

# Nguyên Lý Hệ Điều Hành



3F HEDSPI

3 F H E D S P I

Lớp 3F – Việt Nhật K53

Viện CNTT & TT

<http://3f-hedspi.net>

1/6/2010

## MỤC LỤC

<b>CHƯƠNG I: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH.....</b>	<b>3</b>
BÀI 1: KHÁI NIỆM CHUNG VỀ HỆ ĐIỀU HÀNH.....	3
I. Sơ lược về kiến trúc của Hệ thống tính toán (computing system) .....	3
II. Các định nghĩa về hệ điều hành: .....	3
III. Tài nguyên hệ thống tính toán:.....	4
IV. Phân loại HĐH .....	4
BÀI 2: KIẾN TRÚC HỆ ĐIỀU HÀNH.....	6
I. Các thành phần cơ bản: .....	6
II. Kiến trúc hệ thống:.....	6
III. Boot hệ thống:.....	7
IV. Các hình thái giao tiếp trong hệ điều hành: .....	8
BÀI 3: ÔN TẬP.....	10
<b>CHƯƠNG II: QUẢN LÝ TIẾN TRÌNH .....</b>	<b>13</b>
BÀI 1: MỘT SỐ KHÁI NIỆM VỀ TIẾN TRÌNH .....	13
BÀI 2: LẬP LỊCH .....	15
BÀI 3: ĐIỀU ĐỘ TIẾN TRÌNH .....	18
BÀI 4: DEADLOCK (BẾ TẮC/ÙN TẮC/CHẾT ĐÓI).....	22
BÀI 5: NGẮT VÀ QUẢN LÝ NGẮT .....	24
BÀI 6: LUỒNG .....	26
BÀI 7: ÔN TẬP.....	27
<b>CHƯƠNG III: QUẢN LÝ BỘ NHỚ.....</b>	<b>29</b>
BÀI 1: MỘT SỐ KHÁI NIỆM CƠ BẢN .....	29
BÀI 2: CẤU TRÚC CHƯƠNG TRÌNH.....	31
BÀI 3: CÁC CHIẾN LƯỢC QUẢN LÝ BỘ NHỚ .....	35
BÀI 4: QUẢN LÝ BỘ NHỚ TRONG PC.....	40
BÀI 5: ÔN TẬP.....	42
<b>CHƯƠNG IV: QUẢN LÝ FILE VÀ THIẾT BỊ NGOẠI VI.....</b>	<b>43</b>
BÀI 1: HỆ THỐNG QUẢN LÝ TỆP.....	43
BÀI 2: ÔN TẬP.....	50
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>52</b>

# CHƯƠNG I: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

## BÀI 1: KHÁI NIỆM CHUNG VỀ HỆ ĐIỀU HÀNH

### I. Sơ lược về kiến trúc của Hệ thống tính toán (computing system)

#### 1. Môi trường tính toán: (platform)

##### a. Phần cứng (Là môi trường để phần mềm có thể chạy được)

Gồm 3 thành phần:

- Hệ thống máy tính:
  - Thiết bị vào ra
  - Bộ nhớ
  - Bộ xử lí
  - 3 thành phần trên ghép nối với nhau bởi các bus.
- Ghép nối máy tính
- Truyền thông máy tính

##### b. Phần mềm

- VD: trình biên dịch (compiler), soạn thảo (text editor), duyệt web (web browser), ...
- Định nghĩa các tài nguyên được sử dụng để giải quyết yêu cầu của người dùng.

#### 2. Người sử dụng:

- Người lập trình (Programmer)
- Người dùng đầu cuối (End User)
- Điều phối viên (Operator)

### II. Các định nghĩa về hệ điều hành:

#### 1. Tùy theo góc độ nhìn nhận mà ta có các định nghĩa khác nhau.

- End User: HĐH là môi trường giao tiếp người dùng và máy, tự động hóa giải quyết các bài toán, là tập hợp các chương trình, phục vụ khai thác hệ thống tính toán một cách dễ dàng, thuận tiện.
- Người lập trình: HĐH là môi trường cài đặt, tích hợp, phát triển các ứng dụng.
- Kỹ sư phần cứng và kỹ thuật: Là hệ thống cho phép khai thác, mở rộng khả năng của máy tính.
- Người lập trình hệ thống: Là hệ thống mô hình hóa lại hệ thống tính toán.
- Theo Norton: Là hệ thống chương trình đảm bảo giao tiếp người - máy và quản lí tài nguyên của hệ thống tính toán

## 2. Định nghĩa quan trọng nhất phải thuộc:

HDH có:

- Kiến trúc: Là hệ thống chương trình
- Chức năng: Đảm bảo giao tiếp giữa người và máy và quản lí tài nguyên hệ thống tính toán.

## III. Tài nguyên hệ thống tính toán:

- Bao gồm tất cả các: Thiết bị ngoại vi, bộ nhớ của máy tính, bộ xử lí, các chương trình, dữ liệu bên trong hệ thống.
- Có những tài nguyên được dùng chung cho các chương trình của người dùng (bộ nhớ, thiết bị ngoại vi, 1 số dịch vụ của HDH, ...)
- Có tài nguyên là vô hạn khả năng phục vụ, cũng có tài nguyên là hữu hạn khả năng phục vụ. Điều này phụ thuộc vào đặc trưng vật lí và nguyên lý hoạt động của nó. VD: Màn hình: vô hạn khả năng phục vụ, máy in: hữu hạn khả năng phục vụ
- Đối với tài nguyên hữu hạn khả năng phục vụ có thể xảy ra tranh chấp → bế tắc hệ thống. Do đó cần có chương trình giải quyết tranh chấp (khả năng quản lí của HDH)

## IV. Phân loại HDH

Có nhiều tiêu chí phân loại khác nhau:

- HDH đơn chương (đơn nhiệm): Tại 1 thời điểm chỉ có 1 người dùng và chỉ cho phép chạy 1 chương trình.
- HDH đa chương, đa nhiệm (MutilUser, Multitasking): Cho phép nhiều người dùng khai thác tài nguyên của máy tính. Tại 1 thời điểm cho phép nhiều chương trình chạy song song cùng chia sẻ tài nguyên hệ thống tính toán
- HDH xử lí theo lô, mẻ, đợt (Batch): Mỗi lần, người sử dụng có thể yêu cầu thực hiện 1 dãy các chương trình, hết dãy chương trình này mới thực hiện dãy chương trình khác.
- Hệ điều hành tập trung, phân tán:
  - Tập trung: Giữa các máy tính kết nối với nhau chỉ có 1 máy chủ và xử lí tập trung ở máy chủ.
  - Phân tán: Các máy tính kết nối bình quyền, tại các máy tính có 1 HDH riêng gọi là các trạm làm việc.
- Hệ điều hành phân chia thời gian và HDH thời gian thực:
  - Phân chia thời gian:
    - CPU phân chia thời gian để chia sẻ cho nhiều chương trình
    - Tồn tại những khoảng thời gian mà chương trình không được phục vụ bởi 1 CPU nào (Chương trình bị đóng băng, đi ngủ)

- Thời gian thực: 1 chương trình được luân phiên phục vụ bởi nhiều bộ xử lý nhưng tại 1 thời điểm trong hệ thống, 1 chương trình luôn được phục vụ bởi 1 CPU nào đó.

## **BÀI 2: KIẾN TRÚC HỆ ĐIỀU HÀNH**

### **I. Các thành phần cơ bản:**

- Các môđun của HĐH được chia làm 2 lớp:
  - Điều khiển hệ thống
  - Phục vụ hệ thống

#### **1. Điều khiển hệ thống:**

##### **a. Các supervison: Quản lí tài nguyên**

- Quản lí thời gian của CPU (Lập lịch làm việc cho CPU)
- Quản lí bộ nhớ:
  - Theo nguyên lí Von-Numman
  - Muốn thực hiện chương trình thì bắt buộc cần nạp vào bộ nhớ → phân chia và cấp phát bộ nhớ. (Nếu có 100 chương trình thì bộ nhớ cần chia làm 100 mảnh)
- Quản lí thiết bị
- Quản lí tiến trình

##### **b. Chương trình Monitor (Điều phối chính, Tương tác với OP)**

##### **c. Chương trình điều phối công việc (Job)**

#### **2. Phục vụ hệ thống**

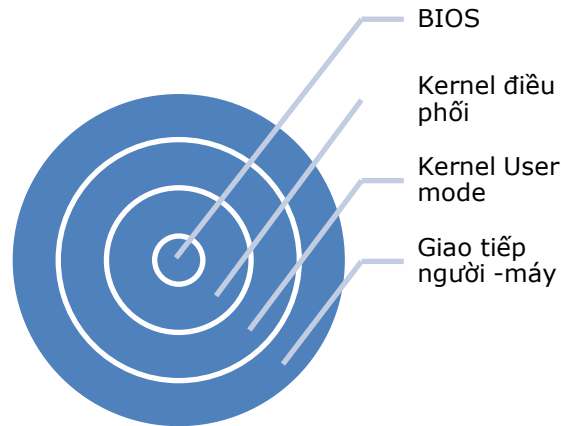
- Ghi nhật kí (tạo LOG): (LOG ghi lại trạng thái của hệ thống từ khi khởi động hệ thống, phục hồi dữ liệu, giám sát hệ thống, chống xâm nhập)
- Chương trình Editor hệ thống: Soạn thảo văn bản đơn giản, có tác dụng can thiệp và sửa đổi cấu hình hệ thống.
- Các tiện ích hệ thống.

### **II. Kiến trúc hệ thống:**

#### **1. Kiến trúc chung:**

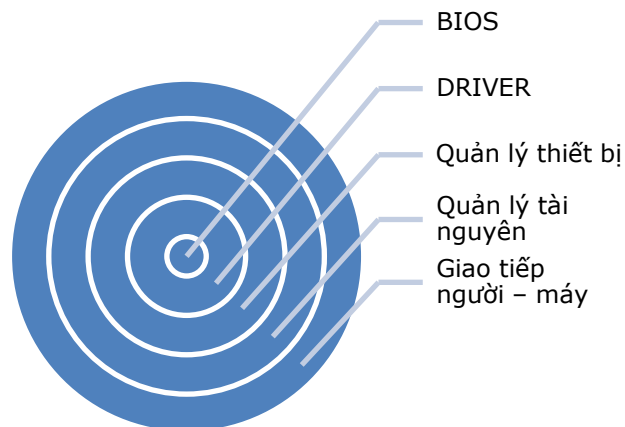
Mọi HĐH đều có kiến trúc phân lớp:

- Đối với End User:



Có thể biết mã nguồn mở ở 2 lớp đầu

- Đối với người lập trình hệ thống



2 vòng ngoài là tầng Logic

3 vòng trong là tầng vật lí.

## 2. Kiến trúc vi nhân (Micro kernel)

- Đây là 1 dạng của HĐH, trong hệ thống chỉ gồm những thành phần tinh giản nhất với các dịch vụ phục vụ cho lớp bài toán xác định. Được sử dụng trong các lĩnh vực sau:
  - Nghiên cứu lí thuyết về HĐH
  - Phát triển mở rộng thành hệ thống lớn như HĐH trong các hệ thống nhúng.

## 3. Kiến trúc máy ảo (Virtual Machine)

- Một dạng HĐH trong đó có chứa mô phỏng của môi trường nhằm thực hiện 1 ứng dụng nào đó.

## III. Boot hệ thống:

Sơ đồ:

IPL → Boot strap → Khởi tạo nhân → Trình giao tiếp

- IPL (Init Program Loader): Tìm đọc chương trình môi trên đĩa khởi động
- Boot strap (chương trình môi): Tìm nạp các tệp tin liên quan tới
- Khởi tạo nhân

#### IV. Các hình thái giao tiếp trong hệ điều hành:

Trong HĐH thường tích hợp nhiều hình thức giao tiếp:

##### 1. Command line:

###### a. Ưu điểm:

- Không đòi hỏi cấu hình máy
- Viết phần mềm quản lý đơn giản
- Có thể gán tham số chương trình bất kì một cách đơn giản.

###### b. Nhược điểm

- Tốc độ đưa lệnh vào chậm (do phải gõ phím)
- Khó khăn với người dùng hạn chế về ngôn ngữ
- Chỉ thực hiện theo chế độ mở (từng đợt, từng trình, ...)
- Phải nhớ các câu lệnh

##### 2. Bảng chọn:

- Menu: Mỗi lựa chọn là 1 pad
- Popup: Hộp chọn (bar)
- Kết hợp Menu + Popup → bảng chọn nhiều mức.
- 2 phương thức ON & ON select

###### a. Ưu điểm:

- Không đòi hỏi cấu hình máy phức tạp
- Không phải nhớ tập lệnh

###### b. Nhược điểm:

- Không gán đc tham số cho 1 trình bất kì
- Vẫn chỉ thực hiện đc ở chế độ mở
- Không thuận tiện với người, hạn chế về ngôn ngữ.

##### 3. Giao diện dạng biểu tượng: ICON, WINDOW, DESKTOP

- ICON: Lờn chú biểu diễn cho trình hoặc nhóm công việc.
- WINDOW:



- DESKTOP: Lưu trữ các biểu tượng có chức năng hay đc sử dụng.

a. Ưu điểm:

- Thân thiện với người dùng, không còn rào cản ngôn ngữ
- Chọn lệnh nhanh
- Cùng một thời điểm có thể chạy nhiều chương trình (phù hợp với hệ thống đa nhiệm)

b. Nhược điểm:

- Đòi hỏi cấu hình máy phức tạp
- Không thể gán tham số cho 1 chương trình bất kì

## BÀI 3: ÔN TẬP

### 1. Nêu các tính chất của HĐH? Vì sao HĐH phải là 1 hệ thống mở và phải tích hợp nhiều hình thái giao tiếp? Nguyên lý xây dựng 1 hệ thống mở?

#### a. Các tính chất cơ bản của HĐH: HĐH có tính chất cơ bản:

- Độ tin cậy cao, ổn định (Dội thông báo của HĐH phải chính xác )
- Độ an toàn cao (Dữ liệu không bị xóa hoặc bị thay đổi ngoài ý muốn)
- Hiệu quả sử dụng cao (Khai thác triệt để tiềm năng của máy tính).
- Tổng quát theo thời gian (Có tính thừa kế, có khả năng thích nghi với những thay đổi có thể xảy ra trong tương lai)
- Thuận tiện, dễ sử dụng.

→ HĐH phải có tính mở để có thể dễ dàng nâng cấp cải thiện hệ thống hay HĐH phải là hệ thống mở.

→ Vì 5 tính chất trên nên HĐH phải là hệ thống mở

#### b. Vì sao HĐH phải tích hợp nhiều hình thái giao tiếp:

- Do yêu cầu của người dùng cũng như trình độ của người dùng (VD: End User: dùng ICON, người quản trị hệ thống: dùng command line ...)
- Mỗi hình thái giao tiếp đều có ưu và nhược điểm riêng. (Ví dụ)

#### c. Nguyên lý xây dựng 1 hệ thống mở: dựa vào 2 nguyên tắc cơ bản:

- Nguyên tắc Môđun:
  - Hệ thống được xây dựng từ những môđun độc lập và tồn tại bộ quy tắc liên kết chúng thành hệ thống có tổ chức.
  - Các môđun quan hệ với nhau thông qua các dữ liệu vào ra, các đối số.
  - Quan hệ phân cấp của môđun được xác lập khi chúng liên kết với nhau để giải quyết những vấn đề phức tạp.
- Nguyên tắc sinh: Mô đun khởi sinh HĐH

### 2. Định nghĩa? Chức năng của HĐH?

#### a. Định nghĩa: Hệ điều hành có:

- Kiến trúc: Là hệ thống chương trình
- Chức năng: Đảm bảo giao tiếp giữa người và máy và quản lí tài nguyên hệ thống tính toán.

#### b. Chức năng chính: 2 chức năng:

- Giúp người sử dụng khai thác các chức năng của phần cứng máy tính dễ dàng và hiệu quả hơn.

- Quản lý tài nguyên hệ thống tính toán.

### 3. Nêu các kiến trúc, phân loại HĐH

#### a. Kiến trúc:

- Theo phân lớp:
  - Đối với End User
  - Đối với người lập trình hệ thống
- Theo ứng dụng:
  - Vi nhân
  - Đa nhân: Ứng dụng trong các hệ thống lớn: các ứng dụng hoạt động song song cùng chia sẻ tài nguyên hệ thống tính toán.
  - Máy ảo

#### b. Phân loại (Xem bên trên)

### 4. Nêu các thành phần của HĐH

#### a. Điều khiển hệ thống:

- Các supervisor: Quản lý tài nguyên hệ thống
  - Quản lý thời gian của CPU (Lập lịch làm việc cho CPU)
  - Quản lý bộ nhớ: Theo nguyên lý Von-Neumann (Muốn thực hiện chương trình thì bắt buộc cần nạp vào bộ nhớ). Nhiệm vụ:
    - Cấp phát thu hồi vùng bộ nhớ.
    - Ghi nhận trạng thái của bộ nhớ.
    - Bảo vệ bộ nhớ.
    - Quyết định tiến trình nào được nạp vào bộ nhớ.
  - Quản lý tiến trình:
    - Tạo lập, hủy bỏ tiến trình
    - Tạm dừng tái kích hoạt tiến trình.
    - Tạo cơ chế thông tin liên lạc giữa các tiến trình.
    - Tạo cơ chế đồng bộ hóa giữa các tiến trình.
  - Quản lý vào ra: 2 thành phần (chương 4): Quản lý tệp và Quản lý thiết bị. Nhiệm vụ chính:
    - Gửi mã lệnh điều khiển đến thiết bị.
    - Tiếp nhận các yêu cầu ngắt từ thiết bị.

- Phát hiện và xử lý lỗi trong quá trình vào ra (đường truyền hỏng, thiết bị chưa sẵn sàng ...)
  - Tối ưu hóa thời gian trao đổi giữa thiết bị ngoại vi với hệ thống.
  - Chương trình Monitor (Điều phối chính; Tương tác với OP)
  - Chương trình điều phối công việc (Job)
- b. Phục vụ hệ thống
- Ghi nhật kí (tạo LOG): (LOG ghi lại trạng thái của hệ thống từ khi khởi động hệ thống, phục hồi dữ liệu, giám sát hệ thống, chống xâm nhập)
  - Chương trình Editor hệ thống: soạn thảo văn bản đơn giản, có tác dụng can thiệp và sửa đổi cấu hình hệ thống.
  - Các tiện ích hệ thống.

## CHƯƠNG II: QUẢN LÝ TIẾN TRÌNH

### BÀI 1: MỘT SỐ KHÁI NIỆM VỀ TIẾN TRÌNH

#### 1. Định nghĩa:

- Đối với EndUser: Tiến trình là chương trình đang thực hiện trong bộ nhớ.
- Đối với người lập trình hệ thống: Là 1 dãy các trạng thái của hệ thống tính toán. Sự thay đổi từ trạng thái này sang trạng thái khác phụ thuộc vào 1 sự kiện nào đó.

$S_i$ : là trạng thái ở thời điểm  $i$

Dãy các trạng thái:  $S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_n$  gọi là 1 tiến trình.

#### 2. Các trạng thái của tiến trình:

Trong hệ thống tiến trình tồn tại ở các trạng thái sau:

- New (Trạng thái khởi tạo):
  - Trạng thái khi tiến trình mới được hình thành, nhờ lệnh nhập vào từ thiết bị ngoại vi.
  - Có thành phần Reader: phân tích chương trình, tách, tạo các khối điều khiển đặt vào dòng xếp hàng của hệ thống.
- Ready: Tiến trình chuyển sang trạng thái Ready khi tiến trình được điều phối. (Tiến trình được cấp tài nguyên từ thiết bị ngoại vi)
- Running: Khi tiến trình được cấp CPU hay lập lịch
  - Khi đang ở trạng thái Running mà bị ngắt, gián đoạn (tức bỏ CPU) thì quay về trạng thái Ready.
  - Khi đang ở trạng thái Running mà có sự kiện vào ra (I/O) thì chuyển sang trạng thái Waiting. Kết thúc sự kiện I/O thì chuyển sang Ready.
- Từ Running chuyển sang trạng thái Terminate: Xảy 2 trường hợp:
  - Kết thúc tốt đẹp
  - Kết thúc bất thường (do ngắt bỏ cố ý của người dùng, hoặc do lỗi chương trình, lỗi thiết bị ngoại vi ...)
- Từ Terminate  $\rightarrow$  dòng xếp hàng ra  $\rightarrow$  Thiết bị ngoại vi
- Như vậy có 3 dòng xếp hàng:
  - Dòng xếp hàng vào
  - Dòng xếp hàng chờ
  - Dòng xếp hàng ra

### 3. Luồng:

- Có thể xem là 1 tiến trình hay 1 dãy các tiến trình hoạt động song song với nhau.
- Có 2 kiểu hoạt động song song:
  - Song song độc lập:
  - Song song bố con (chính phụ).

### 4. Sự tương tác giữa các tiến trình:

- Khởi tạo tiến trình (song song chính phụ): Tiến trình cha sinh ra tiến trình con bằng cách gọi một dịch vụ khác của HĐH
- Hủy bỏ tiến trình (kill): Tiến trình bị hủy bỏ trong 1 số tình huống:
  - Khi cha bị hủy thì con cũng bị hủy
  - Do yêu cầu về điều độ (tránh tắc nghẽn)
- Truyền thông giữa các tiến trình: Trong quá trình hoạt động các tiến trình có thể gửi các thông điệp cho nhau. Có 2 phương thức:
  - Hộp thư (Thường sử dụng vùng nhớ tĩnh)
  - Truyền thông đồng bộ (Gửi xong → nghe → trả lời. Song song luân phiên)

→ **Kết luận quan trọng: Tiến trình muốn chạy phải được cấp CPU**

## BÀI 2: LẬP LỊCH

### 1. CPU Logic và CPU Vật lí

#### a. CPU vật lí:

- Là những CPU có thực được gắn vào máy tính.
- Máy tính có nhiều CPU gọi là máy đa xử lí.  
(Phân biệt với CP. CP là thành phần nói chung của hệ thống. 1 CP có nhiều CPU)
- Có 2 mô hình song song:
  - Song song độc lập: Các bộ xử lí có vai trò như nhau. Việc lập lịch cho CPU là do HĐH đảm nhiệm.
  - Song song chính phụ: Một số bộ xử lí tham gia điều phối hoạt động của bộ xử lí khác.

#### b. CPU Logic:

- Là CPU tồn tại theo quan sát của người dùng

#### c. Các vấn đề cần giải quyết trong bài toán lập lịch:

- Vấn đề 1:  
Trong 1 hệ thống nên tạo bao nhiêu bộ xử lí logic là vừa đủ?
  - Nếu có ít bộ xử lí logic: Tiến trình bị kéo dài trong hệ thống
  - Nếu có nhiều bộ xử lí logic: CPU chờ đợi công việc
- Vấn đề 2:
  - Clock time: Thời gian thực hiện tiến trình
  - CPU time: Thời gian thực được CPU thực hiện
  - Khoảng thời gian gán bộ xử lí cho 1 tiến trình là bao nhiêu thì hợp lí?
- Mô hình hóa bài toán lập lịch: Kí hiệu: Thời gian tiến trình thứ i chuyển từ trạng thái Ready sang Running trong lần chờ đợi thứ j. Tiến trình này phải chờ đợi tổng cộng n lần.
- Như vậy bài toán trên phát biểu như sau: “Cực tiểu hóa thời gian chờ đợi TB của tiến trình” → làm CPU không phải chờ đợi công việc đồng thời tiến trình không phải đợi lâu trong hệ thống.

### 2. Lập lịch trong hệ thống 1 dòng xếp hàng:

Nguyên tắc chung của việc lập lịch: Tổ chức dòng xếp hàng cho các tiến trình, sau đó xác lập 1 chính sách phục vụ cho dòng xếp hàng đó.

#### a. Chiến lược đến trước phục vụ trước (First Come, First Service)

- Ưu tiên ngoài: Mỗi tiến trình trong hệ thống được gán một mức độ ưu tiên ngoài bởi người quản trị hệ thống.

- Khi đến xếp hàng:
  - Cùng ưu tiên ngoài đến trước thì xếp trước.
  - Khác ưu tiên ngoài thì ưu tiên ngoài cao hơn xếp trước.
- Phục vụ theo thứ tự xếp hàng.
- Ưu điểm:
  - Đơn giản, dễ viết chương trình quản lí.
  - Có chú ý đến thứ tự ưu tiên phục vụ tiến trình.
- Nhược điểm: Trọng tải hệ thống phân bố không đều → suy giảm hệ số song song của hệ thống.

b. SJN/SJF (Shortest Job Next/ Shortest Job First)

- Ưu tiên các tiến trình có thời gian phục vụ ngắn hơn.
- Mục đích: Sớm giải tỏa dòng xếp hàng.
- Nhược điểm: Người lập trình phải khai báo thời gian dự kiến của chương trình mình thực hiện → điều này khó với chương trình chạy lần đầu.

c. SRT (Shortest Remain Time)

- Ưu tiên các tiến trình sắp kết thúc.
- Mục đích: Sớm thu hồi lại tài nguyên.

d. Round Robin (RR)

- Phục vụ vòng tròn.
- CPU luân phiên phục vụ các tiến trình trong 1 khoảng thời gian như nhau và khá nhỏ (10 đến 100 micro giây)

→ Nhận xét: Chiến lược đến trước phục vụ trước (*First Come, First Service*) và Round Robin (RR) được dùng phổ biến.

### 3. Lập lịch cho nhiều dòng xếp hàng

Nguyên tắc: Phân hoạch các tiến trình thành nhiều lớp, sau đó xác định thứ tự phục vụ cho các lớp.

a. Các lớp tương đương:  $P_0P_1.....P_K$

- Trình tự ưu tiên: Theo số hiệu lớn tăng dần. Sau đó quay vòng.
- Trong mỗi lớp có thể thực hiện 1 trong 4 chiến lược đã nêu: Thường là FCFS hoặc RR
- Nhược điểm: Không kết hợp được tính ưu tiên.

b. Chiến lược Font



- Lấy 1 lớp làm nền để thực hiện cho các lớp khác.
- VD: 1 lớp gồm các tiến trình tính toán CPU, 1 lớp gồm các tiến trình vào ra I/O. Người ta lấy lớp tính toán làm nền cho lớp vào ra, cụ thể: là phục vụ các tiến trình của 2 lớp theo tỉ lệ:  $4/1(I/O \ 4 \ / \ CPU \ 1)$
- Ưu điểm: Tăng được mức song song và kết hợp được tính ưu tiên.

## BÀI 3: ĐIỀU ĐỘ TIẾN TRÌNH

### 1. Tài nguyên căng và bài toán điều độ

#### a. Tài nguyên căng:

- Một tài nguyên hữu hạn về khả năng phục vụ nhưng trong 1 khoảng thời gian nào đó được nhiều tiến trình yêu cầu đến thì trong khoảng thời gian này tài nguyên đó được gọi là tài nguyên căng.
- Một tiến trình không mất tổng quát có thể chia làm 2 đoạn: Đoạn căng và đoạn còn lại.
- Đoạn căng là đoạn sử dụng tài nguyên căng.

#### b. Mục tiêu của điều độ:

- Đảm bảo các tiến trình qua đoạn căng 1 cách hợp lí nhất.
- Không để xảy ra đống độ.
- Đảm bảo thông lượng giao thông là lớn nhất.

→ Như vậy giải pháp điều độ phải thỏa mãn 4 yêu cầu:

- Đảm bảo tài nguyên căng không phải phục vụ quá khả năng của mình,
- Không để tiến trình nằm vô hạn trong đoạn căng,
- Nếu có xếp hàng chờ thì sớm hay muộn tiến trình cũng qua được đoạn căng,
- Nếu có tiến trình chờ đợi và nếu tài nguyên căng được giải phóng, thì tài nguyên căng phải phục vụ ngay cho tiến trình đang chờ đợi
- Các kĩ thuật điều độ được chia làm 2 mức:
  - Mức sơ cấp: Do người dùng điều độ (Người dùng viết ra )
  - Mức cao cấp: Hệ thống tự động điều độ không cần người dùng quan tâm.

### 2. Kĩ thuật “Khóa trong” (Đèn hiệu)

- Mỗi tiến trình vào đoạn căng thì phải dùng 1 ô nhớ làm dấu hiệu (Tôi vào đoạn căng rồi, các anh đừng vào)
- Khi đến đoạn căng thì tiến trình phải quan sát dấu hiệu của các tiến trình khác.
- Nếu có dấu hiệu xác lập thì tiến trình phải chủ động chờ đợi. Ngược lại nó sẽ xác lập dấu hiệu của mình, thực hiện đoạn căng, sau đó xóa dấu hiệu đã xác lập của mình, rồi tiếp tục thực hiện đoạn còn lại.

```

Var C1,C2:byte;
Begin
    C1:=0; C2:=0
    Parbegin
        TT1:
            Repeat
                While C2=1 do;
                    C1:=1;
                Thực hiện đoạn "găng" TT1
                C1:=0;
                Thực hiện đoạn còn lại TT1
            UntilFalse
        TT2
            Repeat
                While C1=1 do;
                    C2:=1;
                Thực hiện đoạn "găng" TT2
                C0:=0;
                Thực hiện đoạn còn lại TT2
            UntilFalse
    ParEnd;
End.

```

- Chú ý: có thể tổng quát hóa với n tiến trình

```

Parbegin
    TT1,TT2,TT3,..... .TTk
ParEnd

```

- Trong chương trình trên, vòng lặp Repeat.....UntilFalse là vòng lặp không kết thúc diễn tả rằng: Chừng nào tiến trình còn trong hệ thống còn phải tuân thủ nguyên tắc điều độ này.
- Nhận xét:
  - Kỹ thuật điều độ đơn giản nhưng độ phức tạp giải thuật sẽ tăng cao khi có nhiều tiến trình cùng vào đoạn găng.
  - Dễ xảy ra xung đột. (Do sự đồng thời là không đồng bộ. VD khi quan sát không thấy dấu hiệu vào đoạn găng của tiến trình khác, các tiến trình đồng thời bật dấu hiệu của mình lên → xung đột)
- Để khắc phục: Người ta đề nghị là tăng thời gian chờ đợi chủ động của cá tiến trình ở đầu đoạn găng.
  - Trong trường hợp tiến trình quên “mở khóa” (quên xóa dấu hiệu của mình) thì tài nguyên găng sẽ bị khóa.
  - Kỹ thuật này không phù hợp trong trường hợp các tiến trình có tốc độ thực hiện chênh lệch nhau quá lớn.

### 3. Kỹ thuật “Test and Set” (TS)

- Trong HĐH xây dựng thủ tục TS với tham số là dấu hiệu vào đoạn “găng” của tiến trình. Thủ tục này xử lý biến tổng thể có giá trị là số tiến trình xin vào đoạn găng tại thời điểm được gọi. Nếu chấp nhận tiến trình vào đoạn “găng” thì TS xóa dấu hiệu xác lập của tiến trình.

```

Var C1,C2: byte; g:byte
Begin g:=2;
Parbegin
    TT1:
        Repeat
            C1=1;
            While C1=1 do TS(C1);
            Thực hiện đoạn găng TT1
            g:=g-1;
            Thực hiện đoạn còn lạiTT1
        UntilFalse;
    TT2:
        Repeat
            C2=1;
            While C2=1 do TS(C2);
            Thực hiện đoạn găng TT2
            g:=g-1;
            Thực hiện đoạn còn lạiTT2
        UntilFalse;
ParEnd;
End.

```

- Nhận xét:
  - Độ phức tạp không tăng nhiều trong trường hợp có nhiều tiến trình cùng tham gia vào đoạn “găng”.
  - Hiệu quả của điều độ chỉ phụ thuộc vào thủ tục TS.
  - Không phù hợp đối với các tiến trình có thời gian thực hiện ngắn.

### 4. Kỹ thuật “Semaphore”

- Trong HĐH xây dựng 2 thủ tục Ps (Probogen\_giảm), Vs (Verbogen\_Tăng)
- Khi một tiến trình vào đoạn găng nó gọi đến thủ tục P. Thủ tục này gồm 2 bước:
  - Giảm S đi 1
  - Xét  $S \geq 0$ : còn khả năng đáp ứng tài nguyên găng  $\rightarrow$  cấp tài nguyên cho tiến trình để nó thực hiện.  $S < 0$ : Hết khả năng phục vụ, đặt tiến trình vào dòng xếp hàng chờ đợi
- Khi tiến trình ra khỏi đoạn “găng” nó gọi thủ tục V. Thủ tục này gồm 2 bước:
  - Tăng S lên 1
  - Xét  $S \leq 0$  trong dòng xếp hàng có tiến trình đang chờ đợi tài nguyên này. Chọn 1 tiến trình ở dòng xếp hàng chờ cấp tài nguyên cho nó thực hiện.  $S > 0$ : Không có tiến trình nào chờ tài nguyên vừa trả. Hệ thống hoạt động bình thường.

- Nhận xét:
  - Độ phức tạp giải thuật không tăng đáng kể cả trong trường hợp có nhiều tiến trình tham gia vào đoạn “găng”.
  - Hiệu quả điều độ phụ thuộc 2 thủ tục P, V
  - Các lệnh do người lập trình viết trong chương trình. Do đó, nếu sai thứ tự thì hệ thống có thể rơi vào bế tắc. (Thiết lập sai : V trước P sau, toàn V)
- Câu hỏi:
  - Tại sao cặp cặp lệnh mở và đóng tệp lại là P, V?
  - Khi nào có lỗi “Too many Files opened”?

## 5. Kỹ thuật điều độ cao cấp

- Trong thành phần của monitor (điều phối chính) hệ thống có các môđun điều độ.
- Sử dụng 2 thủ tục P, V: với khối điều khiển thiết bị UCB (Unit control block) được xem là thông tin đầu vào của môđun này.
- Bên cạnh đó nó còn có cơ chế tương tác với người điều hành trong quá trình điều phối.

## BÀI 4: DEADLOCK (BẾ TẮC/ÙN TẮC/CHẾT ĐÓI)

### 1. Hiện tượng:

Có 2 hay nhiều tiến trình cùng chờ đợi 1 sự kiện nào đó ở đoạn vắng. Nếu không có sự can thiệp từ bên ngoài → chờ đợi vô hạn. Hiện tượng này gọi là hiện tượng máy “treo” (chaos).

### 2. Nguyên nhân:

Bế tắc xảy ra khi cả 4 điều kiện sau:

- Trong hệ thống có tài nguyên “găng”.
- Có 2 hay nhiều tiến trình cùng chờ đợi qua đoạn vắng.
- HĐH thiếu thành phần thu hồi và cấp phát lại tài nguyên vắng.
- Có hiện tượng chờ vòng tròn qua các đoạn vắng.

3 điều kiện đầu là điều kiện cần, điều kiện thứ 4 là điều kiện đủ.

### 3. Kỹ thuật phòng ngừa:

#### a. Nguyên tắc:

Tìm nguyên nhân phát sinh bế tắc và tác động đến nguyên nhân đó.

#### b. Biện pháp cụ thể:

- Tác động đến điều kiện 1: Làm giảm mức vắng của tài nguyên vắng. Có 2 cách:
  - Bổ sung thiết bị vào trong hệ thống: tốn kém về kinh tế.
  - Phân luồng: Sử dụng kỹ thuật SPOOLIN: Mô phỏng các chế độ vào ra ảo trực tuyến. (Simulate Peripheral Out/Input Online)
- Tác động đến điều kiện 2: Tổ chức dòng xếp hàng cho các tiến trình đồng thời phải có mô đun quản lý dòng xếp hàng đó.
- Tác động đến điều kiện 3: Trong hệ thống phải có thành phần làm nhiệm vụ thu hồi cấp phát lại tài nguyên vắng.
- Tác động đến điều kiện 4: Nguyên nhân chờ đợi vòng tròn là do sự phân phối bình quân về tài nguyên.
  - Cách 1 (Cực đoan): Chỉ cấp tài nguyên vắng cho 1 tiến trình duy nhất khi đã có đủ nguồn tài nguyên yêu cầu của nó → Làm giảm hệ số song song của toàn hệ thống
  - Cách 2: Phân mức tài nguyên vắng. Tiến trình muốn xin tài nguyên mức cao thì phải trả lại tài nguyên mức thấp.

*Nhận xét chung: Chi phí thực hiện kỹ thuật này tương đối lớn, chỉ thực hiện khi khả năng xảy ra bế tắc lớn, thiệt hại do bế tắc gây ra là đáng kể.*

#### 4. Kỹ thuật dự báo vòng tránh

##### a. Nguyên tắc:

Khi có 1 tiến trình xin tài nguyên gắng thì HĐH phải xem xét n thời điểm ở phía trước (của tài nguyên gắng). Nếu suốt n thời điểm này bế tắc không có khả năng xảy ra thì HĐH mới cấp phát tài nguyên gắng đó cho tiến trình. Nếu đến 1 thời điểm nào đó khả năng bế tắc xuất hiện thì HĐH phải đưa ra giải pháp hợp lí.

##### b. Biện pháp: Áp dụng giải thuật có tên: “Người quản đốc ngân hàng Tây”

- Bước 1: Thu thập thông tin về tài nguyên gắng ở thời điểm này
  - Tổng số khả năng phục vụ đang có
  - Tổng số khả năng phục vụ tiến trình yêu cầu
  - Tổng số khả năng phục vụ đã đáp ứng
  - Tổng số khả năng phục vụ còn lại để đáp ứng tiếp
- Bước 2 : Lặp lại bước này khi  $i=1$  cho đến  $n$ 
  - Xem xét khả năng kết thúc của tiến trình.
  - Xét khả năng thu hồi, cân đối với mức cung cầu tài nguyên “gắng”. Nếu không bị vi phạm thì chuyển sang thời điểm kế tiếp. Ngược lại xảy ra mất cân bằng thì phải đưa ra giải pháp an toàn cho hệ thống.

*Nhận xét: Khi áp dụng kỹ thuật này bế tắc không thể xảy ra được, tuy nhiên khối lượng tính toán sẽ tăng cao, làm chậm tốc độ của hệ thống → kỹ thuật này chỉ áp dụng khi khả năng xảy ra bế tắc tuy nhỏ nhưng lại gây thiệt hại lớn.*

#### 5. Kỹ thuật “Nhận biết, khắc phục”

##### a. Nguyên tắc:

Người điều hành phải giám sát được trạng thái hệ thống tính toán ở tại thời điểm bất kì. Nếu nhận biết được nguy cơ bế tắc xảy ra thì có thể can thiệp phá vỡ thế bế tắc.

##### b. Biện pháp:

Trong monitor (điều phối chính) của HĐH phải có các môđun làm nhiệm vụ cung cấp thông tin cho người điều hành, cung cấp cho người điều hành công cụ giám sát và can thiệp.

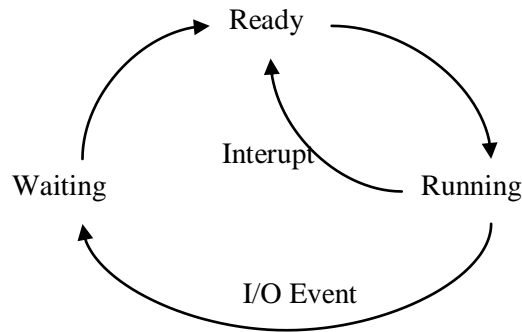
*Nhận xét: dùng trong trường hợp khả năng xảy ra bế tắc là nhỏ và thiệt hại do bế tắc gây ra là không đáng kể.*

## BÀI 5: NGẮT VÀ QUẢN LÝ NGẮT

### 1. Hiện tượng ngắt (gián đoạn)

Hiện tượng CPU đang phục vụ 1 tiến trình thì rời bỏ tiến trình đó, sang phục vụ 1 tiến trình khác gọi là hiện tượng ngắt.

Ngắt là dấu hiệu phát sinh trong hệ thống tính toán làm gián đoạn



### 2. Dấu hiệu ngắt:

Đây là dấu hiệu phát sinh trong hệ thống tính toán làm xảy ra hiện tượng ngắt.

Để nhận biết và xử lý: Phân loại hệ thống này theo các cách sau:

- Cách 1: Dựa vào cấu trúc hệ thống:
  - Ngắt trong
  - Ngắt ngoài
- Cách 2:
  - Ngắt cứng: Ngắt đồng hồ, ngắt các thiết bị ngoại vi
  - Ngắt mềm: Lời gọi dịch vụ từ trong chương trình
- Cách 3:
  - Ngắt che được: Có khả năng khắc phục bởi phần mềm
  - Ngắt không che được: Do lỗi thiết bị.
- Cách 4: Theo cơ chế quản lí
  - Ngắt yêu cầu: Do CPU phát sinh để khởi động phép toán vào ra trên thiết bị.
  - Ngắt báo cáo: Do thiết bị phát sinh khi kết thúc vào ra.

### 3. Số hiệu ngắt

Là số hiệu được gán cho các dấu hiệu ngắt.

VD: Ở PC

- Ngắt số 0: Ngắt khi thực hiện phép chia cho 0.



- Ngắt 1C: Ngắt đồng hồ (điều khiển hệ thống)

#### **4. Chương trình con xử lý ngắt.**

- Là chương trình được gọi đến khi có dấu hiệu ngắt phát sinh.
- Một dấu hiệu ngắt gọi 1 chương trình con ngắt.
- Một chương trình con ngắt có nhiều dịch vụ (function)
- Đặc trưng của chương trình con ngắt:
  - Thường trú trong bộ nhớ.
  - Duy nhất trong bộ nhớ. Trong ngắt không gọi đệ quy.
  - Không ngắt được chương trình con ngắt
  - Nếu chương trình con ngắt có tham số thì các tham số này luôn có kiến trúc là hệ thống các thanh ghi của bộ xử lý.

#### **5. Bảng ngắt**

- Là bảng địa chỉ, số hiệu các phần tử của bảng là 0,1 tương ứng với số hiệu của ngắt.
  - Nội dung mỗi phần tử chứa địa chỉ liên hệ đến chương trình con xử lý ngắt.
  - Quy trình xử lý ngắt như sau: Khi có dấu hiệu ngắt phát sinh, HĐH phát hiện đặc trưng của nó, tìm ra số hiệu ngắt, tra bảng tìm địa chỉ của chương trình con ngắt, sau đó chuyển quyền điều khiển đến chương trình con ngắt
  - Có thể thấy quy trình gồm 5 bước:
    - Bước 1: Lưu đặc trưng của ngắt (xác định chương trình con ngắt)
    - Bước 2: Lưu trạng thái của tiến trình bị ngắt
    - Bước 3: Gọi chương trình con ngắt
    - Bước 4: Thực hiện chương trình con này
    - Bước 5: Phục hồi lại tiến trình bị ngắt và phục vụ tiếp
- HĐH không trực tiếp xử lý ngắt.

## BÀI 6: LUỒNG

### 1. Khái niệm:

- Là 1 dãy các tiến trình song song nhưng đều là thành phần của 1 tác vụ (Task) nào đó. (1 công việc có thể bị chia thành 2 hay nhiều thành phần, mỗi thành phần có thể chạy song song với nhau)
- Thông thường luồng được sinh ra do kết quả các lời gọi hệ thống.
- Việc chia luồng đảm bảo 2 lợi điểm sau:
  - Thứ nhất: Rút ngắn thời gian thực hiện 1 tác vụ.
  - Thứ hai: Kết hợp với cơ chế gọi hệ thống, tạo ra cá thư viện dùng chung. Thư viện này có thể được hình thành dưới dạng các nhân.

### 2. Các mô hình luồng:

- Luồng đơn: tương tự như tiến trình.
- Luồng một nhiều: 1 luồng có thể gọi đến nhiều dịch vụ hệ thống.
- Luồng nhiều một: Nhiều luồng đều gọi đến 1 dịch vụ hệ thống ở cùng 1 thời điểm.
- Luồng nhiều nhiều: Tại 1 thời điểm 1 luồng có thể gọi đến nhiều dịch vụ trong nhân và có những dịch vụ mà được nhiều luồng gọi đến.

### 3. Phân loại các luồng: 2 loại

- Luồng người dùng: Được hỗ trợ các thư viện người dùng (Kernel User mode).
- Luồng nhân: Được hỗ trợ trực tiếp bởi các dịch vụ trong nhân mức thấp của HĐH.

### 4. Quản lí luồng:

- Bài toán: Cấp phát, sinh thêm luồng.
- Để sinh thêm luồng, HĐH sử dụng các dịch vụ quản lí tiến trình bao gồm: nạp chương trình (nạp chương trình, khởi tạo môi trường để chương trình hoạt động)
- Hủy bỏ 1 luồng:
  - Đồng bộ: Khi hủy bỏ 1 luồng con thì toàn bộ luồng bị hủy bỏ ngay lập tức.
  - Không đồng bộ (Trì hoãn): Khi 1 luồng con bị hủy luồng chính vẫn có thể kéo dài hoạt động cho tới khi mọi luồng con của nó đều được kết thúc.
- Để quản lí luồng người ta sử dụng phương thức “tín hiệu” (tương tự như kĩ thuật “đồng bộ khóa trong” trong đồng bộ tiến trình)

## BÀI 7: ÔN TẬP

### 1. Vì sao phải quản lí tiến trình? Mục tiêu của quản lí tiến trình?

#### a. Phải quản lí tiến trình vì:

- Trong hệ thống luôn tồn tại nhiều luồng tiến trình.
- Mặt khác trong hệ thống có những tài nguyên hữu hạn khả năng phục vụ nhưng lại có nhiều tiến trình muốn sử dụng các tài nguyên đó.

→ Dẫn đến xung đột, bế tắc giao thông trong máy bị đình trệ → treo máy.

Do đó ta phải quản lí tiến trình.

#### b. Nhiệm vụ của quản lí tiến trình:

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

#### c. Mục tiêu

- Hạn chế tối đa xung đột và bế tắc có thể xảy ra, đưa ra giải pháp nếu xảy ra các tình huống đó.
- Tận dụng tối đa khả năng của CPU (bài toán lập lịch)

### 2. Giải pháp quản lí tiến trình và yêu cầu đối với giải pháp đó?

- Kỹ thuật khóa trong
- Kỹ thuật Test and Set
- Kỹ thuật Semaphore
- Kỹ thuật điều độ cấp cao.

### 3. Phương thức quản lí dòng xếp hàng trong hệ thống?

### 4. Bế tắc và giải quyết?

### 5. Trong trường hợp khả năng xảy ra bế tắc là lớn, hậu quả gây ra nguyên trọng thì cần sử dụng kĩ thuật nào? (đề giữa kì)

### 6. Phân tích sự tự chủ của tiến trình trong quá trình đồng bộ thông qua các kĩ thuật điều độ. (đề giữa kì)

Tự chủ đồng nghĩa với di trú được

### 7. Thế nào là lời gọi hệ thống (System Call)? Phân biệt Shell và (System Call)?

Lời gọi hệ thống là lời gọi từ HĐH đến chương trình để tạo môi trường giao tiếp giữa chương trình và HĐH hoặc gọi đến 1 chương trình con ngắt để quản lý tiến trình. Chương trình dùng lời gọi hệ thống để yêu cầu dịch vụ của HĐH.

Phân biệt

- Shell: Tạo môi trường giao giữa người sử dụng với HĐH
- System Call: Tạo môi trường giao tiếp giữa Chương trình với HĐH

**8. Công cụ để quản lý tiến trình là gì? Quản lý công cụ đó như thế nào?**

**9. Công cụ để quản lý tiến trình là Ngắt (Interrupt).**

Quản lý ngắt:

- Dựa vào các dấu hiệu ngắt, số hiệu ngắt và bảng ngắt.
  - Khi có dấu hiệu ngắt phát sinh: Hệ điều hành phát hiện đặc trưng của dấu hiệu đó, tìm ra số hiệu ngắt → tra bảng tìm ra địa chỉ của chương trình con ngắt → Chuyển điều khiển đến chương trình con ngắt.
  - Ta có quy trình 5 bước:
    - B1: Lưu đặc trưng của ngắt (Xác định chương trình con ngắt)
    - B2: Lưu trạng thái của tiến trình bị ngắt.
    - B3: Gọi CTC ngắt
    - B4: Thực hiện CTC ngắt này.
    - B5: Phục hồi lại tiến trình bị ngắt và thực hiện tiếp.
- HĐH Không trực tiếp xử lý ngắt mà thông qua CTC ngắt

## CHƯƠNG III: QUẢN LÝ BỘ NHỚ

### BÀI 1: MỘT SỐ KHÁI NIỆM CƠ BẢN

#### 1. Phân loại

Bộ nhớ chia thành 3 loại

- Bộ nhớ chính (main memory): Lưu trữ các lệnh chương trình và dữ liệu đang được sử dụng.
  - Đắt
  - Dung lượng thấp
  - Hiệu năng cao
- Bộ nhớ thứ cấp/bộ nhớ ngoài (Secondary storage): Lưu trữ các chương trình và dữ liệu chưa cần sử dụng.
  - Rẻ
  - Dung lượng cao
  - Chậm
- Bộ nhớ tốc độ cao (cache): Tốc độ truy cập cao, thường nằm ngay trong bộ xử lý. Lưu trữ bản sao của các dữ liệu thường được sử dụng để truy cập nhanh hơn.

#### 2. Tổ chức bộ nhớ:

- Bộ nhớ máy tính đc chia thành các ngăn nhớ và đánh địa chỉ liên tiếp
- Việc truy cập vào bộ nhớ là truy cập theo địa chỉ chứ không phải theo nội dung.
- Chương trình muốn thực hiện phải nạp vào bộ nhớ máy tính.

#### 3. Địa chỉ thực và ảo:

##### a. Địa chỉ thực

- Là địa chỉ vật lí.
- Dựa trên cách đánh địa chỉ các ngăn nhớ.
- Đối với dòng máy PC thì cách đánh địa chỉ như sau: Bộ nhớ được chia thành các đoạn hay các segment (64K). Các bytes trong các segment này được đánh địa chỉ từ 0 (gọi là các offset).  
→ Địa chỉ vật lí của byte xác định dựa trên (segment, offset).

##### b. Địa chỉ ảo (địa chỉ logic)

- Được HĐH tạo ra để nạp các chương trình vào bộ nhớ 1 cách mềm dẻo, linh hoạt. Đồng thời cho phép người lập trình tách biệt được quá trình lập trình với kiến trúc của 1 máy tính cụ thể.

- Với người lập trình thì dữ liệu được gắn bởi các tên gọi.
- Để biến đổi từ các tên gọi sang các địa chỉ bộ nhớ HĐH sử dụng ánh xạ bộ nhớ.
- Mặt khác để tạo ra cơ chế nạp chương trình và quản lí vùng bộ nhớ nạp chương trình thì hệ thống gắn thêm địa chỉ cơ sở vào địa chỉ thực (địa chỉ cơ sở, (segment, offset)).

#### 4. Quá trình dịch chương trình từ ngôn ngữ bậc cao.

Tổng quát:



Kết quả bước dịch: Ngoài môđun đích ta còn nhận được bảng tên trong (Ánh xạ tên)

Bảng tên trong có dạng như sau:

Tên ngoài	Tên trong	Địa chỉ tương đối
A	T <sub>0</sub>	0
B	T <sub>1</sub>	20
C	T <sub>2</sub>	24
2	T <sub>3</sub>	28

Khai báo:

Int A[10];

Float B;

Float C;

A[6]=C\*2+B

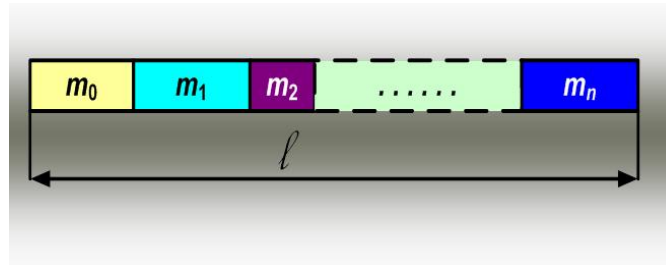
Lệnh            A[6] = C\*2 + B

Ô nhớ        12    = 24\*28+20

(Nội dung ô nhớ 12 = nội dung ô nhớ ..... nội dung ô nhớ 20)

## BÀI 2: CẤU TRÚC CHƯƠNG TRÌNH

### 1. Kiến trúc tuyến tính (độc lập)



Chương trình chỉ có 1 môđun duy nhất được nạp vào bộ nhớ 1 lần. Vùng bộ nhớ được giải phóng khi chương trình thực hiện xong.

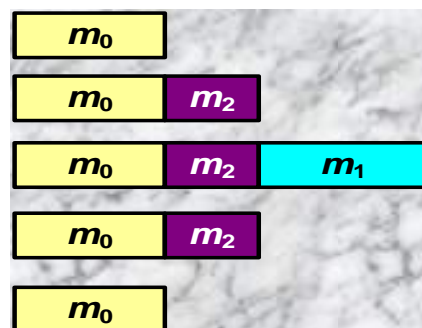
Ví dụ: Các tệp \*.com trong HĐH MSDOS

#### a. Ưu điểm:

- Đơn giản, dễ tổ chức, biên tập và định vị
- Thời gian thực hiện là nhanh
- Thời gian thực hiện chương trình thường được dùng làm chuẩn để đánh giá, so sánh các phương pháp tổ chức khác nhau.
- Tính linh động cao: dễ dàng sao chép chương trình mang sang các máy khác cùng loại và cùng hệ thống mà vẫn duy trì khả năng thực hiện.

#### b. Nhược điểm: Lãng phí bộ nhớ, mức lãng phí tỉ lệ với kích thước chương trình

### 2. Kiến trúc động



Chia chương trình thành nhiều môđun:  $m_1, m_2, \dots, m_k$

Cách nạp chương trình:

- Lúc đầu nạp  $m_1$  vào bộ nhớ.
- Kết thúc  $m_1$ ,  $m_1$  nạp  $m_2$  vào bộ nhớ và tổ chức cho  $m_2$  thực hiện.

( $m_1$  kết thúc giải phóng vùng nhớ của  $m_1$ ,  $m_2$  có thể sử dụng vùng nhớ đó). Quá trình diễn ra liên tục đến khi kết thúc ở môđun  $p$  ( $1 \leq p \leq k$ )

#### a. Ưu điểm

- Nếu quản lý bộ nhớ tốt thì cấu trúc này rất tiết kiệm bộ nhớ
- Giải phóng bộ nhớ theo các chu trình (giai đoạn)

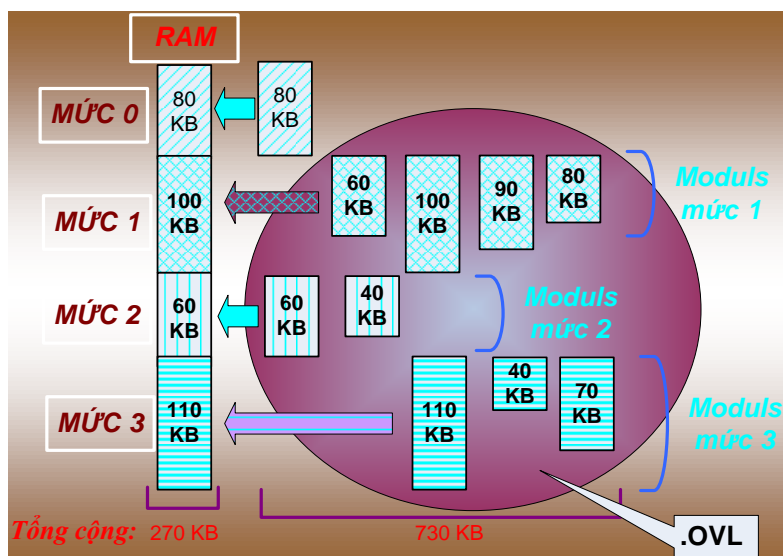
b. Nhược điểm

- Tốn thời gian nạp chương trình (do nhiều lần giao tiếp ngoại vi)
- Đòi hỏi người sử dụng phải biết quản lý, cũng như cần có kỹ năng, kiến thức về lập trình hệ thống

### 3. Kiến trúc Overlay

Hầu hết các chương trình ứng dụng đều có kiến trúc Overlay. Tất cả các ngôn ngữ lập trình đều hỗ trợ Overlay

Khi sử dụng kỹ thuật “chia để trị” để thiết lập giải thuật cho bài toán, ta xây dựng lược đồ cho cấu trúc chương trình.



Ta nhận thấy rằng, tại 1 thời điểm chương trình chỉ thực hiện theo 1 nhánh nào đó của cây. Như vậy tại thời điểm này chỉ cần nạp các môđun trên nhánh đó vào bộ nhớ.

Nhận xét:

- Giải sử số nút là n.
- Độ sâu của cây là h.
- Ở 1 thời điểm chỉ cần nạp tối đa h vùng nhớ để nạp 1 nhánh cây ( $h < n$ ).
- h vùng nhớ này trong quá trình chương trình hoạt động sẽ được sử dụng luân phiên để nạp các nhánh cây. Nhánh nạp sau phủ lên nhánh nạp trước.
- Hầu hết các chương trình trong máy tính đều có cấu trúc này.
- VD: Các file \*.exe

a. Ưu điểm



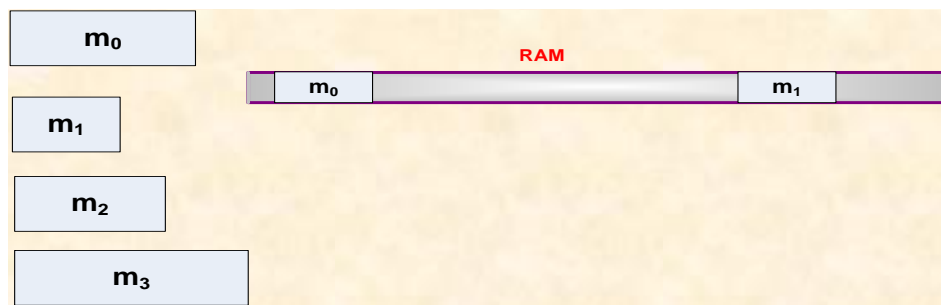
- Tiết kiệm vùng nhớ
- Cấu trúc Overlay dễ dàng xác lập.
- Không đòi hỏi kiến trúc sâu về bộ nhớ

b. Nhược điểm

- Tốn thời gian nạp các môđun (Do có thể nạp đi nạp lại 1 môđun nhiều lần (Re\_Entrance)).  
Cách khắc phục: Tạo vùng nhớ đệm (cache) trong bộ nhớ chứa môđun
- Nếu khai báo sai cấu trúc Overlay thì hệ thống bị rối loạn

#### 4. Cấu trúc môđun

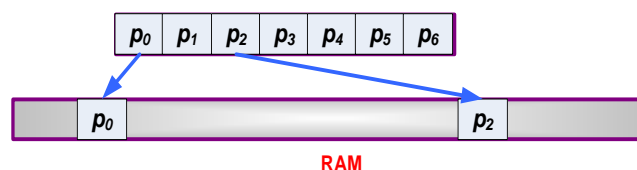
- Biên tập riêng từng môđun
- Tạo bảng quản lý môđun để điều khiển thực hiện



- Đặc điểm:
  - Tự động hoàn toàn
  - Không cần phân phối bộ nhớ liên tục
  - Hiệu quả phụ thuộc vào cấu trúc ban đầu của CT nguồn
  - Dễ dàng sử dụng chung môđun.

#### 5. Cấu trúc phân trang

- CT biên tập như cấu trúc tuyến tính
- Chia thành các phần bằng nhau – trang
- Tạo bảng quản lý trang



- Đặc điểm:
  - Tiết kiệm bộ nhớ
  - Hiệu quả không phụ thuộc và cấu trúc ban đầu của CT nguồn

## 6. Các tệp lệnh máy \*.COM, \*.EXE và nạp chương trình vào trong PC

### a. Tệp .COM

- Không chứa các lệnh
  - Call far địa chỉ
  - JMP far địa chỉ
- Đặc điểm: Khi nạp vào bộ nhớ thì lệnh đầu tiên sẽ được trao quyền ngay

### b. Tệp .EXE

- Được hình thành bởi các cấu trúc Overlay.
- Chương trình gồm nhiều đoạn.
- Đầu tệp (header) có đoạn ghi 32 bytes mô tả kiến trúc tệp và xác định đoạn điều khiển.
- Cho phép 2 lệnh: Call far và JMP far

### c. Nạp chương trình vào bộ nhớ trong PC:

1 chương trình ứng dụng được nạp vào bộ nhớ chiếm 3 phần liên tiếp nhau:

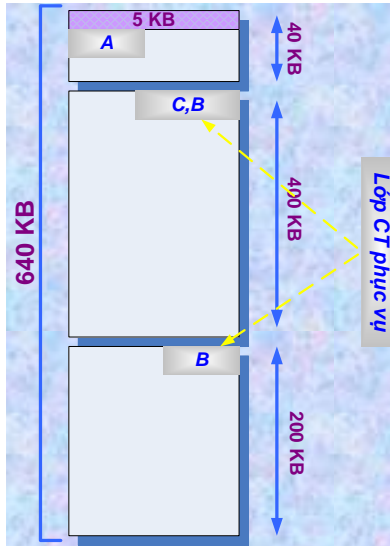
	Heap	Code	Stack
--	------	------	-------

- Khi nạp vào bộ nhớ, HĐH luôn tạo ra 1 khối nhớ PSP 256 bytes (khối tiền tố của chương trình) chứa tham số về môi trường sử dụng tệp lệnh máy. Ví dụ cá tham số liên quan đến ngăn xếp các địa chỉ (stack) để kết nối chương trình.
- Trong byte 2C và 2D của khối này sẽ trỏ đến 1 vùng nhớ vùng nhớ này chứa xâu kí tự kết thúc là kí tự '\0'. Xâu này là tên tệp lệnh máy được nạp vào đoạn bộ nhớ trên.

## BÀI 3: CÁC CHIẾN LƯỢC QUẢN LÝ BỘ NHỚ

### 1. Chế độ phân vùng cố định

#### a. Khái niệm



- Bộ nhớ được chia thành n phần, mỗi phần có kích thước cố định (không nhất thiết phải bằng nhau)
- Mỗi phần được sử dụng như một bộ nhớ độc lập, phục vụ thực hiện cho một chương trình

#### b. Đặc điểm

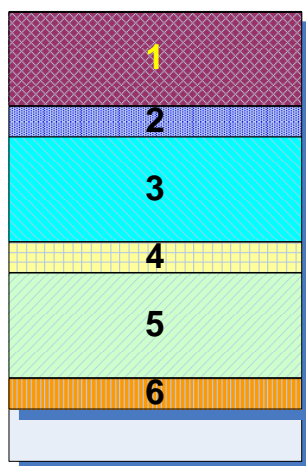
- Mỗi vùng có một danh sách quản lý bộ nhớ tự do
- Mỗi vùng: thực hiện một chương trình ứng dụng
- Sơ đồ bảo vệ thông tin: theo toàn vùng
- Một số chương điều khiển phải được copy vào từng vùng
- Chương trình người dùng sẽ được gán 1 số hiệu đặc trưng cho 1 lớp bài toán nào đó.
- Chương trình người dùng chỉ được nạp vào trong các chương còn rảnh và có lớp bài toán phù hợp.
- Kích thước chương không nhỏ hơn kích thước chương trình người dùng yêu cầu.

#### c. Ưu, nhược điểm:

- Ưu điểm: Cách quản lý bộ nhớ đơn giản.
- Nhược điểm: Hiệu suất sử dụng thấp, dễ gây hiện tượng thiếu bộ nhớ giả tạo.

### 2. Chế độ phân vùng động

#### a. Khái niệm



- Khi khởi tạo HĐH bộ nhớ là 1 vùng liên tục.
- Ở 1 thời điểm nào đó, chương trình xin bộ nhớ, HĐH sẽ cấp cho nó 1 vùng bộ nhớ liên tục nếu có.
- Khi chương trình kết thúc, vùng nhớ được giải phóng và có thể dùng để nạp chương trình khác. (Không phân chương trước, dùng đến đâu tạo bộ nhớ đến đó)

b. Đặc điểm:

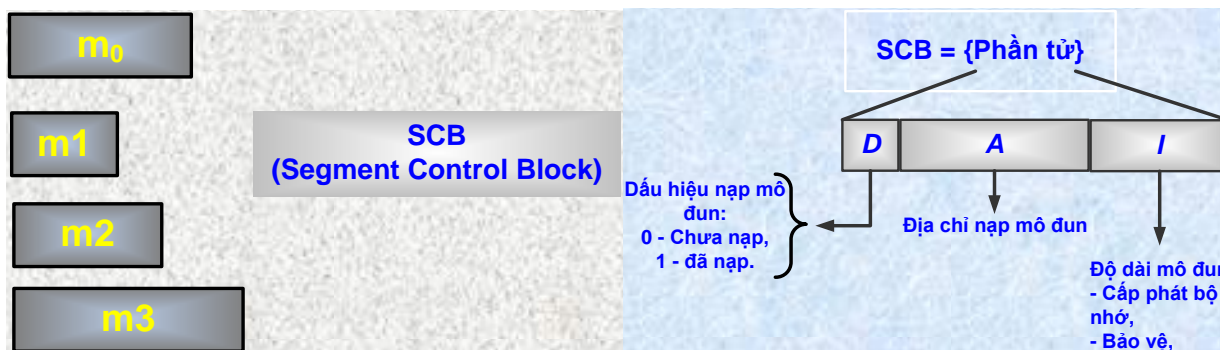
- Sau 1 thời gian thực hiện, bộ nhớ xuất hiện hiện tượng phân mảnh.
- Dễ phát sinh hiện tượng thiếu bộ nhớ giả tạo.
- Để khắc phục: Đưa ra phương pháp dồn chỗ: tạm ngừng 1 số chương trình, thực hiện dồn chỗ, để tạo vùng nhớ liên tục.

c. Ưu và nhược điểm:

- Ưu điểm: linh hoạt, mềm dẻo.
- Nhược điểm: Vẫn phát sinh hiện tượng thiếu bộ nhớ giả tạo, tuy khắc phục được (dồn chỗ) nhưng làm chậm thời gian tính toán.

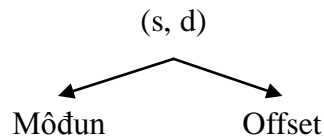
### 3. Chế độ quản lý theo môđun

a. Khái niệm



- Phân đoạn: Người ta xây dựng một bảng quản lý đoạn. Mỗi khoảng này gọi là một khối, địa chỉ đoạn khối này đặt ở đầu mỗi đoạn

- Địa chỉ dữ liệu:



- SCB nạp vào các thành ghi đoạn (CS, DS, SS, ES)
- Để đọc, ghi dữ liệu cần 2 lần truy nhập.

b. Đặc điểm

- Do khi nạp chương trình, mỗi môđun nạp vào 1 đoạn bộ nhớ chỉ cần thỏa mãn điều kiện là đoạn đó đang còn rảnh và kích thước đoạn  $\geq$  kích thước môđun  $\rightarrow$  Không cần bộ nhớ liên tục
- Không đòi hỏi công cụ đặc biệt  $\rightarrow$  có thể áp dụng cho mọi MTĐT
- Dễ dàng sử dụng chung môđun giữa các CT,
- Hiệu quả phụ thuộc vào cấu trúc CT nguồn,
- Tồn tại hiện tượng phân đoạn ngoài (External Fragmentation)
- Do các môđun nạp không liên tục nhau (không nhất thiết theo thứ tự) nên có thể xảy ra thiếu bộ nhớ, phân đoạn ngoài  $\rightarrow$  swapping

c. Swapping

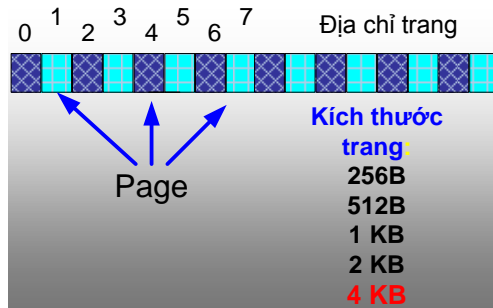
- Nội dung:
  - Đưa một hoặc một số môđun ra bộ nhớ ngoài, giải phóng chỗ nạp môđun mới
  - Do hệ thống đảm nhiệm
  - Không mất thông tin
- Cách chọn môđun đưa ra
  - Môđun tồn tại lâu nhất trong bộ nhớ
  - Môđun có lần sử dụng cuối cùng cách đây lâu nhất
  - Môđun có tần xuất sử dụng thấp nhất.

d. Ưu và nhược điểm:

- Ưu điểm: Sử dụng kỹ thuật này cho phép tận dụng lỗ trống trong bộ nhớ mà không phải dọn chỗ phù hợp với cấu trúc đa đoạn (Overlay).
- Nhược điểm: để thực hiện 1 lệnh ta phải thực hiện qua ánh xạ nhiều bộ nhớ. Do đó tốn thời gian.

## 4. Chế độ phân trang

### a. Khái niệm



- Chia chương trình thành m trang có kích thước bằng nhau.
- Chia bộ nhớ thành n trang có kích thước bằng nhau.
- Giả sử với điều kiện  $m \gg n$  (tức là không thể nạp cùng lúc vào bộ nhớ)
- Nạp:
  - Ở một thời điểm, chỉ nạp được tối đa là n trang chương trình vào bộ nhớ
  - Khi cần đến 1 trang chương trình nào ta nạp trang chương trình đó bổ sung vào bộ nhớ (cần đến đâu nạp đến đó)
- Hiện tượng lỗi trang (Fault Page): Khi thực hiện một chương trình mà toán hạng của lệnh tham chiếu đến một trang chương trình chưa được nạp vào bộ nhớ. Để khắc phục hiện tượng trên dừng thực hiện chương trình, nạp trang bị tham chiếu vào trong bộ nhớ. Như vậy trang vừa nạp sẽ thay thế 1 trang trong bộ nhớ. Có 3 cách thay trang (swapping):
  - Thay trang chương trình cho trang trong bộ nhớ có tần suất sử dụng thấp nhất
  - Thay trang chương trình đã được nạp vào bộ nhớ và ở đó lâu nhất
  - Thay trang chương trình được sử dụng cách đây lâu nhất

### b. Đặc điểm

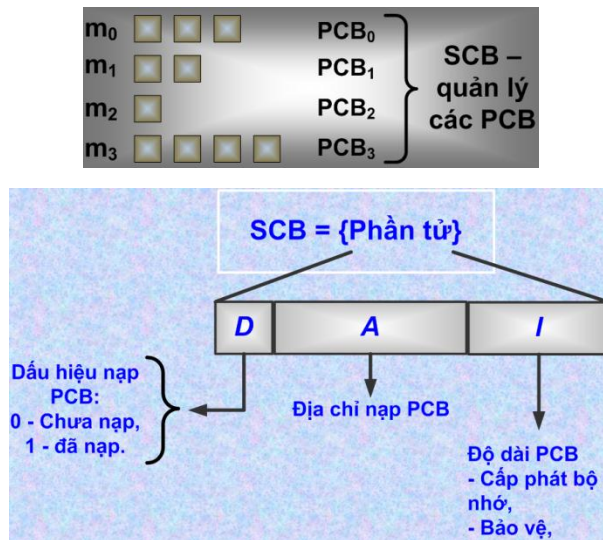
- Không cần phân phối bộ nhớ liên tục
- Phải có công cụ kỹ thuật hỗ trợ định vị trang
- Không sử dụng chung môđun giữa các chương trình
- Hiệu quả không phụ thuộc vào cấu trúc chương trình nguồn
- Bảng PCB có thể rất lớn
- Không bị phân đoạn ngoài
- Thiếu bộ nhớ (mọi trang đều đã được sử dụng) → Swapping

### c. Ưu nhược điểm

- Ưu điểm:

- Tiết kiệm bộ nhớ
  - Tạo bộ nhớ có cảm giác vô hạn
- Nhược điểm: Tốc độ chậm do thời gian thay trang lớn. Để khắc phục là sử dụng cache để hỗ trợ

## 5. Chế độ kết hợp



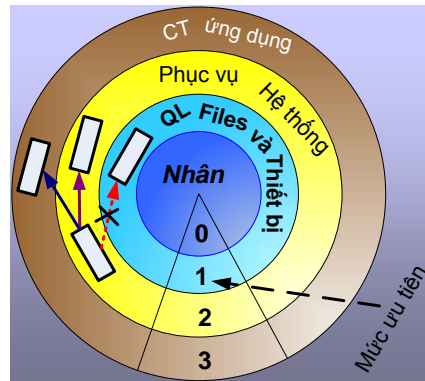
- Kết hợp giữa 2 chiến lược phân đoạn và phân trang. Bộ nhớ được chia làm nhiều đoạn Logic. 1 Đoạn bao gồm nhiều trang
- Tích hợp ưu điểm của cả 2 chiến lược trên.
- Tuy nhiên cũng tích hợp nhược điểm của 2 chiến lược trên.
- Khắc phục: Sử dụng phần cứng để hỗ trợ tra bảng.

## BÀI 4: QUẢN LÝ BỘ NHỚ TRONG PC

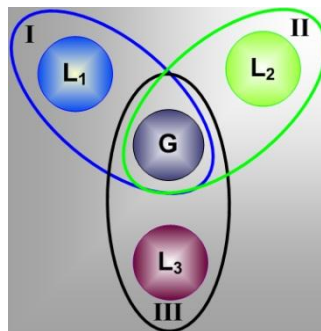
### 1. Khái niệm

Bốn mức ưu tiên từ thấp đến cao (0: cao nhất, 3: thấp nhất)

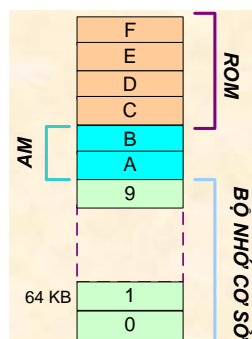
- Ứng dụng người dùng
- Phục vụ hệ thống
- Quản lý file và thiết bị
- Phần Nhân



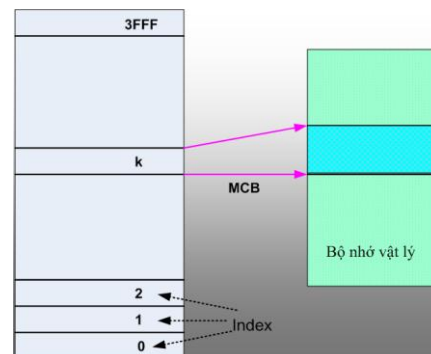
- Một chương trình chỉ được phép truy cập vào các chương trình có cùng mức đặc quyền hoặc mức đặc quyền thấp hơn
- Bộ nhớ phân phối cho CT - 2 loại: bộ nhớ chung (G) và bộ nhớ riêng (L)



- 2 chế độ: Chế độ thực (XT) và chế độ bảo vệ (AT)



Chế độ thực



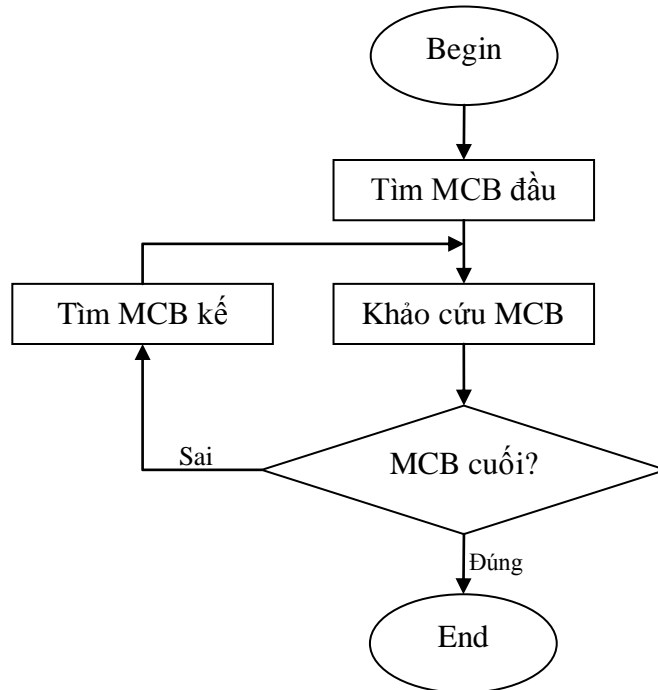
Chế độ bảo vệ



- PC sử dụng chế độ kết hợp để quản lý bộ nhớ
- Mỗi đoạn bộ nhớ trong PC là 1 paragraph: 16 bytes

## 2. Khối MCB

- Mỗi khối được dùng để quản lý một đoạn bộ nhớ. Với bộ nhớ chung gọi là GDT, riêng là LDT
- MCB gồm 16 bytes
- Khi thực hiện chương trình, dựa vào bảng này, hệ điều hành thực hiện ánh xạ địa chỉ



## BÀI 5: ÔN TẬP

### 1. Vì sao phải quản lí bộ nhớ? Nhiệm vụ, mục tiêu của quản lí bộ nhớ?

Quản lí bộ nhớ vì:

- Bộ nhớ là tài nguyên hữu hạn khả năng phục vụ.
- Mọi chương trình muốn thực hiện đều phải nạp vào bộ nhớ.
- Mọi hệ điều hành mong muốn có nhiều hơn các tiến trình trong bộ nhớ.

Nhiệm vụ của quản lí bộ nhớ:

- Nhiệm vụ chính của thành phần quản lí bộ nhớ của HĐH :
  - Hiệu suất cao tức là nạp được và nạp được nhiều chương trình của người sử dụng vào bộ nhớ, mà không phụ thuộc nhiều vào giới hạn của bộ nhớ vật lí.
  - Tránh xung đột: khi chia sẻ tài nguyên bộ nhớ.
- Nhiệm vụ cụ thể:
  - Thực hiện việc tái định vị chương trình, tiến trình: Các mô đun trong chương trình có thể bị vào ra lại nhiều lần (Re-Entrance) do đó HĐH cần xác định đúng vị trí mà nó được nạp trước đó khi đưa mô đun đó vào lại trong bộ nhớ.
  - Bảo vệ bộ nhớ: 1 tiến trình được bảo vệ chống lại sự truy xuất bất hợp lệ của tiến trình khác vào vùng nhớ của nó.
  - Chia sẻ bộ nhớ. Cho phép nhiều tiến trình có thể truy cập đến cùng 1 địa chỉ trên bộ nhớ (tính mềm dẻo).
  - Tổ chức bộ nhớ logic: Làm cho HĐH và phần cứng giao dịch 1 cách hiệu quả với các mô đun, và dữ liệu trong các mô đun của chương trình.
  - Tổ chức bộ nhớ vật lí: Tổ chức luồng thông tin giữa bộ nhớ chính và bộ nhớ phụ.

### 2. Các phương thức quản lí bộ nhớ? (Xem bên trên)

### 3. Phân biệt môi trường đa nhiệm và hệ thống thường trú?

- Giống nhau: Đều có các chương trình hoạt động song song
- Khác:

	Đối với đa nhiệm:	Đối với thường trú:
Khởi tạo, kết thúc chương trình	Do HĐH đảm nhiệm.	Do người dùng đảm nhiệm
Cấp phát tài nguyên cho chương trình	Do HĐH đảm nhiệm.	Do người dùng đảm nhiệm
Tính duy nhất	Không. 1 chương trình có thể có nhiều phiên bản trong bộ nhớ (do có chia sẻ tài nguyên)	Có. Một chương trình chỉ tồn tại 1 phiên bản trong bộ nhớ (do không chia sẻ tài nguyên)

## CHƯƠNG IV: QUẢN LÝ FILE VÀ THIẾT BỊ NGOẠI VI

### BÀI 1: HỆ THỐNG QUẢN LÝ TẬP

#### 1. Nguyên lí chung về quản lí vào ra trong HĐH

- a. Phân cấp quản lí giữa CPU và ngoại vi (trung ương với địa phương)
  - CPU chỉ thực hiện vào ra ở mức logic
  - Ngoại vi thực hiện vào ra ở mức vật lí
  - Khi cần thực hiện vào ra CPU tạo ra ngắt yêu cầu bằng cách đưa ra lời gọi ngắt (ngắt mềm)
  - Thực hiện vào ra xong ngoại vi phát sinh ngắt báo cáo (ngắt cứng)
  - Máy thế hệ I và II: Processor làm việc trực tiếp với thiết bị ngoài → tốc độ thấp
  - Máy thế hệ III trở lên: Processor → Thiết bị điều khiển → Thiết bị ngoại vi
- b. Để thực hiện vào ra, HĐH xây dựng 2 hệ thống con:
  - Quản lý thiết bị ngoại vi: Cần đảm bảo hệ thống thích nghi với:
    - Số lượng nhiều
    - Chất lượng đa dạng
    - Thuận tiện cho người dùng
  - Quản lý files: Cho phép người dùng:
    - Tạo files ở các loại bộ nhớ ngoài
    - Tìm kiếm, truy nhập files
    - Đảm bảo độc lập giữa CT và thiết bị

#### 2. Các thành phần cơ bản

- Thành phần giao tiếp giữa người với máy:
  - Yêu cầu đối với thành phần này: thân thiết, thuận tiện, dễ sử dụng
  - HĐH hỗ trợ người sử dụng ở 2 mức :
    - Mức điều hành: Hỗ trợ các macro cho điều hành viên (delete, copy, cut)
    - Mức lập trình: Hỗ trợ các macro lập trình: open, close, read, write
- Thành phần quản lí các truy nhập tập: Căn cứ vào các đặc trưng vật lí của thiết bị để xây dựng các phương pháp truy nhập nhằm đảm bảo thời gian truy nhập thông tin trên thiết bị.
- Thành phần liên kết các trình thiết bị điều khiển:

- Làm nhiệm vụ chuyển giao các dữ liệu, yêu cầu xử lý từng ứng dụng người dùng cho đến các thiết bị
- Yêu cầu: nội dung phần này có tính chuẩn hóa cao, tốc độ thực hiện nhanh.

### 3. Cách dịch vụ cơ bản

- Dịch vụ đặt tên: HĐH cung cấp cho người sử dụng quy cách đặt tên cho tệp tin. Quy cách này tùy thuộc vào từng HĐH khác nhau
- Dịch vụ thư mục (DIRECTORY) hỗ trợ người sử dụng quản lý tệp tin, đảm bảo tính riêng tư của người sử dụng, đưa ra chế độ bảo mật trong Windows. Khái niệm FOLDER có ý nghĩa rộng hơn.
- Dịch vụ quản trị hệ thống: hỗ trợ phân quyền người sử dụng, cho phép người sử dụng truy nhập tệp tin như thế nào? Quyền truy cập là gì? Các dịch vụ mở rộng như: dịch vụ truyền File FTP, các dịch vụ mạng.
- Hệ quản trị Cơ sở dữ liệu (CSDL) :
  - CSDL là kho dữ liệu có cấu trúc được tổ chức theo mô hình dữ liệu xác định. Mô hình dữ liệu là 1 phương thức tổ chức như:
    - Mô hình đẳng cấp
    - Mô hình phân cấp
    - Mô hình mạng
    - Mô hình quan hệ
  - Hệ quản trị CSDL (DBMS) là công cụ cho phép tạo lập, cập nhật, truy vấn tới Database.

### 4. Vấn đề tối ưu hóa vào ra:

a. Mục tiêu: Giảm thời gian trao đổi giữa thiết bị ngoại vi với hệ thống.

b. Biện pháp:

❖ Kết khối dữ liệu:

- Gom kết nhiều bản ghi logic thành 1 bản ghi vật lý sao cho phù hợp với kiến trúc vật lý của thiết bị.
- Phương thức chung: việc kết khối có thể tiến hành ở mức người dùng hoặc HĐH, ai kết khối thì người đó phải phân rã khối.
- Có 5 phương pháp kết khối:
  - Dạng F (Fixed): mỗi bản ghi vật lý có 1 bản ghi logic, độ dài bản ghi Không thay đổi.
  - Dạng FB (Fixed Block) một khối nhiều bản ghi độ dài giống nhau.
  - Dạng V (Variant) bản ghi có độ dài thay đổi → đầu vào mỗi bản ghi có 1 trường ghi lại độ dài.

- Dạng VB (Variant Block) ở đầu mỗi khối có 1 trường ghi độ dài khối.
  - Dạng U (Undefined) Bản ghi có độ dài không xác định → cuối bản ghi luôn có kí hiệu kết thúc.
- ❖ Kỹ thuật vùng đệm: (Buffer)
- Khái niệm: Vùng đệm là vùng bộ nhớ liên tục ở bộ nhớ trong hoặc trên đĩa từ, được dùng làm nơi trung chuyển thông tin giữa CPU với ngoại vi.
  - Vai trò của vùng đệm:
    - Hỗ trợ phân cấp quản lý giữa CPU với ngoại vi
    - Tăng hệ số hoạt động song song của các thiết bị
    - Đồng bộ hóa giữa truyền và xử lý thông tin: có 2 tình huống cần đồng bộ hóa có thể xảy ra:
      - Tốc độ truyền tin chậm, CPU phải đợi chờ → cần buffer
      - Tốc độ truyền tin nhanh, phức tạp, CPU Không đáp ứng kịp → cần buffer
    - Trong vùng đệm có các loại phòng đệm: Vào, Ra, Xử lý. Ngoài ra còn có Spool (1 loại phòng đệm đặc biệt)
- ❖ Chi tiết:
- Quản lý vòng đệm vào: Nguyên tắc: Khi gặp lệnh nhập tin, HĐH quan sát bộ đệm, nếu còn thông tin thì đưa vào xử lý ngay (thiết bị không phải làm việc), ngược lại phát sinh ngắt yêu cầu (ngắt mềm), thiết bị thực hiện vào ra xong → phát sinh ngắt báo cáo (ngắt cứng)
  - Quản lý vòng đệm ra: Nguyên tắc: khi đưa thông tin ra ngoài, HĐH quan sát bộ đệm, nếu còn chỗ, đặt tiếp thông tin vào bộ đệm, nếu hết chỗ thì phát sinh ngắt yêu cầu để thiết bị giải phóng bộ đệm, sau đó mới đặt tiếp thông tin vào bộ đệm
  - Spool:
    - Có 3 dòng xếp hàng:
      - Dòng xếp hàng vào
      - Dòng xếp hàng chờ
      - Dòng xếp hàng ra.
    - Mỗi dòng xếp hàng có 2 thành phần:
      - Dòng xếp hàng các khối điều khiển
      - Dòng xếp hàng nơi lưu trữ dữ liệu

- Spool cũng là một loại vòng đệm đặc biệt để dùng tổ chức các dòng xếp hàng trong hệ thống. Về bản chất thì nó là những vùng bộ nhớ liên tục, thường ở trên đĩa từ. Nó được phục vụ bởi chương trình điều phối công việc và theo nguyên tắc lập lịch.

❖ Các phương pháp truy nhập:

Tùy thuộc vào đặc trưng vật lý của thiết bị, HĐH lựa chọn phương pháp điều khiển truy nhập thích hợp nhằm tối ưu hóa thời gian vào ra.

Từ mục đích đó ta có 5 phương pháp:

- QSAM (Queue Sequential Access Method): Áp dụng cho mọi thiết bị
  - Ghi tuần tự, đọc tuần tự
  - Bổ sung vào cuối tệp (phải chép sang tệp mới và bổ sung ở tệp mới)
  - Sửa 1 phần tử (phải chép sang tệp mới và sửa ở tệp mới)
- QISAM (Queue Index Sequential Access Method): Dùng cho thiết bị từ tính như băng từ đĩa từ ...
  - Khi truy nhập một tập tin theo một khóa nào đó, người ta xây dựng một thẻ chỉ mục tương ứng với khóa đó (2 trường: khóa, vị trí vật lý)
  - Bước 1: Truy cập qua tệp chỉ dẫn (tìm kiếm nhị phân) → xác định vị trí vật lý lưu khóa cần tìm
  - Bước 2: Truy cập trực tiếp vào tệp: Chỉ áp dụng cho địa chỉ vật lý, sử dụng bộ đệm vào ra
- BSAM (Basic Sequential Access Method): Áp dụng cho thiết bị từ tính có sử dụng phòng đệm khi truy nhập. Cũng truy cập tuần tự nhưng có sử dụng bộ đệm vào ra và ngay từ khi gặp lệnh mở tệp để đọc thì dữ liệu đã tự động lấp đầy bộ đệm
- BDAM (Basic Direct Access Method): Áp dụng cho các thiết bị đĩa từ và trống từ, vật mạng tin chuyển động, thực hiện phép toán trực truy
  - Phương pháp truy nhập: Có sử dụng bộ đệm, các thao tác đọc, ghi ngẫu nhiên. Có thao tác bổ sung vào cuối, sửa đổi phần tử trực tiếp trên tệp (không phải sao ra tệp mới).
- BPAM (Basic Partition Access Method): Dữ liệu được phân đoạn. Tổ chức tiến hành lưu trữ trên các đoạn, việc truy nhập được tiến hành song song trong các đoạn. Chỉ áp dụng cho đĩa từ và trống từ. Ứng dụng RAID

## 5. Khối FCB (File Control Block)

- Khi mở 1 tệp, HĐH sẽ cấp phát 1 khối FCB để phục vụ truy nhập tệp.
- Không gian lưu trữ các khối này được khởi tạo ngay từ khi hình thành HĐH (Trong tệp config.sys với lệnh File= "...") quy định số tệp được phép truy nhập đồng thời. Số tệp được phép truy nhập đồng thời của mỗi HĐH là khác nhau.

→ Trong quá trình truy nhập tệp có thể xảy ra lỗi: “too many files be opened”. Có 2 giải pháp để sửa lỗi trên:

- Tắt các file không cần thiết đang mở.
- Sửa tệp config.sys
- Khối này chứa các thông tin sau:
  - Tên tệp và đường dẫn.
  - Thuộc tính tệp.
  - Địa chỉ lưu trữ tệp:
    - Địa chỉ logic của thiết bị lưu trữ (ổ A, B, C, D ...)
    - Địa chỉ vật lí trên thiết bị đó.
  - Phương pháp truy nhập
  - Trạng thái tệp (Vị trí đầu ghi, đọc)
  - Chương trình điều khiển thiết bị, phục vụ vào ra tệp.

## **6. An toàn truy nhập tệp**

- Hệ điều hành phải có độ an toàn cao
- Mục tiêu :CIAA
  - C: Bí mật với người không có thẩm quyền
  - I: Toàn vẹn:
    - Vật lí
    - Giá trị (Tránh sự truy nhập bất hợp pháp)
  - A: Sẵn dùng cho người có thẩm quyền (Người có thẩm quyền thì muốn dùng là có ngay)
  - A: Tính xác thực: Tính đúng đắn của thông tin.
- Nguyên lí xây dựng 1 hệ thống thông tin: 5 bước:
  - Bước 1: Nghiên cứu hệ thống
  - Bước 2: Nghiên cứu nguy cơ, điểm yếu. Phân tích khả năng rủi ro (Tính đến các trường hợp xấu nhất)
  - Bước 3: Xây dựng chính sách an toàn
  - Bước 4: Xây dựng các giải pháp
  - Bước 5: Cài đặt, kiểm nghiệm các giải pháp.
- 4 Nguy cơ dẫn đến mất an toàn :

- Thông tin dùng chung (Thông tin có thể bị sao chép hoặc bộ nhớ bị ghi đè)
- Thiết bị dùng chung (Chung thiết bị gây có thể làm phong tỏa tài nguyên gây → lỗi)
- Các bộ dữ liệu dùng chung
- Dịch vụ dùng chung
- Các chính sách an toàn:
  - Sử dụng phần cứng
  - Sử dụng phần mềm
  - Sử dụng hỗn hợp
- Giải pháp an toàn:
  - Phân mức bảo vệ.
  - Share all or not share.
  - Bảo mật dựa trên các quyền điều khiển truy nhập.
- a. Kỹ thuật sao lưu và phục hồi:
  - Tạo bản sao cho các tệp và phục hồi lại các tệp bị lỗi
  - Phục hồi lại dữ liệu nhờ các bản sao đó.
- b. Che dấu dữ liệu
  - Các kỹ thuật vật mang
  - Sử dụng cấu trúc của các tệp (khóa dữ liệu, cấu trúc bộ dữ liệu)
  - Mã hóa trực tiếp thông tin
- c. An toàn ở dạng lưu trữ:
  - Giải pháp chung: Kỹ thuật mã hóa.
    - Mã hóa theo thiết bị lưu trữ (Kiến trúc đĩa)
    - Mã hóa theo cấu trúc tệp tin
    - Mã mật
  - Đối với mã mật: có 2 loại:
    - Mã đối xứng: mã hóa và giải mã theo cùng 1 phương pháp. Ví dụ: mã cổ điển, mã khối DES
    - Mã công khai: mã hóa và giải mã theo các phương pháp khác nhau (mã bất đối xứng). Ví dụ: Mã RSA
- d. Bảo vệ qua thuộc tính: Dựa vào 6 thuộc tính của 1 tệp



e. Bảo vệ qua điều khiển truy nhập:

- Xác lập quyền thao tác trên các tệp dữ liệu: Có 7 quyền (Tạo mới, Sao chép (Copy), Xóa (Delete), Cut, Quản trị, Bổ sung mới, Sửa đổi (Modify)).
- 2 chính sách DAC, RBAC
  - DAC: Bảo mật rời rạc: Gom nhóm các đối tượng và nhóm người dùng.

$S_{ij}$ : Bộ các quyền mà nhóm người sử dụng thứ  $i$  tác động lên đối tượng  $j$

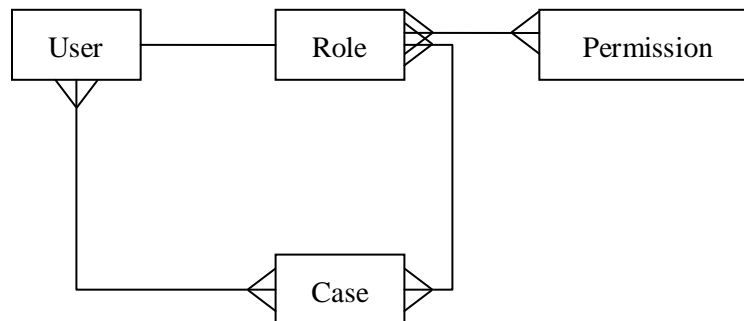
	$O_1$	$O_2$	...	$O_n$
$U_1$				
...			$S_{ij}$	
$U_m$				

Danh sách các quyền:

- 4 quyền đầu là đối với toàn bộ đối tượng (tạo mới, sao chép, hủy bỏ, quản trị đối tượng)
- 3 quyền sau là đối với thành phần (bổ sung mới, xóa, sửa đổi)

Nếu số lượng sử dụng lớn, đa lớp thì DAC phù hợp

- RBAC: 5 quan hệ này đều là quan hệ nhiều – nhiều



## BÀI 2: ÔN TẬP

### 1. Khi nào có lỗi “Too many Files Opened”? Nguyên nhân, khắc phục?

- Lỗi “Too many Files Opened” xuất hiện khi có quá nhiều truy nhập vào hệ thống tệp đồng thời. HĐH quy định số tệp truy nhập đồng thời. Mỗi HĐH khác nhau thì số tệp khác nhau. Quy định này được ghi trong file config.sys (đối với winXP là config.nt) với lệnh “files=”
- Cách khắc phục: 2 cách:
  - Đóng những tệp không cần thiết
  - Sửa tệp config.sys

### 2. Tại sao cặp lệnh mở và đóng tệp lại là cặp lệnh P,V

- Khi mở một tệp, hệ điều hành sẽ cấp phát một khối FCB để phục vụ cho việc truy nhập tệp. Về không gian lưu trữ các khối FCB này, nó được hình thành ngay từ khi nạp hệ điều hành (được khai báo trong file config.sys, ở dòng FCBs = ...)
- Ta thấy, các khối FCB này về nguyên tắc cũng giống như tài nguyên gắng (tài nguyên hữu hạn khả năng phục vụ nhưng trong một khoảng thời gian nào đó được nhiều tiến trình yêu cầu, cụ thể là các tiến trình mở tệp tin mới, yêu cầu cấp phát khối FCB). Khi có quá nhiều tệp tin mở đồng thời, sẽ xuất hiện lỗi “Too many files open!”
- Phải có giải pháp để điều độ các tiến trình qua đoạn găng này. Hệ điều hành sử dụng kỹ thuật Semaphore để giải quyết bài toán này. Vì thế, các cặp lệnh mở và đóng tệp là các cặp lệnh p,v

### 3. Quản lý vào ra là gì? Nhiệm vụ? Vấn đề tối ưu hóa trong quản lý vào ra?

- Nhiệm vụ chính:
  - Gửi mã lệnh điều khiển đến thiết bị.
  - Tiếp nhận các yêu cầu ngắt từ thiết bị.
  - Phát hiện và xử lý lỗi trong quá trình vào ra (đường truyền hỏng, thiết bị chưa sẵn sàng...)
  - Tối ưu hóa thời gian trao đổi giữa thiết bị ngoại vi với hệ thống.
- Tối ưu hóa thời gian vào ra (kết khối dữ liệu và kỹ thuật vùng đệm - đã trình bày)

### 4. Quản lý thiết bị là gì? Mục tiêu, nhiệm vụ?

### 5. Truy nhập thông tin trên đĩa từ như thế nào?

### 6. An toàn tệp trong HĐH?

### 7. Virus là gì? Cơ chế tồn tại, ẩn mình và phá hoại của virus. Phân loại spyware với Virus?

Virus: là những chương trình hay đoạn mã được thiết kế để tự nhân bản và sao chép chính nó vào các đối tượng lây nhiễm khác.

- Cơ chế tồn tại: Tự nhân bản
- Cơ chế phá hủy: sao chép chính nó vào các đối tượng lây nhiễm khác để phá hủy đối tượng đó (chiếm quyền các khối điều khiển)
- Cơ chế ẩn mình:
- Tác hại: Làm cạn kiệt bộ nhớ, phá hoại các chương trình, dữ liệu của máy tính.

❖ Phân biệt Virus với Spyware:

- Giống:
  - Xuất phát vì những mục đích không tốt.
  - Phá hoại máy tính.
  - Có cơ chế lây nhiễm
- Khác:
  - Spyware thường là phần mềm chuyên thu thập thông tin từ máy chủ (mục đích thương mại)
  - Virus là đoạn mã phá hoại máy tính

**8. Khi in tài liệu thì đường kẻ băng bị mờ đi hoặc 1 phần của ảnh bị mờ đi. Giải thích nguyên nhân và cho cách xử lý?**

- Hệ điều hành Windows sử dụng một cơ chế spool để phục vụ việc in ấn. Khi in một tài liệu, đầu tiên trình ứng dụng gửi tạm dữ liệu vào đĩa cứng. Mặc định, windows sẽ tạo ra tập tin spool trong folder windows/spool/printers. File spool này là vùng trên đĩa mà windows sử dụng để lưu trữ các printing jobs khi chúng được thực hiện in. Cơ chế spool giúp cho windows có thể trả quyền điều khiển ngay lập tức cho chương trình khi tài liệu đã được in.
- Hiện tượng đã miêu tả ở trên là do: File hoặc ảnh mà bạn đang cố gắng in đòi hỏi quá nhiều bộ nhớ. Cho nên khi vào folder spool thì file hoặc ảnh đó không được đưa vào hết, dẫn đến khi in ra bị mờ hoặc mất một phần.

❖ Cách xử lý:

- Tắt tất cả các ứng dụng khác mà bạn đang sử dụng
- Tắt tất cả các tài liệu khác trừ tài liệu mà bạn đang muốn in
- Tắt tất cả các chương trình đang mở và khởi động lại máy
- Xem thử xem bạn có đủ dung lượng đĩa cần thiết không. Nếu thiếu thì có thể sử dụng các chương trình như disk defragment hoặc disk cleanup để giải phóng ổ đĩa.

## TÀI LIỆU THAM KHẢO

1. Đề cương ôn tập Hệ Điều Hành (ali\_hobao – <http://hedspi.net>)
2. Slide Nguyên Lý Hệ Điều Hành + Bài giảng của thầy Uy
3. ...