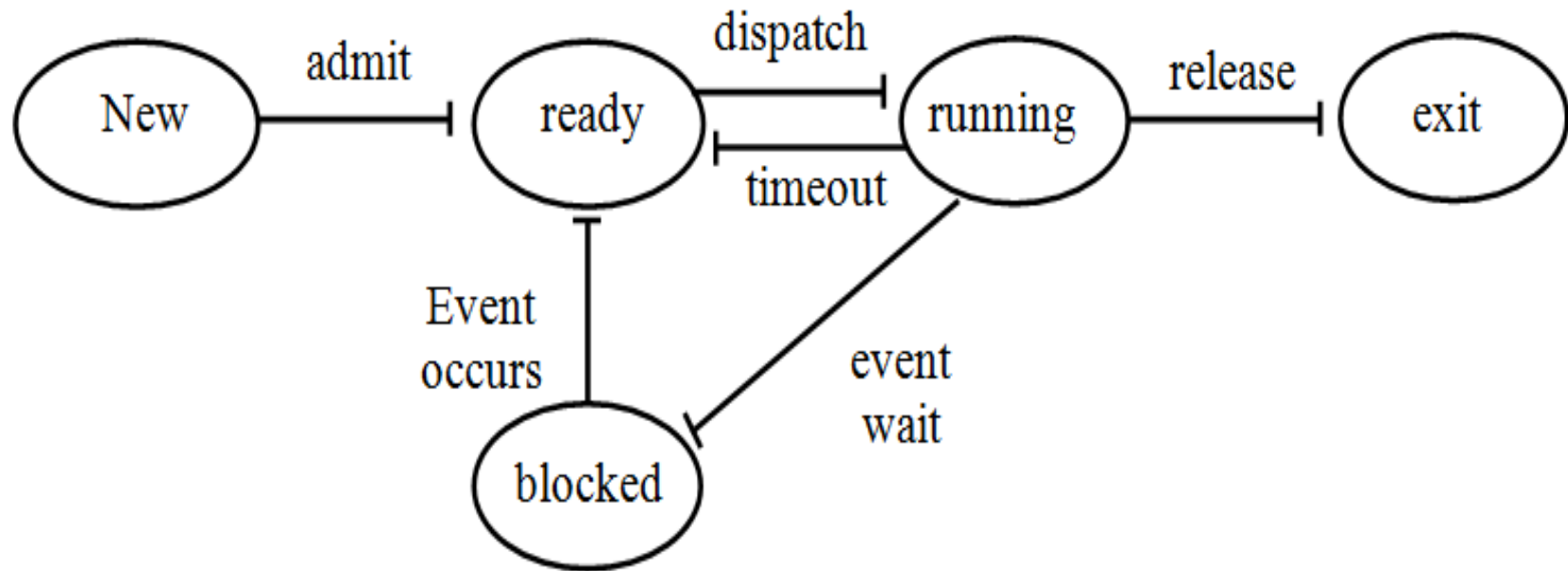
Quản lí Tiến trình

Quan hệ giữa hai tiến trình

- Độc lập: 2 tiến trình không có quan hệ trực tiếp gì với nhau → bảo vệ thông tin sao cho một tiến trình không làm hỏng dữ liệu và chương trình của tiến trình khác
- Tiến trình trao đổi thông tin: một tiến trình có thể gửi thông báo cho tiến trình khác, tổ chức các vùng nhớ làm hòm thư
- Phân lớp: trong quá trình hoạt động của một tiến trình có thể khởi tạo một tiến trình khác hoạt động song song
- Tiến trình đồng mức: những tiến trình có một số tài nguyên sử dụng chung theo nguyên tắc lần lượt

Quản lí Tiến trình

■ Trạng thái tiến trình



Quản lí Tiến trình

- **Trạng thái tiến trình**
 - **Sẵn sàng (Ready):** tiến trình được phân phối đầy đủ tài nguyên và chờ để được xử lý
 - **Thực hiện (Running):** tiến trình dành được vi xử lý để thực hiện
 - **Ngắt (Blocked/Waiting):** tiến trình không thể thực hiện tiếp vì bị thiếu một vài điều kiện nào đó, tiến trình sẽ nằm trong trạng thái ngắt. Nếu tiến trình gọi tới một module nhưng module chưa được nạp và định vị trong bộ nhớ, khi đó tiến trình rơi vào trạng thái chờ (wait)

Quản lí Tiến trình

Quản lí tiến trình: tiến trình chỉ được thực hiện khi nó được phân phối đầy đủ tài nguyên và có quyền sử dụng Processor → Phân phối tài nguyên và Processor cho các tiến trình

- Phân phối tài nguyên cho các tiến trình
 - Điều độ tiến trình qua đoạn Găng
 - Phòng tránh tắc nghẽn
- Phân phối Processor cho các tiến trình
 - Phân phối Processor cho các tiến trình
 - Điều độ tiến trình

Quản lí Tiến trình

- Phân phối tài nguyên cho các tiến trình - Điều độ tiến trình qua đoạn Găng
 - Tài nguyên Găng: tài nguyên phân phối cho một tiến trình
 - Đoạn chương trình có sử dụng tài nguyên Găng: đoạn Găng
 - Điều độ tiến trình qua đoạn Găng:
 - Tại một thời điểm có không quá một tiến trình nằm trong đoạn Găng
 - Không một tiến trình nào được phép ở lâu vô hạn trong đoạn Găng
 - Không một tiến trình nào phải chờ vô hạn ngoài đoạn Găng

Quản lí Tiến trình

■ Điều độ tiến trình qua đoạn Găng

Ví dụ:

TTA ghi nội dung biến Dem vào TgA (biến cục bộ)

TTB ghi nội dung biến Dem vào TgB

TTA tăng TgA

TTB tăng TgB

cả 2 tiến trình *A* và *B* đều tăng nội dung *Dem*, song nội dung này chỉ tăng 1 → *Dem*: tài nguyên Găng, đoạn chương trình: đoạn Găng

Quản lí Tiến trình

- Điều độ tiến trình qua đoạn Găng
 - Công cụ điều độ cấp thấp: nằm ngoài tiến trình được điều độ
 - Phương pháp khoá trong
 - Phương pháp kiểm tra và xác lập (Perterson)
 - Kỹ thuật đèn báo (Semaphore)
 - Cấp cao: nằm trong tiến trình
 - Chương trình thư kí (Monitor)

Quản lí Tiến trình

- Điều độ tiến trình qua đoạn Găng
 - Phương pháp khoá trong
 - Hai hay nhiều tiến trình cùng định ghi vào một địa chỉ nào đó của bộ nhớ trong → chỉ cho phép một tiến trình làm việc còn tiến trình khác phải chờ
 - Mỗi tiến trình: sử dụng một byte trong vùng bộ nhớ chung làm khoá, khi vào được đoạn găng → khóa = 1 → thông báo cho các tiến trình khác biết đã có tiến trình sử dụng tài nguyên găng

Quản lí Tiến trình

- Điều độ tiến trình qua đoạn Găng
 - Phương pháp khoá trong: Giải thuật Delker

```

{
  k1= k2= 0; tg=1; kt1= kt2 =1;
  do
  {
    k1=1;
    while (k2==1) do Ct2
    if (tg=2)
    {
      k1= 0;
      while (tg==2) do Ct2
      k1=1;
    };
    kt1=0; tg=1;
  }
  while (kt1==0);

```

...

Quản lí Tiến trình

- Điều độ tiến trình qua đoạn Găng
 - Kiểm tra và xác lập (phương pháp PERTerson)
 - Sử dụng các giá trị kiểm tra là các biến trạng thái: tham số (cục bộ - L, toàn cục - G)
 - Sử dụng thủ tục PAR: gán $L = G$ và gán $G = 1$

Quản lí Tiến trình

■ Điều độ tiến trình qua đoạn Găng

➤ Kiểm tra và xác lập

```
{  
  G=0;  
  { TT=1;  
    do  
    {  
      L1=1;  
      while (L1==1) do PAR(L1);  
      //đoạn giữa tiến trình 1  
      G=0;  
      phần còn lại của tiến trình 1  
    }  
    while false  
  }  
  ...  
}
```

Quản lí Tiến trình

- Điều độ tiến trình qua đoạn Găng
 - Kỹ thuật đèn báo (Semaphore)
 - Sử dụng biến đèn báo nguyên đặc biệt: s , ban đầu s nhận một giá trị bằng khả năng phục vụ của tài nguyên găng.
 - Sử dụng các phép $P(s)$ và $V(s)$ trên s :
 - $P(s)$:
 - Giảm S đi 1 đơn vị
 - Nếu $s \geq 0$ tiếp tục thực hiện tiến trình, ngược lại đưa tiến trình vào dòng xếp hàng
 - $V(s)$:
 - Tăng s lên 1
 - Nếu $s \leq 0$ kích hoạt một tiến trình ra hoạt động

Quản lí Tiến trình

■ Điều độ tiến trình qua đoạn Găng

➤ Kỹ thuật đèn báo

```
{
s=1;

    {
    tt:=1;
    do

        {
        P(s)
        //đoạn giữa tiến trình 1
        V(s);
        //phần còn lại của tiến trình 1
        }

    while false

...
}
```


Quản lí Tiến trình

■ Công cụ điều độ cấp cao

- Chương trình thư kí (Monitor): cấu trúc đặc biệt bao gồm các thủ tục, các biến và cấu trúc dữ liệu hoạt động trong chế độ phân chia thời gian, hỗ trợ việc thực hiện tiến trình, với các thuộc tính:
 - Các biến và cấu trúc dữ liệu trong Monitor chỉ có thể được thao tác bởi các thủ tục định nghĩa bên trong Monitor
 - Tại một thời điểm, một tiến trình duy nhất được làm việc với chương trình thư ký
 - Mỗi lần sử dụng tài nguyên mới, hệ thống gắn chương trình thư ký với tiến trình

Quản lí Tiến trình

■ Công cụ điều độ cấp cao

➤ Chương trình thư kí (Monitor):

Trong một Monitor có thể định nghĩa các biến điều kiện C và hai thao tác là Wait () và Signal ():

- Wait (C): chuyển trạng thái tiến trình sang trạng thái khoá và đặt tiến trình vào hàng đợi trên biến điều kiện C
- Signal (C): nếu có một tiến trình đang bị khoá trong hàng đợi của C thì tái kích hoạt tiến trình đó và tiến trình sẽ rời khỏi Monitor

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

- Khi có nhiều tài nguyên găng trong một tiến trình → các tiến trình chờ đợi lẫn nhau
- Hai hay nhiều tiến trình cùng chờ đợi một sự kiện, nếu không có tác động đặc biệt từ ngoài → chờ đợi vô hạn → tắc nghẽn
 - Phòng ngừa: tránh không để tiến trình rơi vào tình trạng tắc nghẽn
 - Dự báo và tránh: kiểm tra xem tiến trình có rơi vào tình trạng tắc nghẽn hay không → thông báo kịp thời trước khi tắc nghẽn xảy ra
 - Nhận biết và khắc phục: phát hiện các tiến trình bị tắc nghẽn và giải quyết

Quản lí Tiến trình

- Tắc nghẽn và điều độ Tiến trình
 - Phòng ngừa tắc nghẽn: xem xét các điều kiện tắc nghẽn:
 - Thiếu tài nguyên Găng
 - Chờ vô hạn khi chưa được vào đoạn Găng
 - Không có hệ thống phân phối lại tài nguyên
 - Tồn tại chờ đợi vòng

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

➤ Phòng ngừa:

- Khi kết thúc tiến trình, kết quả được chuyển ngược lại tài nguyên vật lý mà server yêu cầu, việc chuyển ngược này theo nguyên tắc lần lượt và do chương trình hệ thống đảm nhận như vậy không xảy ra xung đột
- Phân phối trước tài nguyên ➡ tiến trình chỉ được bắt đầu khi nhận đủ tài nguyên trong một số lần phân phối
- Tạo các điểm gác: hệ thống sẽ lưu lại toàn bộ thông tin trạng thái tiến trình, nếu cần thiết có thể hủy tiến trình, giải phóng tài nguyên, sau đó nếu cho phép sẽ tiếp tục công việc bằng cách khôi phục trạng thái cuối
- Phân lớp tài nguyên ➡ tiến trình chỉ nhận được tài nguyên mức cao hơn sau khi đã trả lại tài nguyên mức thấp

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

- Dự báo và phòng tránh: không phòng ngừa nhưng mỗi lần phân phối tài nguyên kiểm tra xem việc phân phối đó có khả năng đẩy hệ thống vào tình trạng tắc nghẽn không → đưa ra phương pháp giải quyết cụ thể trước khi tắc nghẽn có thể xảy ra
- Phương pháp:
 - n tiến trình
 - k tài nguyên
 - Tiến trình i yêu cầu tối đa một lúc $\max(i)$ tài nguyên để có thể thực hiện, nhưng hiện chỉ nhận được $f(i)$
 - Tiến trình i kết thúc $kt(i) = \text{true}$

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

➤ Dự báo và phòng tránh

```
t = k;  
for (i=0, i<= n, i++)  
{  
    t = t - f(i);  
    cl[i] = max[i];  
    kt[i] = 0;  
}  
flag = 1;
```

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

➤ Dự báo và phòng tránh

```
While (flag == 1)
```

```
{
```

```
    flag = 0
```

```
    for (i=0, i<= n, i++)
```

```
        if not(kt[i]) && (cl[i] <=t)
```

```
        {
```

```
            kt[i] = 1;
```

```
            t = t + cl[i]
```

```
            flag = 1;
```

```
        };
```

```
}
```

```
if (t==k) printf ("An toàn") else printf ("không an toàn")
```


Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

➤ Dự báo và phòng tránh: ví dụ

- Các tiến trình: P1, P2, P3, P4
- Các tài nguyên: R1, R2, R3
- Tổng các tài nguyên của hệ thống: $k = 9R1 + 3R2 + 6R3$
- Trạng thái hiện thời của các tiến trình:

Tiến trình	Max (i)			f (i)			t		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0	4	1	2
P2	6	1	3	2	1	1			
P3	3	1	4	2	1	1			
P4	4	2	2	0	0	2			

Giả sử P2 có yêu cầu 4R1 và 1R3, khi đó việc thoả mãn P2 có đẩy hệ thống tới tình trạng tắc nghẽn hay không?

Quản lí Tiến trình

■ Tắc nghẽn và điều độ Tiến trình

- Nhận biết và khắc phục: quan sát trạng thái các tiến trình đang chờ, xem những tiến trình bị rơi vào tắc nghẽn → tùy tình hình cụ thể áp dụng các biện pháp cần thiết
- Phát hiện tắc nghẽn:
 - Đình chỉ hoạt động của tiến trình liên quan → đưa tiến trình về trạng thái ngắt
 - Thu hồi tài nguyên

Quản lí Tiến trình

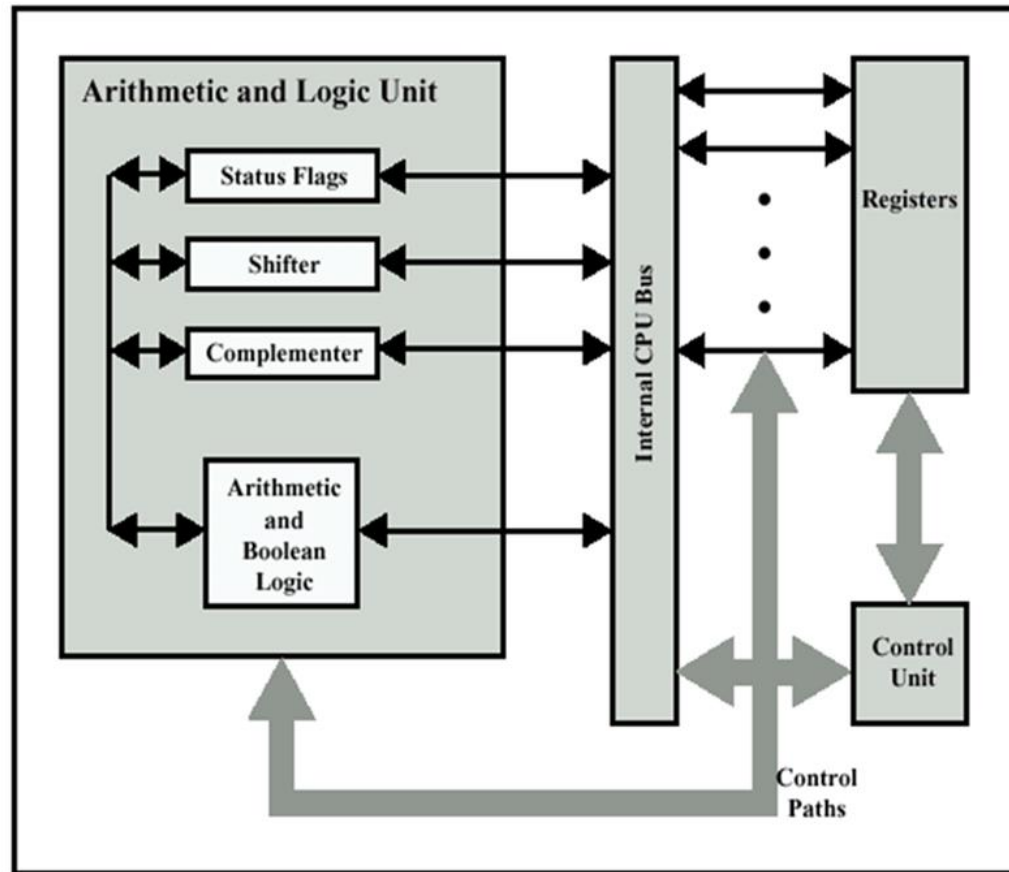
- **Tắc nghẽn và điều độ Tiến trình**
 - **Đưa tiến trình về trạng thái ngất**
 - Độ ưu tiên
 - Thời gian xử lý
 - Số lượng tài nguyên tiến trình đang chiếm dụng
 - Số lượng tài nguyên tiến trình yêu cầu
 - **Thu hồi tài nguyên: thu hồi tài nguyên của một số tiến trình và cấp phát các tài nguyên này cho tới khi loại bỏ được chu trình tắc nghẽn**
 - Lựa chọn tiến trình thu hồi, những tài nguyên nào bị thu hồi
 - Phục hồi trạng thái tiến trình ở trạng thái gần nhất trước đó mà không xảy ra tắc nghẽn
 - Tránh cho một tiến trình nào đó luôn bị thu hồi tài nguyên

Quản lí Processor

- Processor: tài nguyên phục vụ cho việc thực hiện chương trình
- Tiến trình: đối tượng được phân phối Processor để thực hiện
- Processor vật lý: tất cả các hệ điều hành thực hiện song song đều do một Processor của hệ thống - Processor vật lý điều khiển
- Processor logic: người sử dụng đánh giá hoạt động của Processor trên cơ sở quan sát và đánh giá chương trình của mình được thực hiện như thế nào - Processor logic

Quản lí Processor

Processor vật lí ~ Bộ xử lí trung tâm



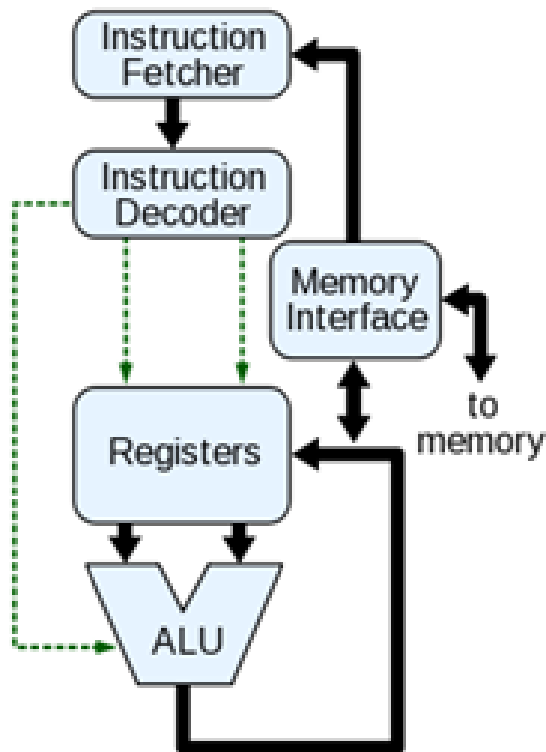
Quản lí Processor

Bộ xử lí trung tâm:

- Đơn vị điều khiển (Control Unit - CU): điều khiển hoạt động của máy tính theo đúng lệnh và thứ tự của lệnh
- Đơn vị số học và logic (Arithmetic and Logic Unit - ALU): thực hiện các phép toán số học và logic
- Tập thanh ghi (Register File - RF): lưu trữ tạm thời dữ liệu và lệnh
- Bus trong (Internal Bus): kết nối các thành phần bên trong CPU với nhau
- Bus ngoài (External bus): kết nối CPU với các thành phần còn lại của máy tính
- Đơn vị nối ghép bus (Bus Interface Unit - BIU): kết nối và trao đổi thông tin bus bên và ngoài

Quản lí Processor

Sự thực thi của CPU khi có yêu cầu xử lí:



- Nhận lệnh (Fetch): CPU đọc các chỉ lệnh từ bộ nhớ
- Giải mã lệnh (Decode): giải mã lệnh → tín hiệu điều khiển việc thực hiện lệnh
- Thực hiện lệnh (Execute): thực hiện lệnh
- Ghi kết quả (Writeback): ghi kết quả vào bộ nhớ hoặc module vào ra

Quản lí Processor

- Điều độ tiến trình - Phân phối Processor cho các tiến trình
 - Nên tạo ra bao nhiêu Processor logic là thích hợp
 - Độ dài khoảng thời gian gắn liên tục Processor vật lý cho Processor logic là bao nhiêu thì hợp lý
 - Sau khi một Processor logic hết quyền sử dụng Processor vật lý thì cần chọn tiến trình nào để phân phối Processor vật lý
- Nguyên tắc
 - Đảm bảo cho mọi tiến trình được phục vụ như nhau, không có một tiến trình nào phải chờ đợi lâu hơn tiến trình khác
 - Đánh giá chất lượng điều độ: dựa vào thời gian chờ đợi trung bình của các tiến trình

Quản lí Processor

- Điều độ tiến trình - Chế độ một dòng xếp hàng
 - Chiến lược phục vụ bình đẳng FCFS (First Come First Served)
 - Chiến lược ưu tiên những tiến trình có thời gian thực hiện ngắn nhất SJN (Shortest Job Next)
 - Chiến lược ưu tiên các tiến trình có thời gian còn lại ít nhất SRN (Shortest Remaining Time)
 - Chiến lược xếp hàng lần lượt RR (Round Robin)

- Điều độ tiến trình - Chế độ nhiều dòng xếp hàng
 - Dựa vào thông tin do người sử dụng cung cấp và kết quả phân tích của hệ thống, phân lớp các tiến trình và đưa ra chiến lược phục vụ tương ứng
 - Các tiến trình có thể được phân thành các lớp:
 - Tiến trình thời gian thực
 - Tiến trình của chế độ sử dụng tập thể phân chia thời gian
 - Tiến trình xử lý lô
 - Chiến lược điều độ: Chương trình thư kí (Monitor)

Quản lí Bộ nhớ

- Tổng quan
- Quản lí bộ nhớ logic
- Quản lí bộ nhớ vật lí



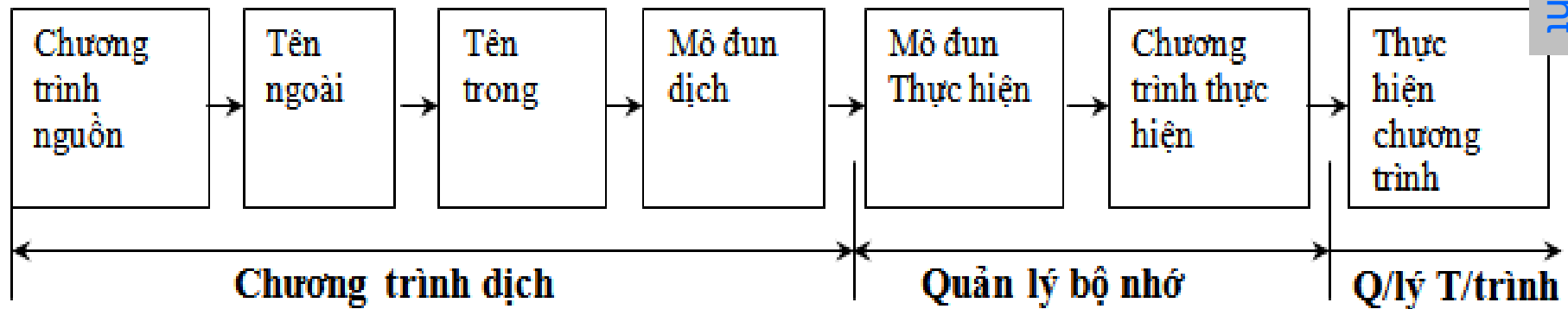
Quản lí Bộ nhớ

■ Bộ nhớ

- Lưu trữ nhân của hệ điều hành - tập các chương trình điều khiển thường xuyên có mặt ở bộ nhớ trong để thực hiện khi cần
- Lưu trữ dữ liệu và chương trình của người sử dụng
- Quản lý bộ nhớ trong các hệ điều hành: quản lý việc cấp phát và sử dụng không gian thực thi cho các chương trình, bảo vệ chương trình và dữ liệu khỏi bị hư hỏng, truy nhập một cách không hợp thức

Quản lí Bộ nhớ

■ Quản lí bộ nhớ logic - cấu trúc một chương trình



Quản lí Bộ nhớ

- Quản lí bộ nhớ logic
 - Cấu trúc tuyến tính
 - Cấu trúc động
 - Cấu trúc Overlay
 - Cấu trúc phân đoạn
 - Cấu trúc phân trang

Quản lí Bộ nhớ

■ Cấu trúc tuyến tính:

- Sau khi biên tập các mô đun được tập hợp thành 1 chương trình hoàn thiện chứa đầy đủ thông tin để có thể thực hiện
- Thực hiện: định vị 1 lần vào bộ nhớ
- Ưu điểm:
 - Đơn giản, chỉ việc tìm các móc nối
 - Không có sự gò bó về thời gian
 - Tính lưu động cao: có thể chuyển từ nơi này tới nơi khác
- Nhược:
 - Lãng phí bộ nhớ vì phải sử dụng vùng bộ nhớ lớn hơn mức cần thiết

Quản lí Bộ nhớ

■ Cấu trúc động:

- Từng mô đun được biên tập riêng biệt, khi thực hiện chỉ việc nạp mô đun đầu tiên vào bộ nhớ
- Khi cần mô đun khác → User sử dụng lệnh macro để nạp hoặc xóa mô đun ra khỏi bộ nhớ
- Ưu điểm:
 - Tiết kiệm bộ nhớ
- Nhược:
 - Yêu cầu người dùng phải biết kích thước hệ thống
 - Thời gian thực hiện lớn vì vừa thực hiện vừa định vị
 - Kém linh động

Quản lí Bộ nhớ

■ Cấu trúc Overlay

- Các mô đun chương trình được chia thành từng lớp → User cung cấp thông tin: số mức, mô đun tương ứng với mức (sơ đồ overlay - OVL)
- Thực thi chương trình: mô đun mức 0 (gốc) được nạp, khi cần nạp mô đun mức thấp hơn → tìm kiếm trong overlay
- Ưu điểm:
 - Tiết kiệm bộ nhớ
 - User không phải can thiệp vào chương trình nguồn
 - Các mô đun không phải lưu trữ nhiều lần
- Nhược:
 - Người dùng phải cung cấp sơ đồ overlay
 - Hạn chế 1 số cách gọi chương trình con

Quản lí Bộ nhớ

■ Cấu trúc phân đoạn

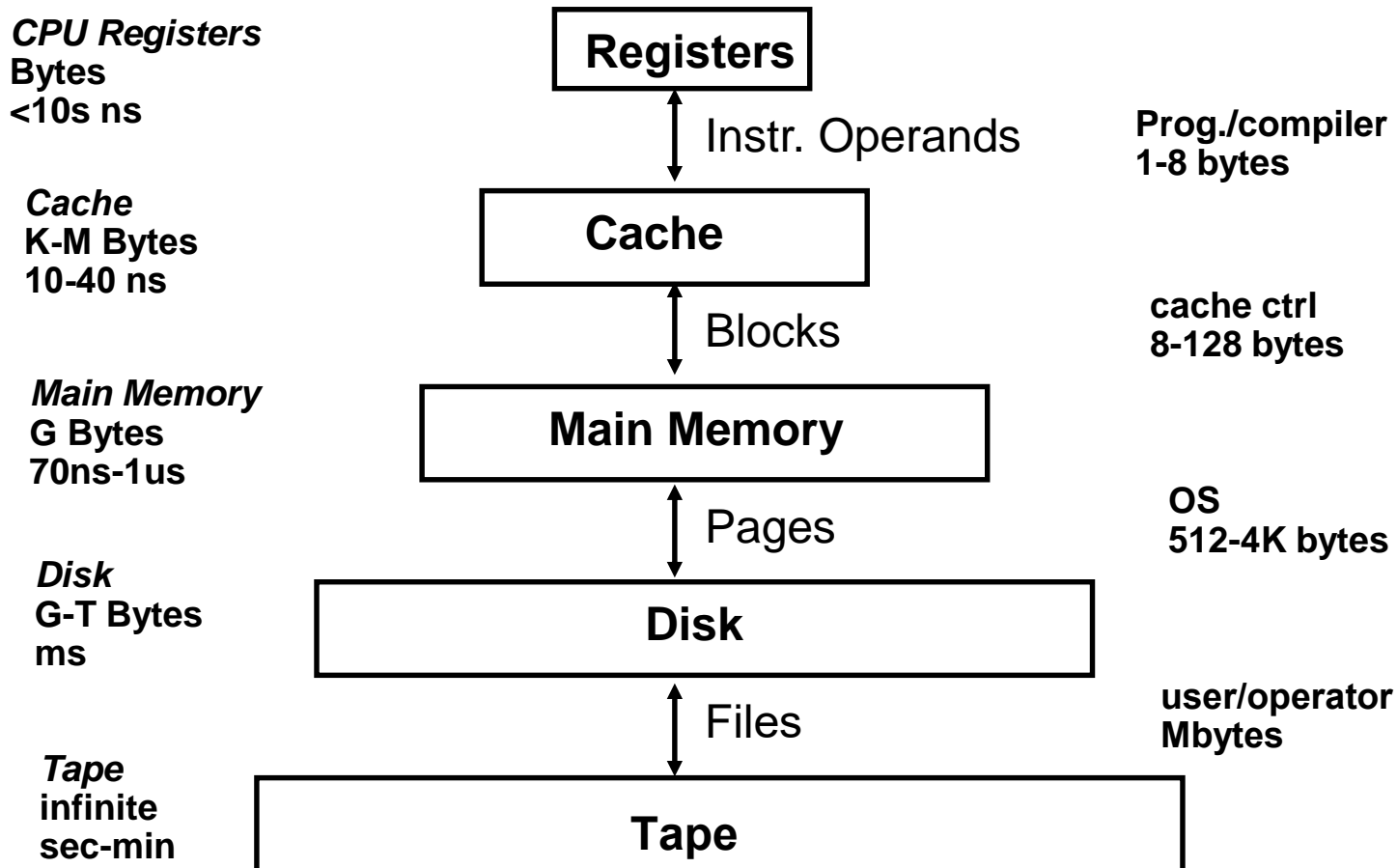
- Chương trình được biên tập thành các mô đun riêng biệt → sử dụng 1 bảng điều khiển cho biết chương trình có thể sử dụng những mô đun nào ~ SCB (Segment Control Block)
- Khi thực hiện chương trình dựa vào SCB kiểm tra xem mô đun có trong bộ nhớ hay không → nạp vào bất kỳ vùng nhớ nào
- Ưu điểm:
 - Các mô đun không cần phải được nạp liên tục và không cần có vị trí cố định
 - Người dùng không cần phải khai báo bất kỳ thông tin phụ nào
 - Hiệu quả tăng dần theo kích thước bộ nhớ
- Nhược:
 - Phụ thuộc cấu trúc ban đầu của chương trình nguồn

Quản lí Bộ nhớ

- Cấu trúc phân trang
 - Chương trình của người dùng được chia thành từng trang có kích thước giống nhau được quản lí bởi bảng quản lí trang (PCB)
 - Khi thực hiện sẽ nạp dần từng trang theo nhu cầu → hạn chế lãng phí bộ nhớ
 - Có sự hỗ trợ của phần cứng

Quản lí Bộ nhớ

Bộ nhớ vật lí



Quản lí Bộ nhớ

- Quản lí bộ nhớ vật lí
 - Phân chương cố định
 - Chế độ phân chương động
 - Chế độ phân đoạn
 - Chế độ phân trang
 - Chế độ kết hợp phân trang và phân đoạn

Quản lí Bộ nhớ

- Cấu trúc phân Chương (cố định)
 - Bộ nhớ được chia thành n phần, mỗi phần sử dụng như 1 bộ nhớ độc lập: Chương nhớ
 - Chương nhớ ~ chương trình, chương trình được nạp cố định vào Chương → tồn tại 1 danh sách quản lý bộ nhớ tự do chưa sử dụng
- Ưu điểm:
 - Đơn giản, dễ bảo vệ
 - Tồn tại công cụ bên trong bộ nhớ có thể phân chia lại hệ thống
 - Có thể phân loại các chương trình trước khi thực hiện → có thể tổ chức phục vụ gần tối ưu
- Nhược:
 - Bộ nhớ bị phân đoạn nên khi phân chia lại sẽ thay đổi đường biên → thông tin bị xóa

Quản lí Bộ nhớ

- Cấu trúc phân Chương (động)
 - Tồn tại 1 danh sách quản lí bộ nhớ tự do cho toàn bộ hệ thống nhớ → mỗi chương trình khi xuất hiện được phân phối 1 vùng nhớ riêng liên tục, sử dụng như 1 bộ nhớ độc lập
 - Ưu điểm:
 - Hệ số song song cao khi số chương trình thực hiện có thể thay đổi và có thể thực hiện 1 chương trình bất kỳ miễn là có đủ bộ nhớ
 - Hệ thống điều khiển không bị sao chép đi nơi khác
 - Nhược:
 - Cơ chế quản lí phức tạp
 - Khi gặp sự cố kỹ thuật → chương trình sẽ bị phá huỷ
 - Xuất hiện hiện tượng phân đoạn ngoài

Quản lí Bộ nhớ

■ Cấu trúc phân Đoạn

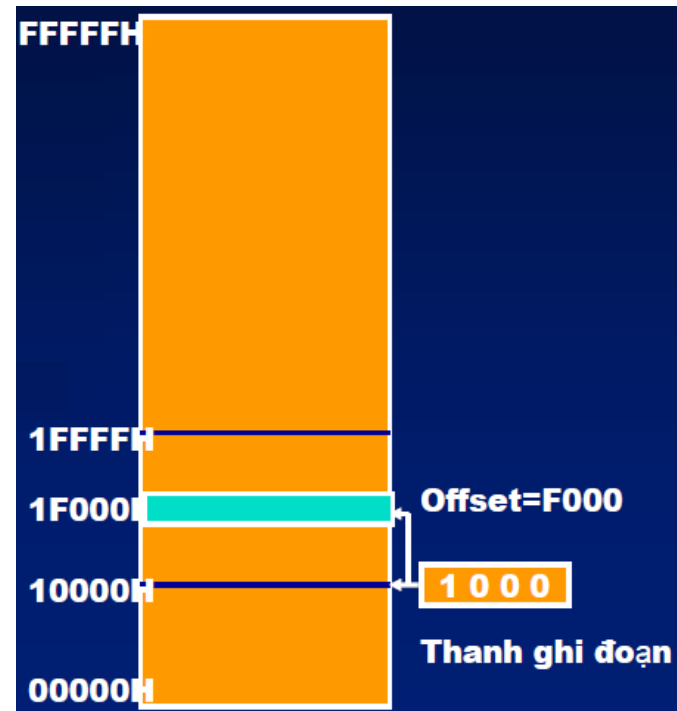
- Chương trình được biên tập theo cấu trúc phân đoạn
- Bộ nhớ được chia thành các phần có kích thước không nhất thiết bằng nhau ~ đoạn nhớ (segment)
- Mỗi chương trình được lưu trữ tại 1 đoạn nhớ → quản lí đoạn: bảng SCB (Segment Control Block)
- Phần tử SCB:
 - D: chương trình đã được nạp vào đoạn hay chưa
 - A: địa chỉ cơ sở (số hiệu) đoạn
 - L: độ dài đoạn

Quản lí Bộ nhớ

■ Cấu trúc phân đoạn

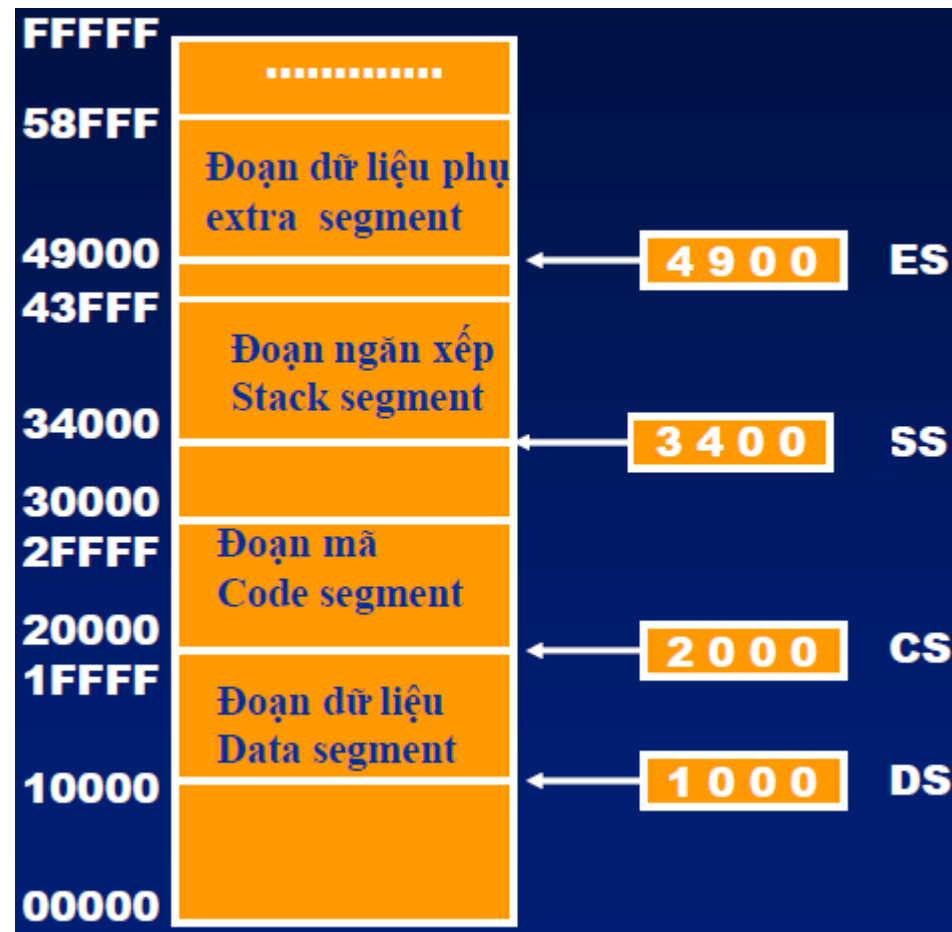
- Địa chỉ cơ sở: lưu trữ trong thanh ghi đoạn (DS, CS, SS, ES)
- Địa chỉ thực (vật lí):
 - Địa chỉ đoạn
 - Độ lệch (offset)

→ Địa chỉ vật lí = Thanh ghi đoạn * 16 + Offset



Quản lí Bộ nhớ

■ Các thanh ghi đoạn



Quản lí Bộ nhớ

■ Cấu trúc phân đoạn

Ví dụ 1: Địa chỉ vật lý 12345H có thể được tạo ra từ các giá trị:

Thanh ghi đoạn	Offset
1000H	2345H
1200H	0345H
1004H	2305H
0300H	F345H

Quản lí Bộ nhớ

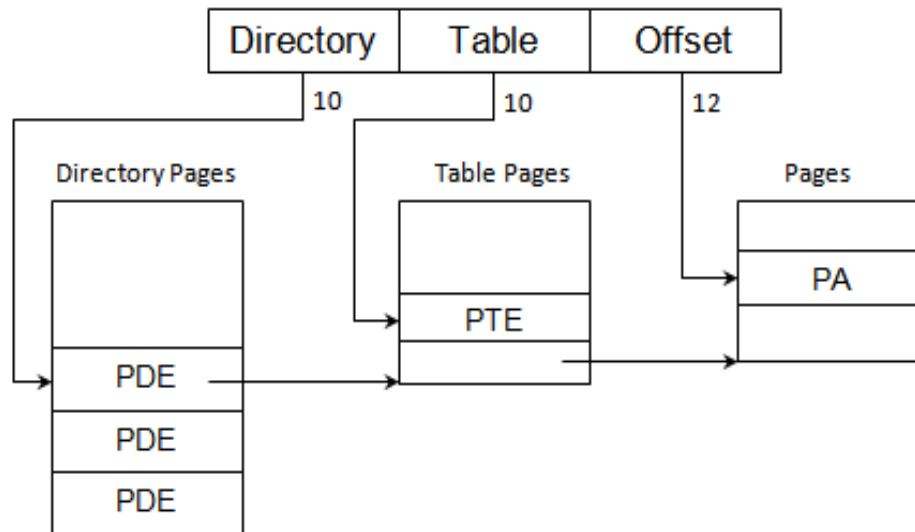
■ Cấu trúc phân Trang

- Chương trình được biên tập theo cấu trúc phân trang
- Bộ nhớ được chia thành các phần có kích thước bằng nhau ~ trang nhớ
- Mỗi trang chương trình được lưu trữ tại 1 trang nhớ → kích thước trang nhớ = trang chương trình
- Quản lí đoạn: bảng PCB (Page Control Block)
- Phần tử PCB:
 - D: chương trình đã được nạp vào trang hay chưa
 - A: địa chỉ cơ sở (số hiệu) trang

→ Địa chỉ vật lí = Địa chỉ trang + Offset

Quản lí Bộ nhớ

■ Cấu trúc phân Trang



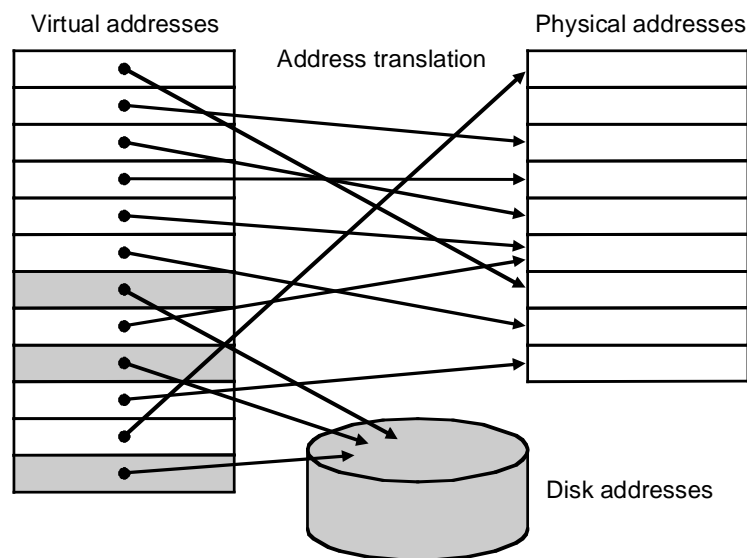
- Địa chỉ offset: vị trí byte nhớ cần truy nhập trong trang nhớ - địa chỉ vật lí của byte nhớ trong bộ nhớ chính (1page = 4Kb)
- Bảng trang (Table): chứa địa chỉ cơ sở của trang nhớ
- Danh mục trang (Directory): chứa địa chỉ cơ sở của bảng trang

Quản lí Bộ nhớ

- Cấu trúc kết hợp phân Trang và phân Đoạn
 - Bộ nhớ được tổ chức theo kiểu phân trang → bảng PCB
 - Chương trình được tổ chức theo kiểu phân đoạn → bảng SCB
 - Mỗi mô đun chương trình được lưu trữ trong nhiều trang nhớ → mỗi phần tử SCB sẽ quản lí nhiều PCB trong nó

Quản lí Bộ nhớ

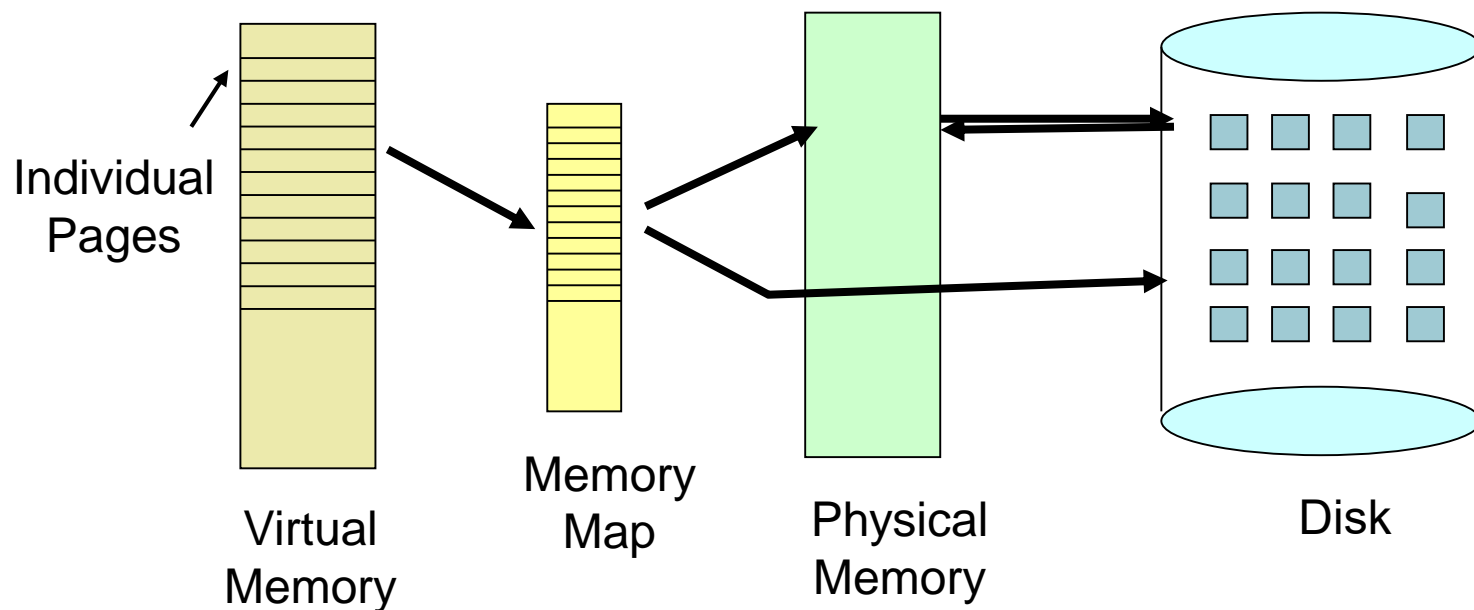
- Bộ nhớ ảo: đồng nhất các bộ nhớ (Cache, Main memory Disk..) thành một hệ thống tổng thể
- Quản lí bộ nhớ ảo qua địa chỉ ảo
 - Cho phép các chương trình có thể chia sẻ bộ nhớ một cách an toàn và hiệu quả
 - Giảm tải cho bộ nhớ chính
 - Cho phép chương trình có thể được nạp vào bất kì bộ nhớ vật lí nào



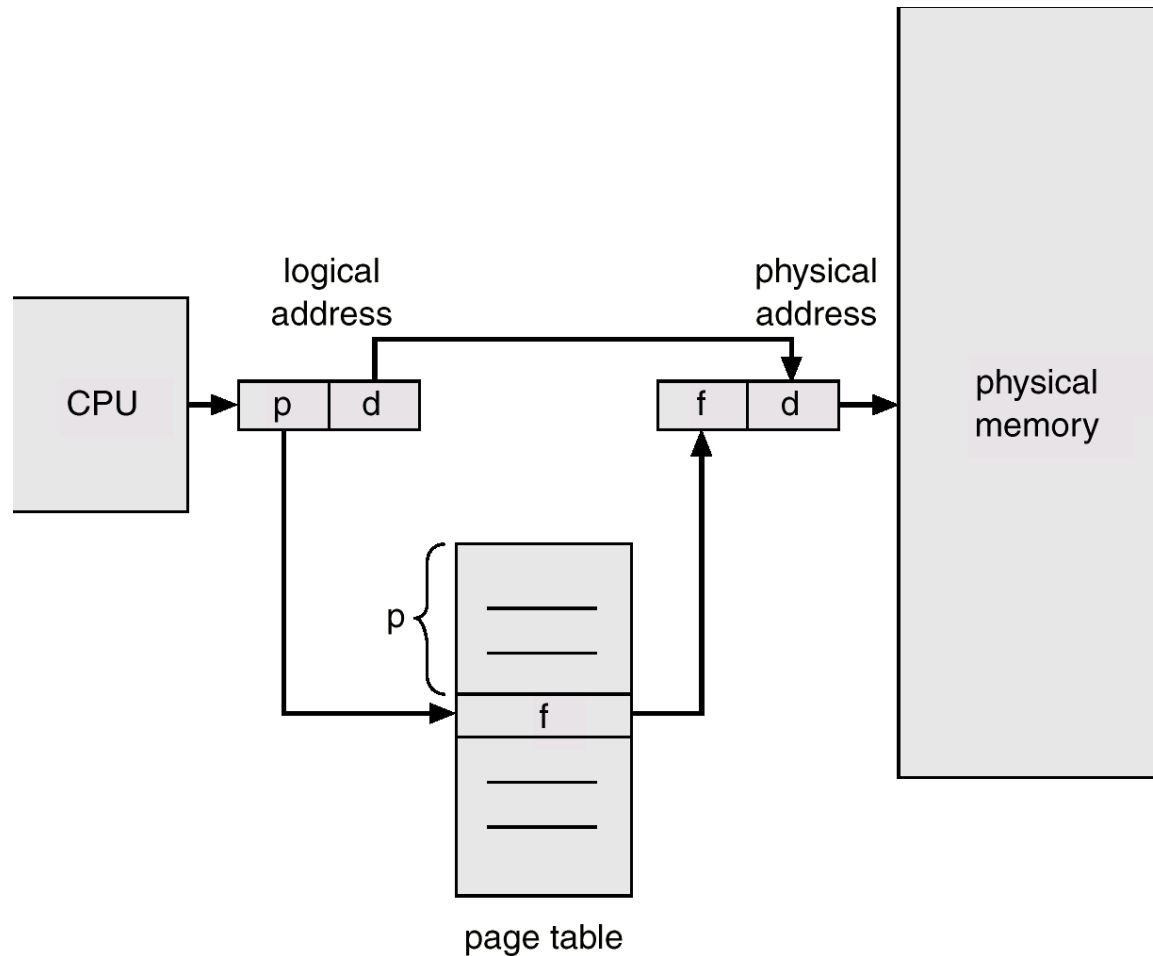
Quản lí Bộ nhớ

■ Quản lí bộ nhớ ảo

- Chia không gian địa chỉ thành các Trang (Page)
- Mỗi chương trình có thể truy nhập tới 1 tập các trang
- Các trang được lưu trữ trong bộ nhớ vật lí hoặc đĩa
- Truy nhập các trang qua địa chỉ ảo



Quản lí Bộ nhớ



Pages = logical blocks; Frames = physical blocks

Quản lí Bộ nhớ

0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Logical Memory

0	5
1	6
2	1
3	2

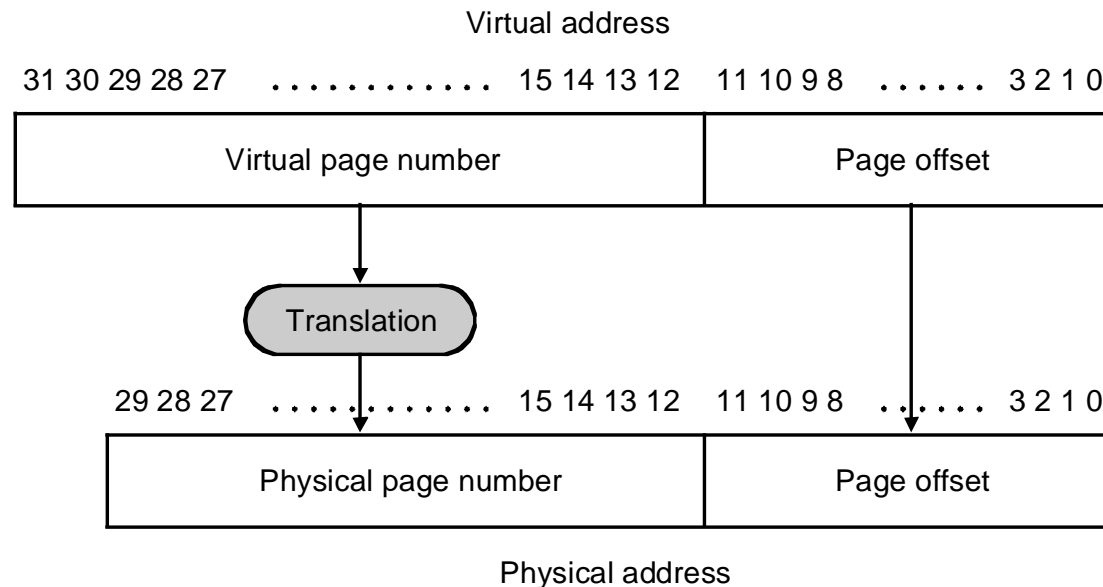
Page Table

0	0
1	4 i
	j
	k
2	8 m
	n
	o
3	12 p
4	16
5	20 a
	b
	c
	d
6	24 e
	f
	g
	h
7	28

Physical Memory

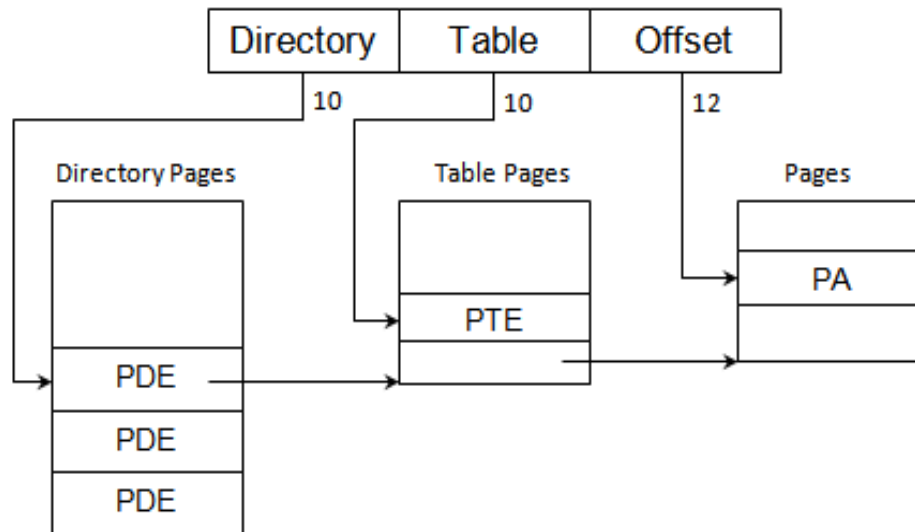
Quản lí Bộ nhớ

- Quản lí bộ nhớ ảo: biên dịch địa chỉ
 - Ánh xạ địa chỉ ảo sang địa chỉ vật lí nhờ kĩ thuật biên dịch địa chỉ
 - Nếu trang nhớ có trong bộ nhớ chính → xử lí
 - Nếu trang nhớ có trong đĩa → “treo” chương trình và nạp trang nhớ từ đĩa



Quản lí Bộ nhớ

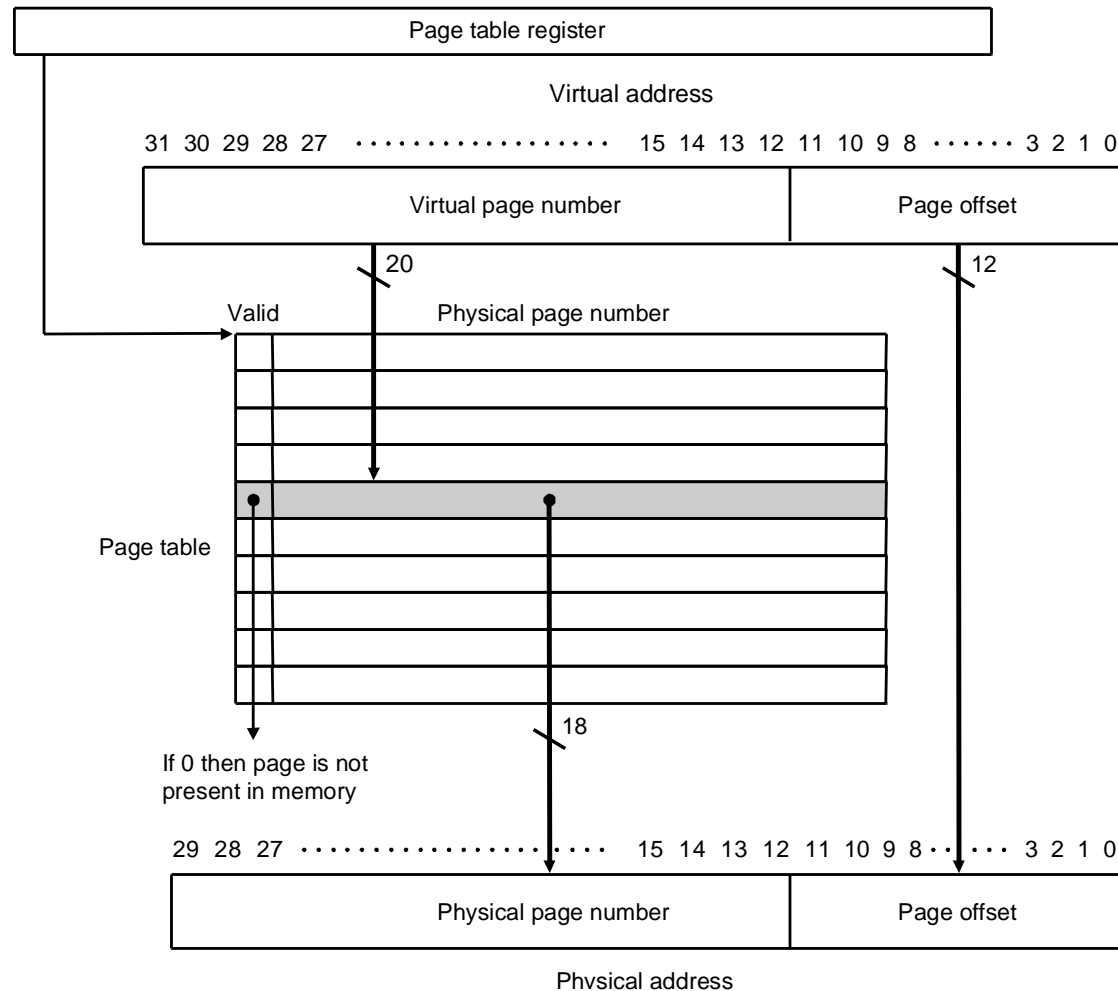
■ Biên dịch địa chỉ dùng danh mục trang



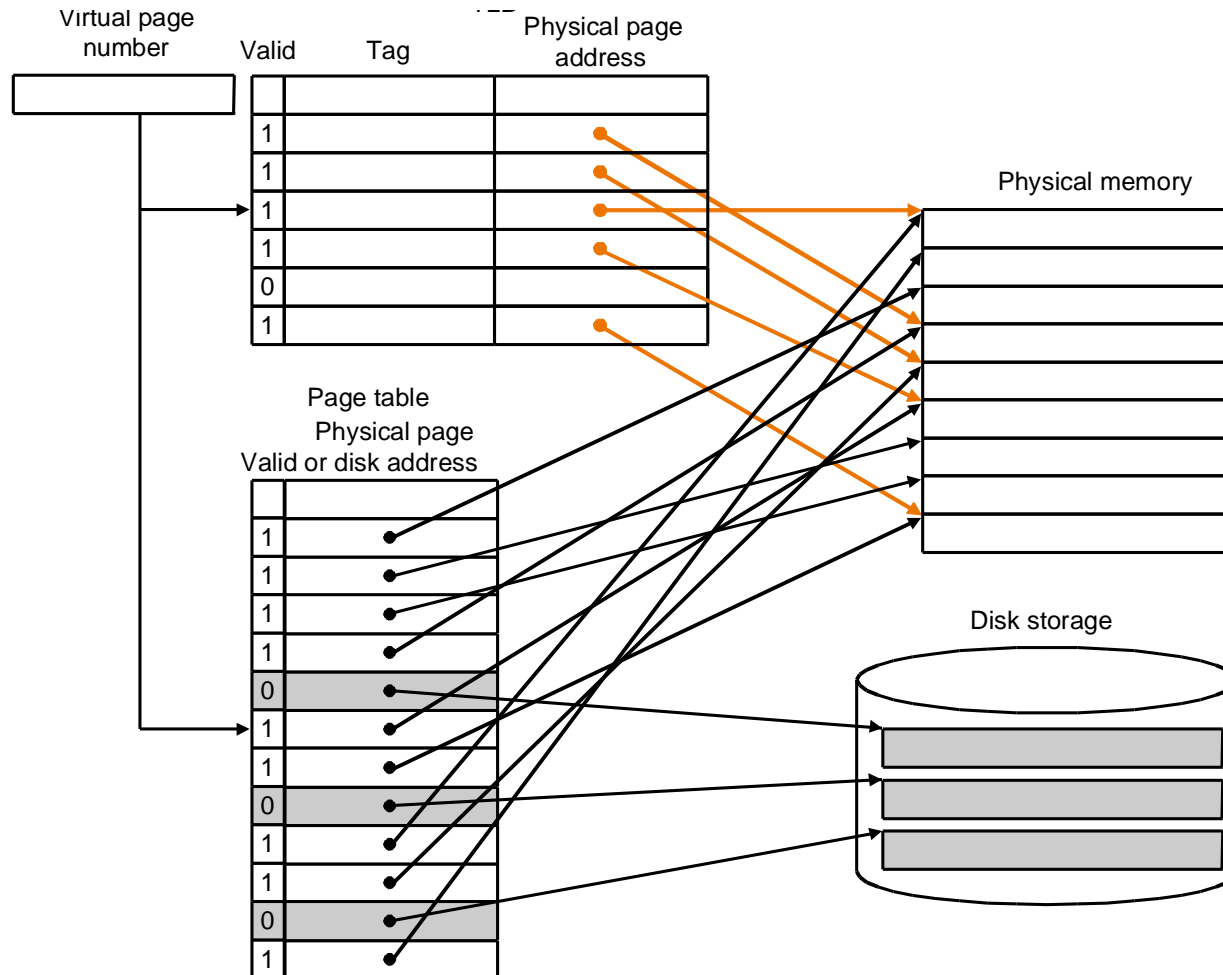
- Địa chỉ offset: vị trí byte nhớ cần truy nhập trong trang nhớ - địa chỉ vật lí của byte nhớ trong bộ nhớ chính (1page = 4Kb)
- Bảng trang (Table): chứa địa chỉ cơ sở của trang nhớ
- Danh mục trang (Directory): chứa địa chỉ cơ sở của bảng trang

Quản lí Bộ nhớ

■ Biên dịch địa chỉ dùng Translation lookaside buffer



■ Biên dịch địa chỉ dùng Translation lookaside buffer



Quản lí thiết bị

- Tổng quan
- Quản lí thiết bị
- Quản lí File



Quản lí thiết bị

- Hệ thống vào/ra: giao tiếp giữa máy tính với môi trường ngoài
- Thành phần:
 - Các thiết bị ngoại vi
 - Mô đun vào/ra

Quản lí thiết bị

- Thiết bị ngoại vi (Peripheral): chuyển đổi dữ liệu từ môi trường ngoài sang dạng phù hợp để đưa vào máy tính xử lí và ngược lại
- Phân loại
 - Các thiết bị thu nhận dữ liệu: bàn phím, chuột, máy quét,...
 - Các thiết bị hiển thị dữ liệu: màn hình, máy in, ...
 - Các thiết bị lưu trữ: đĩa từ, đĩa quang, thẻ nhớ
 - Các thiết bị truyền thông: modem, card mạng, ...

Quản lí thiết bị

- Thiết bị ngoại vi:
 - Bộ chuyển đổi tín hiệu: chuyển đổi dữ liệu giữa bên ngoài sang dạng phù hợp để đưa vào máy tính xử lí và ngược lại
 - Bộ đệm dữ liệu: lưu trữ tạm thời dữ liệu cần trao đổi giữa TBNV và CPU
 - Logic điều khiển: điều khiển hoạt động của thiết bị ngoại vi đáp ứng theo yêu cầu từ module vào-ra

Quản lí thiết bị

- Mô đun vào/ra:
 - Điều khiển và định thời gian cho quá trình trao đổi
 - Trao đổi thông tin với CPU
 - Trao đổi thông tin với TBNV
 - Đệm dữ liệu
 - Phát hiện lỗi

Quản lí thiết bị

■ Quản lí thiết bị

- Các thiết bị vào/ra: đảm nhiệm việc truyền thông tin qua lại giữa các thành phần của hệ thống
- Đặc điểm: phát triển nhanh chóng và đa dạng về chủng loại, phong phú về số lượng → cách thức tổ chức và quản lý:
 - Bộ xử lý chỉ điều khiển các thao tác vào/ra chứ không trực tiếp thực hiện
 - Các thiết bị vào/ra không trực tiếp gắn vào bộ xử lý mà gắn với thiết bị điều khiển chúng
- Thiết bị điều khiển vào/ra ~ máy tính chuyên dụng:
 - Điều khiển hoạt động của thiết bị vào/ra
 - Có ngôn ngữ riêng, lệnh riêng
 - Hoạt động độc lập với nhau và độc lập với bộ xử lý

Quản lí thiết bị

■ Quản lí thiết bị

- Vi xử lý tạo ra một chương trình tương ứng với công việc cần thực hiện → chuyển giao chương trình kênh và dữ liệu tương ứng cho thiết bị điều khiển và tiếp tục thực hiện chương trình của mình
- Các phép vào/ra được điều khiển theo nguyên lý Macroprocessor cho phép trong lúc các phép vào/ra được thực hiện ở thiết bị ngoại vi thì vi xử lý vẫn hoạt động song song (thực hiện các tính toán và điều khiển khác khi chưa cần đến kết quả vào/ra)
- Khi công việc được hoàn thành báo cho vi xử lý biết bằng tín hiệu ngắt

Quản lí thiết bị

- Quản lí thiết bị
 - Cơ chế phòng đệm (Buffer): tổ chức một số vùng nhớ trung gian làm nơi lưu trữ thông tin trong các phép vào/ra để đảm bảo hiệu suất
 - Cơ chế mô phỏng các phép trao đổi vào/ra trong chế độ trực tiếp SPOOL (Simultaneous Peripheral Operation On Line)
- Quản lí file

Quản lí thiết bị

■ Phòng đệm trung gian

■ Lưu trữ tạm thời kết quả vào/ra

■ Phòng đệm vào: nhập thông tin

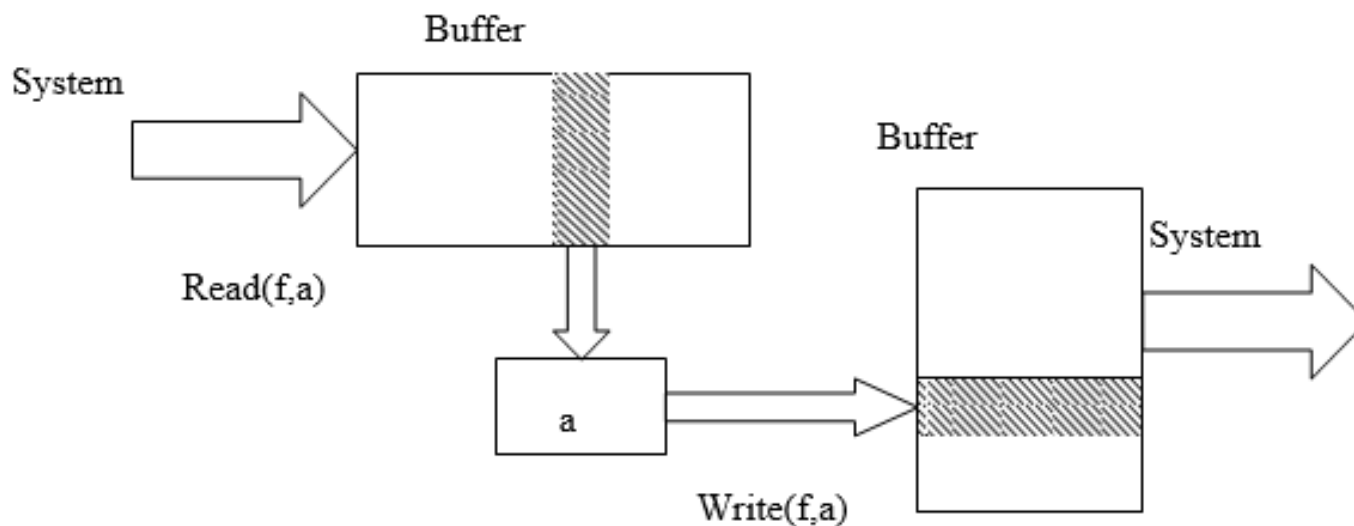
- Chỉ thị đọc (read): thông tin được tách và chuyển từ phòng đệm vào các địa chỉ tương ứng trong Chương trình ứng dụng
- Mỗi giá trị được lưu trữ ở hai nơi trong bộ nhớ (một ở phòng đệm và một ở vùng bộ nhớ trong Chương trình ứng dụng)
- Giảm thời gian chờ đợi: tổ chức nhiều phòng đệm vào, khi hết thông tin ở một phòng đệm, hệ thống sẽ chuyển sang phòng đệm khác

■ Phòng đệm ra:

- Chỉ thị ghi (write): thông tin được đưa vào phòng đệm. Khi phòng đệm ra đầy, hệ thống sẽ đưa thông tin ra thiết bị ngoại vi
- Giảm thời gian chờ đợi: tổ chức nhiều phòng đệm ra.

Quản lí thiết bị

■ Phòng đệm trung gian

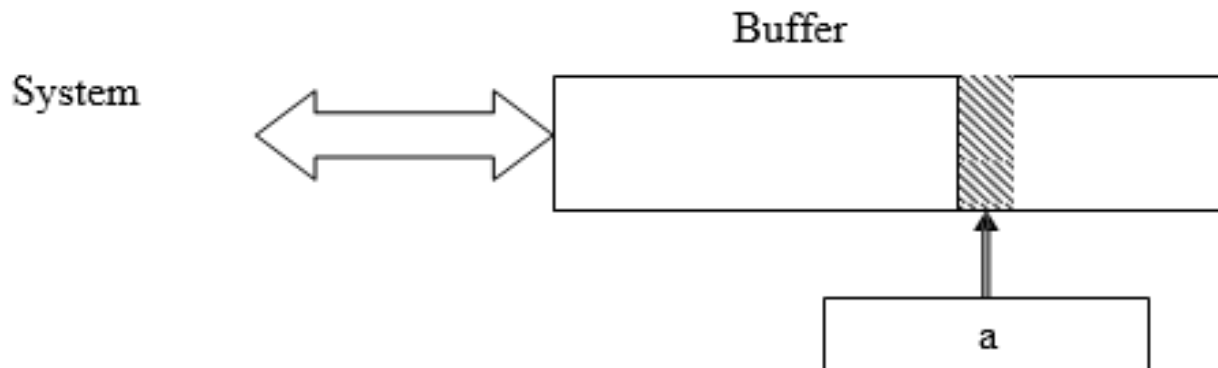


■ Ưu nhược điểm:

- Đơn giản
- Hệ số song song cao (do tốc độ giải phóng vùng đệm lớn)
- Vạn năng, áp dụng cho mọi phép vào/ra
- Tốn bộ nhớ, thời gian trao đổi
- Nhiều lỗi xử lý

Quản lí thiết bị

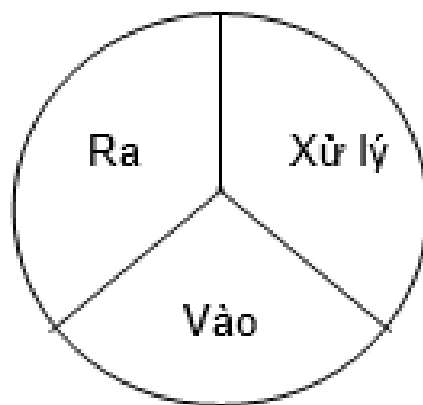
- Phòng đệm xử lí: thông tin được xử lí ngay trong phòng đệm



- Ưu nhược điểm:
 - Tiết kiệm bộ nhớ
 - Không mất thời gian chuyển thông tin ở bộ nhớ trong
 - Hệ số song song thấp
 - Tốc độ giải phóng phòng đệm chậm
 - Tính vụn năng không cao

Quản lí thiết bị

- Phòng đệm vòng: kết hợp 2 loại phòng đệm trên



- Tổ chức:

- Phòng đệm gắn với từng file cụ thể: chỉ được xây dựng khi mở file hoặc đóng file
- Phòng đệm gắn với hệ thống: khi xây dựng hệ thống → xây dựng ngay cơ chế phòng đệm và chỉ gắn vào một file cụ thể nào đó

Quản lí thiết bị

- Cơ chế SPOOL (Simultaneous Peripheral Operation OnLine)
 - Dùng phần mềm để mô phỏng hoạt động của thiết bị ngoại vi và coi nó như một thiết bị ngoại vi ảo
 - Mô phỏng quá trình điều khiển, quản lý thiết bị ngoại vi
 - Mô phỏng các phép trao đổi ngoại vi ngay trong lúc thực hiện
 - Ưu điểm:
 - Làm cho chương trình của người dùng thực hiện nhanh hơn
 - Giảm giá thành chi phí
 - Khai thác thiết bị ngoại vi tốt hơn
 - Giảm yêu cầu về số lượng thiết bị
 - Tạo ra kỹ thuật lập trình tương ứng

Quản lí thiết bị

■ Quản lí File

- Người dùng phải lưu trữ thông tin ở bộ nhớ ngoài → OS phải có vai trò sao cho người dùng truy nhập thuận tiện
- Nhu cầu dùng chung các file dữ liệu

Quản lí thiết bị

■ Hệ quản lí File

- Tính độc lập của file với vi xử lý và với thiết bị ngoại vi
→ quản lí file theo tên
- Bảo vệ dữ liệu không để mất thông tin khi có sự cố kỹ thuật hoặc chương trình thậm chí truy nhập bất hợp lệ
- Tổ chức có hiệu quả đảm bảo tiết kiệm bộ nhớ ngoài và dễ truy nhập
- Mọi thao tác phức tạp phải “trong suốt ” với người dùng đảm bảo công cụ truy nhập tới tay người dùng ở dạng đơn giản nhất

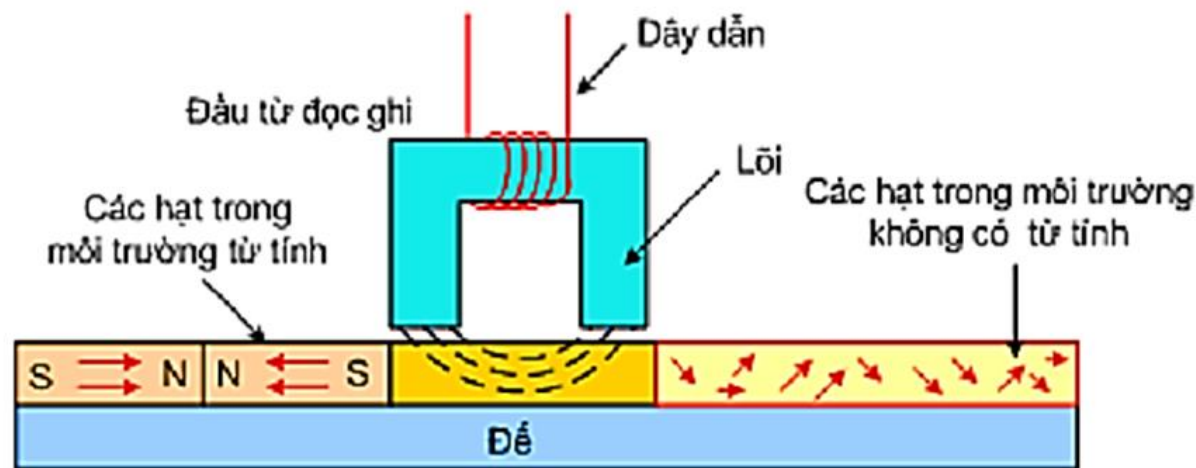
Quản lí thiết bị

■ Hệ quản lí File

- Tồn tại các câu lệnh: đọc, ghi, tạo, đổi tên, đóng, mở file...
- Tổ chức thông tin trên phương tiện mang tin và tự động ghi nhận sơ đồ
- Bố trí file để đáp ứng yêu cầu truy nhập và tìm kiếm
- Có cơ chế bảo vệ file
 - Tĩnh: liên quan tới toàn bộ file và cố định theo thời gian
 - Động: xác lập khi mở file đọc, ghi thông tin
- Xóa dữ liệu trong file
 - Mức vật lý: toàn bộ nội dung file
 - Mức logic: ngắt các móc nối liên hệ với file

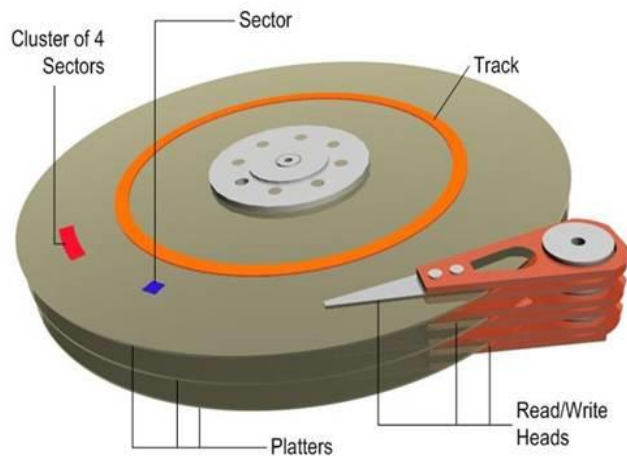
Quản lí thiết bị

- Quản lí File trong các hệ điều hành của MS
 - Tham số đầu đọc
 - Tham số đĩa (tù)
- Nguyên lí ghi/đọc: nam châm điện
 - Nguyên lí ghi thông tin lên đĩa
 - Nguyên lí đọc thông tin trên đĩa



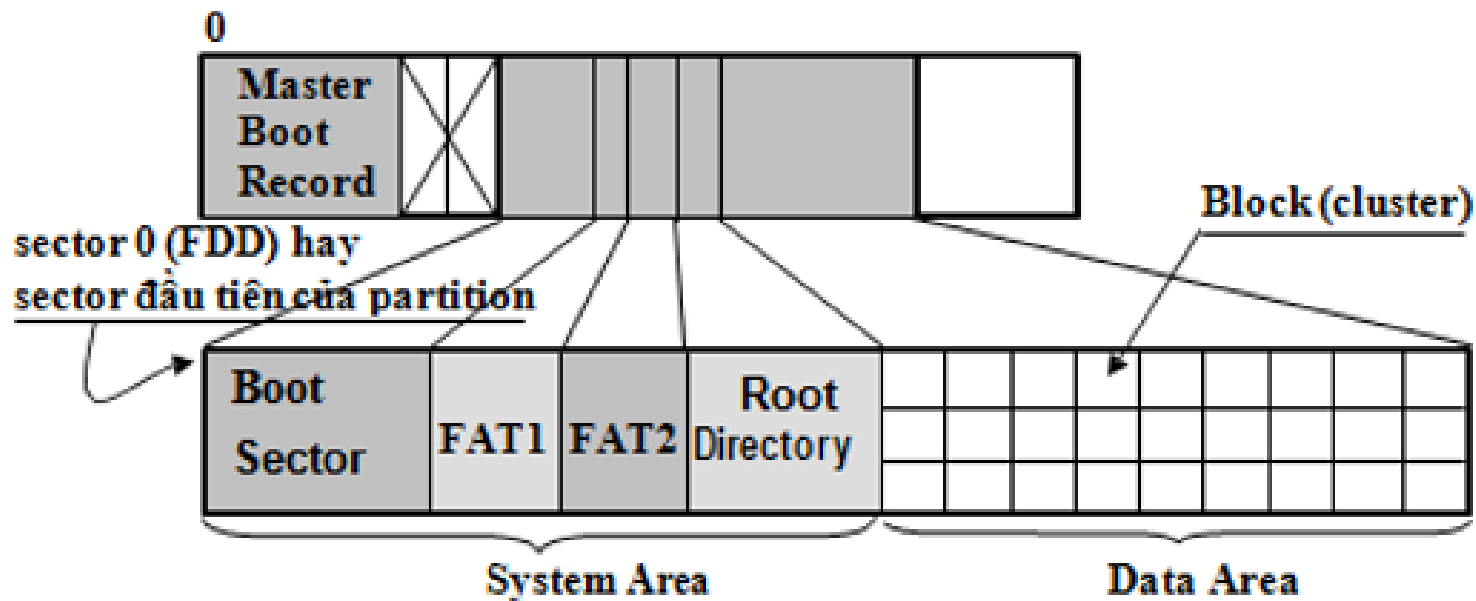
■ Tham số đĩa

- Rãnh từ (Track): các đường tròn đồng tâm được đánh số từ ngoài vào trong (bắt đầu từ rãnh số 0)
- Cung từ (Sector): mỗi rãnh được chia làm nhiều cung, dung lượng 1 sector thường là 512 byte.
- Liên cung (Cluster): tập hợp của 2, 4, 8... cung từ, các cung được đánh số tuần tự
- Từ trụ (Cylinder): các rãnh từ có cùng số thứ tự trên các đĩa từ



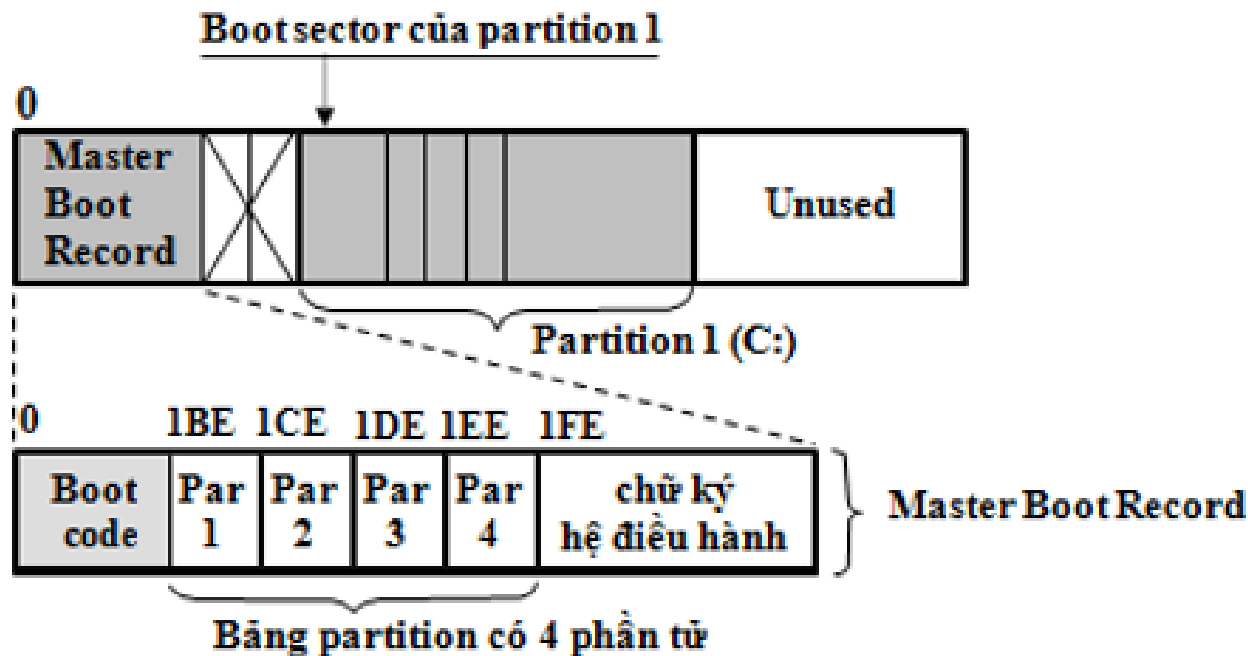
Quản lí thiết bị

- Quản lí File trong các hệ điều hành của MS
 - Cấu trúc các vùng thông tin trên đĩa



Quản lí thiết bị

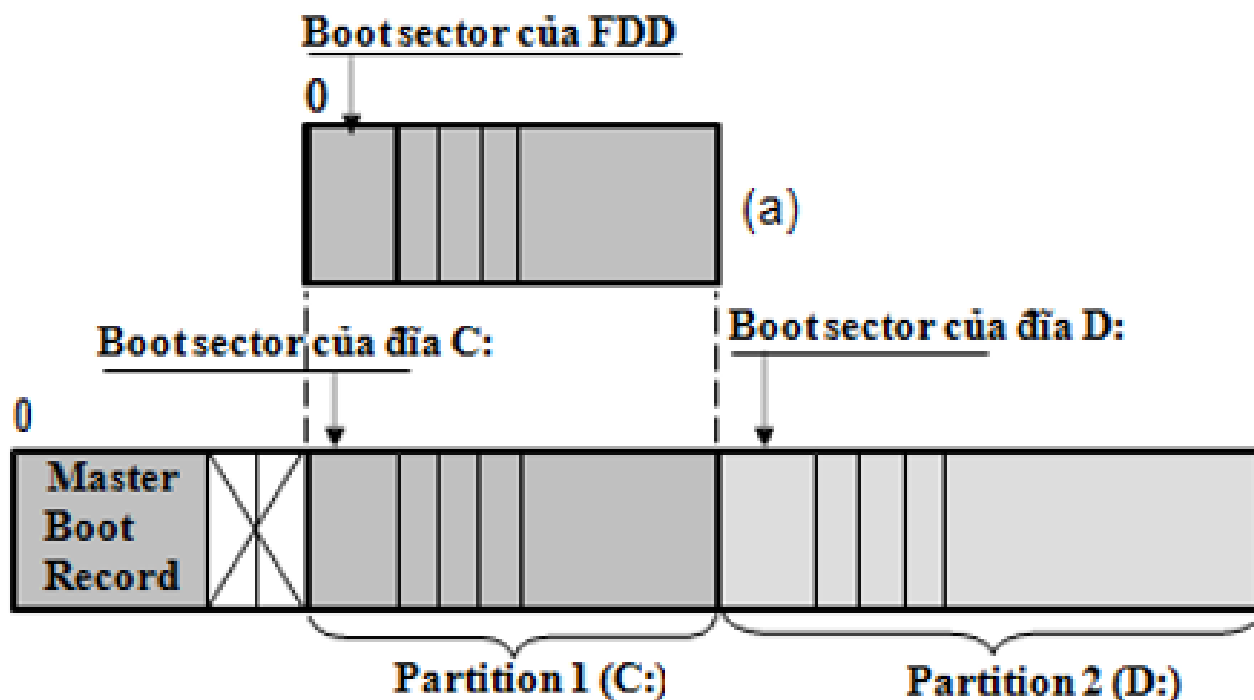
- Quản lí File trong các hệ điều hành của MS
 - Master Boot Boot chính của đĩa từ cứng



- Boot sector:
 - Xác định tham số tổ chức của đĩa: đặc thù cho mỗi đĩa
 - Chương trình khởi phục vụ cho việc nạp hệ điều hành

Quản lí thiết bị

- Quản lí File trong các hệ điều hành của MS
 - Các phân vùng logic



Quản lí thiết bị

- Truy xuất thông tin trên đĩa
 - Số byte cho một sector
 - Số sector trước FAT
 - Số bảng FAT
 - Số mục vào (entry) cho root (32 byte cho một entry)
 - Tổng số sector trên đĩa
 - Số lượng sector cho một bảng FAT
 - Số sector trên một track
 - Số đầu đọc, ghi

Hệ điều hành nhiều Processor

- Tổng quan
- Hệ điều hành nhiều Processor
- Hệ điều hành phân tán



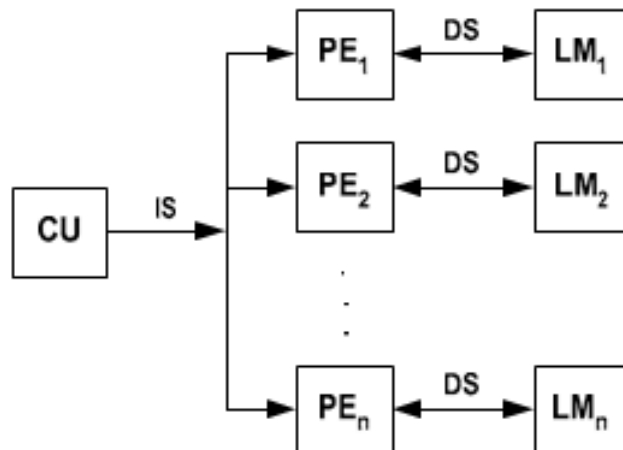
Hệ điều hành nhiều Processor

■ Tổng quan

- Sự kết hợp của các Processor trong một hệ thống tính toán, sự kết hợp của các hệ thống tính toán đơn Processor với mục đích:
 - Chuyên môn hoá các Processor → giảm gánh nặng xử lý
 - Hoạt động ổn định và năng suất cao
 - Làm cho các tài nguyên có giá trị cao, mang tính khả dụng đối với bất kỳ người dùng người dùng nào trên mạng
 - Tăng độ tin cậy của hệ thống nhờ khả năng thay thế khi xảy ra sự cố đối với một máy tính nào đó

Hệ điều hành nhiều Processor

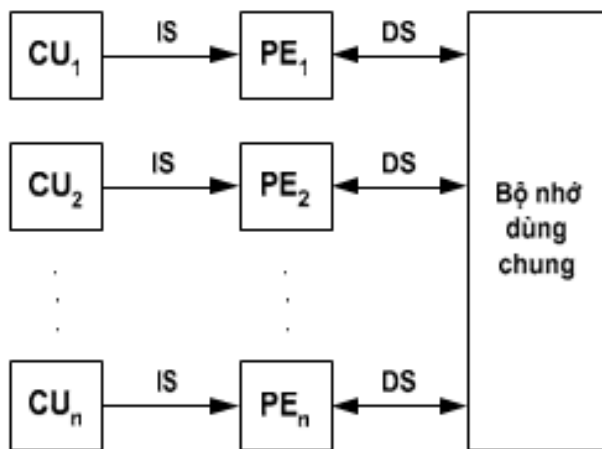
- Kiến trúc đơn dòng lệnh - đa dòng dữ liệu



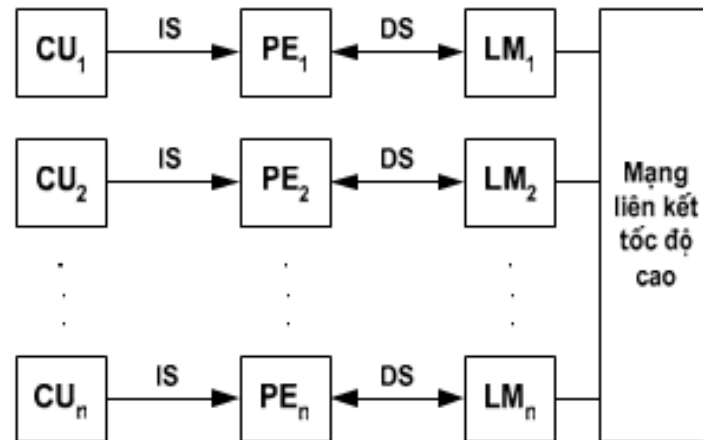
- Tại một thời điểm có thể thực hiện 1 lệnh trên nhiều dòng dữ liệu khác nhau
- Các mô hình SIMD
 - Vector Computer
 - Array processor

Hệ điều hành nhiều Processor

- Kiến trúc đa dòng lệnh - đa dòng dữ liệu
 - Các bộ xử lý đồng thời thực hiện các dãy lệnh khác nhau trên các dữ liệu khác nhau - kiến trúc song song
 - Các mô hình MIMD



Shared Memory
(Multiprocessor)



Distributed Memory
(Multicomputer)

Hệ điều hành nhiều Processor

- Hệ điều hành nhiều Processor
 - Tồn tại một hệ điều hành có chức năng quản lý dữ liệu, tính toán và xử lý một cách thống nhất
 - Tiếp cận:
 - Tập trung: tôn trọng hệ điều hành cục bộ đã có trên các hệ thống tính toán → hệ điều hành nhiều processor được cài đặt như một tập các chương trình tiện ích chạy trên hệ thống
 - Phân tán: bỏ qua hệ điều hành cục bộ đã có trên các hệ thống, cài đặt một hệ điều hành thuần nhất trên toàn mạng

Hệ điều hành nhiều Processor

■ Hệ điều hành nhiều Processor

■ OS tập trung:

- Cung cấp cho mỗi người dùng một tiến trình đồng nhất làm nhiệm vụ cung cấp một giao diện đồng nhất với tất cả các hệ thống cục bộ đã có
- Tiến trình quản lý cơ sở dữ liệu chứa thông tin về hệ thống cục bộ và về các chương trình và dữ liệu của người dùng thuần túy

■ OS phân tán:

- Mô hình tiến trình: mỗi tài nguyên được quản lý bởi một tiến trình nào đó và hệ điều hành điều khiển sự tương tác giữa các tiến trình đó
- Mô hình đối tượng: coi các tiến trình là các đối tượng, mỗi đối tượng có một kiểu, một biểu diễn và một tập các thao tác có thể thực hiện trên nó

Hệ điều hành nhiều Processor

- Hệ điều hành nhiều Processor

- OS tập trung và phân tán:

- Hệ điều hành tập trung: Quan tâm tới việc quản trị tài nguyên hệ thống
 - Hệ điều hành phân tán: Trừu tượng hoá máy tính
 - Hệ điều hành tập trung: chạy trên các hệ thống có một hay nhiều bộ xử lý (Processor)
 - Các hệ thống phân tán: phạm vi rộng, phân tán ở nhiều địa điểm khác nhau đòi hỏi cơ chế quản lý phân tán