Bai tap tong hop He dieu hanh

Hệ điều hành (Trường Đại học Giao thông Vận tải)

CÁC DẠNG BÀI TẬP MÔN HỆ ĐIỀU HÀNH

QUẢN LÝ TIẾN TRÌNH

1. **Xét tập hợp các tiến trình sau:**

Tiến trình	Thời điểm vào RL	Thời gian CPU	Độ ưu tiên
P_1	0	10	3
P_2	1	1	1
P_3	2.5	2	3
P_4	3	1	4
P_5	4.5	5	2

Hãy cho biết kết quả điều phối theo các chiến lược

- FCFS
- SJF
- Round Robin với t = 2
- Độ ưu tiên độc quyền

Vẽ sơ đồ Gantt và tính thời gian chờ cho từng tiến trình trong các chiến lược trên

a/ FCFS

P1	P2	P3	P4	P5

Thời gian chờ:

P_1: 0

P_2: 10 – 1 = 9

P_3: 11 – 2.5 = 8.5

P_4: 13 – 3 = 10

P_5: 14 – 4.5 = 9.5

b/ SJF

P1	P2	P4	P3	P5

Thời gian chờ:

P_1: 0

P_2: 10 – 1 = 9

P_3: 12 – 2.5 = 9.5

P_4: 11 – 3 = 8

P_5: 14 – 4.5 = 9.5

c/ Round Robin

P1	P2	P1	P3	P4	P5	P1	P5	P1	P5	P1

Thời gian chờ:

P_1: 1 + 5 + 2 + 1 = 9

P_2: 2 – 1 = 1

P_3: 5 – 2.5 = 2.5

P_4: 7 – 3 = 4

P_5: 8 + 2 + 2 – 4.5 = 7.5

d./ Độ ưu tiên độc quyền

P1	P2	P5	P3	P4

Thời gian chờ:

P_1: 0

P_2: 10 – 9 = 1

P_3: 16 – 2.5 = 13.5

P_4: 18 – 3 = 5

P_5: 11 – 4.5 = 6.5

Chú ý:

- FCFS vào trước thực hiện trước.
- SJF tiến trình nào có chiều dài CPU burst ngắn thì thực hiện trước.
- RR mỗi tiến trình chỉ được thực hiện trong một thời gian q nhất định, các tiến trình lần lượt thực hiện xoay vòng.
- Điều phối theo độ ưu tiên độc quyền: có độ ưu tiên nhỏ thực hiện trước.

QUẢN LÍ BỘ NHỚ CHÍNH

Bài 1: Trong mô hình cấp phát bộ nhớ liên tục, có năm phân mảnh bộ nhớ theo thứ tự với kích thước là 600KB, 500KB, 200KB, 300KB. Giả sử có 4 tiến trình đang chờ cấp phát bộ nhớ theo thứ tự P1, P2, P3, P4. Kích thước tương ứng của các tiến trình trên là: 212KB, 417KB, 112KB, 426KB. Hãy cấp phát bộ nhớ cho các tiến trình trên theo thuật toán First-fit, Best-first, Worst-fit.

First –Fit

600Kb	500Kb	200Kb	300Kb
P1 P3	P2		

P4 chờ

Best-Fit

600Kb	500Kb	200Kb	300Kb
P4	P2	P3	P1

Worst-Fit

600Kb	500Kb	200Kb	300Kb
P1 P3	P2		

P4 chờ

Chú ý: - First – fit :tìm vùng nhớ đầu tiên đủ lớn để chứa tiến trình

- Best – fit: tìm vùng nhớ nhỏ nhất mà có thể chứa tiến trình
- Worst – fit:tìm vùng nhớ lớn nhất cấp cho tiến trình.

Bài 2: Xét chuỗi truy xuất bộ nhớ sau:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3

Giả sử bộ nhớ vật lí có 4 khung trang. Minh họa kết quả trình thay thế trang với các thuật toán thay thế sau:

a) FIFO b) OPT c) LRU

Giải

a) FIFO

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
*	*	*	*			*	*	*	*		*	*	*	
①	1	1	1	1	1	⑤	5	5	5	5	③	3	3	3
	②	2	2	2	2	2	⑥	6	6	6	6	⑦	7	7
		③	3	3	3	3	3	②	2	2	2	2	⑥	6
			④	4	4	4	4	4	①	1	1	1	1	1

b) OPT

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
*	*	*	*			*	*				*	*	*	
①	1	1	1	1	1	1	1	1	1	1	1	⑦	7	7
	②	2	2	2	2	2	2	2	2	2	2	2	2	2
		③	3	3	3	3	3	3	3	3	③	3	3	3
			④	4	4	⑤	⑥	6	6	6	6	6	6	6

c) LRU

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
*	*	*	*			*	*				*	*	*	
①	1	1	1	1	1	1	1	1	1	1	1	1	⑥	6
	②	2	2	2	2	2	2	2	2	2	2	2	2	2
		③	3	3	3	⑤	5	5	5	5	③	3	3	3
			④	4	4	4	⑥	6	6	6	6	⑦	7	7

Chú ý:

- Thuật toán FIFO: Trong các trang đang ở trong bộ nhớ, chọn trang chọn trang được nạp vào bộ nhớ trước nhất để thay thế.
- Thuật toán OPT: Chọn trang sẽ lậu được sử dụng nhất trong tương lai để thay thế.
- Thuật toán LRU: Chọn trang lâu nhất chưa được sử dụng

PHÂN TRANG

Bài 1: Xét một không gian địa chỉ có 8 trang, mỗi trang có kích thước 1Kbyte. Ánh xạ vào bộ nhớ vật lý có 32 khung trang.

a) Địa chỉ luận lý (logical address) gồm bao nhiêu bit?

b) Địa chỉ vật lý (physical address) gồm bao nhiêu bit?

c) Bảng trang có bao nhiêu mục? Mỗi mục trong bảng trang cần bao nhiêu bit?

Trả lời:

a) 13 bits

Địa chỉ luận lý gồm 2 vùng p và d: Vùng p cho biết biết dữ liệu đang ở trang (page) nào, vùng d cho biết trong trang đó, dữ liệu ở vị trí nào.

p	d

Đề cho 8 trang ($2^3 = 8$) → Cần 3 bits cho vùng p

Mỗi trang có kích thước 1Kbyte = 1024 bytes ($2^{10} = 1024$) → Cần 10 bits cho vùng d

⇨ Địa chỉ luận lý cần 3 + 10 = 13 bits

b) 15 bits

Địa chỉ vật lý gồm 2 vùng f và d: Vùng f cho biết biết dữ liệu đang ở khung trang (frame) nào, vùng d cho biết trong khung trang đó, dữ liệu ở vị trí nào.

f	d

Đề cho 32 khung trang ($2^5 = 32$) → Cần 5 bits cho vùng f.

Mỗi trang có kích thước 1Kbyte → Mỗi frame cũng có kích thước 1Kbyte → Cần 10 bits cho vùng

d

⇨ Địa chỉ luận lý cần 5 + 10 = 15 bits

Bài 2: Cho bảng phân trang (bảng ánh xạ) của một process như hình, hãy cho biết

0	6
1	4
2	5
3	7
4	1
5	9

Bảng trang của P1

a) Địa chỉ vật lý 6578 sẽ được chuyển thành địa chỉ luận lý bao nhiêu? Biết rằng kích thước mỗi frame là 1 KB

b) Địa chỉ luận lý 3654 sẽ được chuyển thành địa chỉ vật lý bao nhiêu? Biết rằng kích thước mỗi frame là 2 KB

Trả lời:

a) Tìm địa chỉ luận lý tương ứng của địa chỉ vật lý 6578

$6578_{(10)} = 1100110110010_{(2)}$

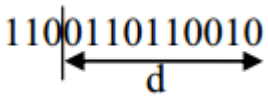
Địa chỉ luận lý:

p	d

Địa chỉ vật lý:

f	d

Kích thước mỗi frame là 1 KB (mỗi page cũng là 1 KB) ➔ vùng d trong địa chỉ luận lý và vật lý là 10 bits

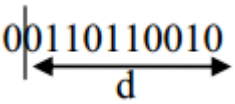


➔ Địa chỉ vật lý 1100110110010 đang ở frame $110_{(2)} = 6_{(10)}$

Tra vào bảng phân trang, frame 6 đang tương ứng với page 0

➔ Địa chỉ luận lý sẽ là (vùng d của vật lý và luận lý giữa nguyên):

$00110110010_{(2)}$ = **434**$_{(10)}$



b) Chuyển địa chỉ luận lý 3654 thành địa chỉ vật lý

$3654_{(10)} = 111001000110_{(2)}$

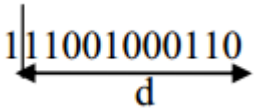
Địa chỉ luận lý:

p	d

Địa chỉ vật lý:

f	d

Kích thước mỗi frame là 2 KB (Mỗi page cũng là 2 KB) ➔ vùng d trong địa chỉ luận lý và vật lý là 11 bits

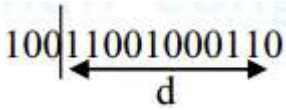


➔ Địa chỉ luận lý 111001000110 đang ở page 1

Tra vào bảng phân trang, tìm đến phần tử/mục thứ 1 của bảng phân trang, thấy 4, tức page 1 trong bộ nhớ luận lý đang được lưu ở frame 4.

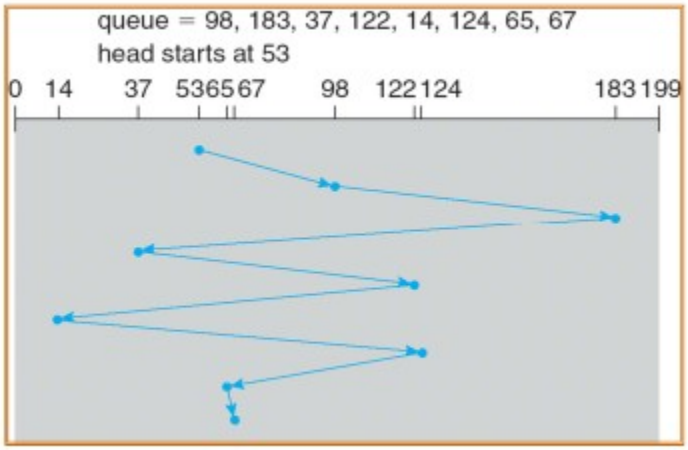
➔ Địa chỉ vật lý sẽ là (vùng d của vật lý và luận lý giữa nguyên):

$10011001000110_{(2)}$ = **9798**$_{(10)}$



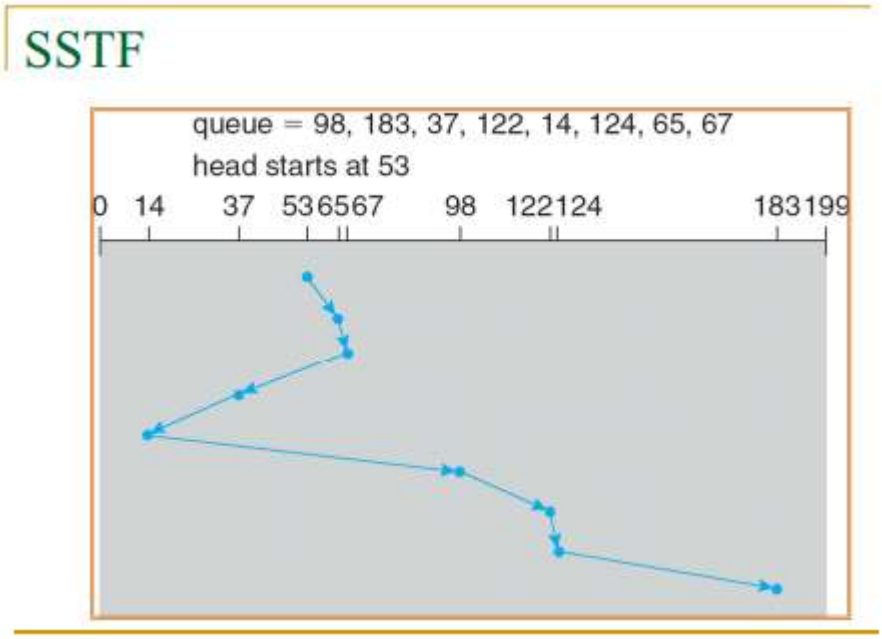
LẬP LỊCH CHO ĐĨA

1. **FCFS**: Đi theo thứ tự

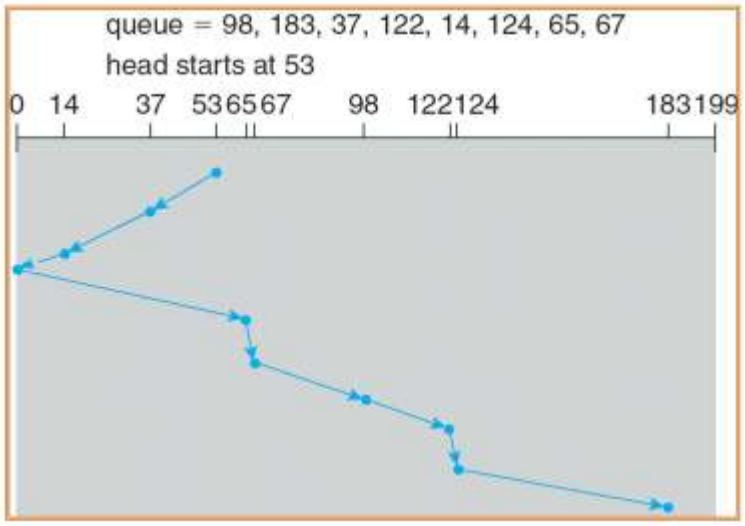


Tổng số di chuyển đầu đọc: 640 cylinder

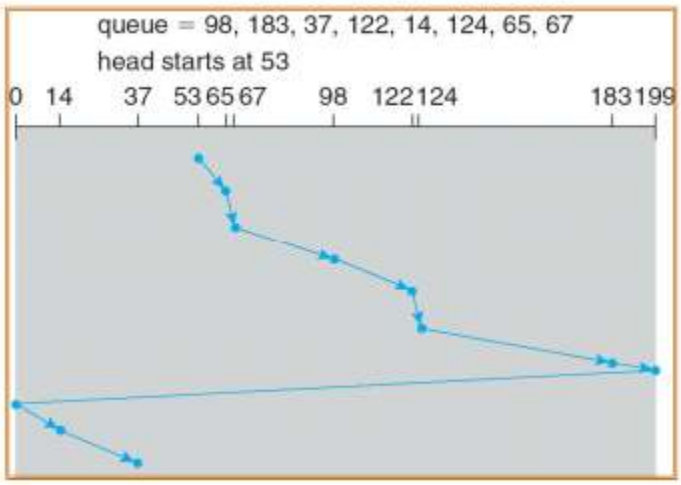
2. **SSTF**: Chọn yêu cầu với khoảng cách tối thiểu với vị trí đầu đọc hiện tại



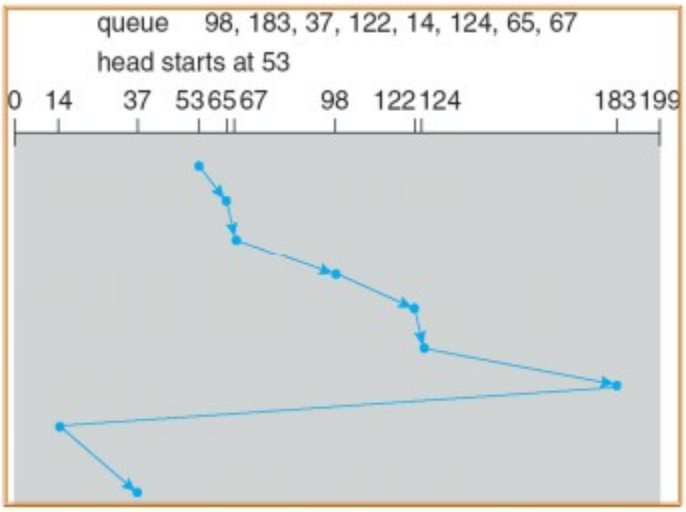
3. **SCAN**: Tay đĩa bắt đầu tại một điểm kết thúc của đĩa và di chuyển hướng tới điểm kết thúc kia. Khi đến điểm kết thúc, đầu đọc di chuyển ngược lại.



4. **C-SCAN**: Đầu đọc cũng di chuyển từ một điểm kết thúc tới điểm kia nhưng khi tới điểm kết thúc kia, nó lập tức di chuyển về điểm bắt đầu mà không phục vụ bất cứ yêu cầu nào trên đường quay về



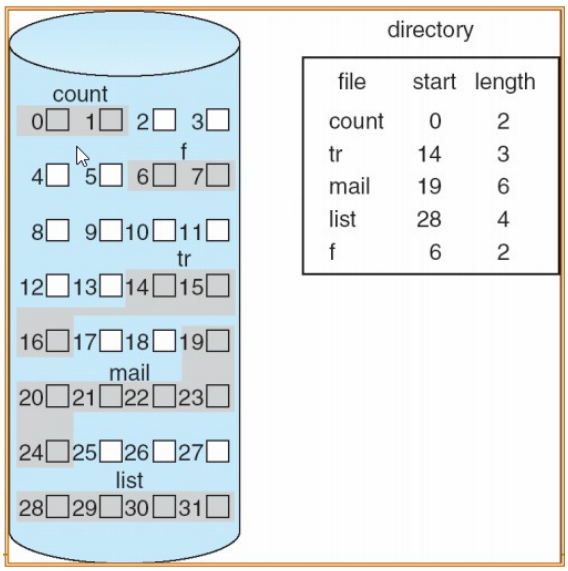
5. **C-LOOK**: Là phiên bản của C-SCAN, Tay đĩa chỉ di chuyển tới yêu cầu cuối cùng trên mỗi hướng sau đó quay lại ngay, mà không đi tới điểm cuối của đĩa



PHÂN PHỐI KHỐI ĐĨA CHO CÁC FILE

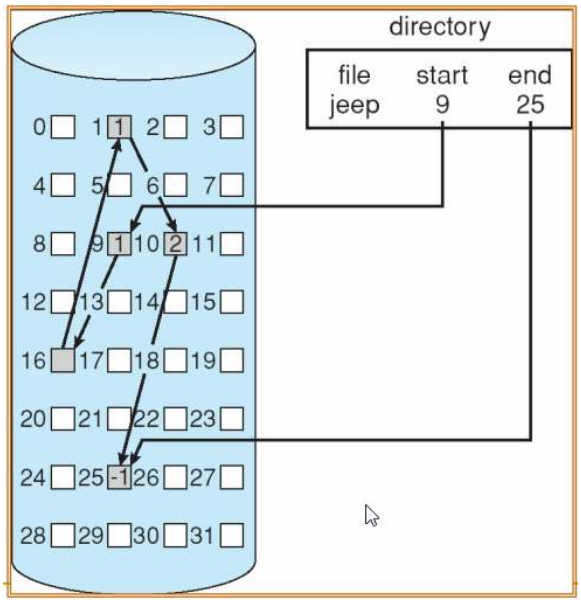
3 phương pháp:

- Liên tục



5 file cấp không gian nhớ tại các ô (0,1);(6,7);(14,15,16);(19,20,21,22,23,24); (28,29,30,31)

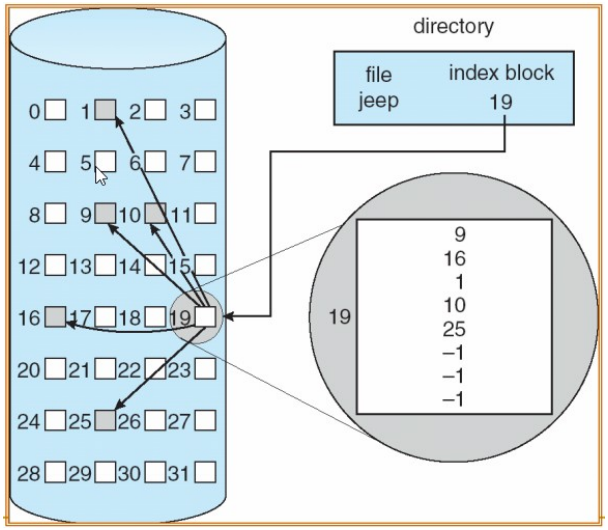
- Liên kết



Số khối cấp phát: 5 khối

Khối đầu:9 – Khối cuối: 25

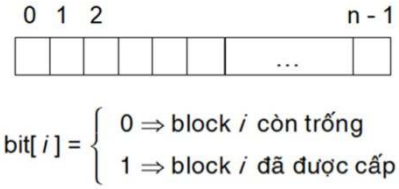
- Chỉ số



Số hiệu khối: 19

Các phương pháp quản lý không gian nhớ tự do: Đếm, Nhóm, Bitmap, Liệt kê

- Bitmap: Không gian nhớ đĩa đc chia thành các khối (block) và đc đánh số từ 0 tới max. Mỗi đĩa khi sử dụng 1 bit để đánh dấu trạng thái. Khối đĩa nào đã sử dụng thì bit trạng thái có giá trị bằng 1, chưa sử dụng thì có giá trị bằng 0. Tập hợp các kí hiệu 0,1 tạo thành bitvector



Ví dụ:

bit vector 00111100…

⇔

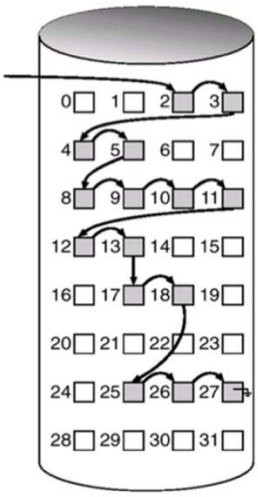
block 0, 1 trống

block 2, 3, 4, 5 đã được cấp

block 6, 7 trống

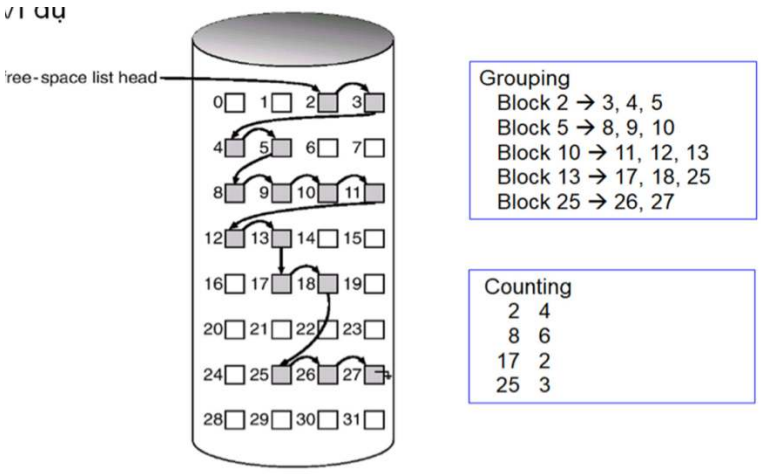
…

- Liệt kê: Sử dụng một danh sách móc nối để liệt kê các khối đĩa tự do. Con trỏ đầu trong danh sách này chỉ tới các khối đĩa tự do đầu tiên, mỗi khối có một con trỏ để trỏ tới khối kế tiếp



-Nhóm: Các khối đĩa tự do liên tiếp thành một nhóm. Khối đĩa tự do đầu tiên trong nhóm lưu trữ địa chỉ của các đĩa tự do trong nhóm. Khối đĩa tự do cuối cùng trong nhóm lưu trữ địa chỉ của khối đĩa tự do đầu tiên của khối tiếp theo

-Đếm: Biến đổi của phương pháp lập nhóm. Hệ thống lập danh sách quản lý địa chỉ của các khối đĩa tự do đầu tiên và số lượng các khối đĩa tự do liên tục kế tiếp các khối đĩa đó.



Grouping and Counting