

---

# Chương I: TỔNG QUAN VỀ HĐH

---

---

# NỘI DUNG:

- 1.1 Khái niệm về HĐH
- 1.2 Phân Loại HĐH
- 1.3 Giới thiệu về cấu trúc của HĐH
- 1.4 Tìm hiểu về lịch sử phát triển của HĐH
- 1.5 Giới thiệu một số HĐH phổ biến hiện nay

---

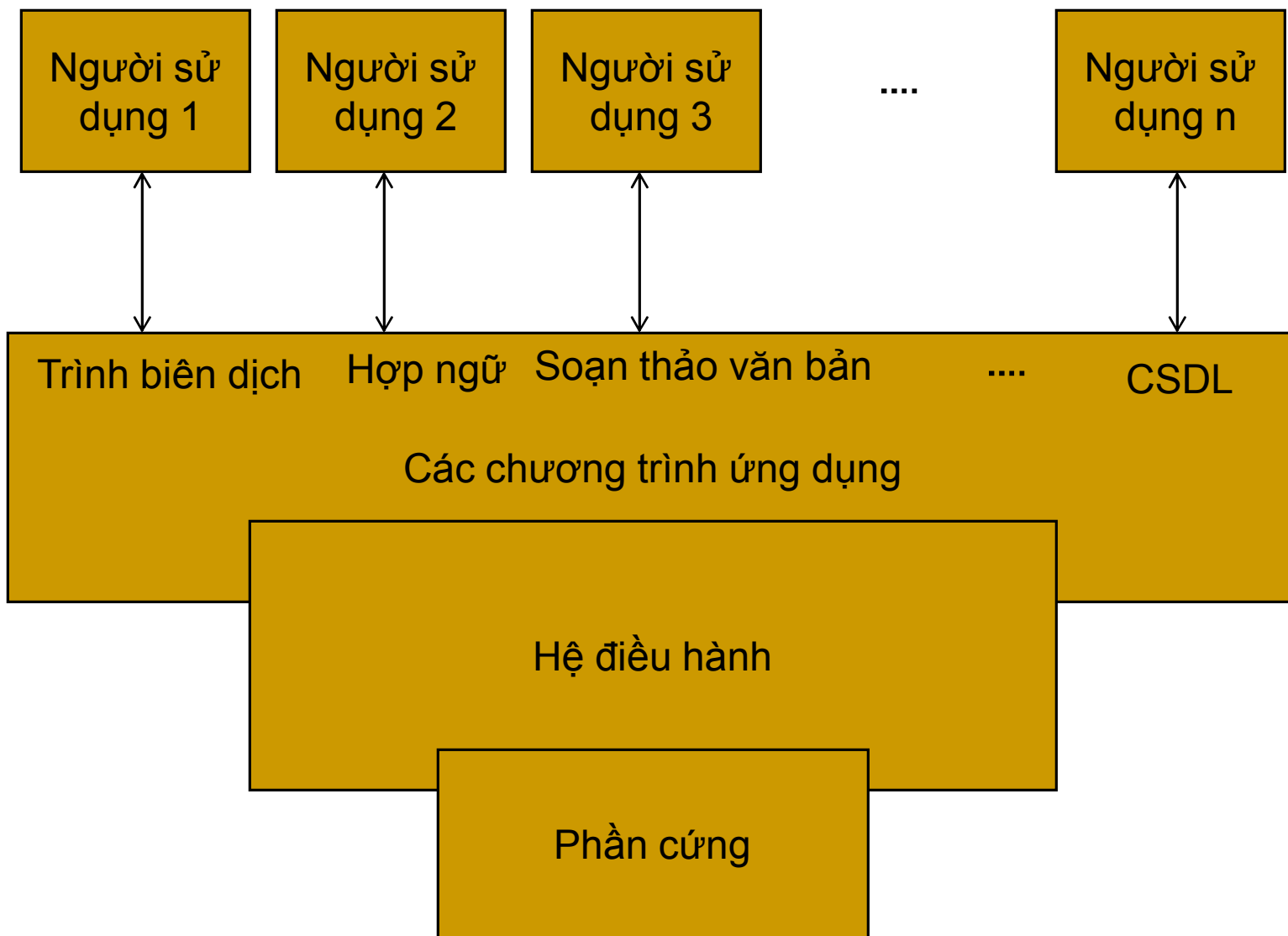
# 1.1 KHÁI NIỆM VỀ HĐH

- Hệ điều hành là một chương trình hay một hệ chương trình phần mềm máy tính, hoạt động ở lớp trung gian giữa người sử dụng và phần cứng máy tính
- Mục tiêu của HĐH là cung cấp môi trường để người sử dụng:
  - Thực thi dễ dàng các chương trình
  - Sử dụng máy tính trở nên dễ dàng, khai thác phần cứng máy tính một cách hiệu quả

---

# 1.1 KHÁI NIỆM VỀ HĐH(tt)

- HĐH là một bộ phận quan trọng của hệ thống máy tính. Một hệ thống máy tính bao gồm 4 phần:
  - Phần cứng: CPU; Bộ nhớ; Các thiết bị xuất/nhập
  - Các chương trình ứng dụng
  - Hệ điều hành
  - Đối tượng sử dụng: Người, thiết bị hoặc máy tính khác



#### 4 Thành phần của hệ thống máy tính

---

## 1.2 PHÂN LOẠI HĐH

- Hệ thống sử lý theo lô đơn giản
- Hệ thống sử lý theo lô đa chương
- Hệ thống chia sẻ thời gian
- Hệ thống song song
- Hệ thống phân tán
- Hệ thống xử lý thời gia thực
- V.v.

---

# HỆ THỐNG XỬ LÝ THEO LỘ ĐƠN GIẢN

- Các tác vụ được đưa vào hàng đợi
  - Thực hiện các tác vụ lần lượt theo những chỉ thị đã được xác định trước
  - Tác vụ tiếp theo tự động được thực hiện khi tác vụ trước kết thúc 1 cách tự động
  - Có bộ giám sát thường trực để giám sát việc thực hiện của các tác vụ trong hệ thống
- ➔ Processor rơi vào trạng thái chờ khi hệ thống truy xuất thiết bị vào ra

---

# HỆ THỐNG XỬ LÝ THEO LỘ ĐA CHƯƠNG

- Thực hiện được nhiều tác vụ đồng thời
- HĐH nạp 1 phần code và data của tác vụ vào bộ nhớ
- Khi có tác vụ đang sử dụng Processor thực hiện truy xuất thiết bị vào ra thì Processor sẽ được chuyển thực hiện tác vụ khác
- Cần có cơ chế lập lịch cho Processor



---

# HỆ THỐNG CHIA SẺ THỜI GIAN

- Các tác vụ, tiến trình được sử dụng Processor luân phiên nhau theo lịch phân chia thời gian sử dụng Processor đã được lập (t rất nhỏ)
- Cung cấp cho mỗi người sử dụng 1 phần nhỏ trong máy tính chia sẻ ->Người sử dụng có thể yêu cầu máy tính thực hiện đồng thời nhiều công việc
- Có cơ chế quản trị vào bảo vệ bộ nhớ, sử dụng bộ nhớ ảo

---

# HỆ THỐNG SONG SONG

- Có nhiều Processor trong cùng một hệ thống máy tính
- Các Processor cùng chia sẻ đường truyền dữ liệu, đồng hồ xung, bộ nhớ và các thiết bị ngoại vi
- Có 2 loại HĐH đa Processor:
  - Đa xử lý đối xứng (Symmetric multiprocessing-SMP)
  - Đa xử lý bất đối xứng (Asymmetric multiprocessing-ASMP)

---

# HỆ THỐNG SONG SONG(tt)

- Đa xử lý đối xứng:
  - Mỗi Processor chạy độc lập trên một bản sao HĐH như nhau
  - Cho phép nhiều tiến trình chạy đồng thời trên một hệ thống
- Đa xử lý bất đối xứng:
  - Mỗi Processor được giao một nhiệm vụ riêng biệt
  - Có một hoặc 2 Processor chủ làm nhiệm vụ lập lịch, xác định công việc cho các Processor thành viên

---

# HỆ THỐNG PHÂN TÁN

- Phân tán sự tính toán trên các bộ xử lý vật lý
- Mỗi bộ xử lý có bộ nhớ cục bộ riêng
- Các bộ xử lý thông tin với nhau thông qua các đường truyền thông tốc độ cao
- Có 2 dạng hệ thống: Client/Server và Peer-to-Peer

---

# HỆ THỐNG XỬ LÝ THỜI GIAN THỰC

- Có khả năng cho kết quả tức thời, chính xác sau mỗi tác vụ
- Tác vụ cần thực hiện không đưa vào hàng đợi mà xử lý tức thời và trả lại ngay kết quả chính xác trong khoảng thời gian bị thúc ép nhanh nhất

---

# 1.3 CẤU TRÚC CỦA HĐH

## 1.3.1 CÁC THÀNH PHẦN CỦA HĐH

- Quản lý tiến trình
- Quản lý bộ nhớ chính
- Quản lý xuất/nhập
- Quản lý bộ nhớ phụ
- Quản lý tập tin
- Thông dịch lệnh
- Bảo vệ hệ thống

---

# NHIỆM VỤ CỦA THÀNH PHẦN QUẢN LÝ TIẾN TRÌNH

- ❑ Tạo lập và hủy bỏ tiến trình
- ❑ Tạm dừng và kích hoạt lại tiến trình đã bị tạm dừng
- ❑ Tạo cơ chế thông tin liên lạc giữa các tiến trình
- ❑ Tạo cơ chế đồng bộ hóa giữa các tiến trình

---

## NHIỆM VỤ CỦA THÀNH PHẦN QUẢN LÝ BỘ NHỚ CHÍNH

- 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



---

# NHIỆM VỤ CỦA THÀNH PHẦN QUẢN LÝ XUẤT/NHẬP

- Làm cho các thao tác trao đổi thông tin trên các thiết bị nhập/xuất được trong suốt với người sử dụng
- Một hệ thống nhập/xuất bao gồm:
  - Hệ thống buffer caching.
  - Bộ giao tiếp điều khiển thiết bị.
  - Bộ điều khiển cho các thiết bị đặc thù.

---

## NHIỆM VỤ CỦA THÀNH PHẦN QUẢN LÝ BỘ NHỚ PHỤ

- Quản lý không gian trống trên đĩa
- Định vị lưu trữ thông tin trên đĩa
- Lập lịch cho vấn đề ghi/đọc thông tin trên đĩa của đầu từ

---

# NHIỆM VỤ CỦA THÀNH PHẦN QUẢN LÝ TẬP TIN

- Tạo/xóa tập tin, thư mục
- Bảo vệ tập tin khi có truy xuất đồng thời
- Cung cấp các thao tác xử lý và bảo vệ tập tin, thư mục
- Tạo cơ chế truy xuất tập tin thông qua tên tập tin,...

---

# NHIỆM VỤ CỦA THÀNH PHẦN THÔNG DỊCH LỆNH

- Đóng vai trò giao tiếp giữa HĐH và người sử dụng
- Một số HĐH thành phần này nằm trong nhân của nó, một số HĐH khác thiết kế dưới dạng 1 chương trình đặc biệt

---

# NHIỆM VỤ CỦA THÀNH PHẦN BẢO VỆ HỆ THỐNG

- Kiểm soát quá trình truy xuất của chương trình, tiến trình, hoặc người sử dụng với tài nguyên của hệ thống

---

## 1.3.2 CẤU TRÚC CỦA HĐH

### a. HỆ THỐNG ĐƠN KHỐI:

- Là một tập hợp các thủ tục, mỗi thủ tục có thể gọi thực hiện một thủ tục khác bất kỳ lúc nào cần thiết
- MS-DOS là một hệ điều hành có cấu trúc đơn giản, nó cung cấp những chức năng lớn nhất cho hệ thống tối thiểu

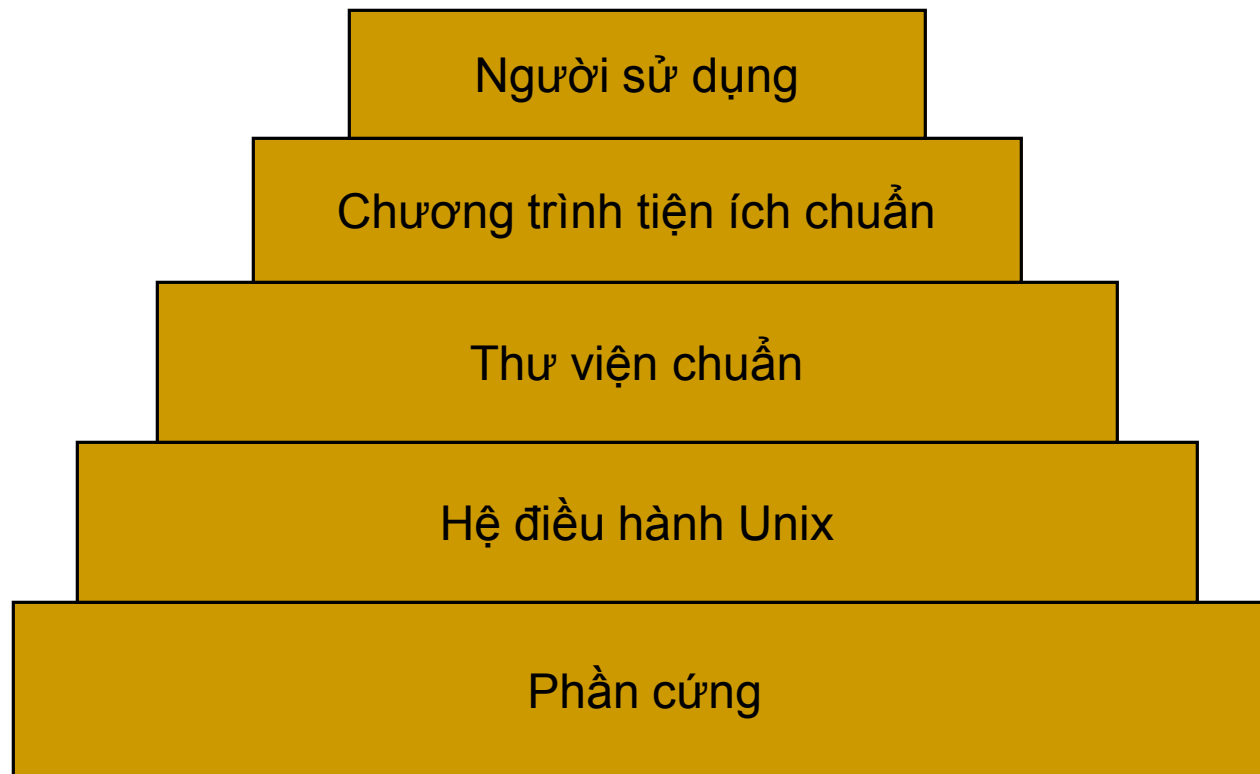
---

## 1.3.2 CẤU TRÚC CỦA HĐH(tt)

### b. HỆ THỐNG PHÂN LỚP:

- Hệ thống được chia thành một số lớp
- Mỗi lớp được xây dựng dựa trên một lớp bên dưới
- Lớp dưới cùng là phần cứng, lớp trên cùng là giao diện với người sử dụng

# Hệ thống phân lớp của UNIX



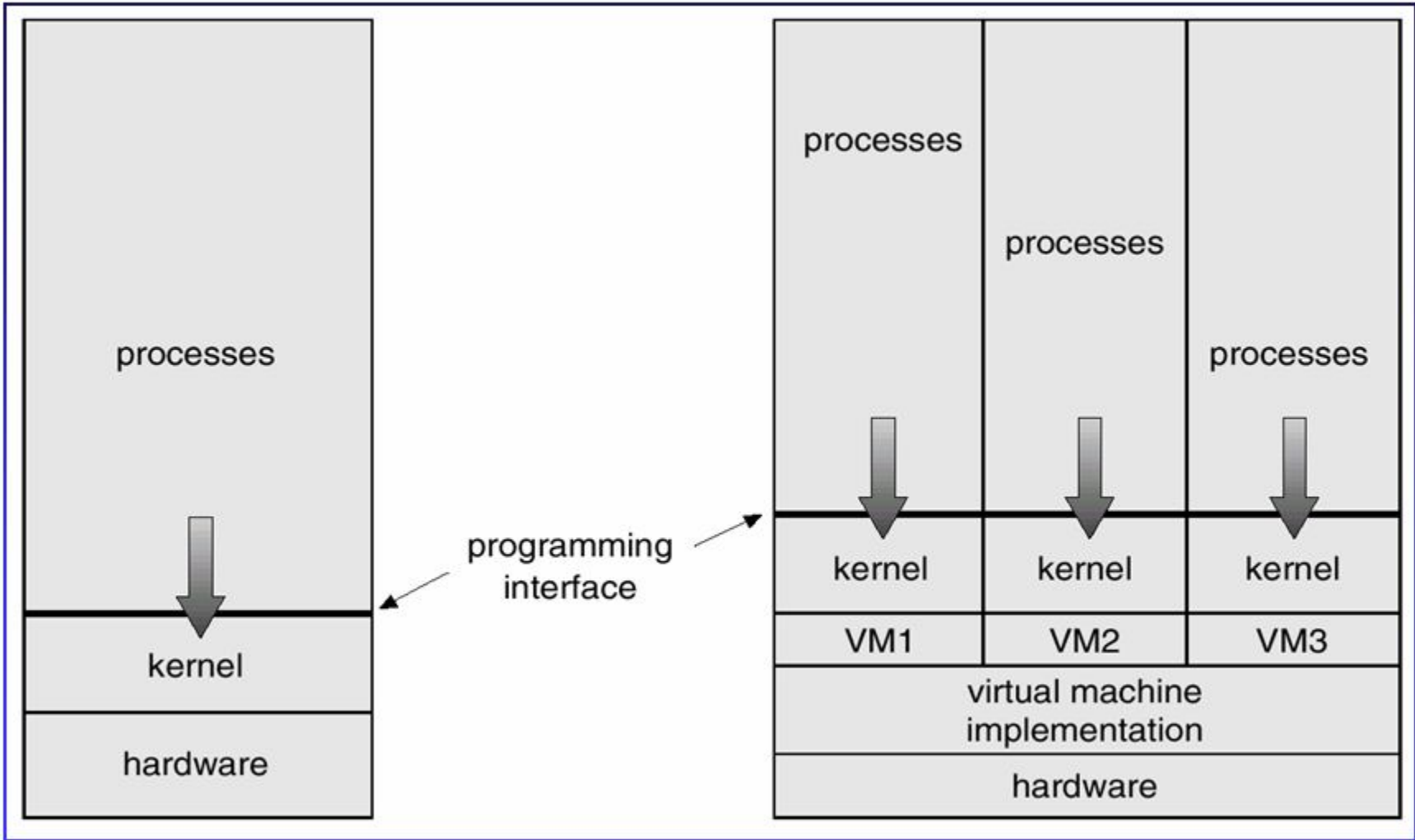


---

## 1.3.2 CẤU TRÚC CỦA HĐH(tt)

### c. MÁY ẢO (Virtual Machine)

- Là bản sao chính xác các đặc tính phần cứng của máy tính thực. Được cung cấp phần cứng và kernel của HĐH như máy thật
- Tài nguyên máy tính vật lý được chia sẻ để tạo ra các máy ảo
- Mỗi tiến trình được thực hiện trên một máy ảo độc lập



---

## 1.3.2 CẤU TRÚC CỦA HĐH(tt)

### d. MÔ HÌNH Client/Server:

- Mô hình này các tiến trình được chia thành 2 loại
  - Tiến trình Client: Là các tiến trình bên ngoài hay tiến trình của chương trình người sử dụng
  - Tiến trình Server: Là các tiến trình của HĐH
- Khi cần thực hiện 1 chức năng của hệ thống tiến trình client gửi yêu cầu đến tiến trình server tương ứng, tiến trình server xử lý và trả về cho client

---

## 1.4 LỊCH SỬ PHÁT TRIỂN CỦA HĐH

### 1. Thế hệ 1(1945-1955)

- Máy tính dùng ống chân không ra đời
- Vận hành máy tính cần 1 nhóm người: Thiết kế, xây dựng chương trình, thao tác, quản lý,...
- Chưa có khái niệm về ngôn ngữ lập trình và HĐH

---

## 1.4 LỊCH SỬ PHÁT TRIỂN CỦA HĐH(tt)

### 2. Thế hệ 2(1955-1965)

- Máy tính dùng bán dẫn ra đời
- Bộ phận sử dụng máy tính được phân chia rõ ràng: người thiết kế, người xây dựng, người lập trình, người vận hành,...
- Ngôn ngữ Assembly và Foxtran ra đời
- Chương trình được viết trên phiếu đục lỗ
- Hệ thống xử lý theo lô ra đời, các yêu cầu thực hiện được lưu trên băng từ, hệ thống đọc và thi hành lần lượt
- Hệ thống xử lý theo lô hoạt động dưới sự điều khiển của 1 chương trình đặc biệt

---

## 1.4 LỊCH SỬ PHÁT TRIỂN CỦA HĐH(tt)

### 3. Thế hệ 3(1965-1980)

- Máy tính được sử dụng rộng rãi
- Ra đời máy tính IBM 360 sử dụng mạch IC
- Thiết bị ngoại vi dùng cho máy tính xuất hiện ngày càng nhiều
- Các thao tác điều khiển máy tính ngày càng phức tạp
- HĐH ra đời nhằm điều phối, kiểm soát hoạt động của hệ thống và giải quyết các yêu cầu tranh chấp thiết bị
- Bắt đầu có khái niệm đa chương, chia sẻ thời gian thực và kỹ thuật spool
- Xuất hiện HĐH Multics và Unix

---

## 1.4 LỊCH SỬ PHÁT TRIỂN CỦA HĐH(tt)

### 4. Thế hệ 4(1980->)

- Máy tính cá nhân ra đời
- HĐH MS-DOS ra đời gắn liền với máy tính IBM PC
- Ra đời và phát triển nhiều HĐH gắn liền với sự phát triển của phần cứng máy tính

---

# BÀI TẬP

1. Hệ điều hành là :
  - a. Một chương trình
  - b. Một chương trình hay hệ chương trình
  - c. Một thiết bị
  - d. ROM-BIOS



---

## *2. Một hệ điều hành bao gồm:*

- a. Hệ thống quản lý tập tin, thiết bị I/O
- b. Hệ thống quản lý tiến trình, bộ nhớ
- c. a và b

---

# CHƯƠNG II: QUẢN LÝ TIẾN TRÌNH

---

---

# 1. TỔNG QUAN VỀ TIẾN TRÌNH

---

---

# 1.1 Tiến trình(process)?

- Tiến trình là một chương trình đang được thực thi, được sở hữu 1 con trỏ lệnh, tập các thanh ghi và các biến
  - Để hoàn thành tác vụ của mình, một tiến trình có thể cần đến một số tài nguyên như CPU, bộ nhớ chính, các tập tin và thiết bị nhập/xuất.
-

# 1.1 Tiến trình(process)?(tt)

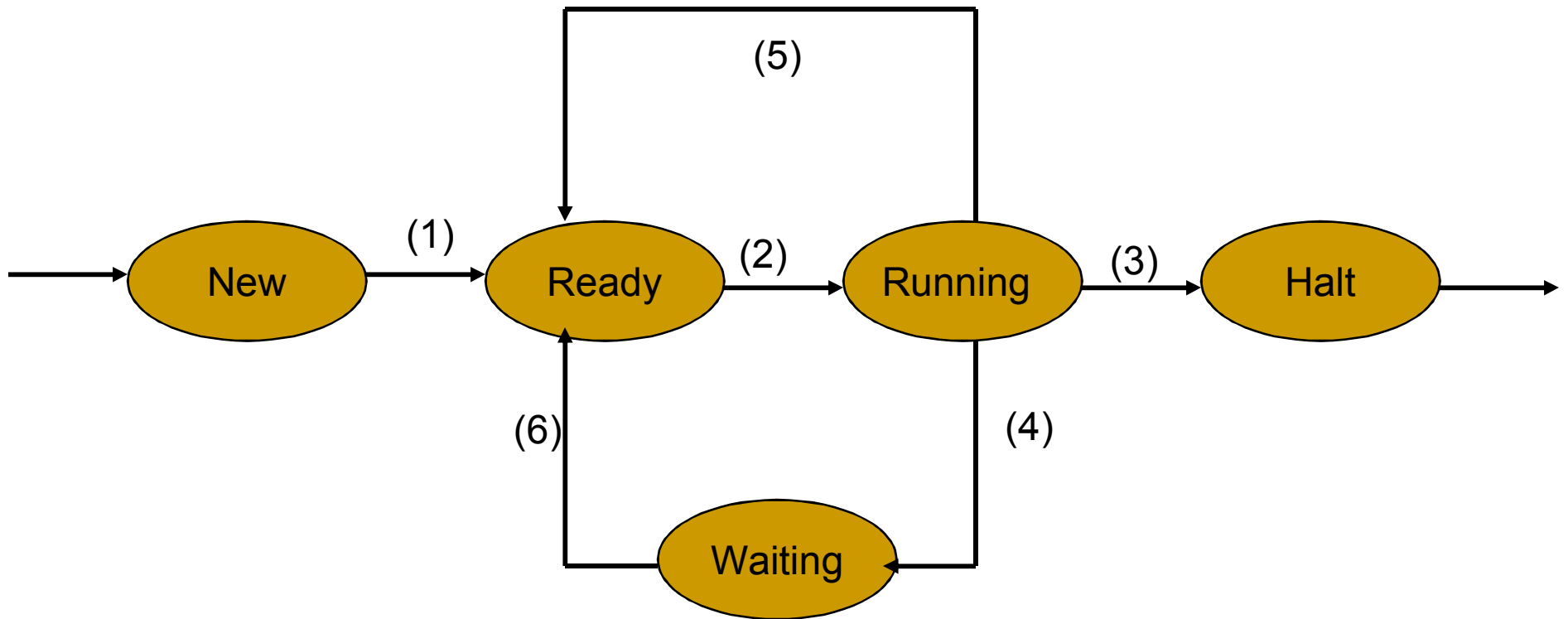
- Tiến trình bao gồm 3 thành phần: Code, Data, Stack
  - Code: Thành phần câu lệnh thực hiện
  - Data: Thành phần dữ liệu
  - Stack: Thành phần lưu thông tin tạm thời
- Các câu lệnh trong code chỉ dùng data và stack riêng của mình ngoại trừ các vùng dùng chung
- Tiến trình được hệ thống phân biệt bằng số hiệu pid (process identification)

---

## 1.2 Các trạng thái của tiến trình

- Trạng thái của tiến trình tại mỗi thời điểm được xác định bởi hoạt động hiện thời của nó:
    - New: tiến trình được tạo lập
    - Ready: tiến trình đã sẵn sàng, đang chờ cấp CPU
    - Running: tiến trình đang được xử lý
    - Waiting: tiến trình tạm dừng và chờ vì thiếu tài nguyên hay chờ 1 sự kiện nào đó
    - Halt: Tiến trình hoàn tất
-

# Mô tả chuyển trạng thái của tiến trình



---

## 1.2 Các trạng thái của tiến trình(tt)

- Tại một thời điểm chỉ có 1 tiến trình ở trạng thái Running trên 1 bộ xử lý bất kỳ và có thể có nhiều tiến trình ở trạng thái Ready và Waiting
-



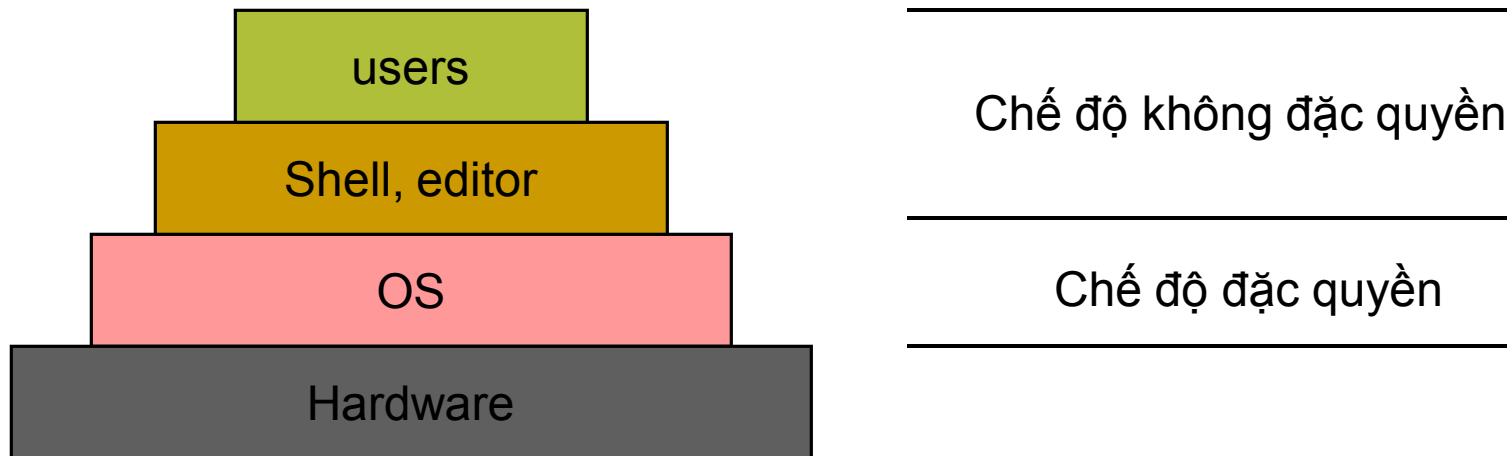
---

## 1.3 Chế độ xử lý của tiến trình

- Tiến trình của HĐH cần được bảo vệ khỏi sự xâm phạm của tiến trình khác
  - Chế độ xử lý được chia thành 2 chế độ nhờ sự hỗ trợ của phần cứng: Đặc quyền và không đặc quyền
  - Tiến trình của HĐH hoạt động trong chế độ đặc quyền và của người sử dụng hoạt động trong chế độ không đặc quyền
-

## 1.3 Chế độ xử lý của tiến trình(tt)

- Tập lệnh của CPU được chia thành 2 tập



## 1.4 Các thao tác điều khiển tiến trình

### a. Khởi tạo tiến trình

- HĐH gán PID và đưa vào danh sách quản lý của hệ thống
- Cấp phát không gian bộ nhớ
- Khởi tạo các thông tin cần thiết cho khối điều khiển tiến trình: Các PID của p cha (nếu có), thông tin trạng thái, độ ưu tiên, ngữ cảnh của processor
- Cung cấp đầy đủ các tài nguyên (trừ processor)
- Đưa tiến trình vào danh sách p nào đó: ready list, suspend list, waiting list

---

## 1.4. Các thao tác điều khiển tiến trình

b. Kết thúc tiến trình HĐH thực hiện các thao tác:

- Thu hồi tài nguyên đã cấp phát cho p
  - Loại bỏ tiến trình ra khỏi danh sách quản lý của hệ thống
  - Hủy bỏ khối điều khiển p
-

---

## 1.4. Các thao tác điều khiển tiến trình

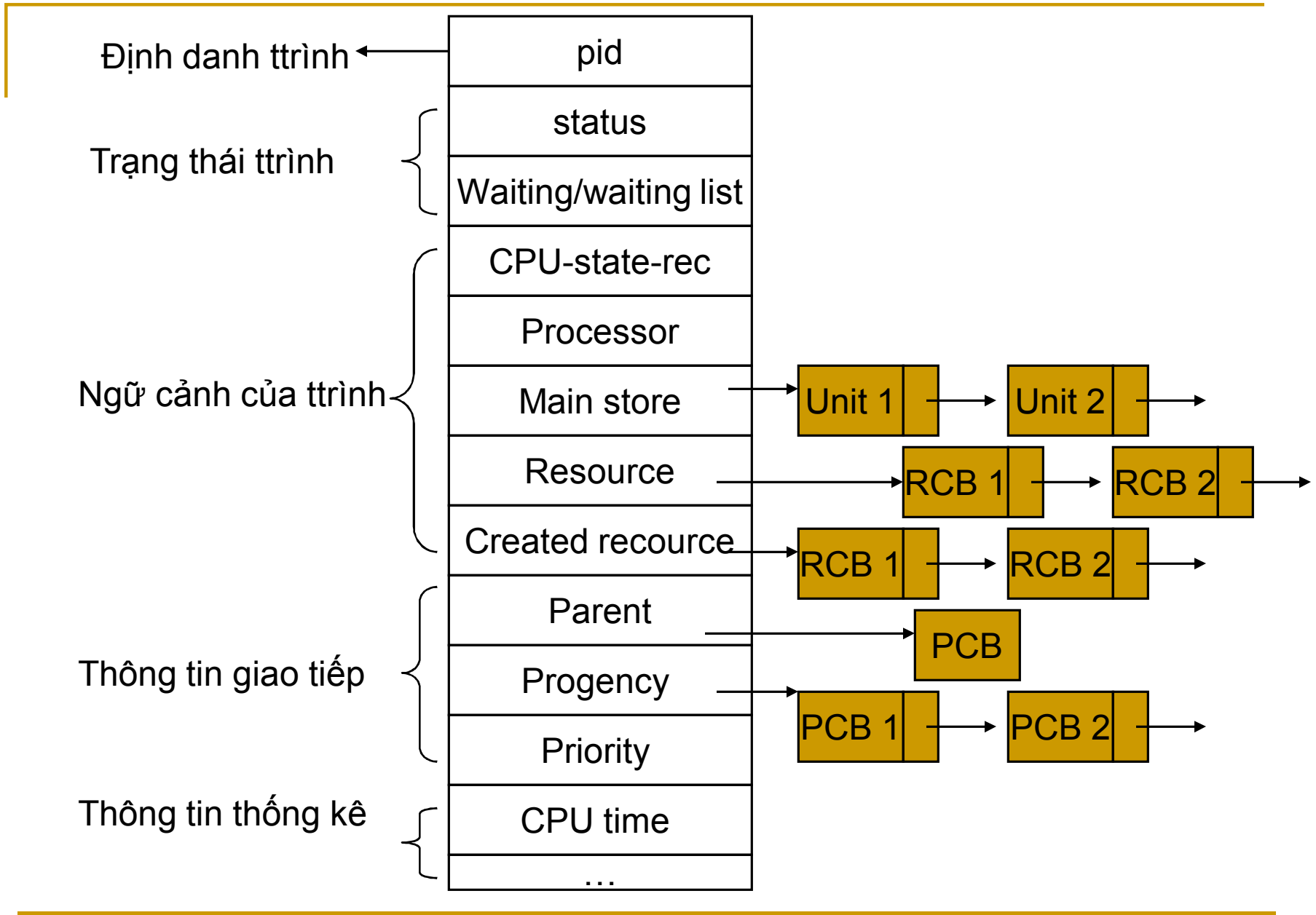
c. Thay đổi trạng thái của p HĐH thực hiện:

- Lưu ngữ cảnh của processor
  - Cập nhật PCB (process control block) của tiến trình sao cho phù hợp với trạng thái của p
  - Di chuyển PCB của p đến 1 hàng đợi thích hợp
  - Chọn tiến trình khác để cho phép nó thực hiện
  - Cập nhật PCB của p vừa thực hiện
  - Cập nhật thông tin liên quan đến quản lý bộ nhớ
  - Khôi phục lại ngữ cảnh của processor
-

---

## 1.5 Khối điều khiển tiến trình(process control block -PCB).

- Quản lý mọi hoạt động của tiến trình
  - Cấu trúc dữ liệu của khối điều khiển bao gồm:
    - Định danh tiến trình
    - Trạng thái của tiến trình
    - Ngưỡng cảnh của tiến trình
    - Thông tin giao tiếp
    - Thông tin thống kê
-



---

# 1.6 Tiểu trình(thread)

- Thông thường mỗi tiến trình có 1 không gian địa chỉ và 1 dòng xử lý
  - Mong muốn có nhiều dòng xử lý cùng chia sẻ 1 không gian địa chỉ và các dòng xử lý hoạt động song song như các tiến trình độc lập
  - Xuất hiện HĐH có cơ chế thực thi mới gọi là tiểu trình
  - Như vậy, tiểu trình là:
    - 1 đơn vị xử lý cơ bản
    - Sở hữu 1 con trỏ lệnh, tập các thanh ghi, 1 vùng nhớ stack riêng
    - Có các trạng thái như tiến trình thật.
-



---

## 2. TÀI NGUYÊN GẮNG VÀ ĐOẠN GẮNG

---

---

## 2.1 Tài nguyên găng(Critical Resource)

- Tài nguyên găng?
    - Những tài nguyên được HĐH chia sẻ cho nhiều tiến trình hoạt động đồng thời dùng chung mà có nguy cơ tranh chấp giữa các tiến trình này khi sử dụng chúng
  - Tài nguyên găng có thể là tài nguyên phần cứng hoặc phần mềm, có thể là tài nguyên phân chia được hoặc không phân chia được
-

---

## 2.1 Tài nguyên găng(Critical Resource)

- Ví dụ: bài toán rút tiền ngân hàng từ tài khoản dùng chung

If (tài khoản – tiền rút  $\geq 0$ )

    tài khoản:=tài khoản – tiền rút

Else

    Thông báo lỗi

endif

---

---

## 2.2 Đoạn găng(Critical Section)

- Các đoạn code trong các chương trình dùng để truy cập đến tài nguyên găng được gọi là đoạn găng
  - Để hạn chế lỗi có thể xảy ra do sử dụng tài nguyên găng, tại 1 thời điểm HĐH chỉ cho 1 tiến trình nằm trong đoạn găng
  - HĐH có cơ chế điều độ tiến trình qua đoạn găng
-

---

## 2.3 Yêu cầu của công tác điều độ tiến trình qua đoạn găng

- Tại 1 thời điểm chỉ cho phép 1 tiến trình nằm trong đoạn găng, các tiến trình khác có nhu cầu vào đoạn găng phải chờ
  - Tiến trình chờ ngoài đoạn găng không được ngăn cản các tiến trình khác vào đoạn găng
  - Không có tiến trình nào phải chờ lâu để được vào đoạn găng
  - Đánh thức các tiến trình trong hàng đợi để tạo điều kiện cho nó vào đoạn găng khi tài nguyên găng được giải phóng
-

---

## 2.4 Điều độ tiến trình qua đoạn găng

a. Giải pháp phần cứng

- Dùng cặp chỉ thị STI(setting interrupt) và CLI (clean interrupt)

Ví dụ:

Procedure P(i: integer)

begin

repeat

CLI;

<đoạn găng của p>;

STI;

<Đoạn không găng>;

until .F.

end;

---

---

- Dùng chỉ thị TSL(Test and set)

Function TestAnhSetLock(Var i:integer):boolean

Begin

  if i=0 then

    begin

      i:=1;

      TestAnhSetLock:=true

    end;

  else

    TestAnhSetLock:=false

End;

---

---

```
Procedure P(lock: integer);  
  begin  
    repeat  
      while (TestAnhSetLock(lock)) do;  
        <đoạn găng của p>;  
        lock:=0  
        <Đoạn không găng>;  
    until .F.  
  end;
```

---



---

## b. Giải pháp dùng biến khóa

### ■ Dùng biến khóa chung

```
Procedure P(lock: integer);  
  begin  
    repeat  
      while lock=1 do;  
        Lock=1  
        <đoạn găng của p>;  
        lock:=0  
        <Đoạn không găng>;  
    until .F.  
  end;
```

---

---

- Dùng biến khóa riêng

```
Var lock1, lock2: byte;
```

```
begin
```

```
lock1:=0; lock2:=1
```

```
    p1: repeat
```

```
        while lock2=1 do;
```

```
            Lock1:=1
```

```
            <đoạn găng của p>;
```

```
            lock1:=0
```

```
            <Đoạn không găng>;
```

```
        until .F.
```

```
    p2: repeat
```

```
        while lock1=1 do;
```

```
            Lock2:=1
```

```
            <đoạn găng của p>;
```

```
            lock2:=0
```

```
            <Đoạn không găng>;
```

```
        until .F.
```

```
end
```

---

## C. Giải pháp được hỗ trợ bởi HĐH và ngôn ngữ lập trình

### ■ Dùng Semaphore(đèn báo)

- Semaphore S là 1 biến nguyên, khởi gán bằng 1 giá trị không âm, là khả năng phục vụ của tài nguyên găng tương ứng với nó
- Ứng với S có 1 hàng đợi F(s) lưu các tiến trình đang chờ trên S
- Thao tác Down giảm S 1 đơn vị, Up tăng S 1 đơn vị
- Mỗi tiến trình trước khi vào đoạn găng cần gọi Down để giảm S và kiểm tra nếu  $S \geq 0$  thì được vào đoạn găng
- Mỗi tiến trình khi ra khỏi đoạn găng phải gọi Up để tăng S lên 1 đơn vị và ktra nếu  $S \leq 0$  thì đưa 1 tiến trình trong F(s) vào đoạn găng

---

Procedure Down(S);

Begin

  S:=S-1;

  If s<0 then

    Begin

      Status(p)=waiting;

      Enter(p,F(s));

    end

End;

---

---

```
Procedure Up(S);  
Begin  
  S:=S+1;  
  If s<0 then  
    Begin  
      Exit(Q,F(S));  
      Status(Q)=ready;  
      Enter(Q,ready-list);  
    end  
  End;
```

---

---

# 3. TẮC NGHẼN VÀ CHỐNG TẮC NGHẼN

---

---

## 3.1 Tắc nghẽn

- Sự xung đột về tài nguyên của các tiến trình hoạt động đồng thời trong hệ thống
  - Tắc nghẽn thường xảy ra với xung đột tài nguyên không phân chia được, ít xảy ra với tài nguyên phân chia được
-

---

## 3.2 Điều kiện hình thành tắc nghẽn

- Sử dụng tài nguyên không thể chia sẻ
  - Chiếm giữ và yêu cầu tài nguyên
  - Không thu hồi tài nguyên từ tiến trình đang chiếm giữ chúng
  - Đợi vòng tròn
-



---

## 3.3 Các mức phòng tránh tắc nghẽn

- Ngăn ngừa
  - Dự báo và tránh tắc nghẽn
  - Nhận biết và khắc phục
-

---

# 4. ĐIỀU PHỐI TIẾN TRÌNH

---

---

## 4.1 Mục tiêu điều phối

- Sự công bằng
  - Tính hiệu quả
  - Thời gian đáp ứng hợp lý
  - Thời gian lưu lại trong hệ thống
  - Thông lượng tối đa
-

---

## 4.2 Cơ chế điều phối

- Độc quyền: Tiến trình toàn quyền sử dụng processor cho đến khi kết thúc hoặc tự động trả lại
    - Quyết định điều phối khi tiến trình chuyển từ Running sang Waiting (blocked) hoặc kết thúc
  - Không độc quyền: Tiến trình đang xử lý thì bị thu hồi processor để cấp cho tiến trình khác
    - Quyết định điều phối khi tiến trình chuyển từ Running sang Waiting (blocked) hoặc ready hoặc kết thúc hoặc từ Waiting sang ready
-

---

## 4.3 Các đặc điểm của tiến trình

- Tính hướng xuất nhập
  - Tính hướng xử lý
  - Tương tác hay xử lý theo lô
  - Độ ưu tiên của tiến trình
  - Thời gian sử dụng CPU
  - Thời gian còn lại để tiến trình hoàn tất
-

---

## 4.4 Tổ chức điều phối

- HĐH sử dụng 2 loại danh sách để tổ chức lưu trữ các tiến trình:
    - Danh sách Ready: Chỉ tồn tại 1 danh sách này
    - Danh sách Waiting: Có thể tồn tại nhiều danh sách này
-

## 4.5 Các chiến lược điều phối

- Chiến lược FIFO: Tiến trình nào được đưa vào danh sách ready trước sẽ được cấp Processor trước
- Ví dụ

Tiến trình	Thời điểm vào	t/g xử lý
P1	0	24
P2	1	3
P3	2	3

Thời điểm cấp processor		
P1	P2	P3
0	24	27

Thời gian chờ:

P1: 0

P2: 23

P3: 25

## 4.5 Các chiến lược điều phối

- Chiến lược phân phối xoay vòng:
  - Tiến trình nào vào danh sách Ready trước được cấp processor trước
  - Mỗi tiến trình chỉ được sử dụng processor trong 1 khoản thời gian bằng nhau được gọi là Quantum

- Ví dụ

Tiến trình	Thời điểm vào	t/g xử lý
P1	0	24
P2	1	3
P3	2	3
Quantum=4		



---

Tiến trình	P1	P2	P3	P1	P1	P1	P1	
Thời điểm	0	4	7	10	14	18	22	

---

---

## 4.5 Các chiến lược điều phối

- Chiến lược theo độ ưu tiên:
    - Mỗi tiến trình được gán cho một độ ưu tiên tương ứng, tiến trình có độ ưu tiên cao nhất sẽ được chọn để cấp phát CPU đầu tiên
    - Độ ưu tiên của tiến trình do HĐH gán và có thể bị thay đổi
    - Giải thuật điều phối với độ ưu tiên có thể theo nguyên tắc độc quyền hay không độc quyền
    - Điều phối với độ ưu tiên và không độc quyền sẽ thu hồi processor từ tiến trình hiện hành để cấp cho tiến trình mới nếu độ ưu tiên của tiến trình này cao hơn
    - Điều phối với độ ưu tiên và độc quyền sẽ chỉ chèn tiến trình mới vào danh sách sẵn sàng tại vị trí phù hợp
-

## 4.5 Các chiến lược điều phối

Ví dụ

Tiến trình	Độ ưu tiên	t/g xử lý
P1	3	24
P2	2	3
P3	1	3

Thời điểm cấp processor		
P1	P2	P3
0	24	27

Nhược điểm: Tiến trình có độ ưu tiên thấp dễ rơi vào trạng thái chờ vô hạn

=> Cần giảm độ ưu tiên của tiến trình sau mỗi lần được cấp processor

---

## 4.5 Các chiến lược điều phối

- Chiến lược công việc ngắn nhất (shortest job first - SJF):
    - Đây là một trường hợp đặc biệt của giải thuật điều phối với độ ưu tiên
    - độ ưu tiên  $p$  được gán cho mỗi tiến trình là nghịch đảo của thời gian xử lý  $t$  mà tiến trình yêu cầu :  $p = 1/t$
    - CPU được sẽ được cấp phát cho tiến trình yêu cầu ít thời gian nhất để kết thúc tiến trình
    - Giải thuật này cũng có thể độc quyền hoặc không độc quyền
-

---

## 4.5 Các chiến lược điều phối

- Chiến lược nhiều cấp độ ưu tiên
    - Phân lớp các tiến trình tùy theo độ ưu tiên của chúng để có cách thức điều phối thích hợp cho từng nhóm
    - Mỗi danh sách bao gồm các tiến trình có cùng độ ưu tiên và được áp dụng một giải thuật điều phối thích hợp để điều phối
    - Ngoài ra, còn có một giải thuật điều phối giữa các nhóm, thường giải thuật này là giải thuật không độc quyền và sử dụng độ ưu tiên cố định
    - Một tiến trình thuộc về danh sách ở cấp ưu tiên  $i$  sẽ chỉ được cấp phát CPU khi các danh sách ở cấp ưu tiên lớn hơn  $i$  đã trống
-

---

# CHƯƠNG III: QUẢN LÝ BỘ NHỚ

---

---

# 1. TỔNG QUAN

---

---

## 1.1 Vì sao phải tổ chức, quản lý bộ nhớ?

- CPU chỉ có thể trao đổi thông tin với bộ nhớ chính
  - Các chương trình muốn được thực thi cần được nạp vào bộ nhớ chính, tạo lập tiến trình tương ứng để xử lý
  - Các hệ thống đa chương trên bộ nhớ chính ngoài HĐH có thể có nhiều tiến trình đang hoạt động
  - Kích thước bộ nhớ chính là hữu hạn nhưng yêu cầu bộ nhớ thì vô hạn
  - ...
-



---

## 1.1 Vì sao phải tổ chức, quản lý bộ nhớ?

- Như vậy, HĐH cần phải tổ chức quản lý bộ nhớ một cách hợp lý để có thể:
    - Đưa bất kỳ một tiến trình nào đó vào bộ nhớ khi có yêu cầu, cho dù khi trên bộ nhớ không còn không gian trống
    - Bảo vệ các tiến trình của hệ điều hành và các tiến trình trên bộ nhớ, tránh các trường hợp truy xuất bất hợp lệ xảy ra.
-

---

## 1.2 Nhiệm vụ của bộ phận quản lý bộ nhớ

- Tái định vị
  - Bảo vệ bộ nhớ
  - Chia sẻ bộ nhớ
  - Tổ chức bộ nhớ logic
  - Tổ chức bộ nhớ vật lý
-

---

# Tái định vị

- Trong các hệ thống đa chương không gian bộ nhớ chính thường được chia sẻ cho nhiều tiến trình và yêu cầu bộ nhớ của các tiến trình luôn lớn hơn không gian bộ nhớ vật lý mà tiến trình mà hệ thống hiện có
    - ⇒ Cần thực hiện cơ chế hoán đổi (Swap):
      - ⇒ Một chương trình đang hoạt động trên bộ nhớ sẽ bị đưa ra đĩa (swap-out) và sẽ được đưa vào lại (swap-in) tại thời điểm thích hợp
-

---

# Tái định vị(tt)

- Khi thực hiện swap-in 1 chương trình vào lại bộ nhớ HĐH phải định vị nó đúng vào vị trí mà trước khi nó bị swap-out
  - ⇒ HĐH phải có cơ chế ghi lại tất cả các thông tin liên quan đến 1 chương trình bị swap-out. Các thông tin này là cơ sở để hệ điều hành swap-in chương trình vào lại bộ nhớ chính và cho nó tiếp tục hoạt động.
-

---

# Bảo vệ bộ nhớ

- Mỗi tiến trình phải được bảo vệ để chống lại sự truy xuất bất hợp lệ vô tình hay có chủ ý của các tiến trình khác.
  - Mỗi tiến trình chỉ được phép truy suất đến không gian địa chỉ mà HĐH đã cấp cho nó
    - ⇒ Bộ phận Qlý bộ nhớ phải biết không gian địa chỉ của tất cả các tiến trình trên bộ nhớ
    - ⇒ Khi tiến trình đưa ra địa chỉ truy xuất bộ phận Qlý bộ nhớ phải kiểm tra tất cả các yêu cầu truy xuất bộ nhớ của mỗi
-

---

# Chia sẻ bộ nhớ

- Bất kỳ một chiến lược nào được cài đặt đều phải có tính mềm dẻo để cho phép nhiều tiến trình có thể truy cập đến cùng một địa chỉ trên bộ nhớ chính
-

---

# Tổ chức bộ nhớ logic

- Bộ nhớ chính của hệ thống máy tính được tổ chức như là một dòng hoặc một mảng
  - Không gian địa chỉ bao gồm một dãy có thứ tự các byte hoặc các word.
  - Bộ nhớ phụ cũng được tổ chức tương tự
  - Cách tổ chức này có sự kết hợp chặt chẽ với phần cứng máy tính nhưng lại không phù hợp với cách xây dựng của chương trình
  - Đa số các chương trình được tổ chức thành các modul
-

---

# Tổ chức bộ nhớ vật lý

- Bộ nhớ máy tính được tổ chức theo 2 cấp:
    - Bộ nhớ chính: tốc độ truy xuất nhanh, nhưng giá thành cao và dữ liệu không thể tồn tại lâu dài trên nó.
    - Bộ nhớ phụ: giá rẻ, dung lượng lớn, dữ liệu được lưu trữ lâu dài nhưng tốc độ truy xuất chậm.
  - ⇒ Theo giản đồ 2 cấp này, việc tổ chức luồng thông tin giữa bộ nhớ chính và bộ nhớ phụ là nhiệm vụ quan trọng của hệ thống
-



---

## 1.3 Không gian địa chỉ và không gian vật lý

- Địa chỉ logic: còn gọi là địa chỉ ảo, là tất cả các địa chỉ do bộ xử lý tạo ra.
  - Địa chỉ vật lý: là địa chỉ thực tế mà trình quản lý bộ nhớ nhìn thấy và thao tác.
  - Không gian địa chỉ: là tập hợp tất cả các địa chỉ ảo phát sinh bởi một chương trình.
  - Không gian vật lý: là tập hợp tất cả các địa chỉ vật lý tương ứng với các địa chỉ ảo
-

---

## 1.4 Các cấu trúc chương trình

- Cấu trúc chương trình tuyến tính
  - Cấu trúc chương trình động
  - Cấu trúc chương trình Overlay
  - Cấu trúc chương trình phân trang
  - Cấu trúc chương trình phân đoạn
-

---

# Cấu trúc chương trình tuyến tính

- Tất cả các modun, thư viện sử dụng trong chương trình khi biên dịch sẽ được biên dịch thành 1 modun duy nhất
  - Khi thực hiện HĐH phải nạp toàn bộ modun này vào bộ nhớ
  - Cấu trúc chương trình này có tính độc lập cao và có tốc độ thực thi cao
  - Làm lãng phí bộ nhớ vì kích thước chương trình tăng lên khi biên dịch
-

---

# Cấu trúc chương trình động

- Chương trình được viết dưới dạng các modun riêng rẽ
  - Được biên dịch thành các modun riêng rẽ, các thư viện chuẩn của HĐH và của NNlập trình không được tích hợp trong modun chính của chương trình
  - Khi thực thi chương trình chỉ 1 modun chính được nạp vào bộ nhớ, các modun khác khi cần sẽ được nạp vào sau
  - Cấu trúc này tiết kiệm được không gian nhớ nhưng thực thi chậm hơn cấu trúc tuyến tính
-

---

# Cấu trúc chương trình Overlay

- Chương trình được biên dịch thành các modul riêng rẽ
  - Các modul chương trình được chia thành các mức khác nhau:
    - Mức 0: Chứa modul gốc dùng để nạp chương trình
    - Mức 1: Chứa các modul được gọi bởi mức 0
    - Mức 2: Chứa các modul được gọi bởi mức 1
    - ...
    - Mức  $i$ : Chứa các modul được gọi bởi mức  $i-1$
-

---

# Cấu trúc chương trình Overlay(tt)

- Các modun trong cùng một mức có thể có kích thước khác nhau, kích thước của modun lớn nhất trong lớp được xem là kích thước của mức
  - Bộ nhớ dành cho chương trình cũng được tổ chức thành các mức tương ứng với các chương trình
  - Khi thực hiện chương trình HĐH nạp sơ đồ overlay của chương trình vào bộ nhớ sau đó nạp các modun cần thiết ban đầu vào bộ nhớ
  - HĐH dựa vào sơ đồ overlay để nạp các modun khác nếu cần
-

---

# Cấu trúc chương trình phân trang

- Các modul chương trình được biên dịch thành 1 modul duy nhất nhưng sau đó được chia thành các phần có kích thước bằng nhau được gọi là các trang
  - Bộ nhớ phải được phân trang, tức chia thành các không gian nhớ bằng nhau gọi là khung trang
  - HĐH phải xây dựng bộ điều khiển trang(PCT-page control table)
-

---

# Cấu trúc chương trình phân đoạn

- Chương trình được biên dịch thành nhiều modun độc lập, được gọi là các đoạn
  - Bộ nhớ phải được phân đoạn, tức chia thành các không gian có kích thước có thể không bằng nhau tương ứng với kích thước của các đoạn chương trình
  - Khi thực hiện chương trình HĐH có thể nạp tất cả các đoạn hoặc 1 vài đoạn cần thiết vào các phân đoạn nhớ liên tiếp hoặc k liên tiếp
  - HĐH phải xây dựng bộ điều khiển đoạn(SCT-Segment control table)
-



---

## 2. KỸ THUẬT CẤP PHÁT BỘ NHỚ

---

---

## 2.1 Kỹ thuật phân vùng cố định

- Không gian địa chỉ được chia thành 2 vùng cố định
    - Vùng địa chỉ thấp dùng để chứa HĐH
    - Vùng còn lại (tạm gọi là user program) cấp cho các tiến trình được nạp vào bộ nhớ chính
-

---

## 2.1 Kỹ thuật phân vùng cố định(tt)

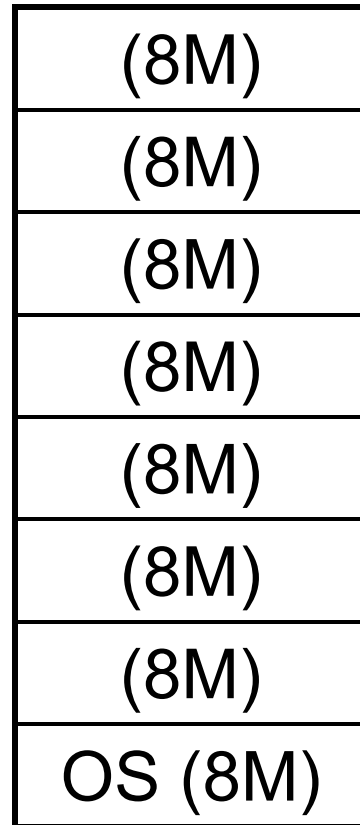
- Với hệ thống đơn chương:
    - Việc quản lý bộ nhớ đơn giản vì vùng nhớ user program chỉ cấp cho 1 chương trình
    - HĐH sử dụng 1 thanh ghi giới hạn để ghi địa chỉ ranh giới giữa HĐH và chương trình người sử dụng
    - Khi chương trình người sử dụng đưa ra địa chỉ cần truy xuất, HĐH sẽ so sánh với giá trị giới hạn được ghi trong thanh ghi giới hạn
      - Nếu nhỏ hơn giá trị giới hạn thì HĐH từ chối việc truy xuất
      - Ngược lại, nếu lớn hơn sẽ cho phép truy xuất
-

---

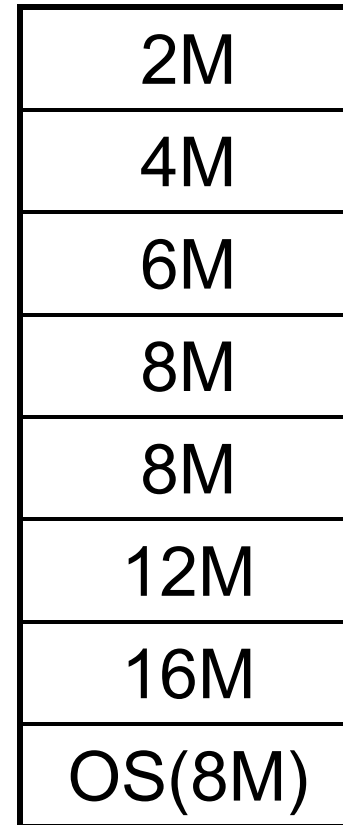
## 2.1 Kỹ thuật phân vùng cố định(tt)

- Với hệ thống đa chương:
    - Vùng nhớ user program được chia n phần không nhất thiết phải bằng nhau. Mỗi phần được gọi là 1 phân vùng
    - Mỗi tiến trình có thể được nạp vào 1 phân vùng bất kỳ nếu kích thước của nó  $\leq$  kích thước của phân vùng và phân vùng này còn trống
    - Khi có tiến trình cần được nạp vào bộ nhớ mà không còn phân vùng trống thì HĐH sẽ swap-out 1 tiến trình tại 1 phân vùng nào đó có kích thước vừa đủ, không chứa tiến trình đang ở trạng thái ready hoặc running và không có quan hệ với tiến trình đang ở trạng thái running khác để nạp tiến trình vừa có yêu cầu
-

## 2.1 Kỹ thuật phân vùng cố định(tt)



Phân vùng kích thước  
bằng nhau



Phân vùng kích thước  
không bằng nhau

Hình 3.1 Ví dụ về phân vùng cố định của bộ nhớ 64MByte

## 2.1 Kỹ thuật phân vùng cố định(tt)

- Có 2 khó khăn với việc dùng phân vùng cố định có kích thước bằng nhau
  - Thứ 1: Nếu chương trình có kích thước quá lớn so với 1 kích thước của phân vùng, để giải quyết việc này thì:
    - Người lập trình phải thiết kế chương trình theo cấu trúc overlay
    - Chỉ 1 phần cần thiết của chương trình mới được nạp vào bộ nhớ lúc nạp chương trình. Khi cần module nào đó mà không sẵn có trong bộ nhớ người sử dụng phải nạp nó vào đúng phân vùng của chương trình và sẽ ghi đè lên bất kỳ chương trình hoặc dữ liệu ở trong đó

---

## 2.1 Kỹ thuật phân vùng cố định(tt)

- Thứ 2: Khi kích thước của chương trình nhỏ hơn kích thước của 1 phân vùng hoặc lớn hơn kích thước của phân vùng nhưng không phải là bội số của kích thước phân vùng. Điều này gây ra sự ***phân mảnh nội vi***, lãng phí bộ nhớ
-

---

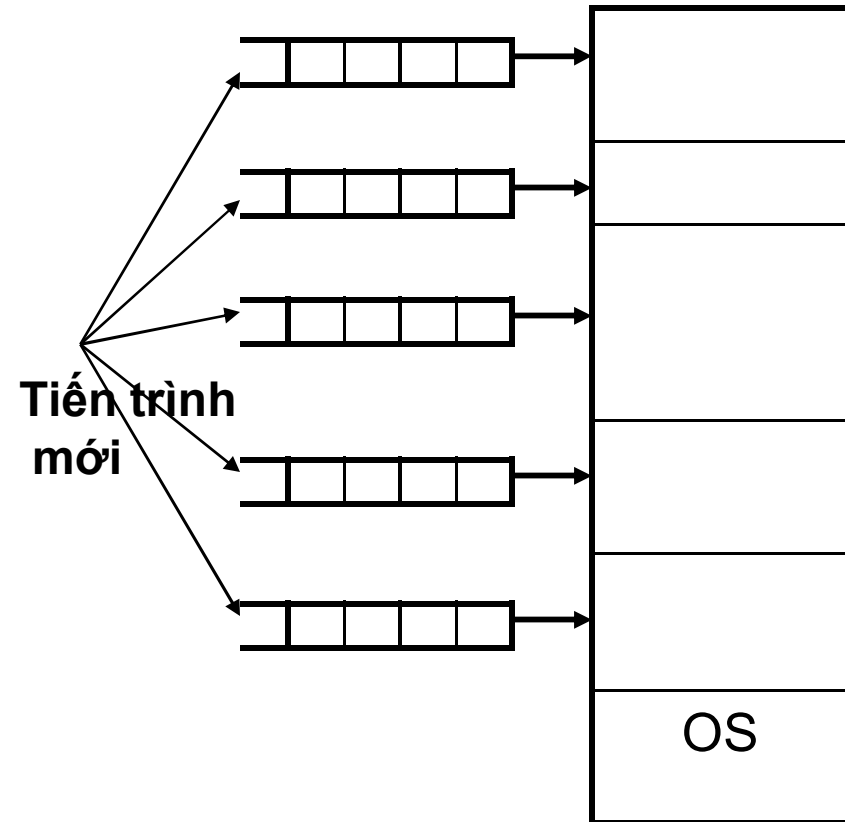
## 2.1 Kỹ thuật phân vùng cố định(tt)

- Để khắc phục nhược điểm này có thể sử dụng phân vùng cố định có kích thước không bằng nhau
  - Có 2 lựa chọn để đưa tiến trình vào dạng phân vùng này
-



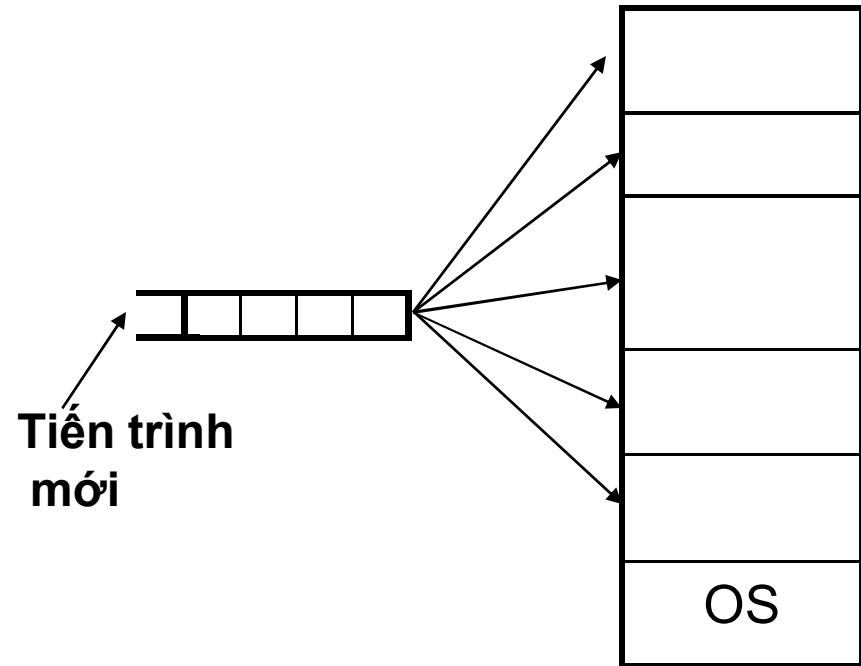
## 2.1 Kỹ thuật phân vùng cố định(tt)

- Lựa chọn 1:
  - Mỗi phân vùng có một hàng đợi tương ứng
  - Khi 1 tiến trình cần được nạp vào bộ nhớ sẽ đưa vào hàng đợi của phân vùng có kích thước vừa đủ để chứa nó để được đưa vào phân vùng
  - Nhược điểm: Có thể có phân vùng đang trống nhưng lại có nhiều tiến trình đang chờ để vào phân vùng khác



## 2.1 Kỹ thuật phân vùng cố định(tt)

- Lựa chọn 2:
  - Dùng 1 hàng đợi chung cho tất cả các phân vùng
  - Khi có tiến trình muốn nạp vào bộ nhớ nhưng chưa được nạp sẽ được đưa vào hàng đợi
  - Khi có phân vùng trống, HĐH sẽ chọn tiến trình có kích thước vừa đủ để đưa vào phân vùng
  - Phương pháp này gây khó khăn trong việc lựa chọn tiến trình để nạp vào phân vùng



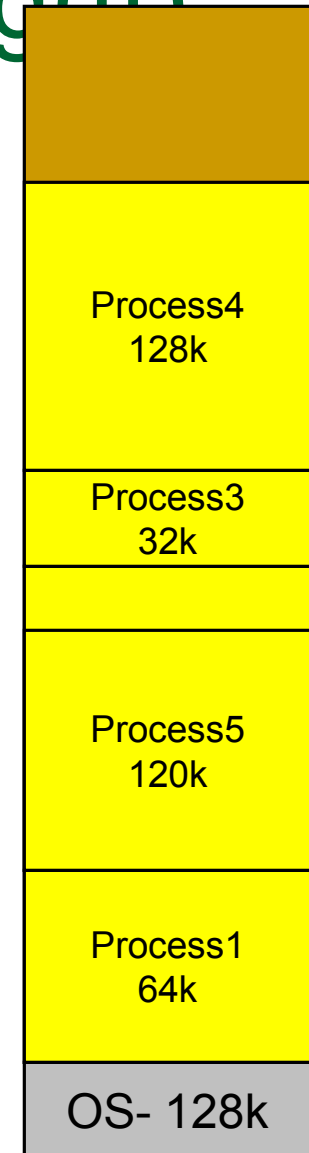
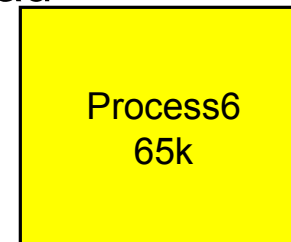
---

## 2.2 Kỹ thuật phân vùng động

- Vùng nhớ user program không được phân chia trước
  - Khi có tiến trình nạp vào bộ nhớ HĐH cấp cho nó không gian nhớ đúng kích thước của nó
  - Khi tiến trình kết thúc vùng nhớ của nó sẽ được thu hồi để HĐH cấp cho tiến trình khác kể cả tiến trình mới có kích thước nhỏ hơn vùng nhớ của tiến trình đã giải phóng đã giải phóng
-

## 2.2 Kỹ thuật phân vùng động(tt)

1. Tiến trình 1,2,3,4 lần lượt được nạp vào bộ nhớ
2. Tiến trình 2 kết thúc, vùng nhớ được giải phóng
3. Tiến trình 5 được nạp vào vùng nhớ của tiến trình 2 vừa giải phóng
4. Tiến trình 6 yêu cầu được nạp vào bộ nhớ nhưng không thể vì không có vùng nhớ trống phù hợp để nạp trong khi tổng dung lượng nhớ còn trống lớn hơn kích thước mà tiến trình yêu cầu




## 2.2 Kỹ thuật phân vùng động(tt)

- Trong kỹ thuật phân vùng động, HĐH phải đưa ra các cơ chế thích hợp để quản lý các khối nhớ đã cấp phát hay còn trống trên bộ nhớ.
- HĐH sử dụng 2 cơ chế: Bản đồ bit và Danh sách liên kết.
- Hai cơ chế HĐH đều chia không gian nhớ thành các đơn vị cấp phát có kích thước bằng nhau, các đơn vị cấp phát liên tiếp nhau tạo thành 1 khối nhớ, HĐH cấp phát các khối nhớ này cho các tiến trình

		A					B						C		D						
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21

## 2.2 Kỹ thuật phân vùng động(tt)

- Cơ chế bản đồ Bit: Mỗi đơn vị cấp phát được đại diện bởi một Bit trong bản đồ bit. Đơn vị cấp phát còn trống đại diện bằng bit 0, ngược lại đại diện bằng bit 1



		A							B						C				D			
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	

0	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bản đồ bit

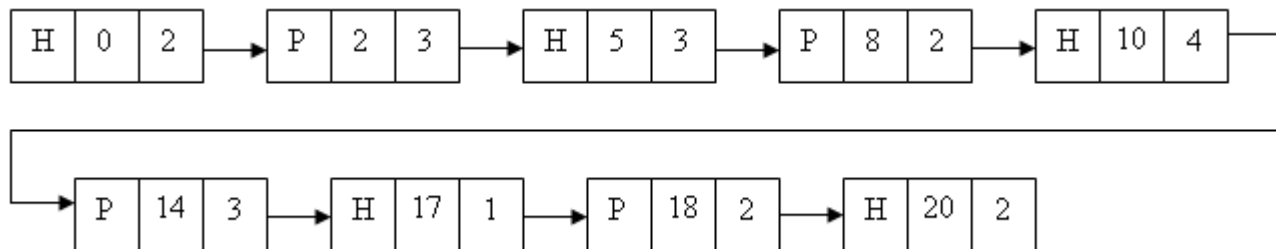
---

## 2.2 Kỹ thuật phân vùng động(tt)

- Cơ chế danh sách liên kết:
    - Mỗi khối trên bộ nhớ được đại diện bởi một phần tử trong danh sách liên kết
    - Mỗi phần tử gồm 3 trường chính:
      - Trường đầu tiên: cho biết khối nhớ đã cấp phát (kí hiệu P) hay còn trống (kí hiệu H)
      - Trường thứ 2: cho biết thư tự của đơn vị cấp phát đầu tiên trong khối
      - Trường thứ 3: cho biết đơn vị tổng số đơn vị cấp phát trong khối
-

## 2.2 Kỹ thuật phân vùng động(tt)

		A						B						C				D			
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21





---

## 2.2 Kỹ thuật phân vùng động(tt)

- Khi có một tiến trình cần được nạp vào bộ nhớ mà bộ nhớ có nhiều hơn một khối nhớ trống có kích thước lớn hơn kích thước của tiến trình đó, HĐH phải quyết định chọn một khối nhớ phù hợp để nạp tiến trình sao cho việc lựa chọn này dẫn đến việc sử dụng bộ nhớ chính là hiệu quả nhất.
  - Có 3 thuật toán mà HĐH sử dụng trong trường hợp này: Best-fit, First-fit, và Next-fit
-

---

## 2.2 Kỹ thuật phân vùng động(tt)

- **Best-fit:** chọn khối nhớ có kích thước vừa đúng bằng kích thước của tiến trình cần được nạp vào bộ nhớ.
  - **First-fit:** HĐH sẽ bắt đầu quét qua các khối nhớ trống bắt đầu từ khối nhớ trống đầu tiên trong bộ nhớ, và sẽ chọn khối nhớ trống đầu tiên có kích thước đủ lớn để nạp tiến trình.
  - **Next-fit:** tương tự như First-fit nhưng ở đây HĐH bắt đầu quét từ khối nhớ trống kế sau khối nhớ vừa được cấp phát và chọn khối nhớ trống kế tiếp đủ lớn để nạp tiến trình
-

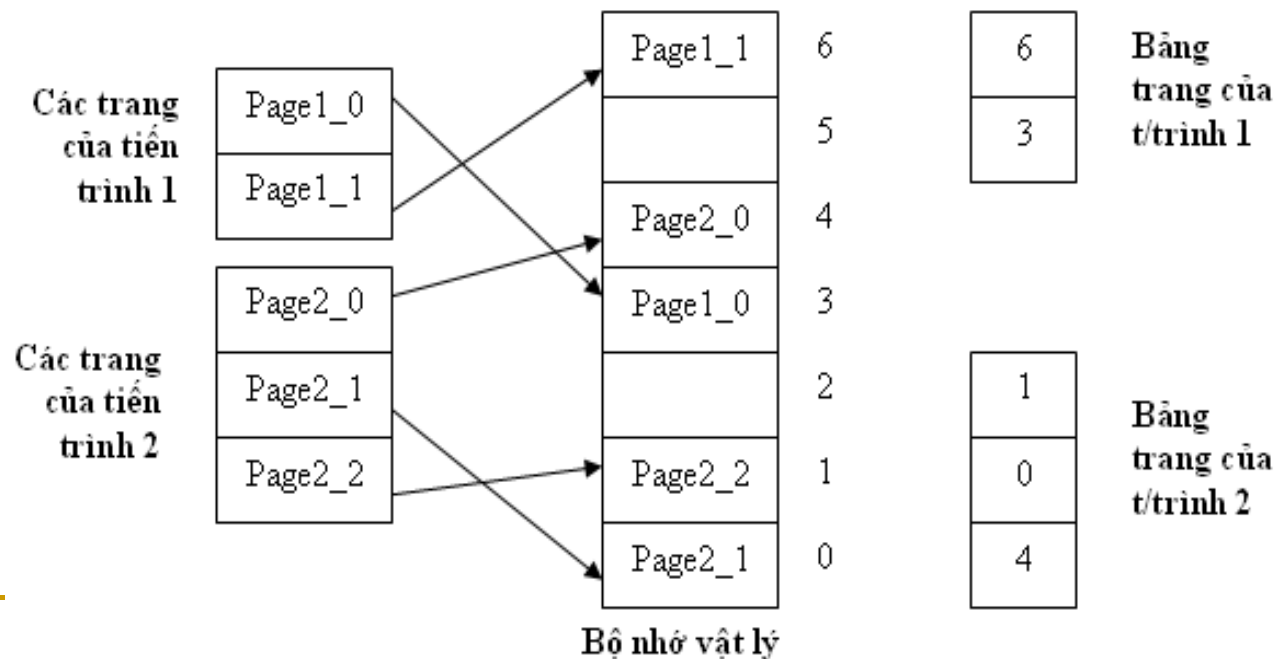
---

## 2.3 Kỹ thuật phân trang đơn

- Bộ nhớ chính được chia thành các phần bằng nhau và cố định, được đánh số bắt đầu từ 0 và được gọi là các khung trang
  - Không gian địa chỉ của các tiến trình cũng được chia thành các phần có kích thước bằng kích thước của một khung trang được gọi là các trang
  - Khi tiến trình nạp vào bộ nhớ thì các trang được nạp vào các khung trang bất kỳ còn trống có thể không liên tiếp nhau
-

## 2.3 Kỹ thuật phân trang đơn(tt)

- HĐH sử dụng các bảng trang(PCT) để theo dõi vị trí các trang của tiến trình trên bộ nhớ. Mỗi tiến trình có bảng trang riêng



---

## 2.3 Kỹ thuật phân trang đơn(tt)

- Sự phân mảnh trong cơ chế này?
  - Nếu kích thước của tiến trình không phải là bội số của kích thước 1 khung trang thì sẽ xảy ra hiện tượng phân mảnh nội vi



---

## 2.4 Kỹ thuật phân đoạn đơn

- Bộ nhớ chính được chia thành các phần cố định có kích thước không bằng nhau, được đánh số bắt đầu từ 0 được gọi là các phân đoạn
  - Mỗi phân đoạn bao gồm số hiệu phân đoạn và kích thước của nó
  - Không gian địa chỉ của các tiến trình kể cả các dữ liệu liên quan cũng được chia thành các đoạn có kích thước không nhất thiết phải bằng nhau
-

---

## 2.4 Kỹ thuật phân đoạn đơn(tt)

- Khi tiến trình được nạp vào bộ nhớ thì tất cả các đoạn của nó được nạp vào các phân đoạn còn trống trên bộ nhớ, các phân đoạn này có thể không liên tục nhau
  - Để theo dõi các đoạn của các tiến trình khác nhau trên bộ nhớ HĐH sử dụng các bảng phân đoạn (SCT), thông thường mỗi tiến trình có 1 bảng phân đoạn riêng
-

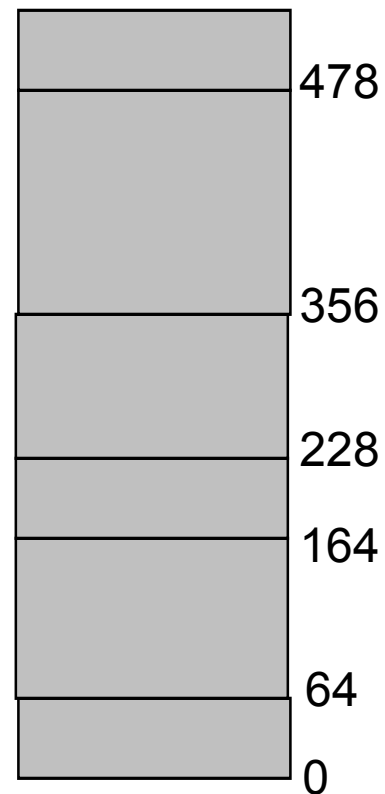
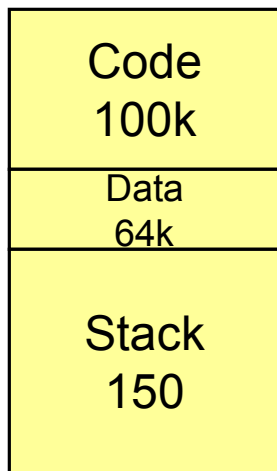
---

## 2.4 Kỹ thuật phân đoạn đơn(tt)

- Mỗi phần tử  $t$  trong bảng phân đoạn tối thiểu gồm 2 trường
    - Trường thứ nhất: cho biết địa chỉ cơ sở của phân đoạn mà đoạn chương trình tương ứng được nạp
    - Trường thứ 2: cho biết độ dài của phân đoạn
-



## 2.4 Kỹ thuật phân đoạn đơn(tt)



base	limit
64	100
164	64
356	150

---

# 3. KỸ THUẬT BỘ NHỚ ẢO

---

---

## 3.1 Khái niệm nhớ ảo

- Để thực thi chương trình có kích thước lớn hơn bộ nhớ vật lý cấp phát cho nó
    - cần xây dựng chương trình theo cấu trúc Overlay
    - gây khó khăn cho người lập trình
  - Để khắc phục khó khăn cho người lập trình, ý tưởng sử dụng bộ nhớ ảo ra đời
  - Kỹ thuật bộ nhớ ảo cho phép xử lý một tiến trình không được nạp toàn bộ vào bộ nhớ vật lý
-

---

## 3.1 Khái niệm nhớ ảo(tt)

- Bộ nhớ ảo mô hình hoá bộ nhớ như một bảng lưu trữ rất lớn và đồng nhất, tách biệt hẳn khái niệm không gian địa chỉ và không gian vật lý
  - Người sử dụng chỉ nhìn thấy và làm việc trong không gian địa chỉ ảo, chuyển đổi sang không gian vật lý do hệ điều hành thực hiện với sự trợ giúp của các cơ chế phần cứng
-

---

## 3.2 Cài đặt bộ nhớ ảo

- Có thể cài đặt bộ nhớ ảo theo 2 kỹ thuật
    - Phân trang theo yêu cầu: Sử dụng kỹ thuật phân trang kết hợp với kỹ thuật swap
    - Phân đoạn theo yêu cầu: sử dụng kỹ thuật phân đoạn kết hợp với kỹ thuật swap
-

---

## 3.2.1 Phân trang theo yêu cầu

- Sử dụng kỹ thuật phân trang kết hợp với kỹ thuật swap
  - Một chương trình được xem như 1 tập hợp các trang thường trú trên bộ nhớ ngoài
  - Khi thực thi hệ thống không nạp toàn bộ chương trình vào bộ nhớ trong mà chỉ nạp những trang cần thiết trong thời điểm hiện tại
    - ⇒ Một trang chỉ được nạp vào bộ nhớ trong khi cần thiết
-

---

## 3.2.1 Phân trang theo yêu cầu(tt)

- Cần có cơ chế phần cứng để phân biệt các trang đang ở bộ nhớ trong và các trang đang ở bộ nhớ ngoài
    - ⇒ Tổ chức bảng trang như kỹ thuật phân trang đơn nhưng 1 phần tử trong bảng trang chứa nhiều thông tin phức tạp hơn
    - ⇒ Cần có 1 bit cho biết trang tương ứng của tiến trình có hay không trong bộ nhớ chính và 1 bit cho biết trang có bị sửa đổi hay không so với lần nạp gần nhất
-

---

# Hiện tượng lỗi trang

Khi hệ thống truy xuất tới 1 trang được đánh dấu là bất hợp lệ sẽ làm phát sinh lỗi trang, HĐH xử lý lỗi trang như sau:

**Bước 1:** Kiểm tra truy xuất đến bộ nhớ là hợp lệ hay bất hợp lệ

- Nếu truy xuất bất hợp lệ : kết thúc tiến trình
- Ngược lại : đến bước 2

**Bước 2:** Tìm vị trí chứa trang muốn truy xuất trên đĩa.

**Bước 3:** Tìm một khung trang trống trong bộ nhớ chính

- Nếu tìm thấy: đến bước 4
  - Ngược lại, thực hiện cơ chế swap out 1 trang thích hợp trên bộ nhớ chính sau đó cập nhật bảng trang tương ứng rồi đến bước 4
-



---

# Hiện tượng lỗi trang(tt)

## Bước 4:

- Chuyển trang muốn truy xuất từ bộ nhớ phụ vào bộ nhớ chính tại khung trang đã xác định được
  - Cập nhật nội dung bảng trang tương ứng.
  - Tái kích hoạt tiến trình người sử dụng
-

# Thay thế trang

- Khi các khung đã đầy mà cần nạp thêm trang thì phải thay thế một trang đang có trên khung
- Nếu trang bị thay thế có thay đổi nội dung thì cần phải đưa ra đĩa
- Có các phương pháp chọn phần tử thay thế:
  - Optimal: Thay thế trang sẽ lâu được sử dụng nhất trong tương lai
  - FIFO: trang ở trong bộ nhớ lâu nhất sẽ được chọn thay thế
  - LRU (Least Recently Used ): trang được chọn để thay thế sẽ là trang lâu nhất chưa được truy xuất

---

## 3.2.2 Phân đoạn đoạn theo yêu cầu

- Bộ nhớ ảo bao gồm các đoạn (segment) có kích thước không cố định
  - Khi nạp đoạn vào bộ nhớ thì hệ điều hành tìm khoảng trống đủ để nạp đoạn
  - Có bảng đoạn quản lý các đoạn
-

---

### 3.2.3 Phân đoạn kết hợp phân trang

- Kết hợp các ưu điểm của phân đoạn và phân trang
  - Bộ nhớ ảo bao gồm các đoạn
  - Trong mỗi đoạn thực hiện phân trang
-

---

# Tài liệu tham khảo

- Trần Hạnh Nhi, Giáo trình HĐH nâng cao, ĐH Khoa học Tự nhiên Tp.HCM, 1998
  - Nguyễn Gia Định-Nguyễn Kim Tuấn, Nguyên Lý HĐH, NXB Khoa học kỹ thuật, 2005
  - William Stallting, Operating Systems, Prentice Hall, 1995
-

---

# CHƯƠNG IV: QUẢN LÝ FILE VÀ ĐĨA

---

---

# 1. CÁC KHÁI NIỆM CƠ BẢN

---

---

# File?

- File hay còn gọi là tập tin, là tập hợp thông tin/dữ liệu được tổ chức theo một cấu trúc nào đó.
  - Nội dung của tập tin có thể là chương trình, dữ liệu, văn bản,...
  - Mỗi tập tin được lưu trên thiết bị lưu trữ đều được đặt tên.
  - Mỗi hệ điều hành có quy ước đặt tên khác nhau, tên tập tin thường có 2 phần: phần tên (name) và phần mở rộng (extension).
-



---

# Các thuộc tính trên file

- **Tên** (name)
  - **Định danh** (identifier)
  - **Kiểu** (type)
  - **Vị trí** (location)
  - **Kích thước** (size)
  - **Giờ** (time), **ngày** (date) và **định danh người dùng** (user identification)
  - Các thông tin tập tin được lưu trữ trên cấu trúc thư mục và được duy trì trên thiết bị
-

---

# Các thao tác trên file

- Tạo
  - Mở
  - Đóng
  - Ghi
  - Đọc
  - Di chuyển
  - Xóa
  - Tìm
  - Lấy thuộc tính
  - Đổi tên
  - .V.v.
-

---

# Các kiểu file

- **File thường:** là file văn bản hay file nhị phân chứa thông tin của người sử dụng
  - **Thư mục:** là những file hệ thống dùng để lưu giữ cấu trúc của hệ thống file
  - **File có ký tự đặc biệt:** liên quan đến nhập/xuất thông qua các thiết bị nhập/xuất tuần tự như màn hình, máy in,..
  - **File khối:** dùng để truy xuất trên thiết bị đĩa
-

---

# Cấu trúc file

Các hệ điều hành thường hỗ trợ ba cấu trúc file thông dụng là:

- **Không có cấu trúc:** file là một dãy tuần tự các byte
  - **Có cấu trúc:** File là một dãy các mẫu tin có kích thước cố định
  - **Cấu trúc cây:** File gồm một cây của những mẫu tin không cần thiết có cùng chiều dài, mỗi mẫu tin có một trường khoá giúp việc tìm kiếm nhanh hơn
-

---

## 2. CÁC PHƯƠNG PHÁP TRUY XUẤT

- Truy xuất tuần tự
- Truy xuất trực tiếp



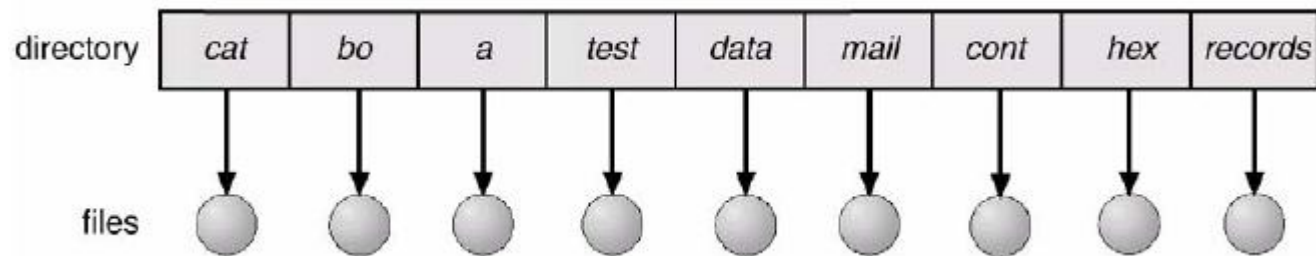
---

# 3. CẤU TRÚC THƯ MỤC

---

## 3.1 Cấu trúc thư mục đơn cấp

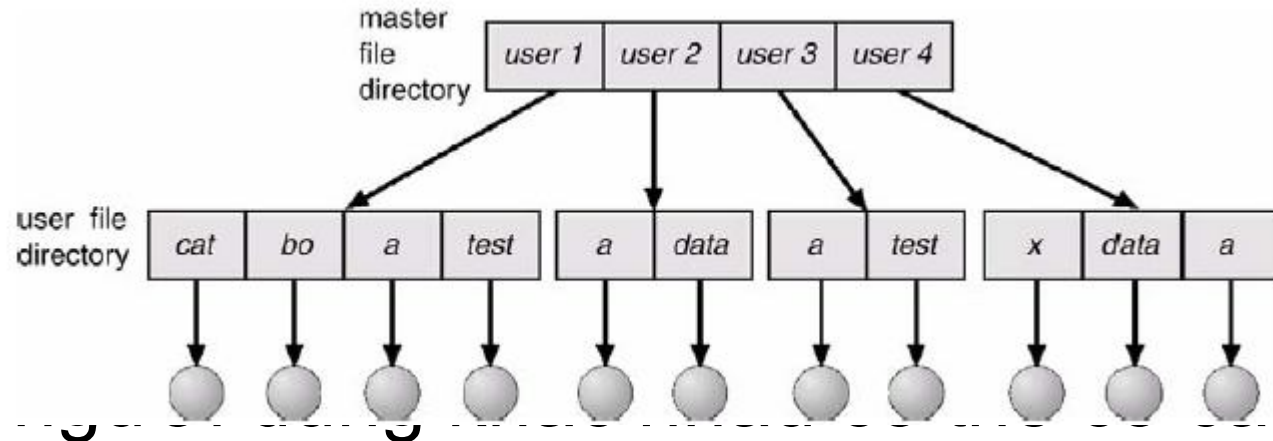
- Một thư mục cho tất cả các tập tin



- Thư mục đơn cấp có nhiều hạn chế khi số lượng tập tin tăng. Vì tất cả tập tin được chứa trong cùng thư mục, chúng phải có tên khác nhau.

## 3.2 Cấu trúc thư mục dạng hai cấp

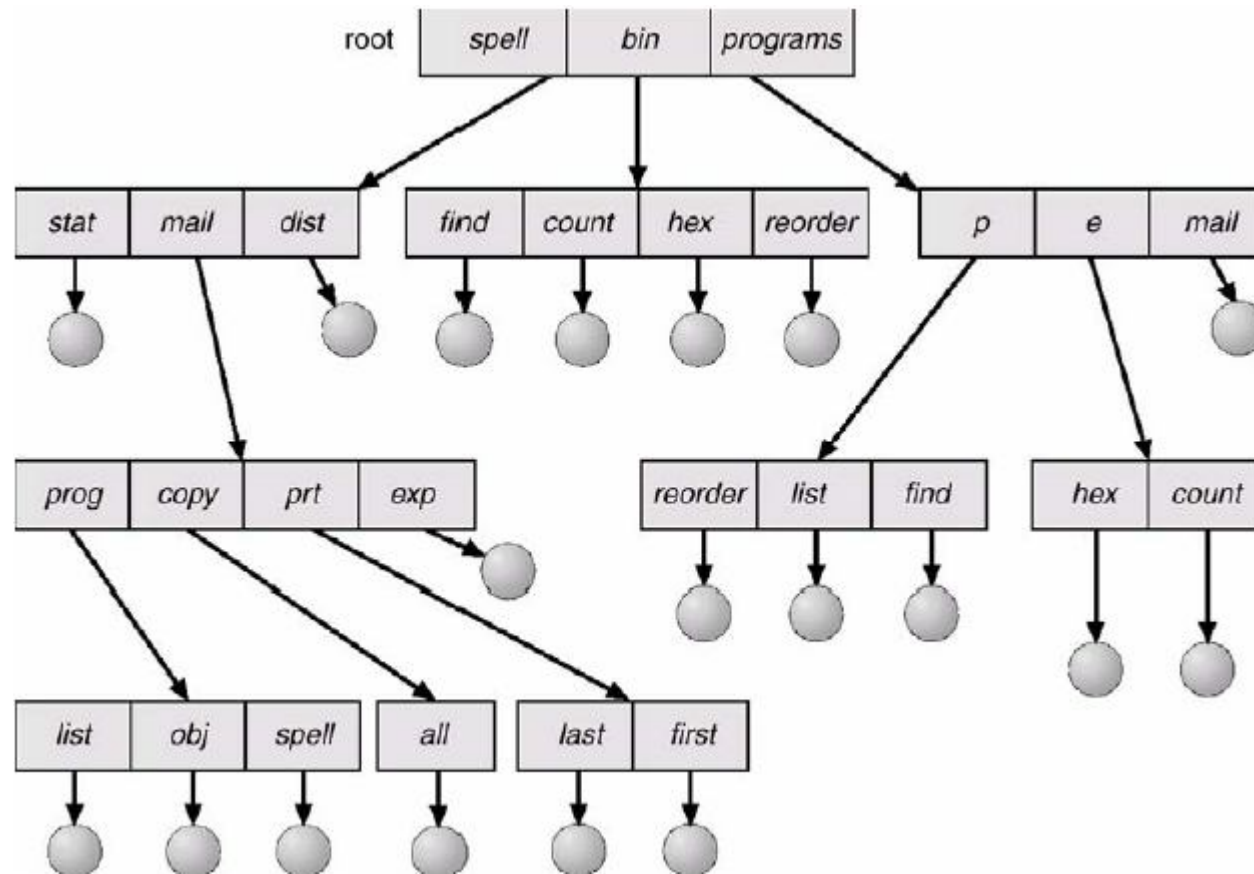
- Mỗi người dùng có 1 thư mục riêng



- các tập tin với cùng một tên
- Cấu trúc này cô lập một người dùng từ người dùng khác.

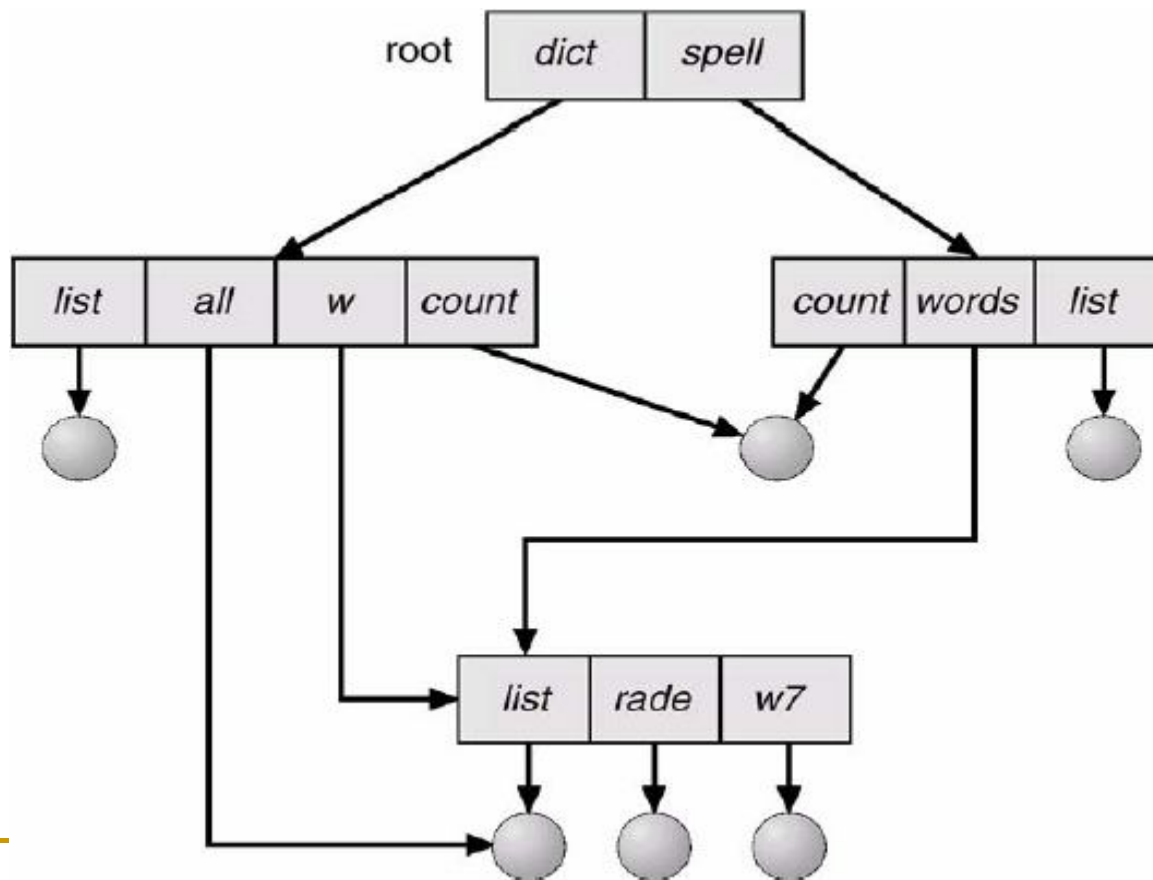


## 3.3 Cấu trúc thư mục dạng cây

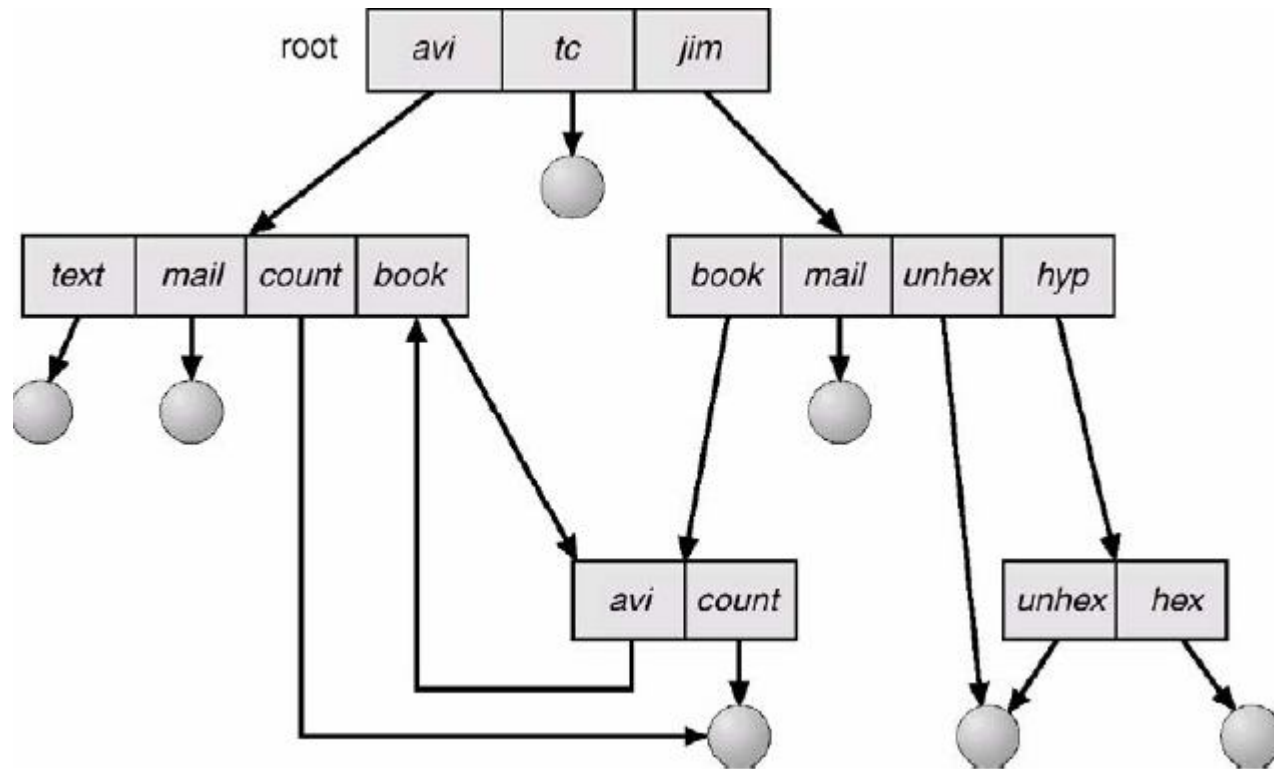


## 3.4 Cấu trúc thư mục dạng đồ thị không chứa chu trình

- Có chung nhau thư mục con và các file



## 3.5. Cấu trúc thư mục dạng đồ thị tổng quát



---

## **4. CÁC PHƯƠNG PHÁP CÀI ĐẶT HỆ THỐNG QUẢN LÝ TẬP TIN**

---

---

## 4.1 BẢNG DANH MỤC QUẢN LÝ THƯ MỤC, TẬP TIN

- Lưu trữ các thông tin liên quan đến các tập tin và các thư mục đang tồn tại trên đĩa (hoặc thiết bị lưu trữ khác)
  - Bảng danh mục gồm nhiều entry, mỗi entry sẽ lưu thông tin về tên, thuộc tính, vị trí lưu trữ, ... của một tập tin hay thư mục.
  - Khi có tập tin/thư mục được tạo ra, HĐH sẽ dùng một entry trong bảng danh mục để chứa các thông tin của nó
  - Khi một tập tin/thư mục xóa khỏi đĩa thì HĐH sẽ giải phóng entry của nó trong bảng danh mục
-

---

## 4.1 BẢNG DANH MỤC QUẢN LÝ THƯ MỤC, TẬP TIN(tt)

- Số lượng entry trong bảng danh mục có thể cố định hoặc không cố định
  - Bảng danh mục thường được lưu trữ tại một không gian đặc biệt nào đó trên đĩa
  - Trong quá trình hoạt động bảng danh mục thường được HĐH nạp từ đĩa vào bộ nhớ để sẵn sàng cho việc truy xuất file của HĐH sau này
-

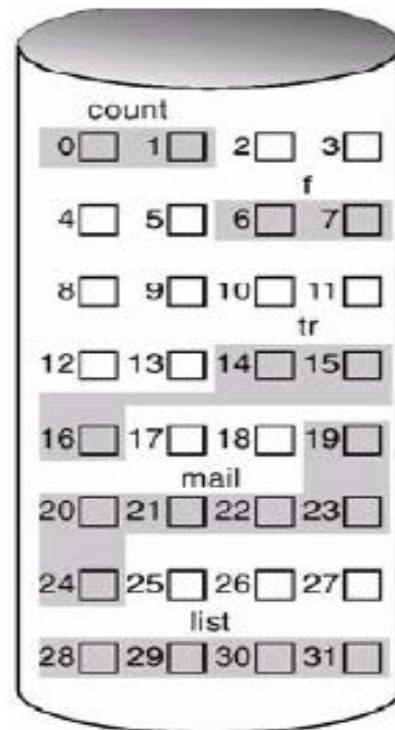
---

## 4.2 Bảng phân phối vùng nhớ

- HĐH chia không gian đĩa thành các khối (block) có kích thước bằng nhau
  - Nội dung file được chia thành các block bằng nhau và bằng kích thước block trên đĩa trừ block cuối cùng
  - Khi lưu tập tin trên đĩa HĐH cấp vừa đủ số block để lưu trữ tập tin
    - ⇒ HĐH tổ chức bảng phân phối vùng nhớ để lưu giữ dãy các khối trên đĩa đã cấp phát cho tập tin hay thư mục
-

## 4.3 Các phương pháp cấp phát vùng nhớ

- Cấp phát liên tục: lưu trữ tập tin trên dãy các block liên tiếp

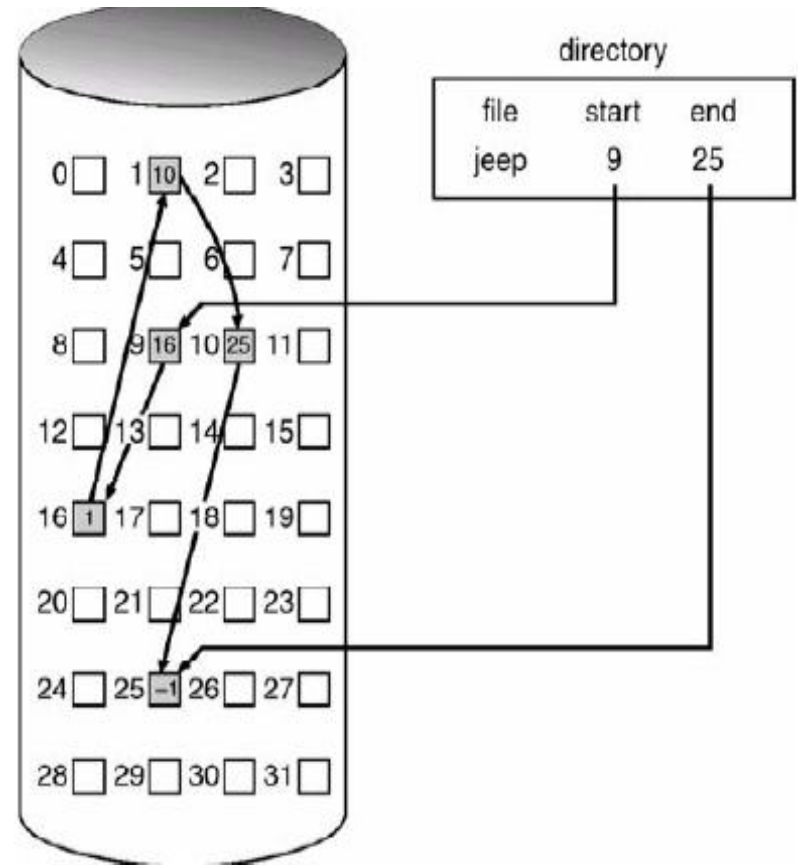


directory		
file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2



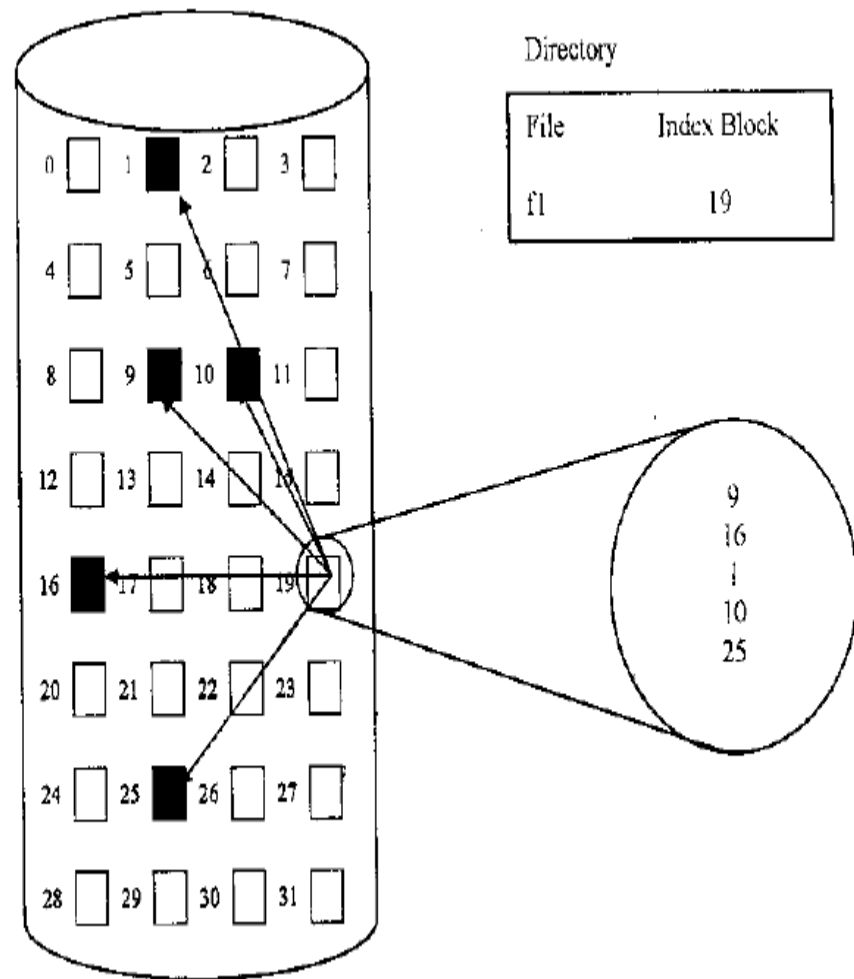
## 4.3 Các phương pháp cấp phát vùng nhớ(tt)

- Cấp phát theo danh sách liên kết:
  - sử dụng danh sách liên kết các block để quản lý các block chứa file
  - Word đầu tiên của mỗi block đĩa được sử dụng như 1 con trỏ trỏ đến block kế tiếp
  - Kích thước của block đĩa lớn hơn kích thước block file 1 word



## 4.3 Các phương pháp cấp phát vùng nhớ(tt)

- Cấp phát theo danh sách liên kết sử dụng Index:
- Tất cả các con trỏ liên kết các block được lưu vào 1 vị trí gọi là khối chỉ mục
- Mỗi tập tin có khối chỉ mục của chính nó, là 1 mảng địa chỉ block đĩa lưu tập tin



---

# I-NODES

- HĐH thiết kế 1 bảng nhỏ để theo dõi các block của 1 file được gọi là I-nodes
  - Một I-nodes gồm 2 phần:
    - Phần 1 chứa các thuộc tính tập tin
    - Phần 2 được chia ra làm 2 phần nhỏ
      - Phần nhỏ thứ nhất gồm 10 phần tử, mỗi phần tử chứa địa chỉ khối dữ liệu của tập tin
      - Phần tử thứ 11 chứa địa chỉ gián tiếp cấp 1 (single indirect)
      - Phần tử thứ 12 chứa địa chỉ gián tiếp cấp 2 (double indirect)
      - Phần tử thứ 13 chứa địa chỉ gián tiếp cấp 3 (double indirect)
-

# I-NODES(tt)

- Địa chỉ gián tiếp cấp 1: Chứa địa chỉ của một khối, trong khối đó chứa một bảng có thể từ  $2^{10}$  đến  $2^{32}$  phần tử mà mỗi phần tử mới chứa địa chỉ của khối dữ liệu của tập tin
- Địa chỉ gián tiếp cấp 2: chứa địa chỉ của bảng các khối địa chỉ gián tiếp cấp 1
- Địa chỉ gián tiếp cấp 3: chứa địa chỉ của bảng các khối địa chỉ gián tiếp cấp 2.

