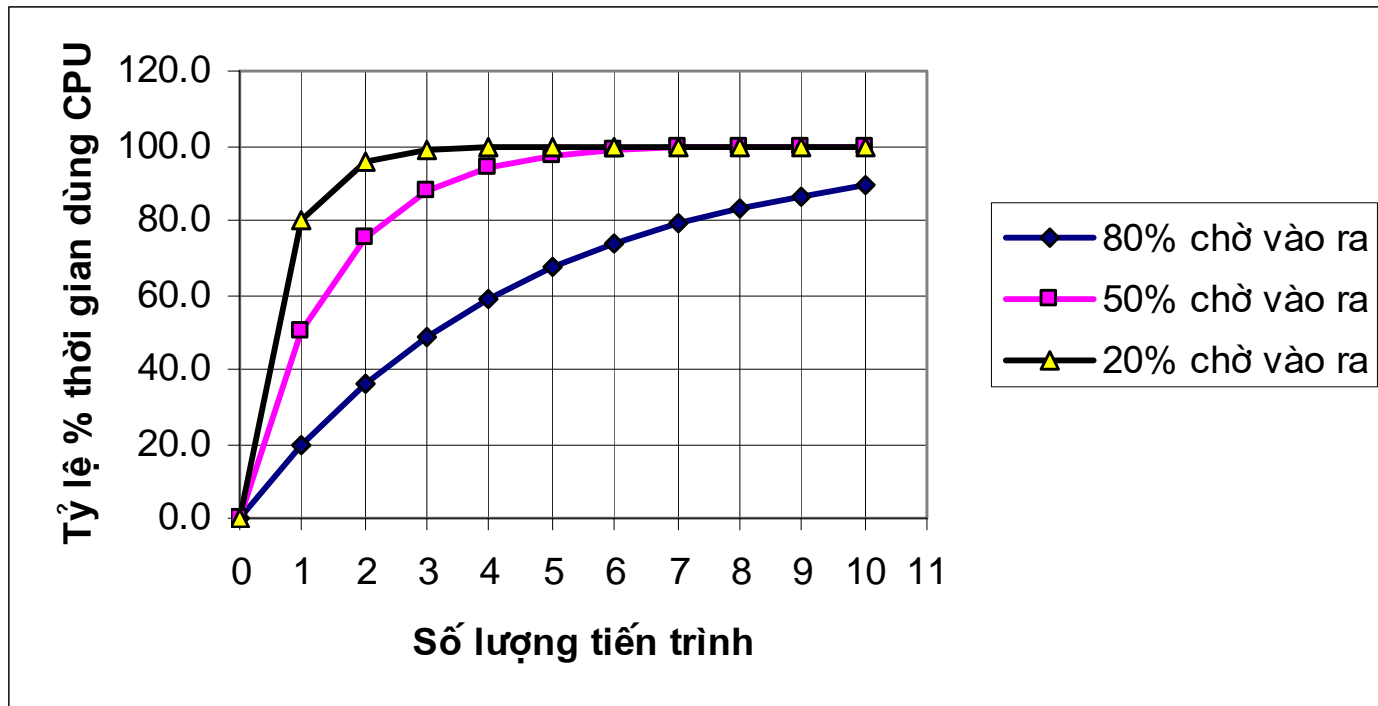


Quản lý bộ nhớ

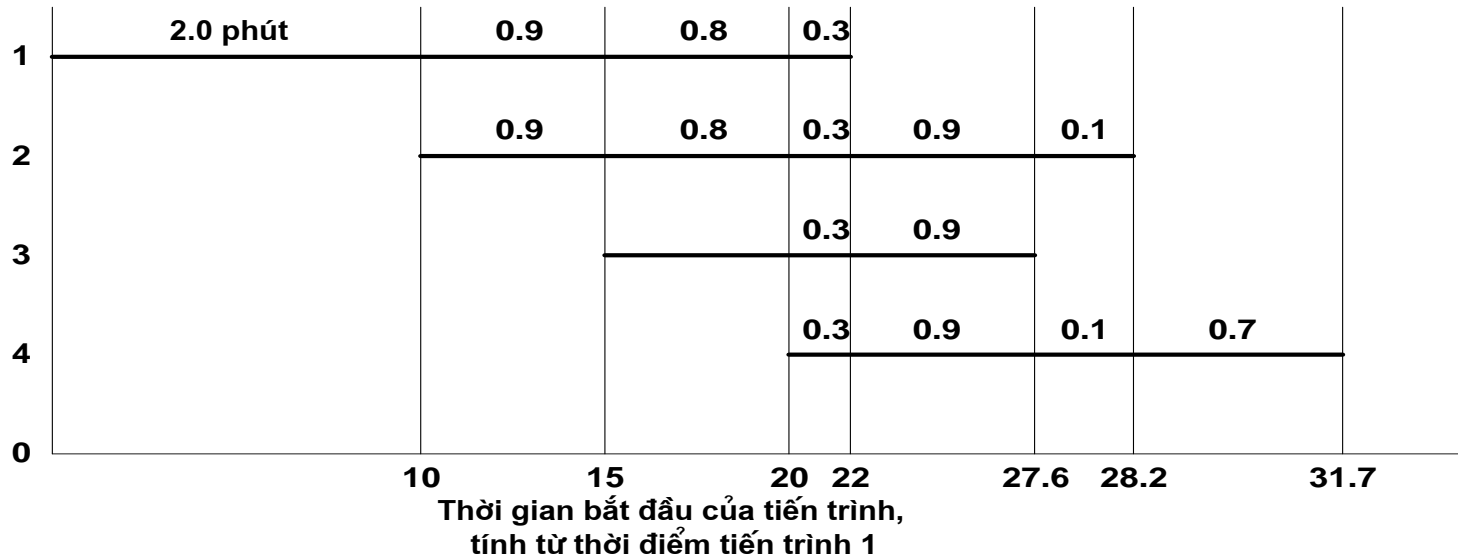
QLBN trong mô hình đa nhiệm



- Tỷ lệ % sử dụng CPU = $1 - p^n$
- n là số tiến trình, p là tỷ lệ % thời gian chờ vào/ra của tiến trình

Đối với mô hình xử lý lô

Tiến trình

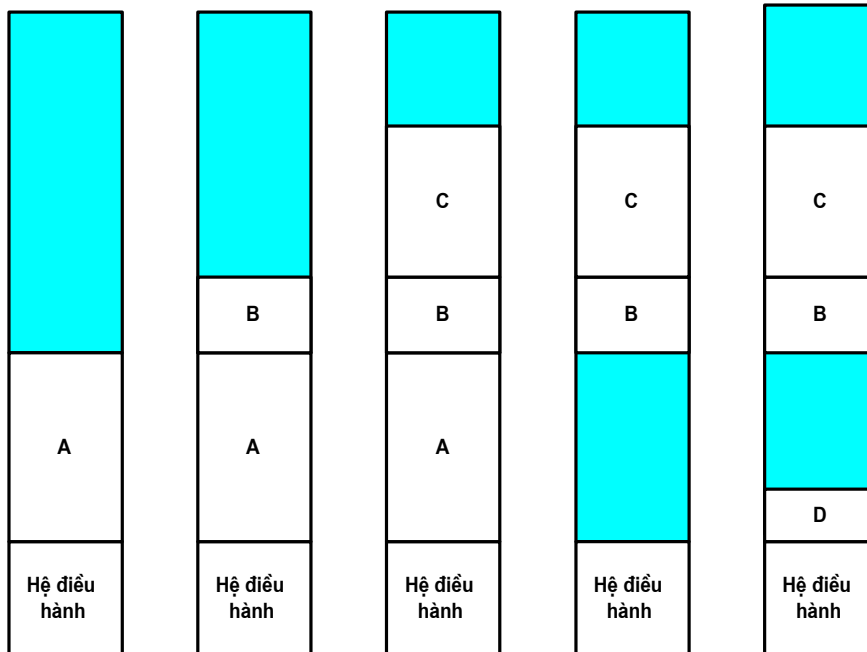


Số lượng tiến trình

Tiến trình	Thời gian bắt đầu	Số phút sử dụng CPU
1	10:00	4
2	10:10	3
3	10:15	2
4	10:20	2

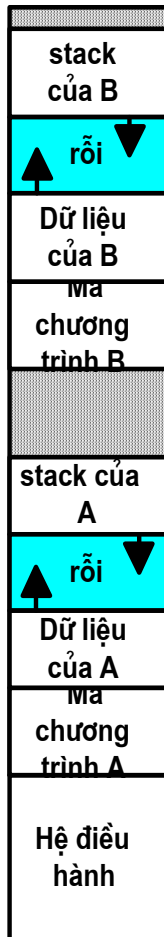
	1	2	3	4
CPU rỗi	.80	.64	.51	.41
CPU bận	.20	.36	.49	.59
CPU/Process	.20	.18	.16	.15

Phân bổ bộ nhớ



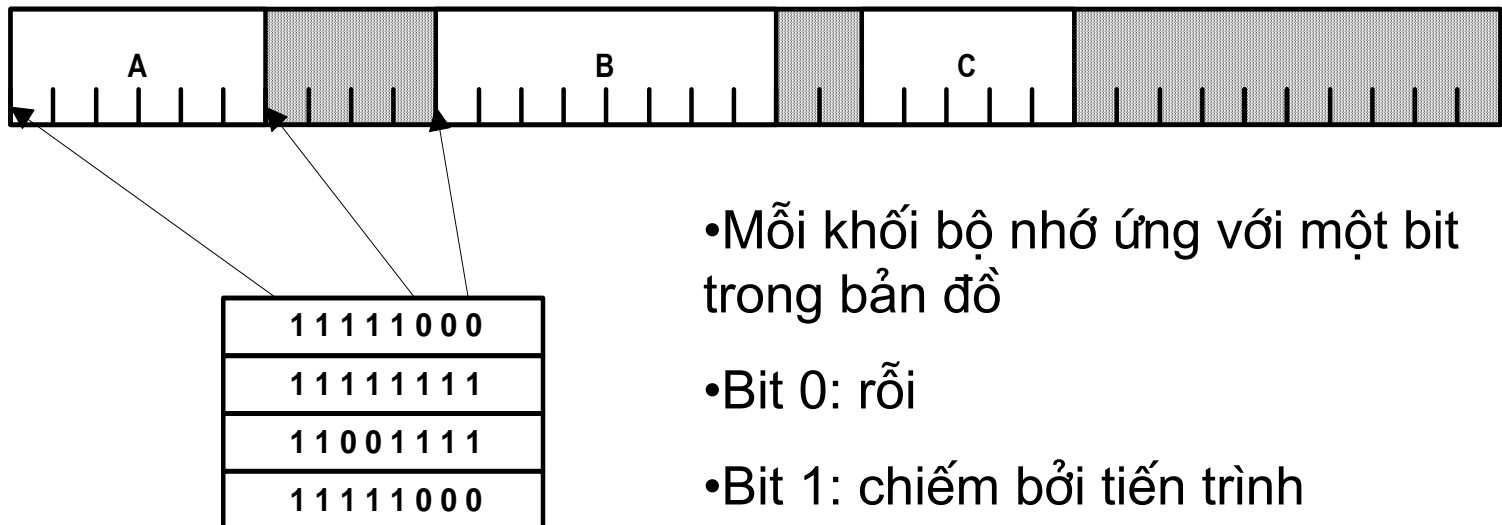
- Tiến trình A được nạp
- Tiến trình B được nạp
- Tiến trình C được nạp
- Tiến trình A kết thúc
- Tiến trình D được nạp

Không gian phụ

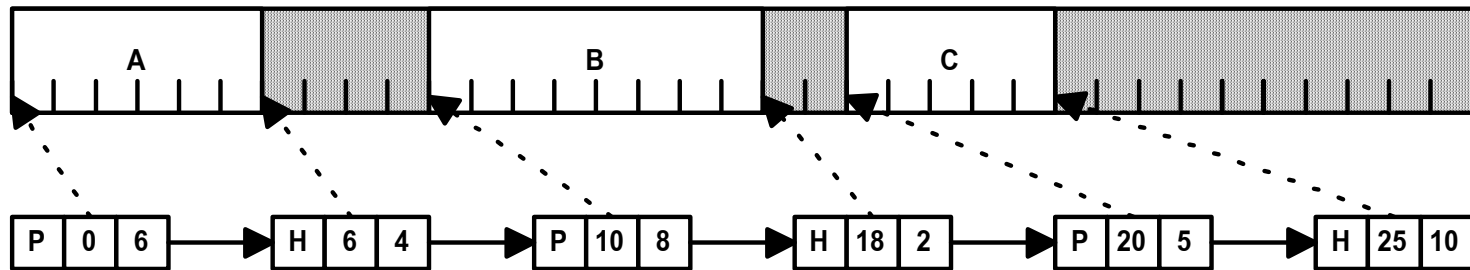


- Tiến trình A và B đang trong bộ nhớ
- Mỗi tiến trình cần có một khoảng trống để dùng thêm khi chạy
- Thông thường: ngăn xếp phát triển xuống, vùng dữ liệu xin cấp động phát triển lên
- Khi xảy ra đụng chạm giữa 2 vùng sẽ gây ra bẫy lỗi thiếu bộ nhớ

Quản lý bằng bản đồ bit



Quản lý bằng DSMN



- P: Process, H: Hole
- Mỗi phần tử của danh sách chứa nơi bắt đầu một vùng và số khối rỗi hoặc đang bận thuộc vùng đó

Th.toán QLBN với DSMN

- First fit: chọn khối rỗi phù hợp đầu tiên
- Next fit: chọn khối rỗi phù hợp đầu tiên tiếp theo
- Best fit: chọn khối rỗi vừa nhất
- Worst fit: chọn khối rỗi thừa thãi nhất
- Quick fit: Lưu 2 danh sách: Holes và Processes

Quản lý bằng phân khối

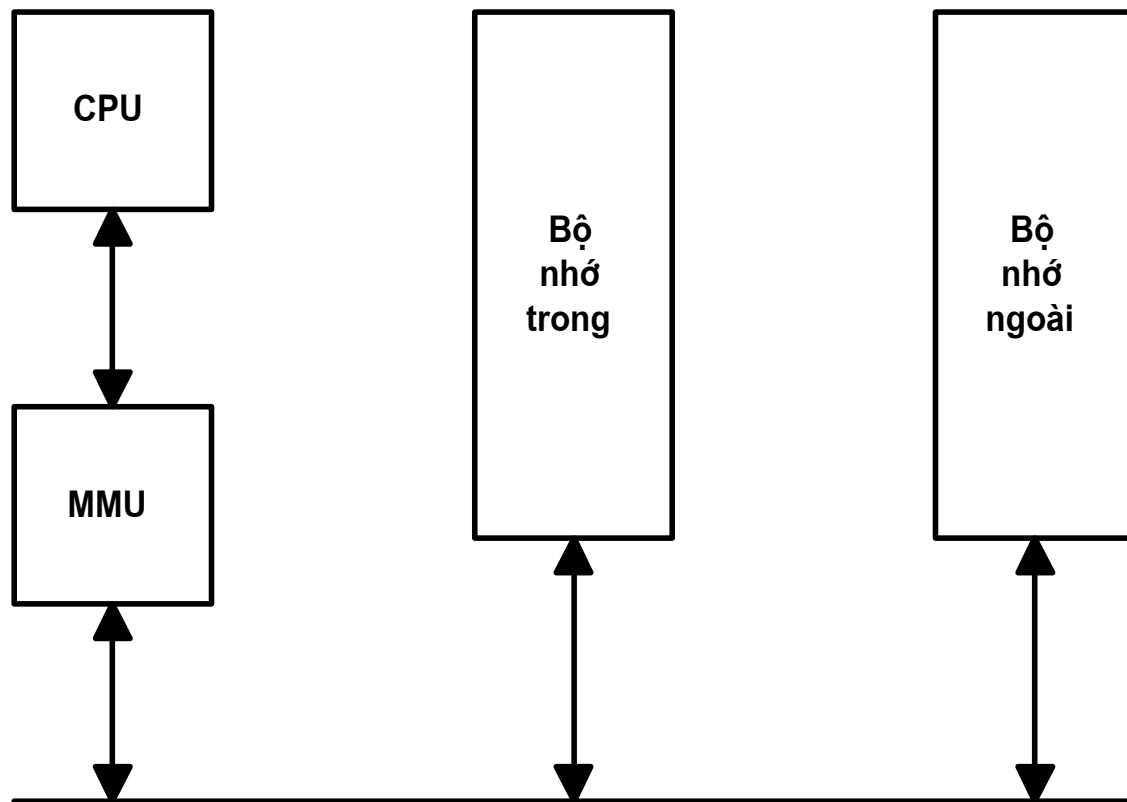
A	128		256		512
A	B	64	256		512
A	B	64	C	128	512
128	B	64	C	128	512
128	B	D	C	128	512
128	64	D	C	128	512
256			C	128	512
1024					

- Bộ nhớ gồm các khối gốc là 1024 bytes
- Kích thước yêu cầu cho tiến trình được làm tròn lên theo mũ số 2 gần nhất và khối rồi có kích thước đó được cấp cho tiến trình
- Một khối rồi lớn có thể được phân thành các khối nhỏ (chẵn mũ 2) để phân cho tiến trình. Khối được phân nhỏ gọi là buddy.

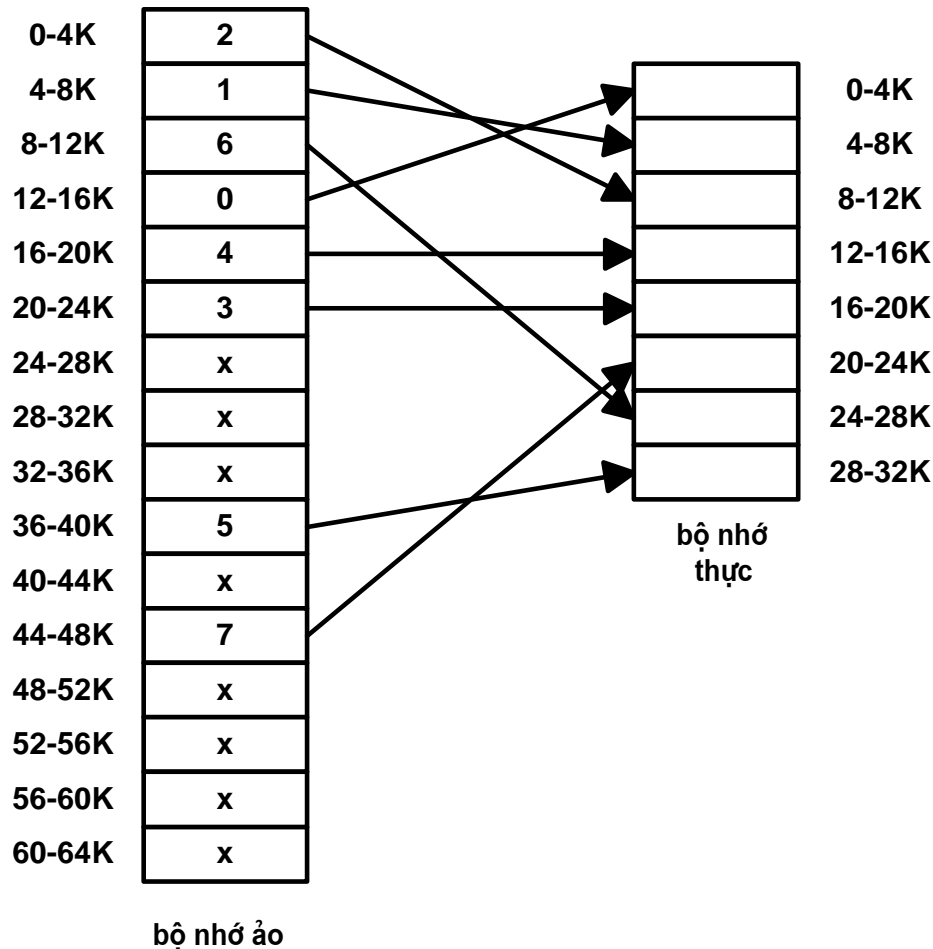
Phân tích hiệu suất hoán đổi²

- Luật 50%: tiến trình nằm giữa bộ nhớ sẽ có khoảng 50% số lần nằm cạnh tiến trình khác và 50% số lần nằm cạnh khối rỗi.
- Với n tiến trình, có khoảng $n/2$ khối rỗi
- Gọi f là tỷ lệ bộ nhớ là khối rỗi; s là kích thước trung bình của n tiến trình, ks là kích thước khối rỗi trung bình ($k > 0$); m là số bytes bộ nhớ
- $(n/2)ks = m - ns$, suy ra $m = ns(1+k/2)$, và:
- $f = (nks/2)/m = k/(k+2)$
- Ví dụ: $k=1/2$, $f=20\%$

Memory Management Unit



Bộ nhớ ảo



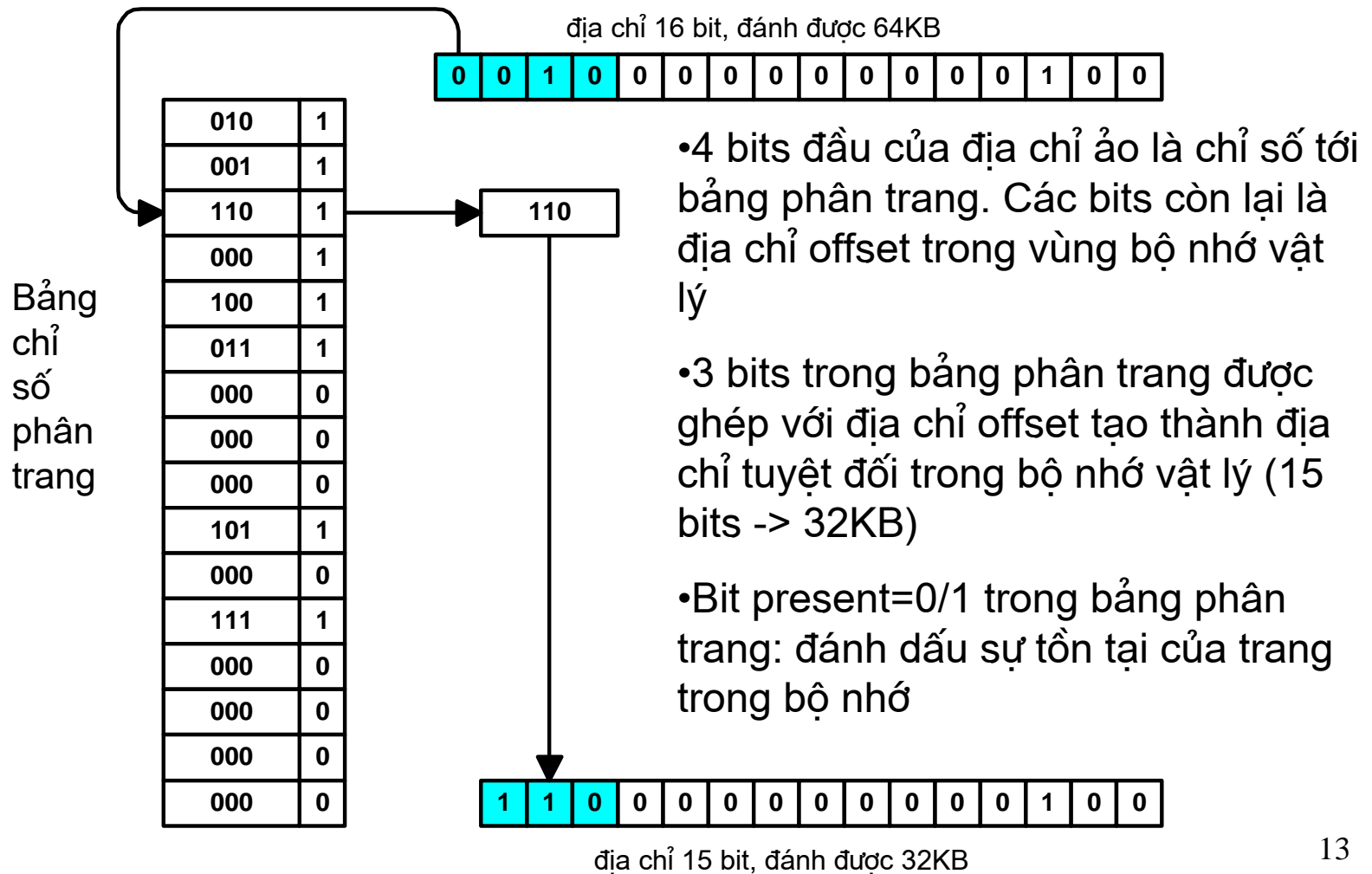
- Kích thước bộ nhớ vật lý: 32KB

- Kích thước bộ nhớ ảo: 64KB

- Độ lớn của mỗi trang bộ nhớ: 4KB

- Các trang 'x' là các trang hiện không được ánh xạ vào bộ nhớ vật lý

Ánh xạ địa chỉ ảo và thực



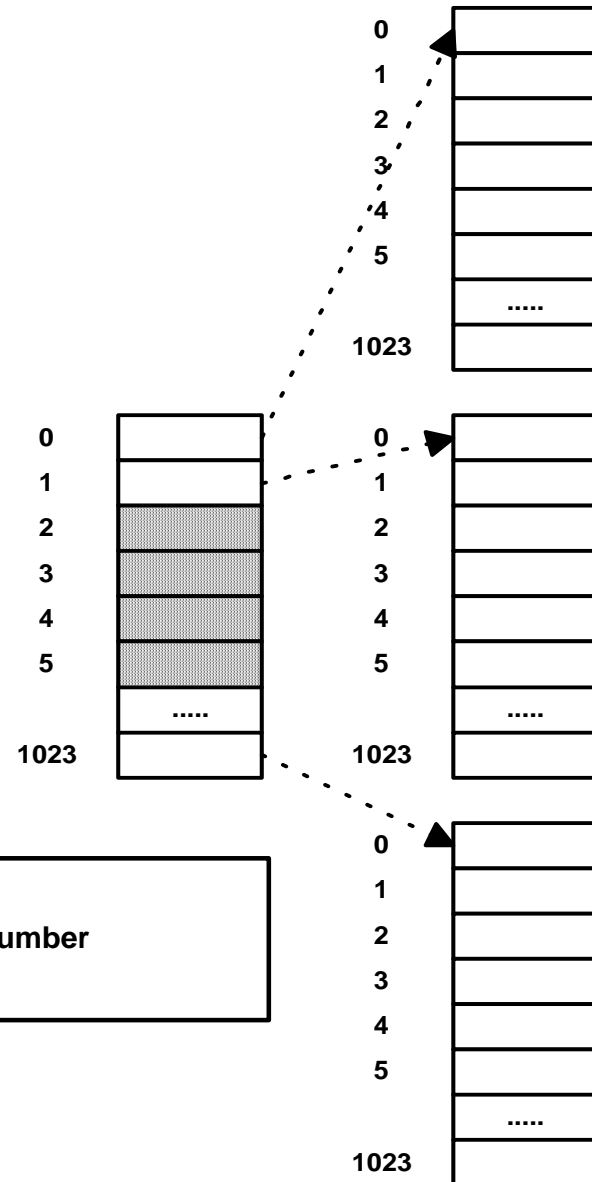
Ánh xạ 2 mức

- Yêu cầu phân trang: **nhANH và nhỏ**
- Địa chỉ ảo gồm 3 phần: 2 chỉ số, 1 offset
- Chỉ cần lưu 4 bảng trong bộ nhớ thay vì tất cả 1024 (3 bảng con ứng với: code, data, stack)

pointer1	pointer1	offset
----------	----------	--------

	caching disable	referenced	Modified	protection (3 bit)	present / absent	page frame number
--	--------------------	------------	----------	-----------------------	---------------------	-------------------

Một dòng chỉ số tới trang bộ nhớ thực

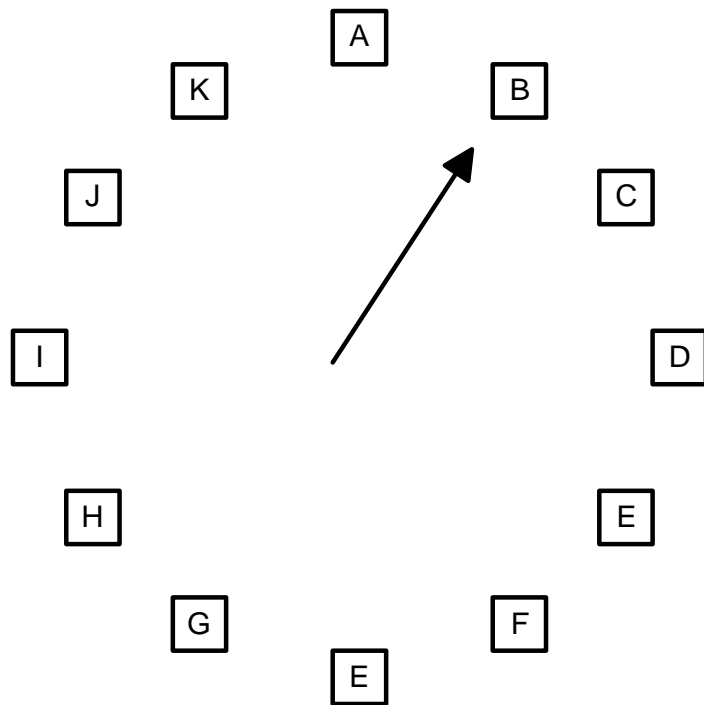


Associative memory

valid entry	virtual page	Modified	Protectio n	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Thao tác so sánh trên associative memory được thực hiện song song bởi phần cứng trên tất cả các dòng

Page replacement



- Dựa vào phương pháp Not Recently Used (NRU)
- Tổ chức các khối bộ nhớ dùng bởi các tiến trình theo danh sách móc nối vòng
- Xóa bớt 'Referenced' và quay 1 vòng, nếu chưa được đặt lại bằng 1 thì khối đó là NRU, có thể swap ra
- Có thể phân loại theo lớp
 - Not referenced, not modified
 - Not referenced, modified
 - Referenced, not modified
 - Referenced, modified

LRU nhờ phần cứng

0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

0	0	1	1
1	0	1	1
0	0	0	0
0	0	0	0

0	0	0	1
1	0	0	1
1	1	0	1
0	0	0	0

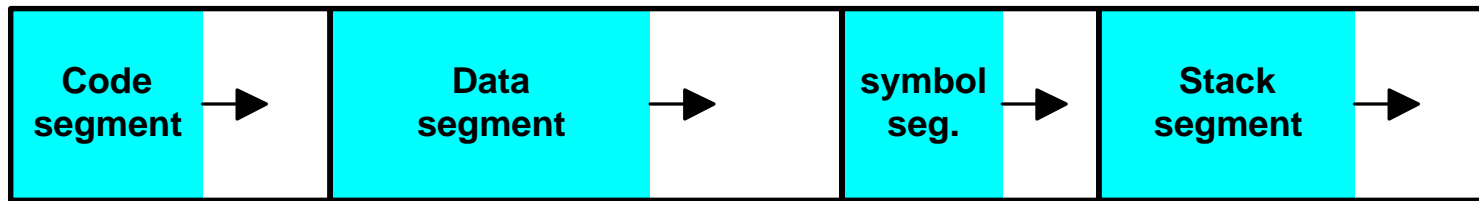
- LRU: Least Recently Used. Hỗ trợ bởi phần cứng
- Kích thước ma trận vuông $n \times n$ với n là số khối trong bộ nhớ vật lý
- Khi khối thứ k được tham khảo:
 - Hàng k được đặt bằng các bit 1
 - Cột k được xóa bởi các bit 0

NFU nhờ phần mềm

	1 0 1 0 1 1	1 0 1 0 1 1	1 0 1 0 1 1	1 0 1 0 1 1	0 1 1 0 0 0
page 0	10000000	11000000	11100000	11110000	01111000
page 1	00000000	10000000	11000000	01100000	10110000
page 2	10000000	01000000	00100000	00010000	10001000
page 3	00000000	00000000	10000000	01000000	00100000
page 4	10000000	11000000	01100000	10110000	01011000
page 5	10000000	01000000	10100000	01010000	00101000
	(a)	(b)	(c)	(d)	(e)

NFU: Not Frequently Used. Mỗi thanh ghi ứng với một khối bộ nhớ. Mỗi nhịp đồng hồ, dịch phải một đơn vị. Điền bit 1 vào bên trái nếu khối được dùng giữa 2 nhịp vừa qua.

Phân chia theo segment

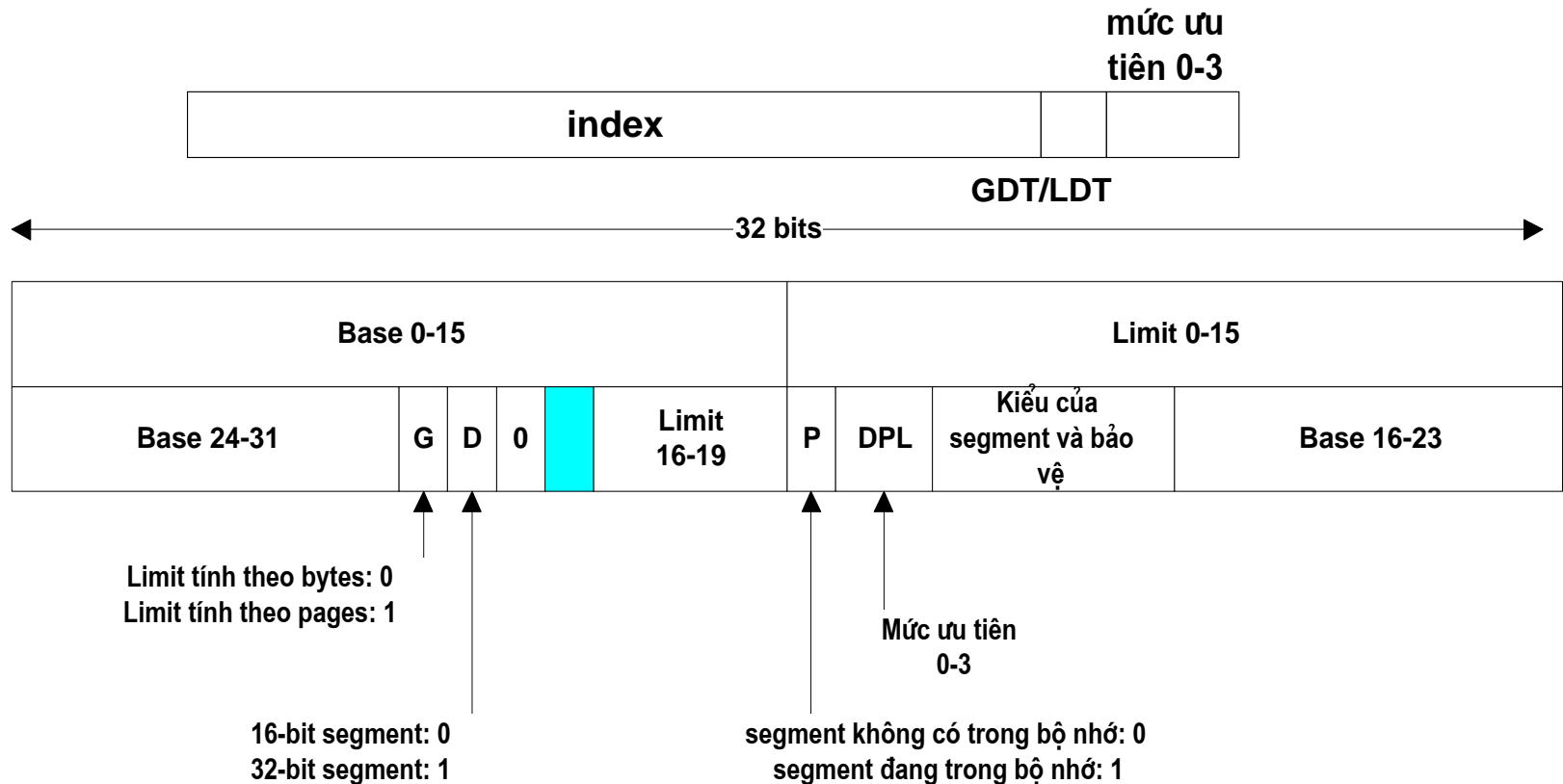


- Giả thiết tiến trình cần 4 bảng riêng biệt, luôn phát triển về kích thước
- Phương pháp phân trang không cho phép dễ dàng mở rộng
- Giải pháp: phân đoạn

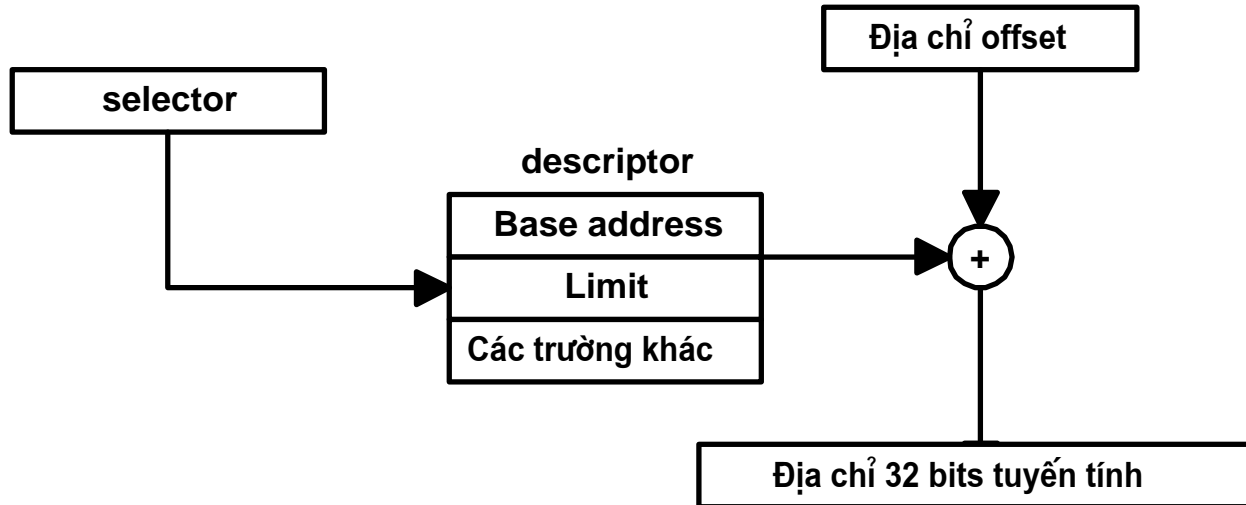
So sánh phân trang/đoạn

Tiêu chí tiêu so sánh	Phân trang	Phân đoạn
Người lập trình phải biết rõ về loại kỹ thuật phân bộ nhớ được dùng	không	có
Số lượng không gian địa chỉ tuyến tính	1	nhiều
Không gian địa chỉ ảo lớn hơn không gian vật lý	có	có
Phân biệt vùng chương trình và DL cũng như có các chế độ bảo vệ khác nhau	không	có
Có thể thay đổi kích thước ba'ng dễ dàng	không	có
Có thể chia sẻ và kết nối dễ dàng các thủ tục giữa các chương trình	không	có
Lý do sử dụng kỹ thuật	tạo không gian ảo lớn hơn không gian vật lý	cho phép chương trình và DL được phân thành các vùng logic riêng biệt với chế độ bảo vệ khác nhau

Phân đoạn trong Intel 386

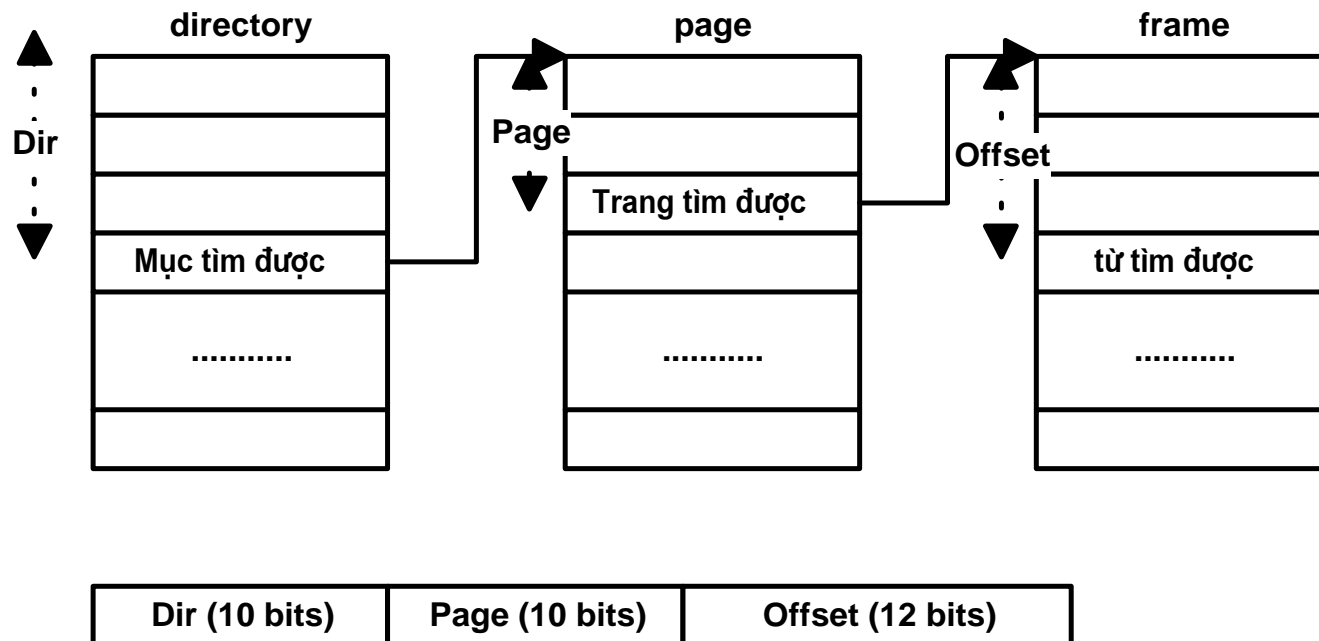


Tính địa chỉ 32-bit

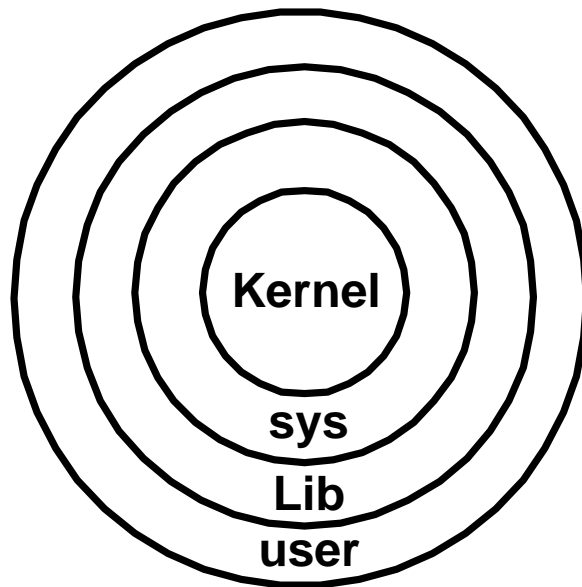


- Địa chỉ 32 bit thu được có thể được dùng như là địa chỉ tuyệt đối trong bộ nhớ hoặc là địa chỉ ảo trong bộ nhớ ảo
- Intel 386 cho phép hoạt động theo chế độ thuần segment, thuần phân trang

Phân trang trong Intel 386



Bảo vệ nhiều mức



Mức 0: nhân hệ thống
Mức 1: Gọi thủ tục hệ thống
Mức 2: Thư viện dùng chung
Mức 3: Chương trình người dùng

- Các thủ tục trong cùng một mức có thể gọi trực tiếp lẫn nhau
- Việc gọi thủ tục từ mức ngoài vào mức trong phải thông qua khái niệm “cổng” – t. là chỉ được gọi những thủ tục được phép và được quản lý
- Khái “cổng” là tương tự như khái niệm bẫy và ngắt.