

HỆ ĐIỀU HÀNH

Giáo viên: Đỗ Tuấn Anh
Bộ môn Khoa học Máy tính
Khoa Công nghệ Thông tin
ĐHBK Hà Nội
anhdt@it-hut.edu.vn
0989095167



MỤC ĐÍCH – YÊU CẦU

- Là giáo trình **cơ sở chuyên ngành**:
 - Xét các vấn đề HĐH bất kỳ phải giải quyết,
 - Phương thức giải quyết các vấn đề đó.
 - Hỗ trợ cho các môn khác trong việc xây dựng cơ sở cho Tin học.
 - Những v/đ xem xét sẽ không lạc hậu trong tương lai.

MỤC ĐÍCH – YÊU CẦU

- Mang yếu tố **chuyên đề**:
 - Minh họa cho các v/đ lý thuyết,
 - Khoảng cách giữa và thực tế công nghệ ở Tin học nói chung và HĐH nói riêng gần như bằng 0.
- Như vậy: đây là một giáo trình khó, khá nặng nề.



TÀI LIỆU

- **A. Tanenbaum** Design and Implementation operating system.
- **A. Tanenbaum** Advanced Concepts to Operating Systems.
- **Microsoft Press** Inside to WINDOWS 2000.
- Nguyên lý hệ điều hành: Hà
- Hệ điều hành: Tác giả: Nguyễn Thanh Tùng

Chương I. CÁC KHÁI NIỆM CƠ BẢN

- **1- Cấu trúc phân lớp của hệ thống tính toán**
- Máy tính điện tử đầu tiên ra đời năm 1944-1945,
- MTĐT được xây dựng và hoạt động theo nguyên lý Von Neuman: **Máy tính được điều khiển bằng chương trình và trong câu lệnh của chương trình người ta chỉ nêu địa chỉ nơi chứa giá trị chứ không nêu trực tiếp giá trị.**



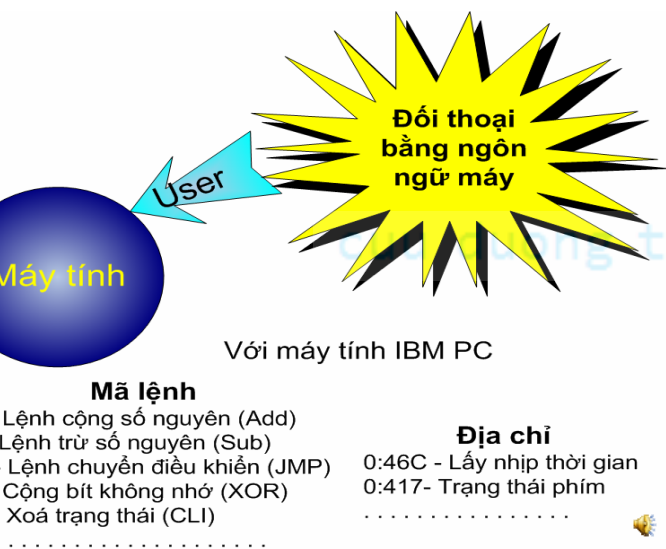
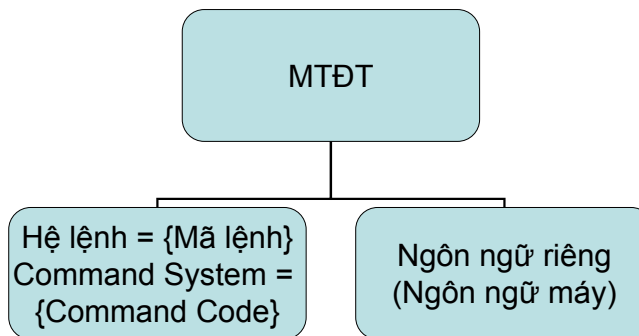
Chương I. CÁC KHÁI NIỆM CƠ BẢN (tt.)

- Thế hệ thứ 2 (1955-1965)
 - Sự ra đời của thiết bị bán dẫn
 - lập trình FORTRAN và hợp ngữ
 - **Hệ thống xử lý theo lô**
- Thế hệ thứ 3 (1965-1980)
 - mạch tích hợp (IC)
 - **hệ điều hành chia sẻ thời gian**
- Thế hệ thứ 4 (1980-nay)
 - máy tính cá nhân (PC-Personal Computer)
 - **hệ điều hành mạng và hệ điều hành phân tán**

Banking System	Airline Reservation	Adventure Game	Application Program
Compilers	Editors	Command Interpreter	
Operating System			System Program
Machine language			Hardware
Micro Programming			
Physical devices			

7

Cấu trúc phân lớp của hệ thống tính toán



Cấu trúc phân lớp của hệ thống tính toán

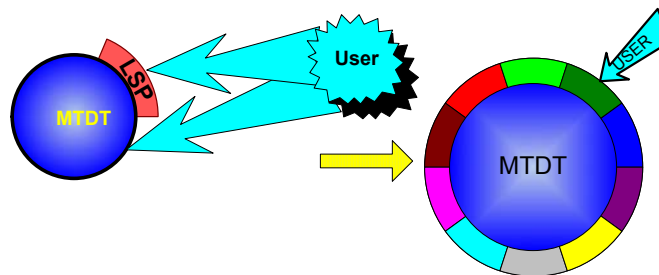
- Người lập trình thường nhầm lẫn → năng suất lập trình thấp,
- Đã áp dụng nhiều biện pháp kích thích:
 - Kỷ luật hành chính,
 - Thưởng phạt kinh tế.
- Năng suất chỉ tăng chút ít và ổn định ở mức 8 câu lệnh/ngày công!
- Kết quả nghiên cứu tâm lý học: Bản chất con người không quen làm các công việc đơn điệu, không có tính quy luật, sớm hay muộn cũng sẽ có sai sót!

10

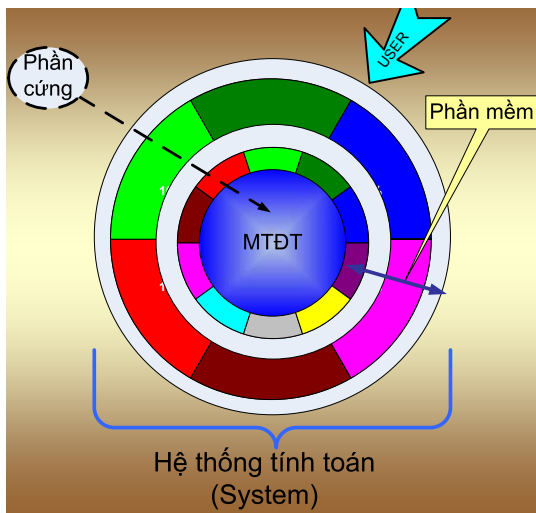
Cấu trúc phân lớp của hệ thống tính toán

- Như vậy, để nâng cao năng suất - cần tác động vào MTĐT.
- \exists các công việc mọi người và \exists CT đều cần (V/d – Trao đổi vào ra) → tạo sẵn CT mẫu (Standard Programs – SP) cung cấp cùng với máy.
- Hình thành LSP = {SP}

11



12



13

Tác động phần mềm lên phần cứng

- Cơ sở hoá hệ lệnh:
 - Các lệnh phức tạp như $x^{1/2}$, e^x , $|x|$. . . dần dần được thay thế bằng CT con,
 - Tăng cường các lệnh xử lý bit.
- Tăng tốc độ của MT,
- Tăng tính vạn năng,
- Tăng độ tin cậy,
- Giảm giá thành,
- Cho phép phân các thiết bị thành từng nhóm độc lập, tăng độ mềm dẻo của cấu hình.

14

Tác động phần mềm lên phần cứng

- Các yếu tố trên có sự tác động của tiến bộ công nghệ, nhưng phần mềm đóng vai trò quan trọng, nhiều khi có tính quyết định:
 - Bàn phím,
 - Máy in.

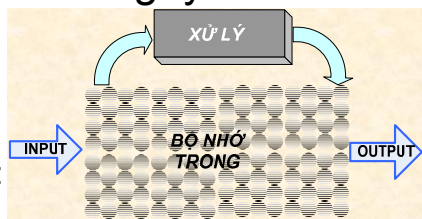
15

Tác động phần mềm lên USER

- Đẩy người dùng ra xa máy, nhưng tạo điều kiện để khai thác triệt để và tối ưu thiết bị



2 – Các tài nguyên cơ bản



a) Bộ nhớ:

- Vai trò,
- Gót chân Asin của hệ thống,
- Quan trọng: sử dụng như thế nào?
- Bảo vệ thông tin?

19

b) PROCESSOR

- Điều khiển máy tính,
- Thực hiện các phép tính số học, lô gic và điều khiển,
- Có tốc độ rất lớn (vài chục triệu phép tính / giây),
- Thông thường có thời gian rảnh (thời gian “chết”) lớn → hiệu suất sử dụng thấp,
- V/đ: tăng hiệu suất sử dụng (giảm thời gian chết).

20

C) THIẾT BỊ NGOẠI VI

- Số lượng: **Nhiều**,
 - Chất lượng: **Đa dạng**,
 - Tốc độ: **Cực chậm** (so với Processor),
- V/đ: Phải đảm bảo:
- Hệ thống **thích nghi** với số lượng và tính đa dạng,
 - Tốc độ thiết bị ngoại vi không ảnh hưởng đáng kể đến **năng suất** hệ thống.

21

D) Tài nguyên chương trình

- Cần phải có các chương trình cần thiết,
- Một chương trình được kích hoạt: phục vụ cho nhiều người dùng (cấu trúc Reenter),
- Khai thác On-Line, RPC,
- Cách tổ chức chương trình: cấu trúc và đảm bảo cho cấu trúc hoạt động,

22

Nhiệm vụ của hệ thống đối với tài nguyên

- 2 nhiệm vụ chung (không phụ thuộc vào loại tài nguyên):
 - Phân phối tài nguyên: Cho ai? Khi nào? Bao nhiêu (với loại chia sẻ được)?
 - Quản lý trạng thái tài nguyên: Còn tự do hay không hoặc số lượng còn tự do?
- Tồn tại nhiều giải thuật → Loại hệ thống:
 - Xử lý theo lô,
 - Phân chia thời gian,
 - Thời gian thực.

23

3 - ĐỊNH NGHĨA HỆ ĐIỀU HÀNH

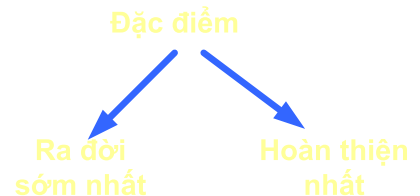
- Có nhiều góc độ quan sát và đánh giá,
- Các đối tượng khác nhau có yêu cầu, đòi hỏi khác nhau đối với OS,
- Xét 4 góc độ:
 - Của người sử dụng,
 - Của nhà quản lý,
 - Của nhà kỹ thuật,
 - Của người lập trình hệ thống.

24



cuu duong than co

- Đối thoại: để hệ thống **gọn nhẹ** + **linh hoạt**,
- Đối thoại → \exists **ngôn ngữ đối thoại** (bằng lời hoặc cử chỉ).
- Ở mô phỏng 2 đối tượng con người → là hệ thống trí tuệ nhân tạo, là **hệ chuyên gia**,



28

Ra đời sớm nhất:

1944/1951 xã hội mới biết và tin vào khả năng giải quyết các bài toán phi số của MT,
1952 - Von Neuman đề xuất tư tưởng xây dựng “CT tự hoàn thiện” ,
1961 – Bell Lab – Các CT trò chơi Animal và Core Ware,
Thách thức thực tế các hệ CG: 1971-1972.
OS – xây dựng từ 1950,
1965 - Hệ ĐH nổi tiếng OS IBM 360

Hoàn thiện nhất:

- Thống kê UNESCO: 73% số công trình không hoàn thành do khâu đặt v/đ,
 - Các HCG khác: Cán bộ chuyên ngành + Cán bộ lập trình,
 - OS:
 - Người lập trình giải quyết bài toán của chính mình
 - Hiểu rõ: V/đ+khả năng công cụ+ khả năng bản thân
- 1974: 3 công trình xây dựng kỹ thuật tiêu biểu đỉnh cao trí tuệ loài người:
Hệ thống ĐT tự động liên lục địa,
Hệ thống Appolo đưa người lên mặt trăng,
OS IBM 360.

4 – TÍNH CHẤT CHUNG CỦA OS

- A) Tin cậy và chuẩn xác,
- B) Bảo vệ,
- C) Kế thừa và thích nghi,
- D) Hiệu quả,
- E) Thuận tiện.

31

Tin cậy và chuẩn xác

- Mọi công việc trong hệ thống đều phải có kiểm tra:
 - Kiểm tra môi trường điều kiện thực hiện,
 - Kiểm tra kết quả thực hiện,
- Nhiều chức năng KT: chuyển giao cho phần cứng.
- Ví dụ: Lệnh **COPY A:F1.TXT B:**
- Sau khi KT cú pháp, bắt đầu thực hiện lệnh. Lần lượt hệ thống sẽ KT gì và có thể có thông báo nào?

32

- Kt CARD I/O,
- Tồn tại ổ đĩa?
- Thiết bị điện tử ổ đĩa?
- Động cơ ổ đĩa?
- Khả năng truy nhập của ổ đĩa?
- Khả năng truy nhập đĩa?
- Tồn tại file F1.TXT?
- Khả năng truy nhập file?
-

• So sánh:

SCANDISK
DEFRAG

NDD

SPEEDISK

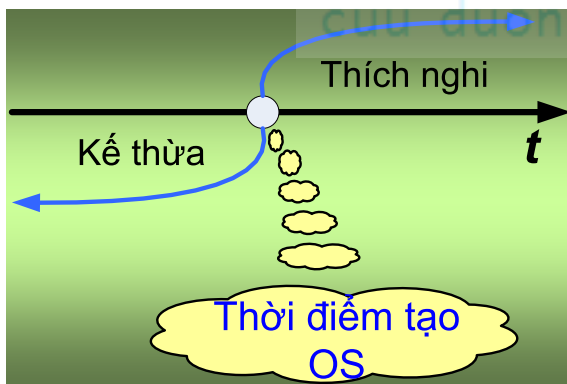
33

BẢO VỆ

- Hạn chế truy nhập không hợp thức,
- Hạn chế ảnh hưởng sai sót vô tình hay cố ý,
- Bảo vệ:
 - Nhiều mức,
 - Nhiều công cụ,
 - Nhiều thời điểm và giai đoạn khác nhau.
- **Chú ý:** *bảo vệ và chống bảo vệ: cùng mức → không thể đảm bảo an toàn tuyệt đối!*

34

Kế thừa và thích nghi



35

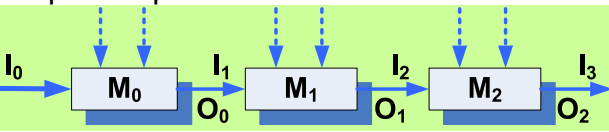
5 - NGUYÊN LÝ TỔ CHỨC VÀ HOẠT ĐỘNG

- Nguyên lý mô đun,
- Nguyên lý phủ chức năng,
- Nguyên lý Macroprocessor,
- Nguyên lý bảng tham số điều khiển,
- Nguyên lý giá trị chuẩn,
- Nguyên lý 2 loại tham số.

36

NGUYÊN LÝ MÔ ĐUN

- Mỗi công việc \Leftrightarrow mô đun CT độc lập,
- Các mô đun – liên kết với nhau thông qua Input/Output:



- Các mô đun được nhóm theo chức năng \rightarrow thành phần hệ thống.

37

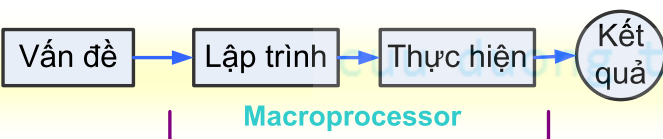
NGUYÊN LÝ PHỦ CHỨC NĂNG

- Mỗi công việc trong hệ thống, thông thường có thể thực hiện bằng nhiều cách với nhiều công cụ khác nhau,
- Lý do:**
- Mỗi mô đun có **hiệu ứng phụ** chức năng,
- Người dùng có quyền khai thác mọi hiệu ứng phụ không phụ thuộc vào việc công bố,
- Lập trình: Phải đảm bảo các tính chất của OS với mọi hiệu ứng phụ,
- Vai trò:**
 - Đảm bảo thuận tiện cho người dùng,
 - Đảm bảo an toàn chức năng của hệ thống,
- Ví dụ:** In một file.

38

NGUYÊN LÝ MACROPROCESSOR

- Trong OS không có sẵn CT giải quyết v/d,
- Khi cần thiết: Hệ thống tạo ra CT và thực hiện CT tạo ra:



- Nguyên lý này áp dụng với cả bản thân **toàn bộ OS**: Trên địa chỉ có các thành phần. Khi cần các thành phần được lắp ráp thành **HỆ ĐIỀU HÀNH** (Nạp hệ thống).

Lưu ý: Các nguyên lý **Phủ chức năng** và **Macroprocessor** trái với lý thuyết lập trình có cấu trúc.

39

NGUYÊN LÝ BẢNG THAM SỐ ĐIỀU KHIỂN

Mỗi đối tượng trong OS \Leftrightarrow Bảng tham số (Control Table, Control Block),

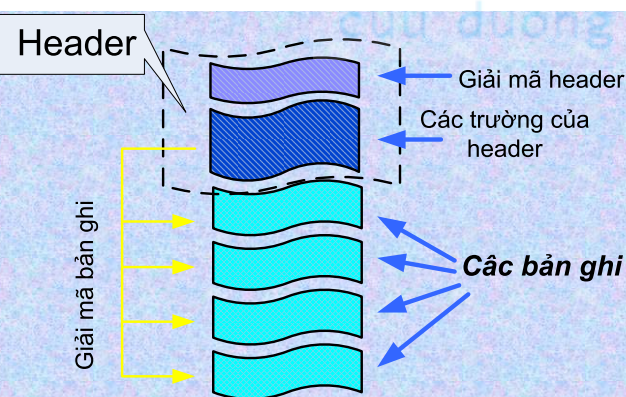
Hệ thống không bao giờ tham chiếu tới đối tượng vật lý mà chỉ tham chiếu tới bảng tham số điều khiển tương ứng.

Với các đĩa từ, CD – bảng tham số ghi ở phần đầu – Vùng hệ thống (System Area),

Với các files – Header.

40

Cấu trúc file định kiểu



Một số loại bảng tham số :

- Cho WINDOWS: Win.ini,
- Cho MS DOS: Config.sys,
- Cho WINWORD: Winword.ini,
- Bảng tham số cấu hình hệ thống: phục vụ cho mọi hệ điều hành: lưu trữ trong CMOS,

41

NGUYÊN LÝ GIÁ TRỊ CHUẨN

- Cách gọi khác: Nguyên tắc ngầm định (Default),
- Hệ thống chuẩn bị bảng giá trị cho các tham số - bảng giá trị chuẩn,
- Khi hoạt động: nếu tham số thiếu giá trị → OS lấy từ bảng giá trị chuẩn.

Vai trò của nguyên lý:

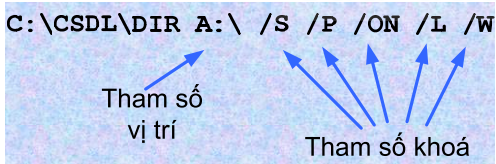
- Thuận tiện: không phải nhắc lại những giá trị thường dùng,
- Người dùng không cần biết đầy đủ hoặc sâu về hệ thống.

Nguyên lý giá trị chuẩn

- Tác động lên giá trị tham số hoặc bảng giá trị chuẩn:
 - Startup,
 - Autoexec.bat,
 - Control Panel
- Ví dụ: c:\csdl>dir
 - Tham số thiếu giá trị:
 - Ổ đĩa?
 - Thư mục?
 - Xem gì?
 - Quy cách đưa ra?
 - Nơi ra?

NGUYÊN LÝ 2 LOẠI THAM SỐ

- 2 loại tham số:
- Tham số vị trí (Position Parameters),
- Tham số khoá (Keyword Param.).



- Tham số khoá – theo trình tự tùy ý.

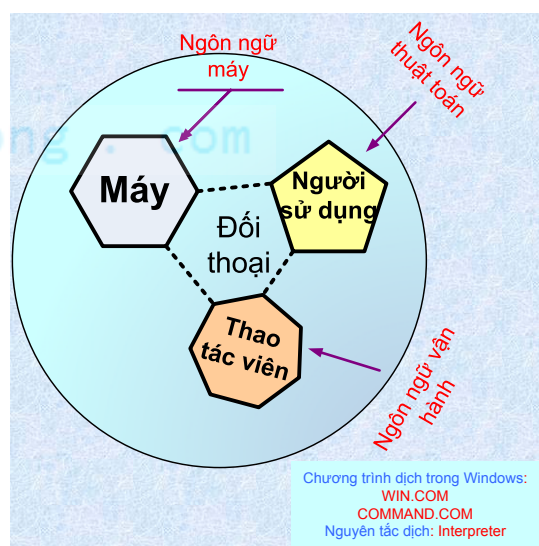
6 – THÀNH PHẦN

- Nhiều các phân chia theo chức năng, mức độ chi tiết,
 - Hệ thống Supervisor,
 - Hệ thống quản lý thiết bị ngoại vi,
 - Hệ thống quản lý files,
 - Hệ thống các chương trình điều khiển:
 - Điều phối nhiệm vụ,
 - Monitor,
 - Biên bản hệ thống,
- Các chương trình phục vụ hệ thống.

Thành phần

Lưu ý: ngôn ngữ không phải là thành phần hệ thống, nhưng trong thành phần hệ thống có một số CT dịch.

Phân biệt: Chương trình phục vụ hệ thống và chương trình ứng dụng



II – QUẢN LÝ FILES VÀ THIẾT BỊ NGOẠI VI

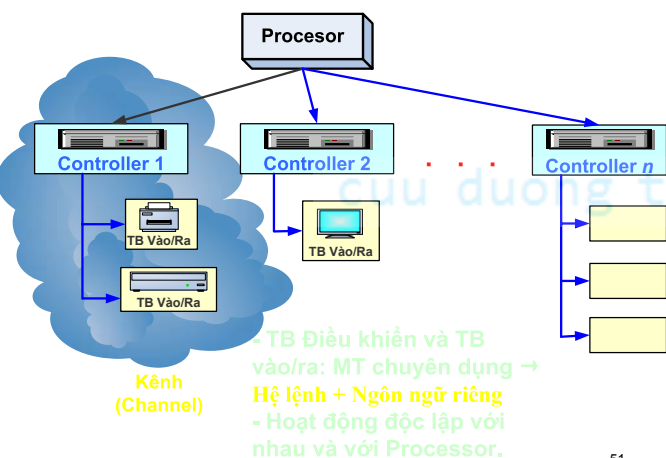
- **Quản lý thiết bị ngoại vi:** Cần đảm bảo hệ thống thích nghi với:
 - Số lượng nhiều,
 - Chất lượng đa dạng,
 - Thuận tiện cho người dùng.
- **Quản lý files:** Cho phép người dùng:
 - Tạo files ở các loại bộ nhớ ngoài,
 - Tìm kiếm, truy nhập files,
 - Đảm bảo độc lập giữa CT và thiết bị

49

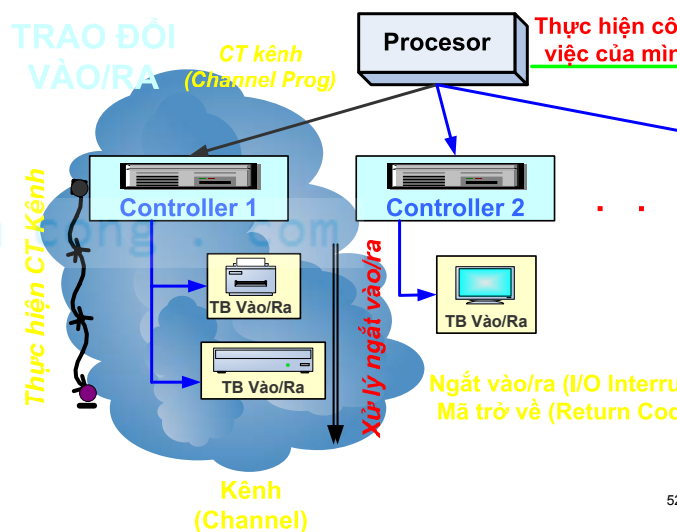
1 – Nguyên tắc phân cấp trong quản lý thiết bị ngoại vi

- **Máy tính thế hệ I và II:** Processor làm việc trực tiếp với thiết bị ngoại vi,
- **Hạn chế:** Tốc độ - Số lượng - Chủng loại,
- **Từ thế hệ III trở lên:**
 Processor → TB điều khiển → TB ngoại vi
 (Control Devices)
 (Controllers)

50



51



52

Nguyên tắc phân cấp trong quản lý thiết bị ngoại vi

Phép trao đổi vào ra: thực hiện theo nguyên lý Macroprocessor,

Với máy vi tính: Thiết bị điều khiển vào ra ≡ I/O Card,

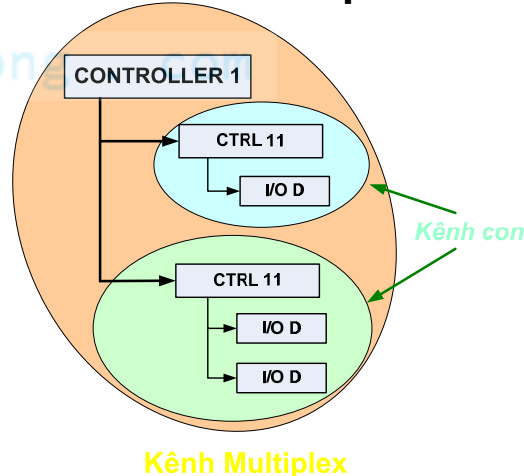
Máy Card on Board,

Lập trình trên Card vào/ra: Viết TOOLS khởi tạo chương trình kênh,

Khái niệm kênh bó (Multiplex), Card Multimedia.

53

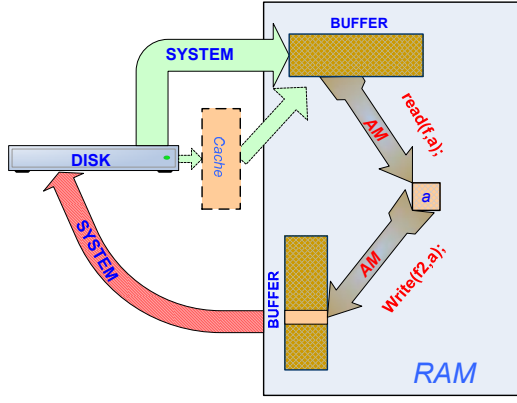
Kênh Multiplex



54

2 - KỸ THUẬT PHÒNG ĐỆM

Khái niệm phòng đệm (Buffer) của OS.



55

KỸ THUẬT PHÒNG ĐỆM

- Cơ chế phục vụ phòng đệm,
- Vấn đề đóng file output, FLUSH(F),
- Vai trò phòng đệm:
 - Song song giữa trao đổi vào ra và xử lý,
 - Đảm bảo độc lập:
 - Thông tin và phương tiện mang,
 - Bản ghi lô gíc và vật lý,
 - Lưu trữ và xử lý,
 - Giảm số lần truy nhập vật lý: Giả thiết mỗi lần truy nhập vật lý: 0.01", truy nhập kiểu BYTE.

56

KỸ THUẬT PHÒNG ĐỆM

	Không có Buffer	Buffer 512B
1B	0.01"	0.01"
512B	~5"	0.01"
5KB	~50"	0.1"
50KB	~8'	1"

57

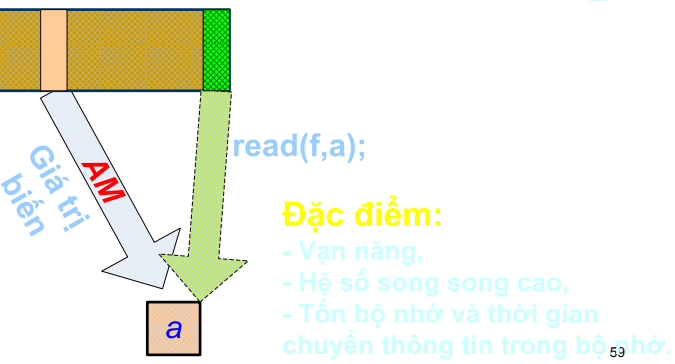
Các loại phòng đệm

- Phòng đệm chung hoặc gắn với file,
- Các Hệ QTCSDL còn hệ thống phòng đệm riêng để nâng độ linh hoạt và tốc độ xử lý,
- Các loại bộ nhớ Cache và phòng đệm.
- Ba kiểu tổ chức chính:
 - Phòng đệm **truy nhập theo giá trị**,
 - Phòng đệm **truy nhập theo địa chỉ**,
 - Phòng đệm **vòng tròn**.

58

Các loại phòng đệm

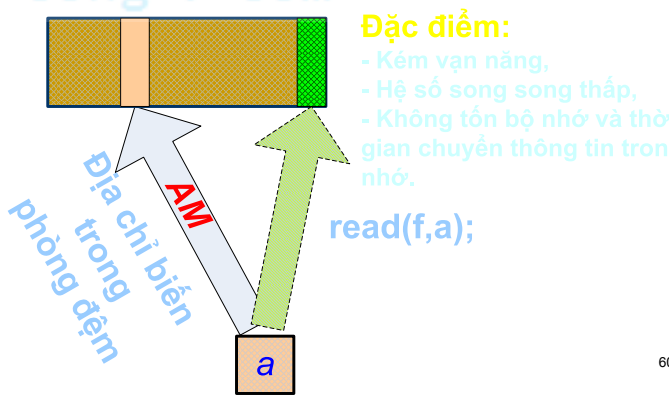
- A) Phòng đệm **truy nhập theo giá trị**:



59

Các loại phòng đệm

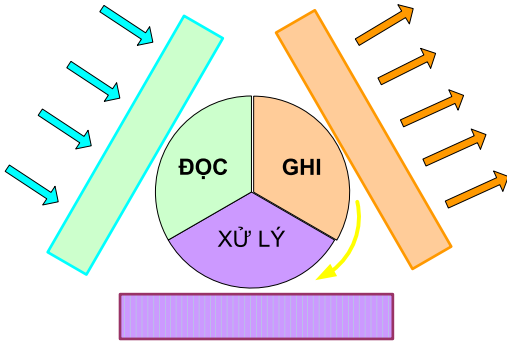
- B) Phòng đệm **truy nhập theo địa chỉ**:



60

Các loại phòng đệm

C) Phòng đệm vòng tròn: thường áp dụng cho các hệ QT CSDL.



61

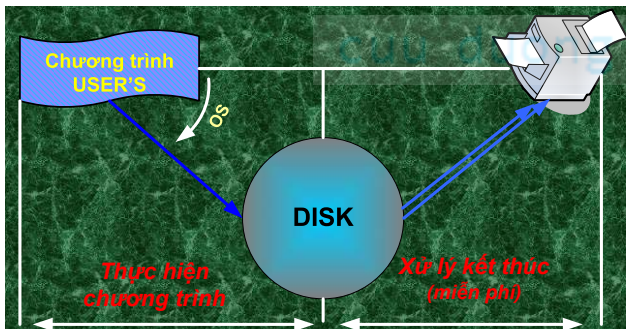
3 - SPOOL

- SPOOL – Simultaneous Peripheral Operations On-Line,
- Không can thiệp vào CT người dùng,
- Hai giai đoạn:
 - Thực hiện: thay thế thiết bị ngoại vi bằng thiết bị trung gian (Đĩa cứng),
 - Xử lý kết thúc:
 - Sau khi kết thúc việc thực hiện CT,
 - Đưa thông tin ra thiết bị yêu cầu.
- Chú ý: Đặc trưng của thiết bị trung gian.

62

SPOOL

- Đảm bảo song song giữa xử lý một CT với trao đổi vào ra của CT khác.



64

SPOOL

- Giải phóng hệ thống khỏi sự ràng buộc về số lượng thiết bị,
- Khai thác thiết bị ngoại vi tối ưu,
- Kỹ thuật lập trình hiệu quả.
- Hệ thống cung cấp các phương tiện để người dùng tạo SPOOL,
- Ai tạo SPOOL – người đó xử lý kết thúc.

64

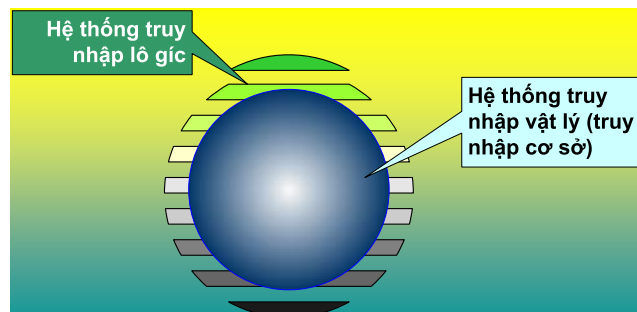
Tổ chức SPOOL

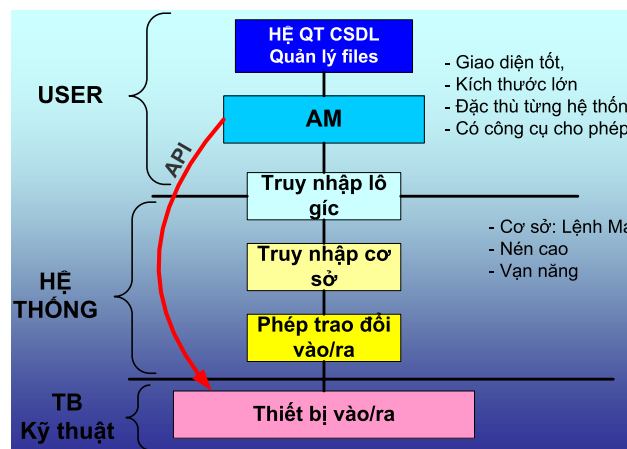
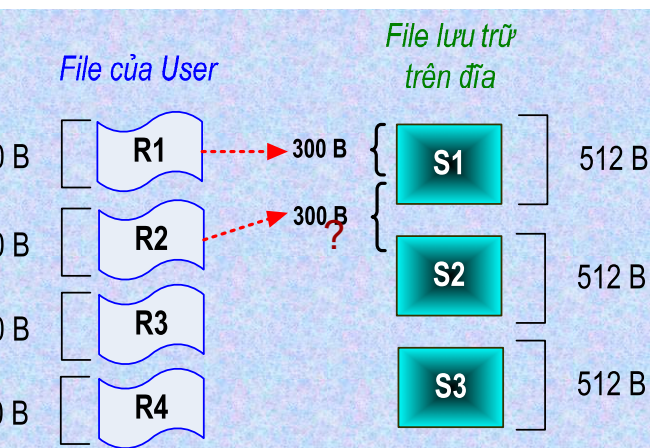
- **Giai đoạn thực hiện:** với mỗi phép trao đổi vào ra hệ thống tạo 2 CT kênh:
 - CT kênh I – theo thiết bị yêu cầu,
 - CT kênh II – phục vụ ghi CT kênh I ra thiết bị trung gian,
- **Xử lý kết thúc:** Đọc CT kênh đã lưu và chuyển giao cho kênh.
- Như vậy, mỗi thiết bị sử dụng → file CT kênh.

65

4 – HỆ THỐNG QUẢN LÝ FILES

- ∃ CSDL quản lý files,
- Hệ thống quản lý files - Hệ QT CSDL.

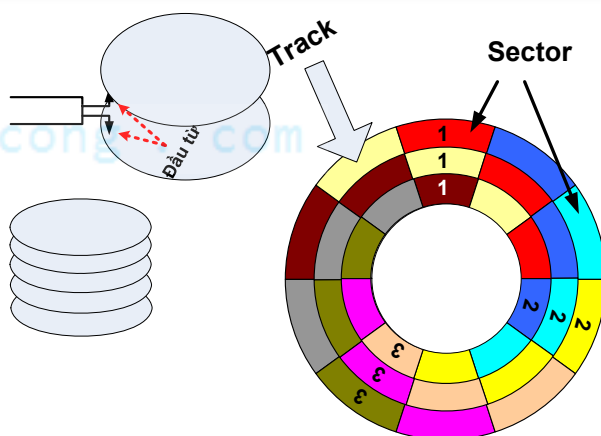




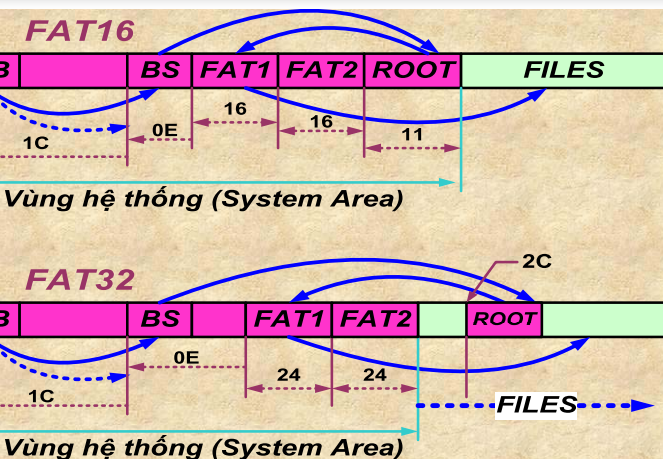
QUẢN LÝ FILE TRONG WINDOWS

- Mục đích:
 - Minh họa nguyên lý bảng tham số điều khiển,
 - Tính kế thừa và thích nghi,
 - Cơ chế bảo vệ,
 - Cách thể hiện một số chế độ quản lý bộ nhớ (chương tiếp theo).

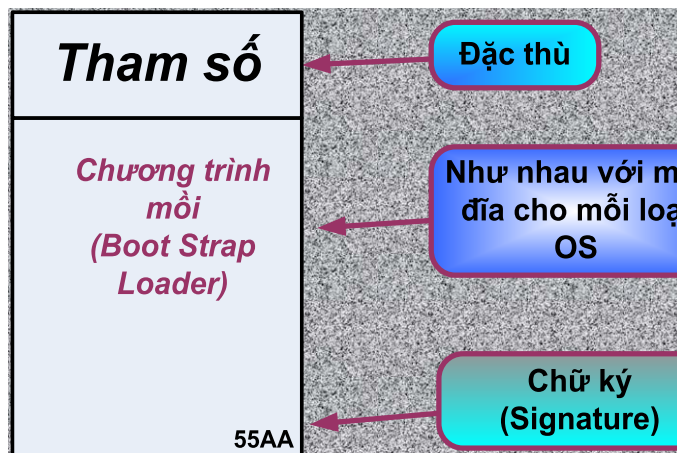
TỔ CHỨC THÔNG TIN TRÊN ĐĨA T



CẤU TRÚC THÔNG TIN TRÊN ĐĨA TỪ



BOOT SECTOR



BOOT SECTOR

Offset	Offs	L (Byte)	Ý Nghĩa
0	3		Lệnh JMP (EB xx 90)
3	8		Tên hệ thống Format đĩa
B	2		Kích thước Sector
D	1		Sec/Cluster
E	2		Địa chỉ tuyệt đối FAT1 trong đĩa lô gíc
10 _H	1		Số lượng bảng FAT
11	2		FAT16: Số phần tử ∈ ROOT FAT32: 00 00
13	2		Σ sect/Disk (<32MB) hoặc 00 00

BOOT SECTOR

9	15	1	Kiểu đĩa từ (F8 – HD, F0 – 1.44MB)
10	16	2	FAT16: Σ Sec/FAT FAT32: 00 00
11	18	2	Sec/Track
12	1A	2	Số đầu từ
13	1C	4	Địa chỉ tuyệt đối BS trong đĩa vật lý
14	20	4	Σ Sec / Disk (≥32MB) hoặc 0
15	24	4	Σ Sec / FAT
16	28	2	Flags
17	2A	2	Version
18	2C	4	Địa chỉ ROOT (Cluster)

30	2	Inf
32	2	Địa chỉ lưu BS
34	12 ₁₀	Dự trữ (00...00)
40	1	Địa chỉ ổ đĩa (80 – C:)
41	1	00
42	1	29 – BIOS mở rộng
43	4	Serial Number
47	11 ₁₀	Volume Name
52	8	FAT32

Boot Sector FAT 16

15	24	1	Địa chỉ ổ đĩa (80 – C:)
16	25	1	00
17	26	1	29 – BIOS mở rộng
18	27	4	Serial Number
19	2B	11 ₁₀	Volume Name
20	36	8	FAT16

Ví dụ

58 90 4D 53 57 49 4E 34 2E 31 00 02 08 2D 00
 00 00 00 00 F8 00 00 3F 00 40 00 3F 00 00 00
 0C 34 00 03 0D 00 00 00 00 00 00 02 00 00 00
 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 29 D1 09 47 32 20 20 20 20 20 20 20 20 20
 20 46 41 54 33 32 20 20 20 20 FA 33 C9 8E 41 BC

79

THƯ MỤC

- Đóng vai trò mục lục tra cứu, tìm kiếm,
- Mọi hệ thống đều phải có với những tên khác nhau (Catalog, Directory, Folder, . . .),
- Bao gồm: Thư mục gốc (ROOT) + Thư mục con,
- Các hệ thống của Microsoft và OS IBM – cấu trúc cây,
- UNIX - cấu trúc phân cấp,
- Thư mục = {Phần tử}, mỗi phần tử: 32_{10} B
- Phần tử \leftrightarrow file,
- Thư mục con và ROOT: File có cấu trúc.

80

Cấu trúc phần tử thư mục tên ngắn (Phần tử 8.3)

Stt	Offs	L	Ý nghĩa
	0	8	Tên (Name)
	8	3	Phần mở rộng (Extention)
	B	1	Thuộc tính (Attribute)
	C	2	Thời điểm tạo file
	E	2	Ngày tạo file
	10 _H	2	Ngày truy nhập gần nhất
	12	1	00 (Cho NT)
	13	1	Số 0.1" của thời điểm tạo file

81

Phần tử 8.3

9	14	2	2 bytes cao của cluster xuất phát
10	16	2	Thời điểm cập nhật cuối cùng
11	18	2	Ngày cập nhật cuối cùng
12	1A	2	2 bytes thấp của cluster xuất phát
13	1C	4	Kích thước file (Byte)

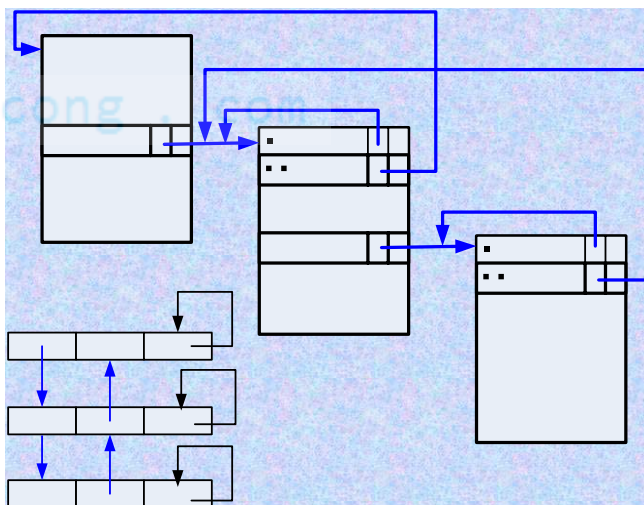
82

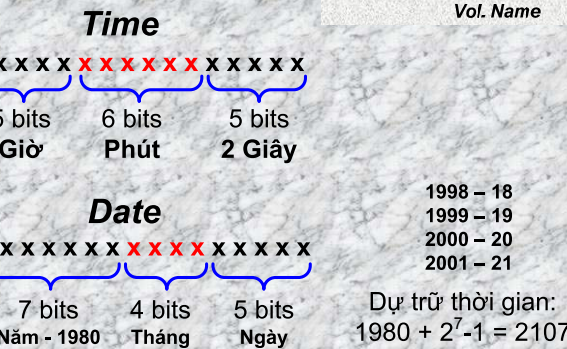
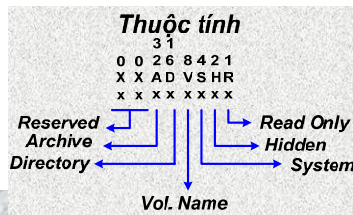
Phần tử 8.3

byte số 0: Vai trò đặc biệt.

0 – Chưa sử dụng, phần tử chưa sử dụng đầu tiên - dấu hiệu kết thúc thư mục,
 5 – (σ) Đã bị xoá,
 5 – Tên bắt đầu bằng ký tự σ,
 E 20 (.) – Phần tử thứ I của thư mục con,
 2E 2E (..) – Phần tử thứ II của thư mục con

xx	
xx	
xx	
xx	
00	





85

Tên dài

- Không quá 255 ký tự,
- Unicode,
- Hệ thống phân biệt theo 66 ký tự đầu tiên,
- Lưu trữ theo cách đưa vào,
- Nhận dạng: Đưa về chữ hoa.

Lưu trữ tên dài

Phần tử tên dài n

Phần tử tên dài n-1

⋮

Phần tử tên dài 1

Phần tử 8.3

Cấu trúc phần tử tên dài

Stt	Offs	L	Ý nghĩa
0	1	1	Số thứ tự i (64+i)
1	10 ₁₀	5	5 ký tự C ₁ - C ₅
B	1	1	Thuộc tính (00001111 _B)
C	1	1	00 - dấu hiệu phần tử tên dài
D	1	1	Σ _K phần tử 8.3
E	12 ₁₀	12	C ₆ - C ₁₁
1A	2	2	00 00
1C	4	4	C ₁₂ - C ₁₃

87

File Allocation Table (FAT)

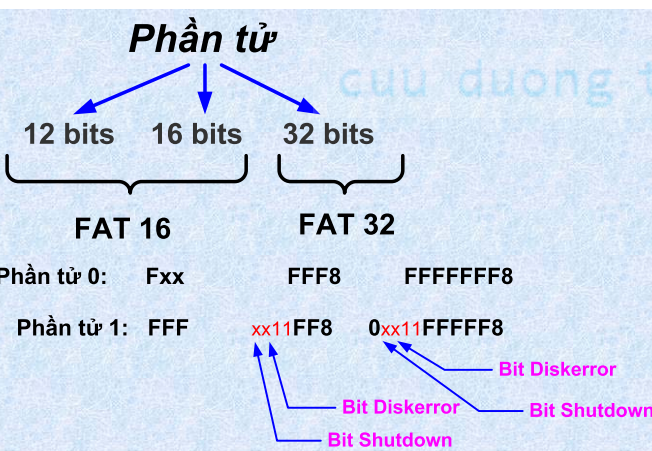
- Chức năng:
 - Quản lý bộ nhớ phân phối cho từng file,
 - Quản lý bộ nhớ tự do trên đĩa,
 - Quản lý bộ nhớ kém chất lượng.
- FAT = {phần tử}
- Phần tử:
 - Đánh số: 0, 1, 2, ...
 - Từ phần tử số 2: phần tử ↔ Cluster

88

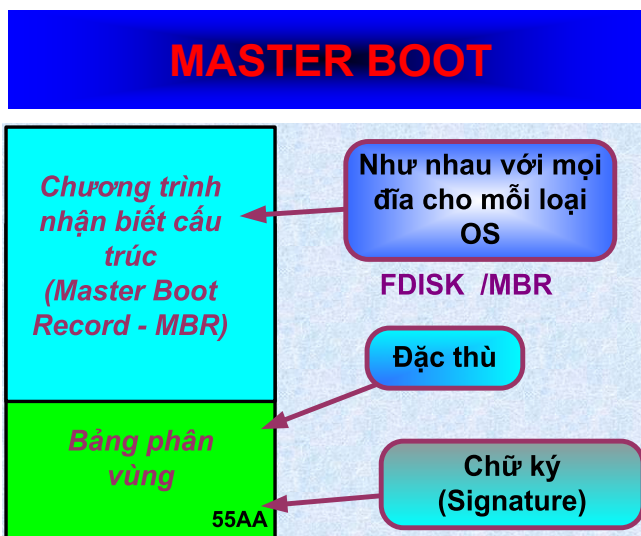
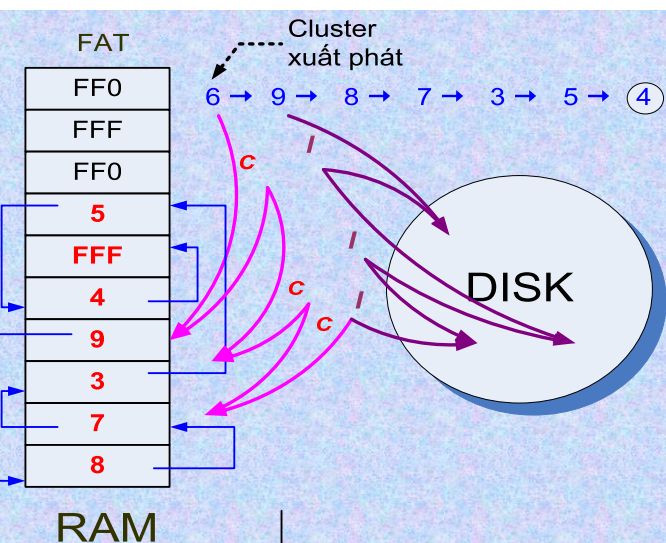
FAT

- Bit Shutdown = 1 - Ra khỏi hệ thống đúng cách
- Bit Diskerror = 1 - không có lỗi truy nhập đĩa ở lần truy nhập cuối cùng.
- Từ phần tử 2 trở đi:
- Giá trị 0 - Cluster tự do,
- FF7 (FFF7, 0FFFFFF7) - Bad cluster,
- Các giá trị khác - đã phân phối,
- Các phần tử tương ứng những Clusters của một file tạo thành một danh sách móc nối,
- EOC (End of Cluster Chain) - FFF (FFFF, FFFFFFFF).

90



89



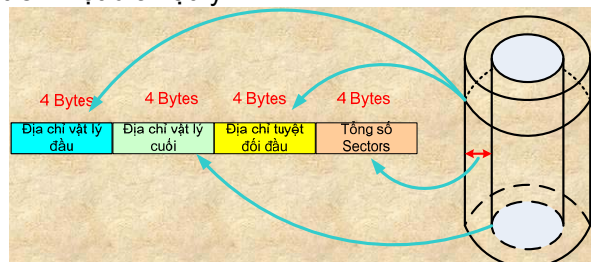
MASTER BOOT

- Nguyên tắc khai thác HD:
 - Chia HD thành các phần, mỗi phần có kích thước cố định,
 - Mỗi phần sử dụng như một đĩa từ độc lập: Đĩa lô gic (Logical Volume).
- OS cho phép tạo các đĩa kích thước động trong mỗi đĩa lô gic.

93

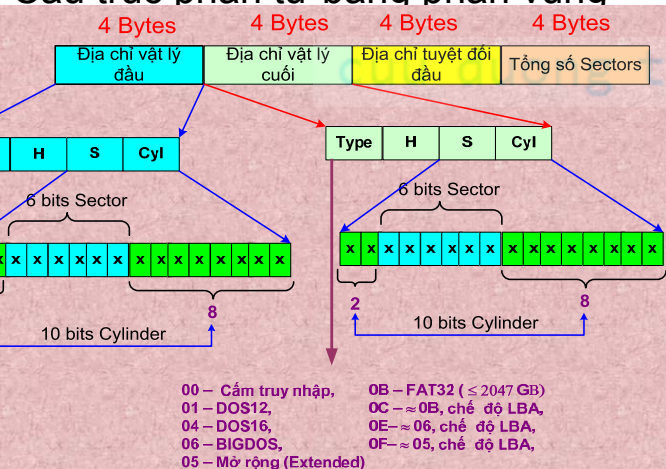
Cấu trúc bảng phân vùng

- Bảng phân vùng bắt đầu từ địa chỉ **1BE_H**,
- Bảng phân vùng = {4 phần tử},
- Mỗi phần tử sử dụng ↔ **Đĩa lô gic**,
- Tồn tại cơ chế cho phép tạo **nhiều hơn 4** đĩa lô gic trên một đĩa vật lý.



94

Cấu trúc phần tử bảng phân vùng



Bảng phân vùng

```

80 01 01 00 0B 3F FF 4D 3F 00 00 00 41 0C 34 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
55AA

80010100 010511BF 11000000 6F4C0000
000001C0 5105511F 804C0000 40260000
00004120 510551DF C0720000 804C0000
000041E0 5105D132 40BF0000 12870000
55AA
  
```


Truy nhập Boot Sector

Dùng các hàm API,

Chương trình đọc và đưa ra màn hình nội dung BS của đĩa mềm A: (Hexa và ASCII):

```
program R_BS_A;
uses Crt, Dos;
const s16: string[16]='0123456789ABCDEF';
var B: array[0..511] of char;
    reg: registers;
    i,j,k: integer;
    c: char;
begin
  clrscr;
  fillchar(b,sizeof(b),0);
  writeln('Cho dia vao o A: va bam phim bat ky.');
```

```
  c:=readkey;
```

```
I := 0;
```

```
Repeat
```

```
  with reg do
```

```
    begin
```

```
      dl := 0; { 0 -> A:, 128 -> C:}
```

```
      dh := 0; {Đầu từ}
```

```
      cl := 1; {Sector}
```

```
      ch := 0; {Cylinder}
```

```
      al := 1; {So Sectors can doc}
```

```
      ah := 2; {2 -> Read; 3 -> Write; . . .}
```

```
      es := seg(b);
```

```
      bx := ofs(b)
```

```
    end;
```

```
  intr($13,reg);
  inc(i)
  Until i = 2;
  for i := 0 to 511 do
    begin
      j := b[i] shr 4 + 1;
      k := b[i] and $0F + 1;
      write(s16[j]:2, s16[k]);
      if (i+1) mod 16 = 0 then
        begin write(' ':5);
          for j := i-15 to i do
            if (b[j] < 32) or (b[j] = 255) then
              write('.')
            else write(chr(b[j]));
```

```
  writeln;
```

```
    if i = 255 then c:= readkey
```

```
  end
```

```
end;
```

```
Repeat
```

```
  Until keypressed
```

```
END.
```

III – QUẢN LÝ BỘ NHỚ

Bộ nhớ tác động nhiều lên độ phức tạp của giải thuật,

phải giải quyết 2 v/d trái ngược nhau:

tiết kiệm bộ nhớ,

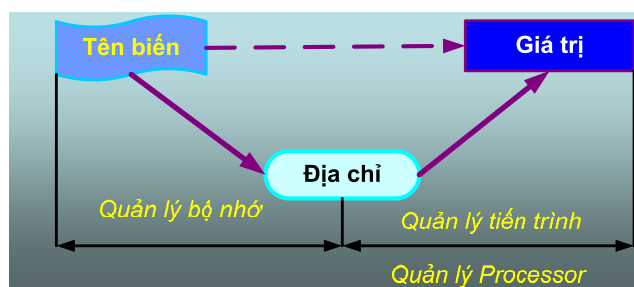
tận dụng tối đa bộ nhớ cho phép.

Phần lớn các chương trình: viết trên ngôn ngữ lập trình: Assembler, VB, JAVA, VC++, ...

Với người lập trình: CT và thực hiện CT là ánh xạ từ tên sang giá trị.

QUẢN LÝ BỘ NHỚ

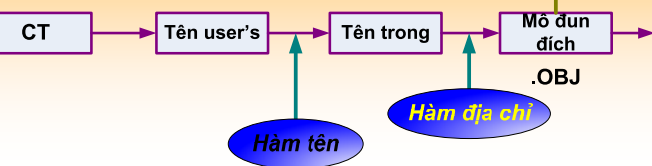
- Với hệ thống:



1 – CÁC BƯỚC XỬ LÝ CT

Lý thuyết chương trình dịch

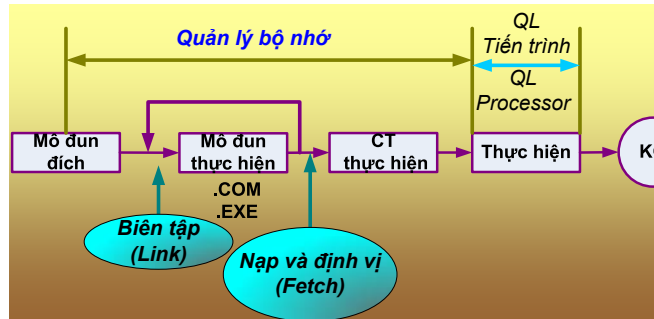
Ph.tích cú pháp + ph. tích ngữ nghĩa + Sinh mã + Tối ưu hoá



I + J
A + B
A + I

103

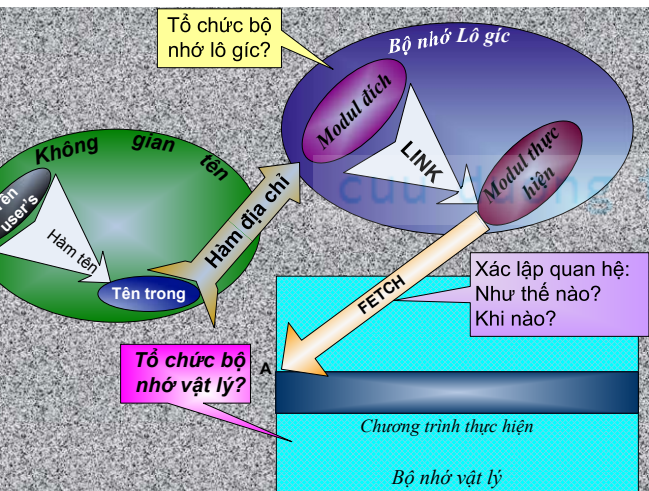
CÁC BƯỚC XỬ LÝ CT



- Vai trò của Biên tập (Input/Output),
- Khái niệm bộ nhớ lô gíc.

104

CÁC BƯỚC XỬ LÝ CT



2 – CẤU TRÚC CHƯƠNG TRÌNH

- Bộ nhớ lô gíc:
 - Không gắn với máy tính cụ thể,
 - Không giới hạn về kích thước,
 - Chỉ chứa 1 mô đun hoặc 1 CT,
 - Chỉ phục vụ lưu trữ, không thực hiện.
- Quản lý bộ nhớ lô gíc ~ tổ chức chương trình,
- Mỗi cách tổ chức CT \Leftrightarrow cấu trúc CT,
- Mọi cấu trúc: đều được sử dụng trong thực tế.

105

CẤU TRÚC CHƯƠNG TRÌNH

Đặc trưng mô đun đích (Object Modul): chứa thông tin về các moduls khác liên quan (các móc nối) \rightarrow kích thước lớn.

Nhiệm vụ biên tập (Linked): Giải quyết các móc nối.

Các loại cấu trúc chính:

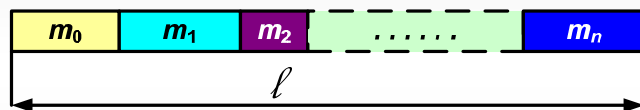
- Cấu trúc tuyến tính,
- Cấu trúc động (Dynamic Structure),
- Cấu trúc Overlay,
- Cấu trúc mô đun,
- Cấu trúc phân trang.

Một chương trình thực hiện có thể chứa nhiều cấu trúc khác nhau.

107

CẤU TRÚC CHƯƠNG TRÌNH

- A) **Cấu trúc tuyến tính**: CT biên tập tìm và lắp ráp các mô đun thành một mô đun duy nhất, chứa đầy đủ thông tin để thực hiện CT



108

Cấu trúc tuyến tính

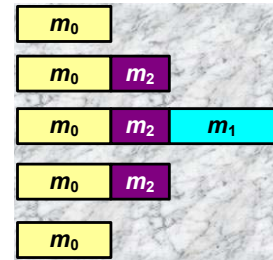
• Đặc điểm:

- Đơn giản,
- Thời gian thực hiện: min,
- Lưu động (mobile) cao,
- Tốn bộ nhớ: với mỗi bộ dữ liệu chỉ có 13% - 17% câu lệnh đóng vai trò tích cực.
- Không dùng chung mô đun CT.

109

B) CẤU TRÚC ĐỘNG

- Trong CT nguồn: phải dùng các lệnh macro hệ thống để nạp, móc nối, xóa (Load, Attach, Delete) . . . các mô đun khi cần thiết,



110

CẤU TRÚC ĐỘNG

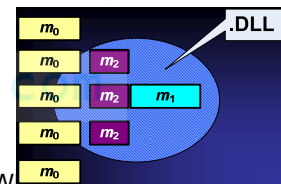
• Đặc điểm:

- Đòi hỏi user phải biết cơ chế và công cụ quản lý bộ nhớ,
- Thời gian thực hiện lớn: song song thực hiện với tìm kiếm, nạp và định vị,
- Tiết kiệm bộ nhớ,
- Kém lưu động → khó nạp, cập nhật, xóa.
- Được sử dụng rộng rãi những năm 60-70 và từ 90 đến nay.
- Thích hợp cho các CT hệ thống.

111

CẤU TRÚC ĐỘNG

- Các mô đun nạp trong quá trình thực hiện → vào các files .DLL (dynamic Link Library)

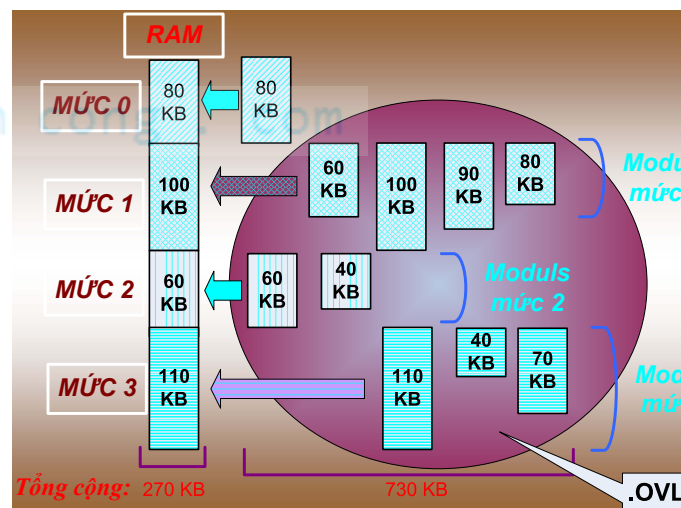


- WINDOWS 98, WINDOWS XP và hệ thống SYSTEM, SYSTEM32,
- Biên bản cài đặt, uninstall.
- Winword, Excel, Vietkey . . .
- Các ngôn ngữ lập trình: ∃ công cụ tổ chức DLL.

112

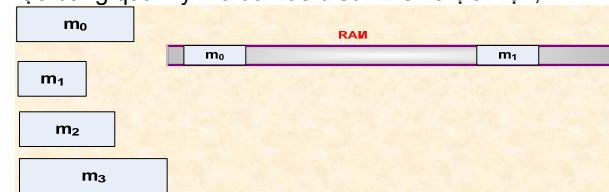
- Moduls → các lớp, lớp = {các moduls không tồn tại đồng thời}
- Moduls lớp i được gọi bởi moduls lớp i-1,
- Thông tin về các lớp: Sơ đồ tổ chức overlay, do user cung cấp cho Link,
- Link tạo sơ đồ quản lý overlay,
- Supervisor Overlay tổ chức thực hiện.
- Đặc điểm:
- Phân phối bộ nhớ theo sơ đồ tĩnh,
- Files .OVL
- Ví dụ: FOXPRO, PCSHELL. . .

113



D) CẤU TRÚC MODULS

Biên tập riêng từng mô đun,
Tạo bảng quản lý mô đun để điều khiển thực hiện,



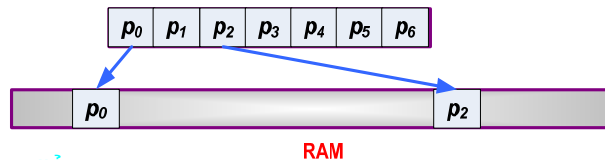
Đặc điểm:

- 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 mô đun.

115

E) CẤU TRÚC PHẦN 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 vào cấu trúc ban đầu của CT nguồn.

116

3 - QUẢN LÝ BỘ NHỚ VẬT LÝ

Đặc điểm:

- Có kích thước cụ thể,
- Có cấu hình sử dụng cụ thể.

Phục vụ giai đoạn thực hiện CT:

- Bảo vệ thông tin,
- Bộ nhớ cho dữ liệu.

Vấn đề:

- Cách tổ chức?
- Xác lập quan hệ với bộ nhớ lô gíc: như thế nào và khi nào?
- Tình huống thiếu bộ nhớ?

117

QUẢN LÝ BỘ NHỚ VẬT LÝ

- Các chế độ quản lý bộ nhớ vật lý:
 - Chế độ phân vùng cố định,
 - Chế độ phân vùng động,
 - Chế độ mô đun,
 - Chế độ phân trang,
- Chế độ kết hợp mô đun và phân trang.
- Mọi chế độ: đều đang được sử dụng.

118

a) Chế độ phân vùng cố định

Bộ nhớ → n phần,

mỗi phần có kích

thước cố định

(thông nhất thiết

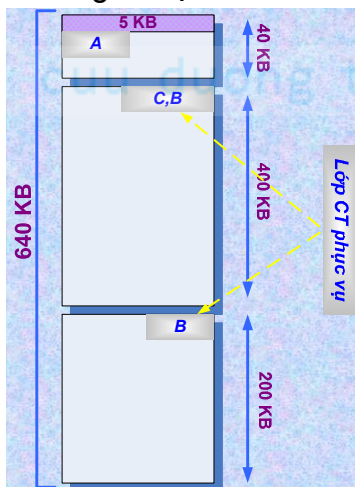
ng nhau),

sử dụng như một

bộ nhớ độc lập,

phục vụ thực hiện

CT.



Chế độ phân vùng cố định

Đặc điểm:

- Mỗi vùng có một danh sách quản lý bộ nhớ tự do
- Mỗi vùng: thực hiện một CT ứng dụng,
- Sơ đồ bảo vệ thông tin: theo toàn vùng.
- Một số CT điều khiển phải được copy vào từng vùng.
- Phân lớp CT phục vụ để hạn chế lãng phí bộ nhớ,
- Mô hình: Tổ chức đĩa cứng.

120

Chế độ phân vùng cố định

Công cụ phân bố lại bộ nhớ (**SWAPPING**):

- Lệnh OP,
- Do OP thực hiện,
- Những vùng nào biến thay đổi: mất thông tin. Lý do: làm lại DSQL bộ nhớ tự do.

Ví dụ: với đĩa cứng: FDISK.

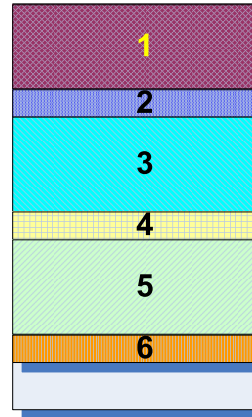
CT điều khiển hệ thống: đơn giản.

Hệ số song song cố định.

121

b) CHẾ ĐỘ PHÂN VÙNG ĐỘNG

- CT → Phân phối vùng bộ nhớ liên tục đủ thực hiện và quản lý như bộ nhớ độc lập.
- ∃ một danh sách QL bộ nhớ tự do duy nhất.



CHẾ ĐỘ PHÂN VÙNG ĐỘNG

Đặc điểm:

- Hệ số song song biến thiên,
- ∃ hiện tượng phân đoạn ngoài (External Fragmentation) → SWAPPING,

Công cụ SWAPPING:

- Lệnh OP,
- Do OP thực hiện,
- Không mất thông tin.

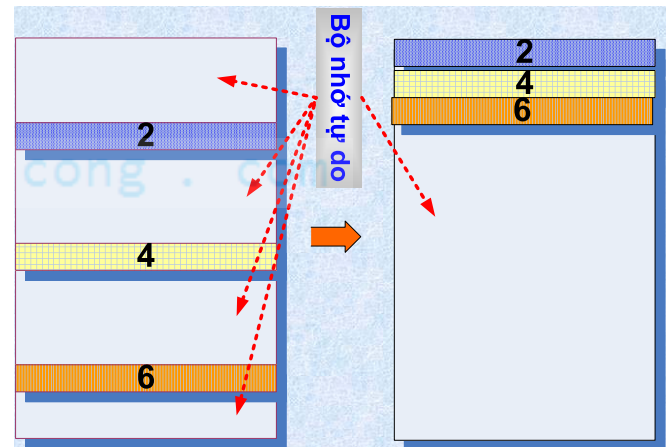
Nội dung SWAPPING.

Phức tạp của Swapping.

Mô hình quản lý đĩa từ SUBST, DRVSPACE

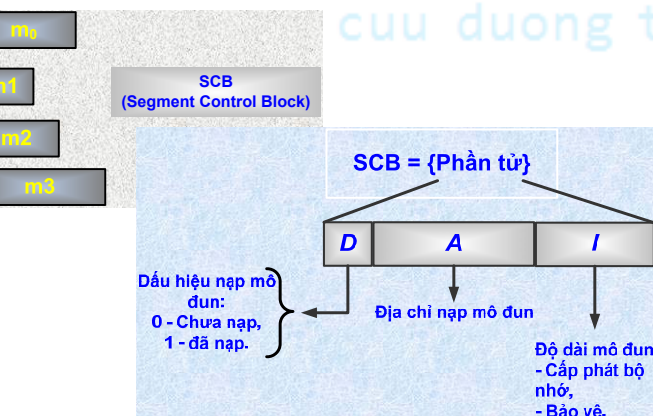
123

CHẾ ĐỘ PHÂN VÙNG ĐỘNG



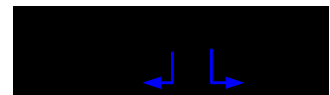
C) CHẾ ĐỘ QUẢN LÝ THEO MÔ ĐUN

CT – cấu trúc mô đun,



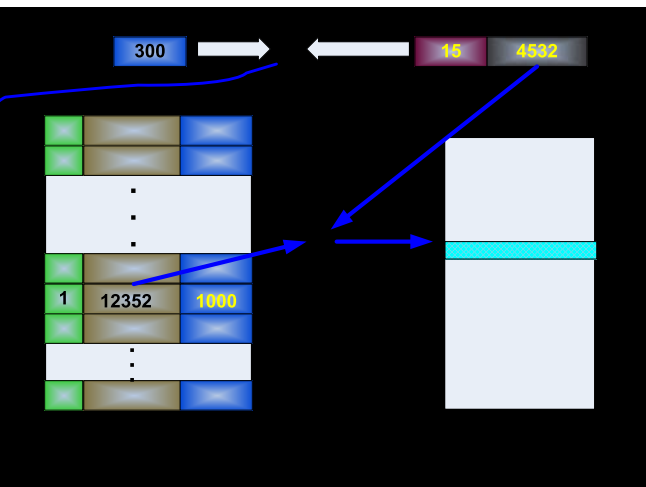
CHẾ ĐỘ QUẢN LÝ THEO MÔ ĐUN

- Thực hiện CT: địa chỉ dữ liệu phải biểu diễn dưới dạng một cặp



- SCB → RAM, địa chỉ đầu của SCB → R_s - Segment Register.
- Để đọc /ghi dữ liệu: cần 2 lần truy nhập tới bộ nhớ:
 - * $(R_s) + s \rightarrow$ truy nhập tới phần tử thứ $s \in$ SCB,
 - ** Khi $D = 1$: $A+d \rightarrow$ truy nhập tới dữ liệu.

124



CHẾ ĐỘ QUẢN LÝ THEO MÔ ĐUN

• Đặc điểm:

- Không cần phân phối bộ nhớ liên tục,
 - Không đòi hỏi công cụ đặc biệt → có thể áp dụng cho mọi MTĐT,
 - Dễ dàng sử dụng chung mô đun giữa các CT,
 - Hiệu quả phụ thuộc vào cấu trúc CT nguồn,
 - Tồn tại hiện tượng phân đoạn ngoài (External Fragmentation).
- Thiếu bộ nhớ, phạm đoạn ngoài → **Swapping**

129

CHẾ ĐỘ QUẢN LÝ THEO MÔ ĐUN

SWAPPING:

- Do hệ thống đảm nhiệm,
- Không mất thông tin,
- Nội dung swapping: đưa một hoặc một số mô đun ra bộ nhớ ngoài, giải phóng chỗ nạp mô đun mới.

Cách chọn mô đun đưa ra: Option

- Mô đun tồn tại lâu nhất trong bộ nhớ,
- Mô đun có lần sử dụng cuối cùng cách đây lâu nhất,
- Mô đun có tần xuất sử dụng thấp nhất.

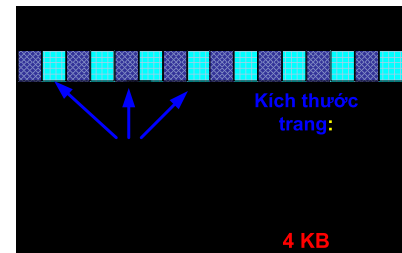
IBM PC 286 trở lên:

- Một trong 2 chế độ của 286 và một trong 3 chế độ của 386 trở lên,
- Swapping - ngầm định – **tiêu chuẩn 2**.

129

D) CHẾ ĐỘ PHÂN TRANG

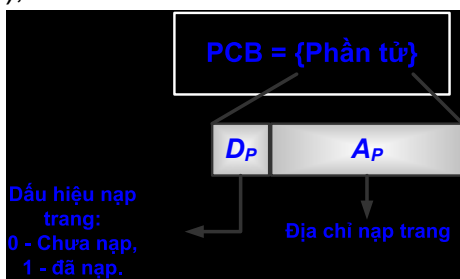
- Bộ nhớ được chia thành các phần bằng nhau – các trang (Pages),
- Các trang – đánh số 0, 1, 2, . . . - địa chỉ trang.



130

CHẾ ĐỘ PHÂN TRANG

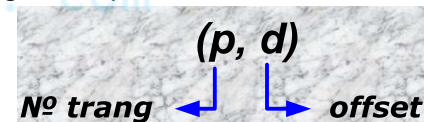
- CT - cấu trúc phân trang,
- Bảng quản lý trang PCB (Page Control Block),



131

CHẾ ĐỘ PHÂN TRANG

- Thực hiện CT: địa chỉ dữ liệu phải biểu diễn dưới dạng một cặp



- PCB → RAM, địa chỉ đầu của PCB → R_p - Page Register.
- Để đọc /ghi dữ liệu: cần 2 lần truy nhập tới bộ nhớ:
 - * $(R_p) + p \rightarrow$ truy nhập tới phần tử thứ $p \in$ PCB,
 - ** Khi $D_p = 1$: $A \cup d \rightarrow$ truy nhập tới dữ liệu.

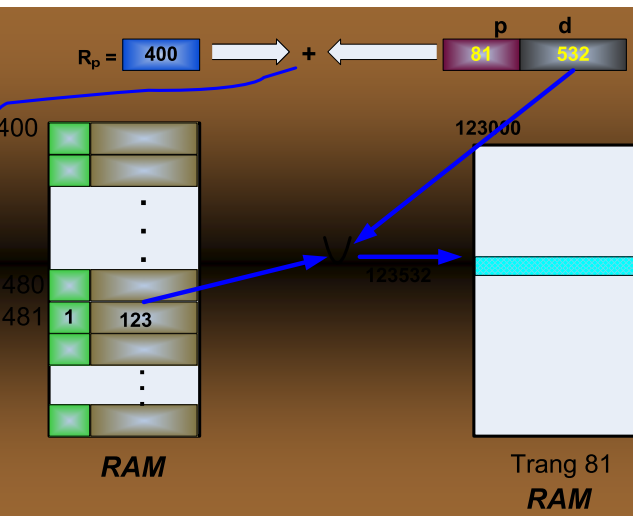
132

CHẾ ĐỘ PHÂN TRANG

• 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 mô đun giữa các CT,
- Hiệu quả không phụ thuộc vào cấu trúc CT nguồn,
- Bảng PCB có thể rất lớn,
- Không bị phân đoạn ngoài.

- Thiếu bộ nhớ (mọi trang đều đã được sử dụng)
→ **Swapping**



CHẾ ĐỘ PHÂN TRANG

• SWAPPING:

- Do hệ thống đảm nhiệm,
- Không mất thông tin,
- Nội dung swapping: đưa một trang ra bộ nhớ ngoài, giải phóng chỗ nạp trang mới.

Cách chọn trang đưa ra: Option

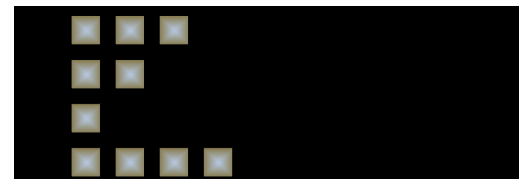
- Trang tồn tại lâu nhất trong bộ nhớ,
- Trang có lần sử dụng cuối cùng cách đây lâu nhất,
- Trang có tần xuất sử dụng thấp nhất.

IBM PC 386 trở lên: ngầm định – **tiêu chuẩn 2.**

135

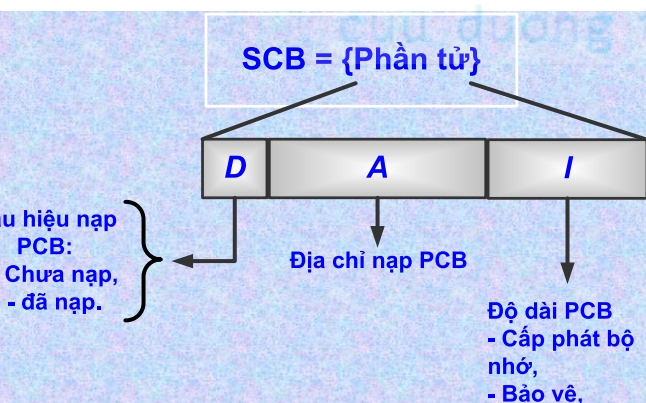
E) CHẾ ĐỘ KẾT HỢP MÔ ĐUN – PHÂN TRANG

- Bộ nhớ vật lý – phân trang,
- CT – cấu trúc mô đun,
- Mỗi mô đun – phân trang:



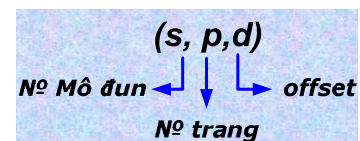
136

CHẾ ĐỘ KẾT HỢP MÔ ĐUN – PHÂN TRANG



CHẾ ĐỘ KẾT HỢP MÔ ĐUN – PHÂN TRANG

- Thực hiện CT: địa chỉ dữ liệu phải biểu diễn dưới dạng một nhóm 3:

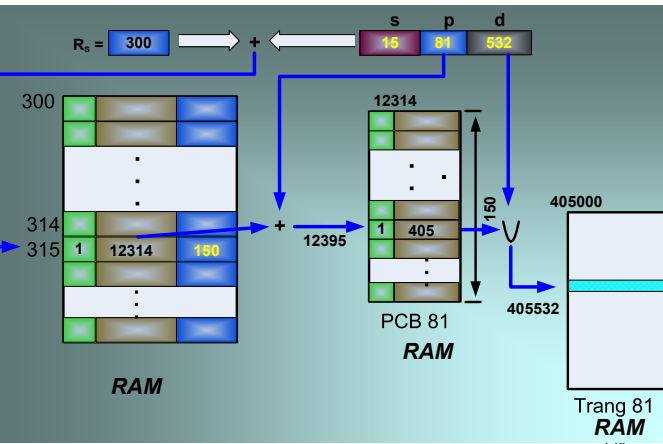
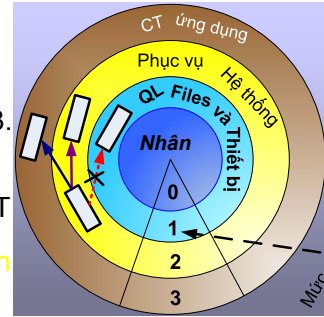


- SCB → RAM, địa chỉ đầu của SCB → R_s - Segment Register.
- Để đọc / ghi dữ liệu: cần 3 lần truy nhập tới bộ nhớ:
 - * $(R_s) + s \rightarrow$ truy nhập tới phần tử thứ $s \in SCB$,
 - ** Khi $D = 1$: $A + d \rightarrow$ truy nhập tới $PCB_s \in SCB$,
 - *** Khi $D_p = 1$: $A \cup d \rightarrow$ truy nhập tới dữ liệu.

137

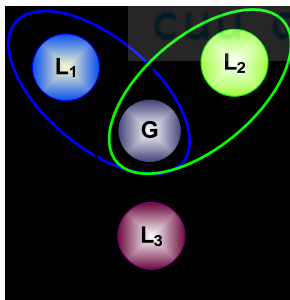
4 - QUẢN LÝ BỘ NHỚ TRONG IBM PC

- Bốn mức ưu tiên (Privilege Levels)
thực hiện CT: $0 \div 3$, cao nhất – 0, thấp nhất – 3.
- Nguyên tắc truy nhập: một CT chỉ được quyền truy nhập tới CT và dữ liệu của CT bằng hoặc kém ưu tiên hơn.



IBM PC

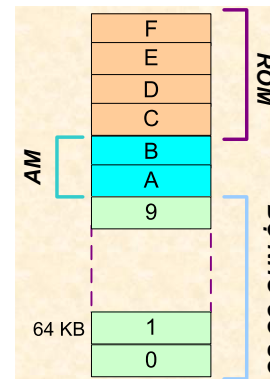
- Bộ nhớ phân phối cho CT - 2 loại: bộ nhớ chung (G) và bộ nhớ riêng (L).



141

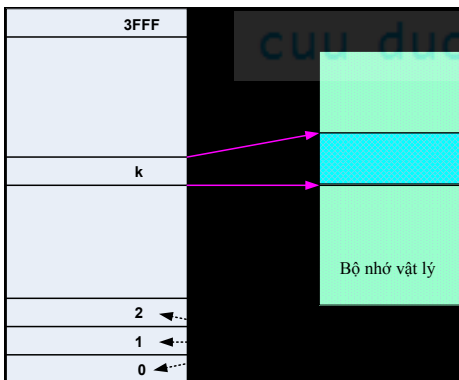
IBM PC

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



142

Chế độ Protected Mode



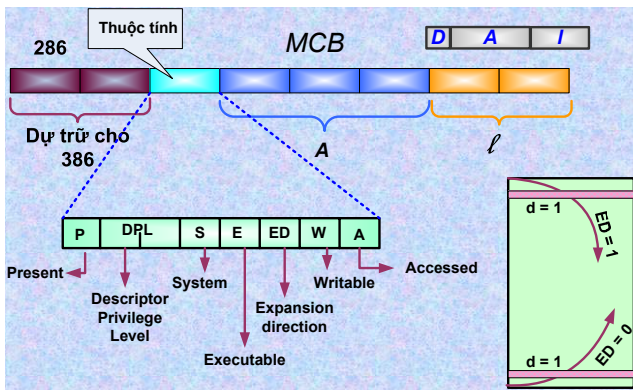
143

Chế độ Protected Mode

- Mỗi khối \Leftrightarrow MCB (Memory Control Block)
- Bộ nhớ chung $\Leftrightarrow \{MCB\} \rightarrow$ **GDT** (Global Descriptor Table).
- Bộ nhớ riêng $\Leftrightarrow \{MCB\} \rightarrow$ **LDT** (Local Descriptor Table).
- MCB: **8 Bytes/phần tử**.
- Thực hiện CT: GDT \rightarrow RAM, **GDTR**
LDT \rightarrow RAM, **LDTR**
Lệnh: **LGDTR**, **SGDTR**, **LLDTR**, **SLDTR**

144

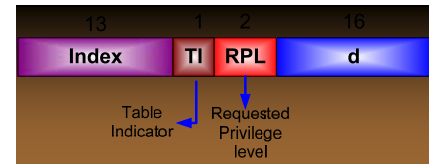
MCB



145

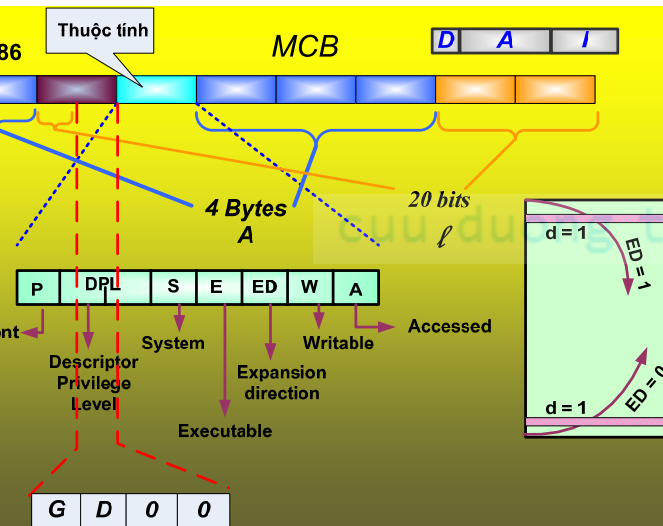
80286

- Địa chỉ tuyến tính: 32 bits.



- Khả năng:
 - Vật lý: AR – 24 bits
 - $V_{ph} = 2^{24}$
 - = 16MB
 - Lô gic: $V_{lg} = 2^{13} \times 2^1 \times 2^{16}$
 - $= 2^{30}$
 - = 1 GB

146



147

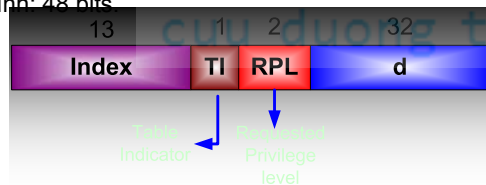
80386 - PENTUM

- G = 0 - Chế độ mô đun, đơn vị tính kích thước khối – Byte $\rightarrow L = 2^{20} = 1 \text{ MB}$.
- G = 1 - Chế độ phân trang, đơn vị tính kích thước khối – trang (4 KB) $\rightarrow L = 2^{20} P = 2^{20} \times 2^{12} = 2^{32} = 4 \text{ GB}$.
- D = 0 - Chế độ dữ liệu 16 bit,
- D = 1 - Chế độ dữ liệu 32 bit.

148

80386 - PENTUM

- Địa chỉ tuyến tính: 48 bits

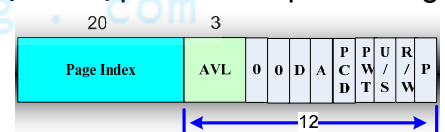


- Khả năng:
 - Vật lý: AR – 32 bits
 - $V_{ph} = 2^{32}$
 - = 4GB
 - Lô gic: $V_{lg} = 2^{13} \times 2^1 \times 2^{32}$
 - $= 2^{46}$
 - $= 2^6 \times 2^{40}$
 - = 64 TB

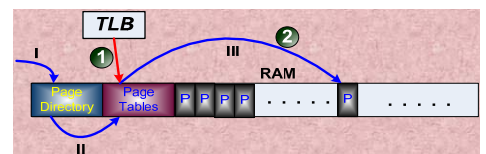
149

80386 - PENTUM

- Chế độ kết hợp mô đun – phân trang:



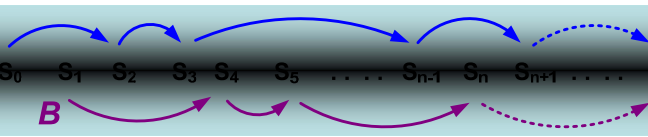
- Phân phối bộ nhớ:



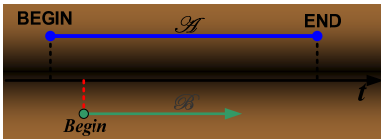
150

IV – QUẢN LÝ TIẾN TRÌNH (PROCESS)

- 1 - **Định nghĩa** tiến trình:

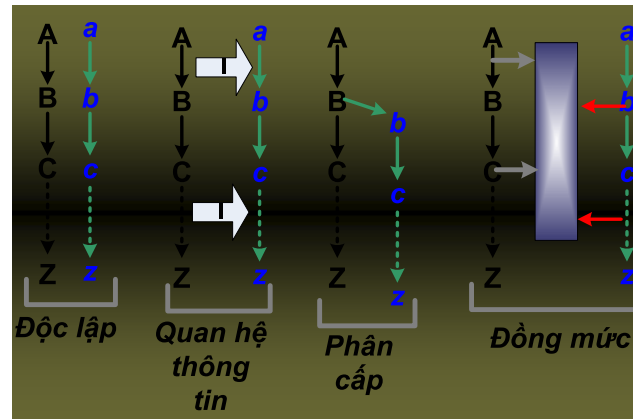


- 2 – **Phân loại**: kế tiếp và song song,
- Tiến trình song song:



151

Phân loại



Phân loại

- a) **Độc lập**: Bảo vệ thông tin,

- b) **Quan hệ thông tin**:

– Tiến trình nhận: Tồn tại? Ở đâu? Giai đoạn nào?

- Cơ chế truyền tin:

- Hòm thư,
- I/O Ports,
- Monitor/

153

Phân loại

- c) **Phân cấp**:

- Tài nguyên cho tiến trình con:

- Hệ thống QL tài nguyên tập trung: từ hệ thống,
- Hệ thống QL tài nguyên phân tán: từ vốn tài nguyên tiến trình chính,

- QL phân tán: Tiến trình chính phải kết thúc sau tiến trình con → POST, WAIT.

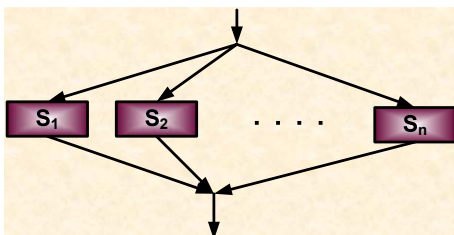
- d) **Đồng mức**:

- Sử dụng chung theo nguyên tắc lần lượt,
- Các hệ thống mô phỏng, trò chơi, . . .

155

3 - BIỂU DIỄN TIẾN TRÌNH SONG SONG

- Giả thiết: S_1, S_2, \dots, S_n – các công việc thực hiện song song (Trên 1 hoặc nhiều máy).



155

BIỂU DIỄN

- 2 cách mô tả phổ biến:

PARBEGIN

$S_1;$
 $S_2;$

.....

S_n

PAREND;

COBEGIN

$S_1;$
 $S_2;$

.....

S_n

COEND;

Các công việc S_i được mô tả chính xác bằng một ngôn ngữ lập trình cụ thể.

155

– TÀI NGUYÊN GẮNG và ĐOẠN GẮNG

Tài nguyên găng: Khả năng phục vụ đồng thời bị hạn chế, thông thường - bằng 1.

Ví dụ: Máy in, quá trình bán vé máy bay . . .

Đoạn găng (chỗ hẹp) của tiến trình,

Điều độ tiến trình qua đoạn găng: Tổ chức cho mọi tiến trình qua được chỗ hẹp của mình.

Giải thuật điều độ phải đảm bảo 4 yêu cầu.

157

Yêu cầu

- i) Đảm bảo tài nguyên găng không phải phục vụ quá khả năng của mình,
- ii) Không để tiến trình nằm vô hạn trong đoạn găng,
- iii) Nếu có xếp hàng chờ thì sớm hay muộn tiến trình cũng qua được đoạn găng,
- iv) Nếu có tiến trình chờ đợi và nếu tài nguyên găng được giải phóng, thì tài nguyên găng phải phục vụ ngay cho tiến trình đang chờ đợi.

158

Công cụ điều độ

Công cụ điều độ: 2 loại:

- **Cấp cao**: do hệ thống đảm nhiệm, nằm ngoài tiến trình được điều độ,
- **Cấp thấp**: cài đặt ngay vào trong tiến trình được điều độ.

Các giải thuật điều độ cấp thấp: **3 lớp giải thuật**:

- Phương pháp khoá trong,
- Phương pháp kiểm tra và xác lập,
- Kỹ thuật đèn báo.

159

5 – CÁC GIẢI THUẬT ĐIỀU ĐỘ CẤP THẤP

• Phương pháp khoá trong:

• Nguyên lý:

- Mỗi tiến trình (TT) đặt tương ứng tài nguyên găng với 1 biến $\in G$,
- TT dùng biến này để đánh dấu việc mình đang sử dụng tài nguyên găng,
- Trước khi vào đoạn găng TT phải kiểm tra biến tương ứng của các TT khác và chỉ vào đoạn găng khi không có TT nào đang sử dụng tài nguyên găng.

160

Phương pháp khoá trong

Môi trường ví dụ: Xét trường hợp:

- 2 tiến trình,
- Mỗi TT có một đoạn găng ở đầu,
- 1 tài nguyên găng với khả năng phục vụ: 1,
- Các tiến trình lặp vô hạn.

Tránh nhầm lẫn giữa 2 khái niệm:

- **Sơ đồ nguyên lý**: nêu ý tưởng chung,
- **Giải thuật điều độ**: sơ đồ hành động để đảm bảo điều độ.

161

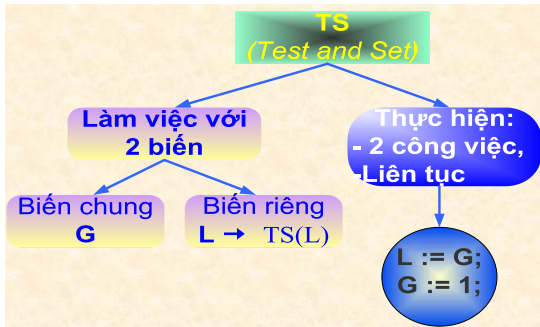
SƠ ĐỒ NGUYÊN LÝ

- **BEGIN**
- Ban đầu $k = 0$; khoá mở
- **PARBEGIN**
- tt1: while $k = 1$ do; <chờ đợi tích cực
- $k := 1$;
- <đoạn găng tt1>
- $k := 0$;
- <phần còn lại của tt1>
- tt2: while $k = 1$ do; <chờ đợi tích cực
- $k := 1$;
- <đoạn găng tt2>
- $k := 0$;
- <phần còn lại của tt2>
- **PAREND**
- **END**

162

KIỂM TRA VÀ XÁC LẬP (TEST and SET)

- Cơ sở: dùng lệnh máy TS có từ các máy tính thế hệ III trở đi.



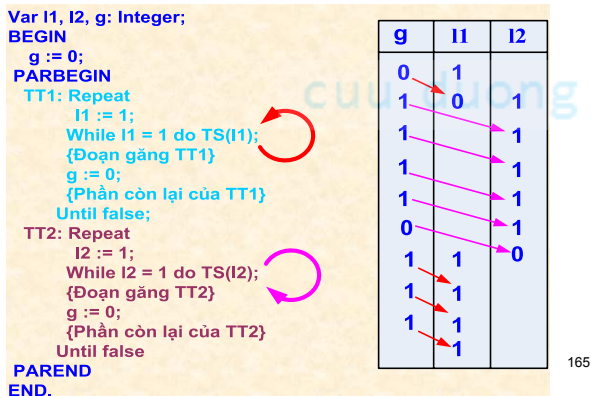
TEST and SET

- IBM 360/370: \exists 1 lệnh TS (mã 92_H),
- IBM PC: Nhóm lệnh BTS (Binary Test and Set)

L :=	G	¬G	G	¬G
G :=	1	0	1	0

TEST and SET

Sơ đồ điều độ:



TEST and SET

Đặc điểm:

- Đơn giản, độ phức tạp không tăng khi số tiến trình tăng. Nguyên nhân: Cục bộ hoá biến và tính liên tục của KT & XL,
- Tồn tại hiện tượng chờ đợi tích cực. Nguyên nhân: Mỗi TT phải tự đưa mình vào đoạn găng.

KỸ THUẬT ĐÈN BẢO (Semaphore)

Dijkstra đề xuất 1972.

Đề xuất:

- Mỗi tài nguyên găng được đặt tương ứng với một **biến nguyên đặc biệt S** (Semaphore),
- Ban đầu: $S \leftarrow$ Khả năng phục vụ t.ng. găng,
- \exists 2 lệnh máy P(S) và V(S) thay đổi giá trị của S, mỗi lệnh làm 2 công việc và làm một cách liên tục.

KỸ THUẬT ĐÈN BẢO

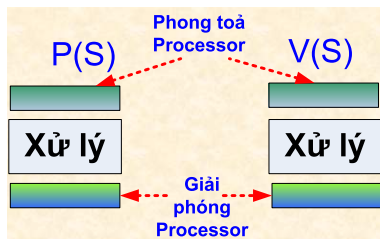
- Nội dung lệnh P(S):
 - * Dec(s);
 - ** If $S < 0$ then *Đưa TT đi xếp hàng.*
- Nội dung lệnh V(S):
 - * Inc(s);
 - ** If $S \leq 0$ then *Kích hoạt TT đang xếp hàng.*

KỸ THUẬT ĐÈN BÁO

Thực hiện:

- Vì nhiều lý do, không thể chế tạo MT với 2 lệnh trên,
- Lệnh P(S), V(S) → *thủ tục* tương ứng.

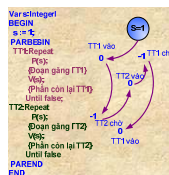
Đảm bảo tính liên tục:



169

KỸ THUẬT ĐÈN BÁO

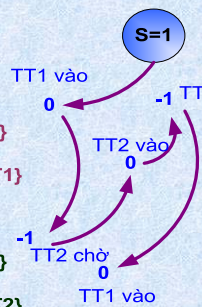
- Sơ đồ điều độ:



```

Var s: Integer
BEGIN
  s := 1;
  PARBEGIN
    TT1: Repeat
      P(s);
      {Đoạn găng TT1}
      V(s);
      {Phần còn lại TT1}
    Until false;
    TT2: Repeat
      P(s);
      {Đoạn găng TT2}
      V(s);
      {Phần còn lại TT2}
    Until false;
  PAREND
END.

```



KỸ THUẬT ĐÈN BÁO

Semaphore nhị phân:

Phần lớn các tài nguyên găng có khả năng phục vụ = 1 → S nhị phân.

P(S):

If s = 0 **then** Xếp_hàng **Else** s := 0;

V(S):

If dòng_xếp_hàng ≠ NULL **then** Kích_hoạt

Else s := 1;

án đề đặt tên các thủ tục P và V.

171

6 – CÔNG CỤ ĐIỀU ĐỘ CẤP CAO

- Đoạn găng quy ước,
- Biến điều kiện quy ước,
- Monitor hỗ trợ điều độ: cung cấp giá trị cho biến điều kiện quy ước.
- Monitor đóng vai trò vỏ bọc bảo vệ ngăn cách giữa tài nguyên găng và công cụ truy cập tới nó.

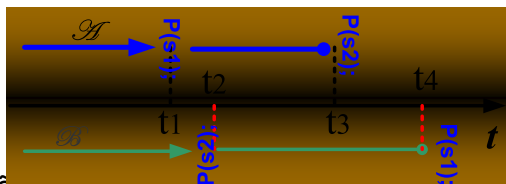
172

7 - BẾ TẮC và CHỐNG BẾ TẮC

Khái niệm bế tắc (Deadlock):

Cùng chờ đợi,

Vô hạn nếu không có tác động từ bên ngoài.



Sẽ không có bế tắc nếu TT 2 bắt đầu đủ sớm hay đủ muộn.

173

BẾ TẮC và CHỐNG BẾ TẮC

- Điều kiện xuất hiện bế tắc: hội đủ *đồng thời* 4 điều kiện:
 - ∃ tài nguyên găng,
 - Có tổ chức xếp hàng chờ đợi,
 - Không phân phối lại tài nguyên,
 - ∃ hiện tượng chờ đợi vòng tròn.
- Chống bế tắc: *3 lớp giải thuật*:
 - Phòng ngừa,
 - Dự báo và tránh,
 - Nhận biết và khắc phục.

174

Phòng ngừa

Điều kiện áp dụng:

- Xác suất xuất hiện bế tắc lớn,
- Các biện pháp Tồn thất lớn.

Biện pháp: tác động lên một hoặc một số điều kiện gây bế tắc để 4 điều kiện không xuất hiện đồng thời.

Các giải pháp: được áp dụng để *nâng cao hiệu quả* của hệ thống.

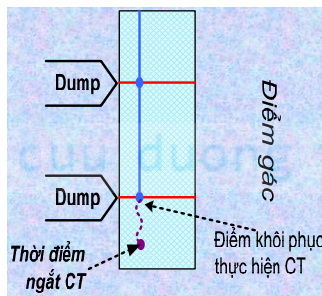
Phòng ngừa

Đặt điểm gác:

- Cố định trong CT,
- Theo tác nhân ngoài (vd: thời gian)

Ứng dụng:

- Hiệu chỉnh CT,
- Thực hiện các CT dài,
- Với toàn bộ hệ thống: Hibernating.



177

Phòng ngừa

- Chống tài nguyên găng:
 - Tổ chức hệ thống tài nguyên lô gíc,
 - 2 mức truy nhập,
 - SPOOL.
- Chống xếp hàng chờ đợi:
 - Chế độ phân phối sơ bộ,
 - Trước khi ngắt TT: lưu trạng thái (Dump),
 - Công cụ:
 - Điểm gác (Control Points),
 - Điểm ngắt (Break Points)

178

Phòng ngừa

- Phân phối lại tài nguyên:
 - Các tài nguyên quan trọng (Bộ nhớ, Processor . . .) luôn luôn được phân phối lại,
 - Chủ yếu: chỉ cần lưu ý các tài nguyên riêng,
 - Hệ thống tài nguyên lô gíc: giảm nhu cầu phân phối lại.
 - Để phân phối lại: Lưu và khôi phục trạng thái tài nguyên.

179

Phòng ngừa

Chống chờ đợi vòng tròn:

- Phân lớp tài nguyên, tạo thành hệ thống phân cấp,
- Nguyên tắc phân phối: Khi chuyển lớp - phải giải phóng tài nguyên lớp cũ.

179

DỰ BÁO VÀ TRÁNH

- Mỗi lần phân phối một tài nguyên: kiểm tra xem việc phân phối này có thể dẫn đến *nguy cơ* bế tắc cho một số tiến trình nào đó hay không và là những tiến trình nào?
- Điều kiện môi trường:
 - Xác suất xảy ra bế tắc nhỏ,
 - Tồn thất (nếu có bế tắc) – lớn.

180

DỰ BÁO VÀ TRÁNH

- Giải thuật tiêu biểu: “*Người chủ ngân hàng*”.
- Giả thiết:
 - Xét 1 loại tài nguyên, số lượng \rightarrow $tstb$,
 - n tiến trình,
 - Max_i ,
 - $Ffoi_i$,
 - Kt_i – boolean,
 - True – chắc chắn kết thúc được,
 - False – trong trường hợp ngược lại.

181

DỰ BÁO VÀ TRÁNH

```
ts := tstb;
{Thống kê}
For i := 1 to n do
begin
  clai[i] := max[i] - ffoi[i];
  ts := ts - ffoi[i];
  kt[i] := false
end;

{Đánh giá}
Flag := true;
While flag do
begin
  flag := false;
  for i := 1 to n do
    if not kt[i] and (clai[i] <= ts) then
    begin
      kt[i] := true;
      ts := ts + ffoi[i];
      flag := true
    end
  end;
  If ts <> tstb then Kh_An_Toan;
```

182

DỰ BÁO VÀ TRÁNH

- Tiêu chuẩn dự báo: ngặt,
- Dựa vào $Kt_i \rightarrow$ biết các TT có nguy cơ bế tắc,
- Kử lý trước khi TT bị bế tắc.
- Đặc điểm giải thuật:
- Đơn giản,
 - Input: Max_i – tin cậy,
 - Mỗi loại tài nguyên \Leftrightarrow thủ tục,
 - Mỗi lần phân phối \rightarrow kiểm tra.

183

NHẬN BIẾT VÀ KHẮC PHỤC

- Định kỳ kiểm tra các TT chờ đợi để phát hiện bế tắc,
- Điều kiện áp dụng:
 - Xác xuất xảy ra bế tắc bé,
 - Tồn thất (nếu có bế tắc) – bé.
- Áp dụng với phần lớn OS trong thực tế,
- Do OP đảm nhiệm.

184

NHẬN BIẾT VÀ KHẮC PHỤC

- Lệnh OP \rightarrow các nhóm lệnh phục vụ nhận biết và khắc phục,
- Nhóm lệnh xem trạng thái (Display Status),
- Nhóm lệnh tác động lên dòng xếp hàng TT,
- Nhóm lệnh tác động lên TT,
- Quan trọng: các lệnh huỷ tiến trình,
- Các biện pháp hỗ trợ và ngăn chặn tự động

185

8 - GỌI TIẾN TRÌNH

- TT có thể cạnh tranh hoặc tương tác với nhau,
- Mối quan hệ tương tác: tuần tự hoặc song song,
- Xác lập quan hệ:
 - Lời gọi,
 - Cơ chế xử lý sự kiện (Sẽ xét ở chương sau),
- Các cách gọi:
 - Trong phạm vi một hệ thống,
 - Giữa các hệ thống:
 - RI (Remote Invocation),
 - RPC (Remote Procedure Call),
 - Lý thuyết chung: RMI (Remote Methods Invocation)

186

GỌI TIẾN TRÌNH

Sơ đồ gọi:

- Không đối xứng,
- Đối xứng.

Kỹ thuật truyền tham số:

- Theo giá trị,
- Theo địa chỉ,
- CR (Call by Copy/Restore).

187

GỌI TIẾN TRÌNH

- Thông tin tối thiểu để lưu và khôi phục TT:
 - Nội dung các thanh ghi,
 - Địa chỉ lệnh,
 - Vùng bộ nhớ RAM liên quan,
 - Vùng bộ nhớ phục vụ của hệ thống,
 - Các sự kiện chưa xử lý.
- Phân biệt sơ đồ gọi đối xứng và đệ quy.

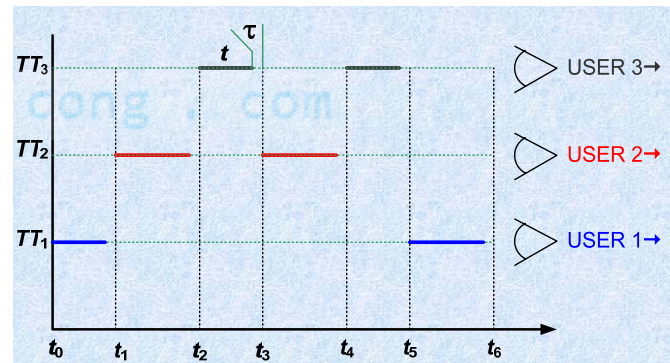
188

V – QUẢN LÝ PROCESSOR

- Mục đích: Giảm thời gian chết của Processor → nâng cao hiệu quả hệ thống,
- Vai trò thiết bị trung tâm: liên kết các bộ phận đọc lập (cứng và mềm) thành hệ thống hoạt động đồng bộ.
- Trong phần này: xét hoạt động của 1 CPU.

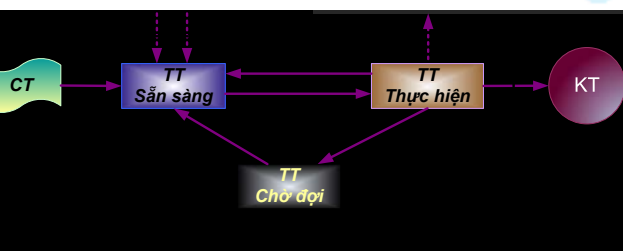
189

1 – PROCESSOR LỘ GÍC



190

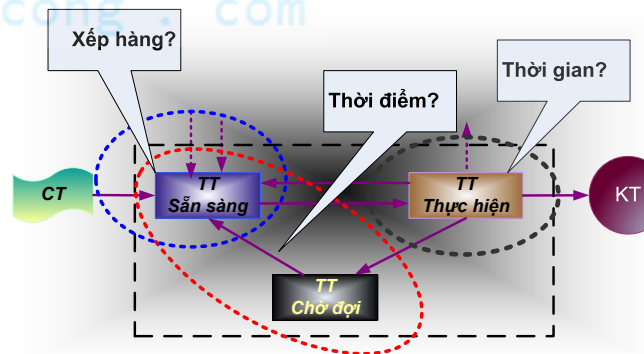
2 – CÁC TRẠNG THÁI CƠ BẢN CỦA TIẾN TRÌNH



Đặc trưng các loại trạng thái,
Vấn đề cần giải quyết: 3 loại.

191

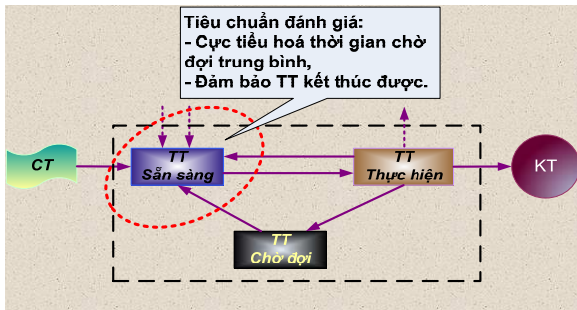
VẤN ĐỀ



192

VẤN ĐỀ

- a) Liên quan tới dòng TT sẵn sàng: Cách tổ chức phục vụ dòng xếp hàng?



193

VẤN ĐỀ

- Trình tự phục vụ tác động lên thời gian chờ đợi trung bình t_w : giả thiết – 3 TT :

TT	P ₁	P ₂	P ₃
Y/c thời gian	60'	15'	1'

$$P_1 = 60', P_2 = 60', P_3 = 60'$$

$$1' = 1', 1' = 1', 1' = 1'$$

$$t_w = \frac{60 + 60 + 15}{3} = 45'$$

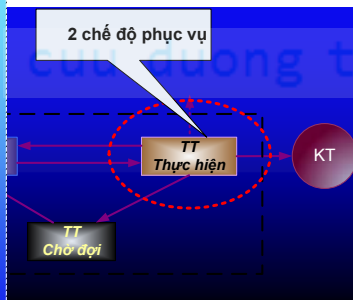
$$t_w = \frac{1 + 1 + 1}{3} = 5.67'$$

194

VẤN ĐỀ

Thời gian thực hiện lên trình:

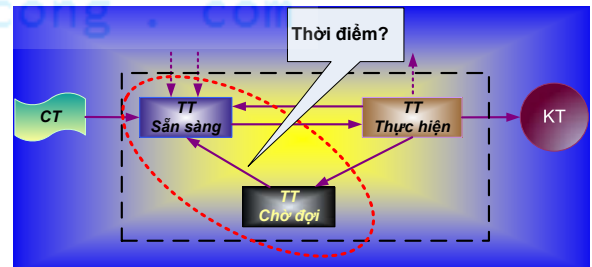
- Không đẩy ra (Non-preemptive), (Xử lý theo lô)
 - Có đẩy ra (Preemptive) (Phân chia thời gian)
- Lượng tử thời gian: $0.03'' \div 0.2''$.



195

VẤN ĐỀ

- c) Thời điểm đưa TT chờ đợi trở lại sẵn sàng? Cơ chế *sự kiện* và *ngắt*.



196

Chế độ một dòng xếp hàng

- TT \Leftrightarrow thứ tự ưu tiên phục vụ,
- Yêu cầu:
 - $t_w \rightarrow \min$.
 - TT kết thúc.
- Chế độ:
 - Một dòng xếp hàng,
 - Nhiều dòng xếp hàng.

- a) FCFS (First come – First served):
 - Đơn giản,
 - \forall TT kết thúc được,
 - Không cần input bổ sung,
 - T_w – lớn,
 - Non-Preemptive.

197

198

Chế độ một dòng xếp hàng

b) SJN (Shortest Job – Next):

- Thời gian thực hiện ít → ưu tiên cao,
- T_w giảm,
- TT dài có nguy cơ không kết thúc được,
- Khó dự báo thời điểm phục vụ TT,
- Non-Preemptive,
- Input: Thời gian thực hiện TT.

199

Chế độ một dòng xếp hàng

c) SRT (Shortest Remaining Time):

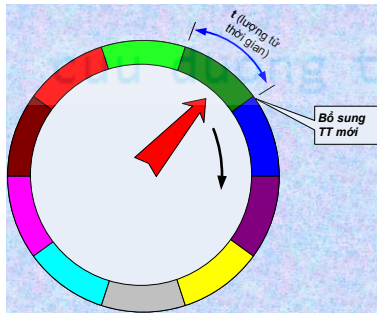
- Thứ tự ưu tiên phục vụ: xác định theo lượng thời gian còn lại cần thiết để kết thúc TT,
- t_w giảm mạnh,
- Các đặc trưng khác: tương tự như SJN,
- TT dài càng có nguy cơ không kết thúc được!
- Ở các chế độ Non-Preemptive: cần có t_{lim} → hủy TT hoặc đưa về thứ tự ưu tiên thấp nhất.

200

Chế độ một dòng xếp hàng

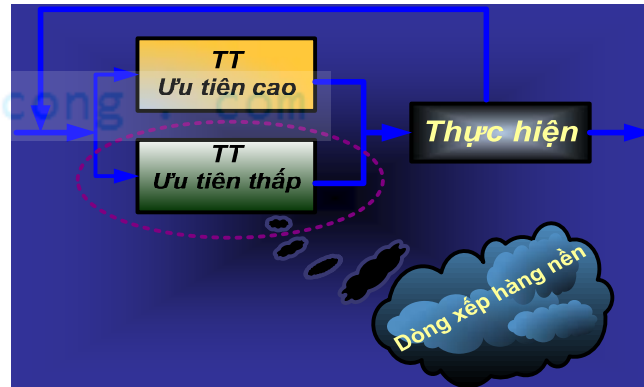
d) RR (Round Robin):

- Preemptive,
- \forall TT - kết thúc được,
- Khả năng đối thoại với TT,
- Ưu tiên thích đáng với TT dài: phân lớp phục vụ với t lớn hơn.



201

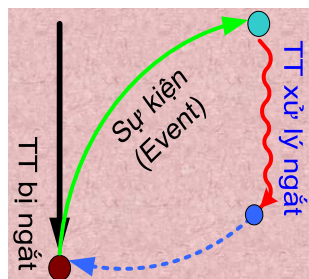
Chế độ nhiều dòng xếp hàng



4 - NGẮT và XỬ LÝ NGẮT

Định nghĩa ngắt (Interrupt):

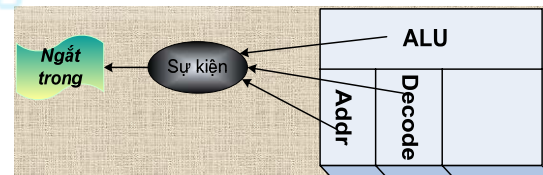
- Cơ chế *Sự kiện* và *Ngắt*: từ MT thế hệ III,
- IBM 360/370 – 7 loại sự kiện,
- IBM PC – 256 loại sự kiện.



203

PHÂN LOẠI NGẮT

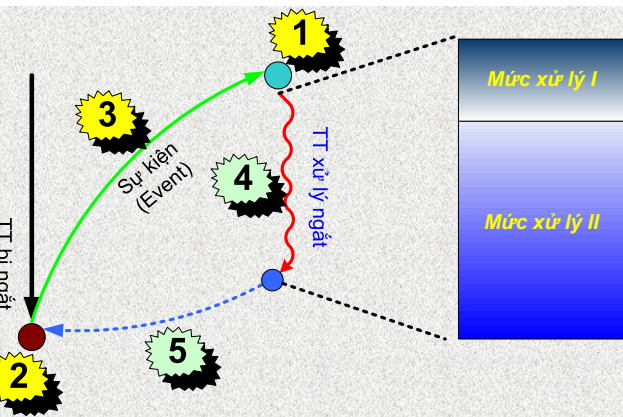
- Ngắt trong và ngắt ngoài,



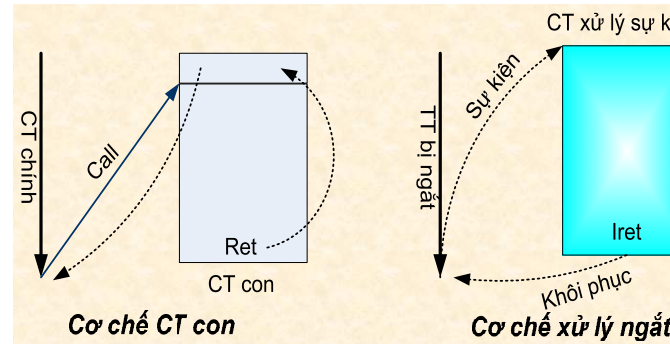
- Ngắt trong: /0, tràn ô, ...
- Ngắt ngoài: I/O Int, Timer, ...
- Ngắt chặn được và không chặn được:
 - Chặn được: i/o Int,
 - Không chặn được: Timer Int.
- Ngắt cứng và ngắt mềm.

204

XỬ LÝ NGẮT



CT con và CT xử lý ngắt



5 - Xử lý ngắt trong IBM PC

Ngắt \Leftrightarrow Pointer (4 bytes),
 Véc tơ ngắt = {Pointers} (1 KB),
 Khối bộ nhớ xử lý ngắt,
 Nét đặc biệt:

- \exists các ngắt | Pointer \rightarrow Bảng tham số (Int 11, 1E, 41, ...),
- Ngắt KT CT – Int 20, Ngắt thường trú CT Int 27,
- Ngắt R/W đĩa theo địa chỉ tuyệt đối – Int 25, 26,
- \exists ngắt tương ứng với việc bấm phím (Int 05, 1B),
- Ngắt OS mô phỏng xử lý các sự kiện (Int 21),
- Một số sự kiện: dành cho user tạo ngắt mềm \rightarrow Lập trình hướng sự kiện (EOP).

VI - CẤU HÌNH VÀ QUẢN LÝ HỆ THỐNG

- 1 - Hệ thống nhiều Processors.
- Các loại cấu hình:
 - Cấu hình phân cấp,
 - Liên kết linh hoạt,
 - Đăng cấu,
- Quản lý tiến trình:
 - S – tài nguyên găng,
 - TS \rightarrow S \rightarrow điều độ,
- Đảm bảo toàn vẹn chức năng và toàn vẹn cấu hình

CẤU HÌNH VÀ QUẢN LÝ HỆ THỐNG

2 - Bảo vệ hệ thống:

Nguyên cơ:

- Mất hoặc hỏng dữ liệu,
- Sử dụng tài nguyên với mục đích xấu,
- Truy nhập không đăng ký,
- Dò rỉ thông tin.

Cơ chế bảo vệ:

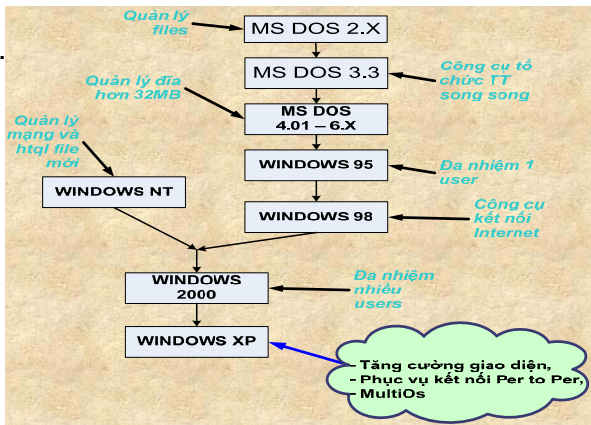
- Nguyên lý ngăn chặn,
- Nguyên lý cho phép.

Giải thuật và biện pháp bảo vệ: linh hoạt, thường xuyên thay đổi.

CẤU HÌNH VÀ QUẢN LÝ HỆ THỐNG

- 3 – Thiết kế và xây dựng hệ thống:
- Nguyên lý tập trung: WINDOWS, UNIX, OS IBM, ...
- Nguyên lý “Thử và sai”: LINUX:
 - Không có đề xuất hướng chung,
 - Mã nguồn mở cho phép mọi người nghiên cứu, bổ sung sửa đổi,
 - Phát triển theo nguyên lý tự điều chỉnh,
 - Giao diện: User tự trang bị.

4 - Hệ thống của Microsoft



cuu duong than cong . com

cuu duong than cong . com