

# Bài tập lý thuyết

## Hệ điều hành

# Tổng quan

1. Nêu ba mục đích chính của hệ điều hành?
2. Mục đích của lời gọi hệ thống (system calls) là gì?
3. Các hoạt động chính của hệ điều hành nhằm quản lý tiến trình (process management) là gì?
4. Các hoạt động chính của hệ điều hành nhằm quản lý bộ nhớ là gì?
5. Các hoạt động chính của hệ điều hành nhằm quản lý lưu trữ thứ cấp (secondary-storage management) là gì?

# Tiến trình

1. Tiến trình là gì, các trạng thái của tiến trình và sơ đồ mô tả mối quan hệ giữa chúng?
2. Mô hình điều phối tiến trình và cơ chế hoạt động?
3. Sơ đồ cài đặt tiến trình trong hệ điều hành?

# Bài toán

## Điều phối tiến trình

- Sơ đồ Gantt
- Thời gian chờ
- Thời gian hoàn thành

## Chiến thuật:

- FCFS, SJF, RR, Độ ưu tiên
- Độc quyền/không độc quyền

# FCFS (FIFO)

- Cho các tiến trình sau

<u>Tiến trình</u>	<u>Thời gian xử lý (CPU burst)</u>
P1	17
P2	8
P3	9

- Hãy cho biết kết quả điều phối theo chiến lược điều phối FCFS dùng biểu đồ Gantt
- Tính thời gian chờ trung bình (waiting time)
- Tính thời gian hoàn thành trung bình (turnaround time)
- Gợi ý: FCFS (First Come First Served)
  - Tiến trình đến trước được xử lý trước
  - Độc quyền

# Round robin

- Hãy cho biết kết quả điều phối, tính thời gian chờ trung bình và thời gian hoàn thành trung bình của các tiến trình sau theo chiến lược điều phối RR với thời lượng quantum = 4

<u>Tiến trình</u>	<u>Thời gian xử lý</u>
P1	13
P2	8
P3	3
P4	5

- Gợi ý: RR (Round Robin – xoay vòng)
  - Tiến trình đến trước được xử lý trước
  - Mỗi tiến trình dùng CPU trong thời gian tối đa quantum
  - Độc quyền

# SJF độc quyền

- Hãy cho biết kết quả điều phối, tính thời gian chờ trung bình và thời gian hoàn thành trung bình của các tiến trình sau theo chiến lược điều phối SJF độc quyền

<u>Tiến trình</u>	<u>Thời gian xử lý</u>	<u>Thời điểm vào RL</u>
P1	6	0
P2	8	2
P3	7	4
P4	3	5

- Gợi ý: SJF (Shortest Job First)
  - Độ dài ngắn thực hiện trước

# SJF không độc quyền (SRTF)

- Hãy cho biết kết quả điều phối, tính thời gian chờ trung bình và thời gian hoàn thành trung bình của các tiến trình sau theo chiến lược điều phối SJF **không độc quyền**

<u>Tiến trình</u>	<u>Thời gian xử lý</u>	<u>Thời điểm vào RL</u>
P1	8	0
P2	5	2
P3	1	4
P4	4	5

- Gợi ý: SJF (Shortest Job First)
  - Độ dài ngắn thực hiện trước



# Ưu tiên độ quyền

- Hãy cho biết kết quả điều phối, tính thời gian chờ và thời gian hoàn thành trung bình của các tiến trình sau theo chiến lược điều phối ưu tiên độ quyền

Tiến trình	Chiều dài CPU burst	Thời điểm vào RL	Độ ưu tiên
P1	2	0	2
P2	5	1	3
P3	3	2	1
P4	4	3	0

- Ưu tiên nhỏ hơn thực hiện trước

# Ưu tiên không độc quyền

- Hãy cho biết kết quả điều phối và tính thời gian chờ trung bình của các tiến trình sau theo chiến lược điều phối ưu tiên **không độc quyền**

Tiến trình	Chiều dài CPU burst	Thời điểm vào RL	Độ ưu tiên
P1	2	0	2
P2	5	1	3
P3	3	2	1
P4	4	3	0

- Ưu tiên nhỏ hơn thực hiện trước, tiến trình có thể dừng giữa chừng, nhường CPU cho tiến trình có độ ưu tiên thấp hơn

# Bài tập 1

- Hãy cho biết kết quả điều phối và tính thời gian chờ trung bình của các tiến trình sau theo **các** chiến lược điều phối FCFS, SJF, RR với  $q = 2$ , ưu tiên độc quyền, ưu tiên không độc quyền

Tiến trình	Vào Ready List	Thời gian xử lý	Ưu tiên
P1	0	10	3
P2	1	1	1
P3	2.5	2	3
P4	3	1	4
P5	4.5	5	2

# Bài tập 2

- Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
<i>P1</i>	2	2
<i>P2</i>	1	1
<i>P3</i>	8	4
<i>P4</i>	4	2
<i>P5</i>	5	3

The processes are assumed to have arrived in the order *P1*, *P2*, *P3*, *P4*, *P5*, all at time 0.

- Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
- What is the turnaround time of each process for each of the scheduling algorithms in part a?
- What is the waiting time of each process for each of these scheduling algorithms?
- Which of the algorithms results in the minimum average waiting time (over all processes)?

# Bài tập 3

- Hãy điều phối
  - CPU: SJF không độc quyền, R1, R2: FCFS

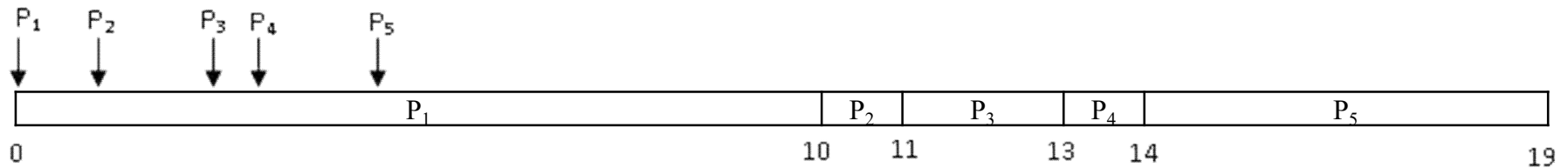
Process	Arrival	CPU 1	IO 1		CPU 2	IO 2	
			Time	IO		Time	IO
P1	0	7	3	R1	1	2	R2
P2	3	2	9	R2	7	1	R1
P3	9	3	4	R2	3	0	NULL
P4	15	2	8	R1	0	0	NULL

# Kiểm tra

- Hãy cho biết kết quả điều phối và tính thời gian chờ trung bình của các tiến trình sau theo **các** chiến lược điều phối FCFS, SJF, RR với  $q = 2$ , ưu tiên độc quyền, ưu tiên không độc quyền

Tiến trình	Vào RL	CPU	Ưu tiên
P1	0	10	3
P2	1	1	1
P3	2.5	2	3
P4	3	1	4
P5	4.5	5	2

# FCFS



- Thời gian chờ:

- P<sub>1</sub>: 0
- P<sub>2</sub>:  $10 - 1 = 9$
- P<sub>3</sub>:  $11 - 2.5 = 8.5$
- P<sub>4</sub>:  $13 - 3 = 10$
- P<sub>5</sub>:  $14 - 4.5 = 9.5$

- Thời gian chờ trung bình

- $(0+9+8.5+10+9.5)/5$

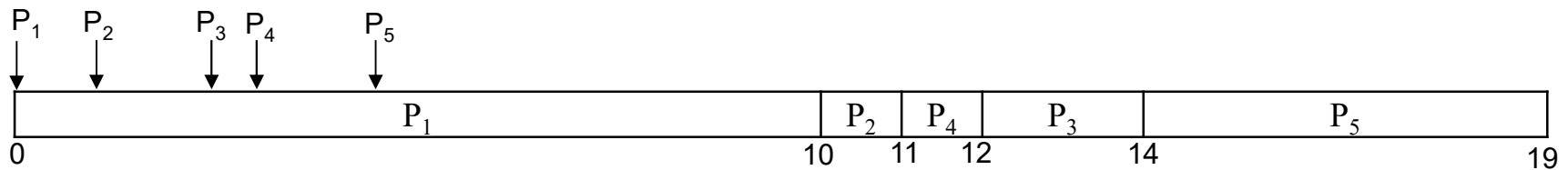
- Thời gian hoàn thành:

- P<sub>1</sub>:  $10 - 0 = 10$
- P<sub>2</sub>:  $11 - 1 = 10$
- P<sub>3</sub>:  $13 - 2.5 = 10.5$
- P<sub>4</sub>:  $14 - 3 = 11$
- P<sub>5</sub>:  $19 - 4.5 = 14.5$

- HT Trung bình

- $(10+10+10.5+11+14.5)/5$

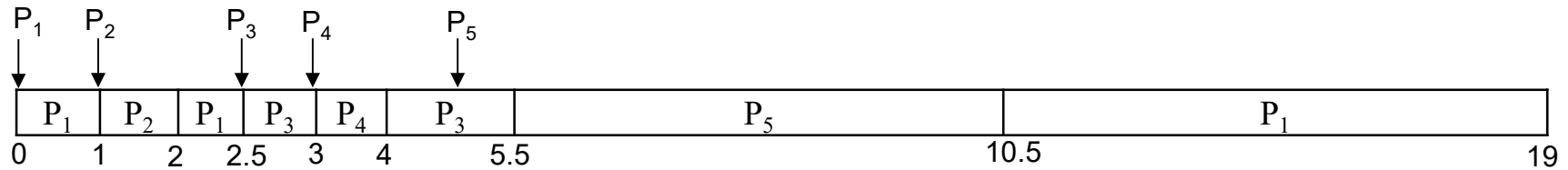
# SJF



- Thời gian chờ:
  - P<sub>1</sub>: 0
  - P<sub>2</sub>:  $10 - 1 = 9$
  - P<sub>3</sub>:  $12 - 2.5 = 9.5$
  - P<sub>4</sub>:  $11 - 3 = 8$
  - P<sub>5</sub>:  $14 - 4.5 = 9.5$
- Thời gian chờ trung bình
  - $(0 + 9 + 9.5 + 8 + 9.5) / 5$
- Thời gian hoàn thành:
  - P<sub>1</sub>:  $10 - 0 = 10$
  - P<sub>2</sub>:  $11 - 1 = 10$
  - P<sub>3</sub>:  $14 - 2.5 = 11.5$
  - P<sub>4</sub>:  $12 - 3 = 9$
  - P<sub>5</sub>:  $19 - 4.5 = 14.5$
- HT Trung bình
  - $(10 + 10 + 11.5 + 9 + 14.5) / 5$

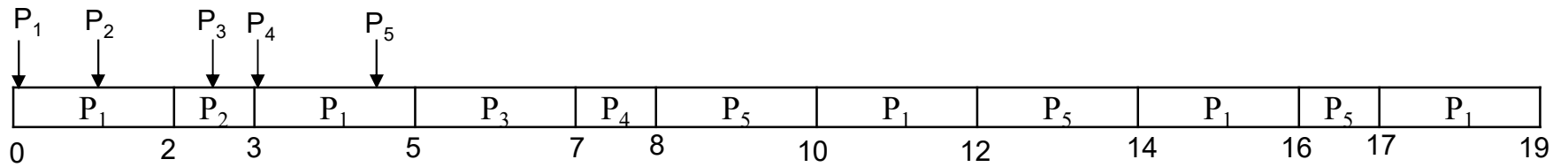


# SJF không độc quyền Hoặc SRTF

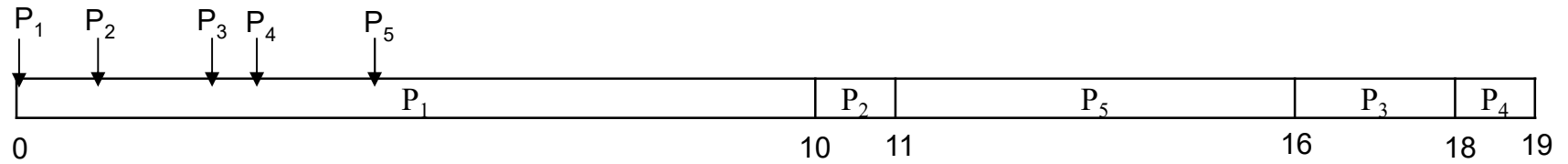


- Thời gian chờ:
  - $P_1$ :  $1+8$
  - $P_2$ : 0
  - $P_3$ : 1
  - $P_4$ : 0
  - $P_5$ :  $5.5 - 4.5$
- Thời gian chờ trung bình
  - $(9+0+1+0+1)/5$
- Thời gian hoàn thành:
  - $P_1$ :  $19 - 0 = 19$
  - $P_2$ :  $2 - 1 = 1$
  - $P_3$ :  $5.5 - 2.5$
  - $P_4$ :  $4 - 3$
  - $P_5$ :  $10.5 - 4.5$
- HT Trung bình
  - $(19+1+2.5+1+6)/5$

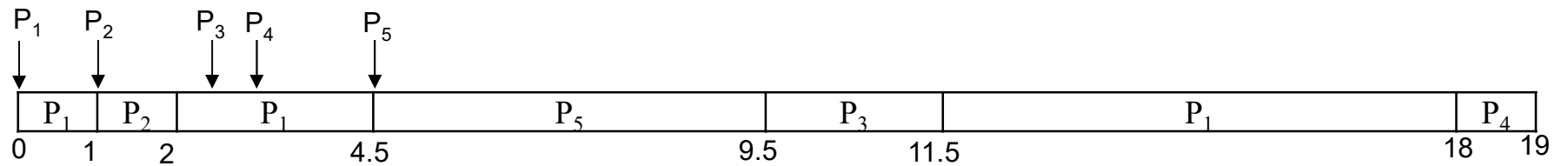
# Round Robin $q = 2$



# Độ ưu tiên độ quyền



# Độ ưu tiên không độc quyền



Bài toán

Đồng bộ tiến trình

# Đồng bộ tiến trình

1. Các điều kiện cho một giải pháp đồng bộ?

# Busy waiting

Hãy cho biết giải pháp sau đây đảm bảo yêu cầu đồng bộ tiến trình hay không? Vì sao?

```
while (TRUE) {  
    int j = 1-i;  
    flag[i]= TRUE;  
    turn = i;  
    while (turn == j && flag[j]==TRUE);  
    critical-section ();  
    flag[i] = FALSE;  
    Noncritical-section ();  
}
```

# Busy waiting

Cho chỉ thị Swap không thể phân chia:

```
void Swap(bool a, bool b)
{
    bool temp;
    temp := a;
    a:= b;
    b:= temp;
}
```

Hãy xây dựng giải pháp đồng bộ hóa tiến trình thỏa mãn yêu cầu truy suất độc quyền dùng chỉ thị Swap.



# Sleep & Wakeup

Xét hai tiến trình sau:

```
process A { while (TRUE) na = na + 1; }  
process B { while (TRUE) nb = nb + 1; }
```

Hãy xây dựng giải pháp đồng bộ hóa hai tiến trình trên sử dụng hai semaphore sao cho:

- a)  $nb < na \leq nb + 50$
- b)  $na < nb + 50$

# Sleep & Wakeup

Xét hai tiến trình cùng thực hiện đoạn code sau:

```
do
{
    X = X + 1;
    if ( X == 20) X = 0;
}while (TRUE);
```

- a) Hãy nêu trường hợp cho thấy X vượt quá 20?
- b) Hãy sửa chữa đoạn code trên để đảm bảo X không thể vượt quá 20 (dùng semaphore).

# Sleep & Wakeup

Xét hai tiến trình sau:

```
process P1 { A1 ; A2 }  
process P2 { B1 ; B2 }
```

Hãy nêu giải pháp đồng bộ hóa sao cho:

- a) Cả A1 và B1 hoàn tất trước khi A2 hoặc B2 bắt đầu.
- b) Các công việc hoàn thành tuần tự theo thứ tự A1, B1, A2, B2.

# Sleep & Wakeup

Xét các tiến trình sau:

```
process P1 { for ( i = 1; i <= 100; i ++ ) Ai }  
process P2 { for ( j = 1; j <= 100; j ++ ) Bj }
```

Hãy xây dựng giải pháp đồng bộ hóa sao cho, với bất kỳ  $1 < k \leq 100$ :

- a)  $A_k$  chỉ có thể bắt đầu khi  $B_{k-1}$  đã hoàn tất
- b)  $A_k$  chỉ có thể bắt đầu khi  $B_{k-1}$  đã hoàn tất và  $B_k$  chỉ có thể bắt đầu khi  $A_{k-1}$  đã hoàn tất.

# Sleep & Wakeup

Sử dụng semaphore để thực hiện đoạn lệnh sau theo mô hình xử lý đồng hành:

$w = x1 * x2;$

$v = x3 * x4;$

$y = v * x5;$

$z = v * x6;$

$y = w * y;$

$z = w * z;$

$ans = y + z;$

# Sleep & Wakeup

Cho mảng: `int x[20];`

Sử dụng cơ chế đồng bộ hoá là semaphore để viết code cho 3 tiến trình A, B, C cùng thực hiện đồng thời các thao tác trên mảng x thoả mãn các yêu cầu sau:

- A tính tổng giá trị các phần tử mảng x có chỉ số chẵn.
- B tính tổng giá trị các phần tử mảng x có chỉ số lẻ.
- C tính tổng giá trị tất cả các phần tử của mảng x, dựa trên kết quả trả về của A và B.

# Deadlock

Giả sử có 3 tiến trình và 3 loại tài nguyên A=4, B=5 và C=6 như sau:

Processes	Allocation			Request		
	A	B	C	A	B	C
P1	1	1	3	0	2	3
P2	2	1	0	1	1	1
P3	0	1	1	0	1	0

- Hãy cho biết hệ thống có bị deadlock không? Chỉ ra chuỗi tiến trình gây deadlock nếu có
- Nếu P1 yêu cầu thêm 2 thể hiện B thì hệ thống sẽ như thế nào?

Processes	Allocation			Request			Available			Finish
	A	B	C	A	B	C	A	B	C	
P1	1	1	3	0	2	3	1	2	2	<div> <div>F</div> <div>F</div> <div>T3+p1</div> <div>I</div> </div>
P2	2	1	0	1	1	1	3	3	2	T1+p2
P3	0	1	1	0	1	0	3	4	3	T2+p3
							4	5	6	

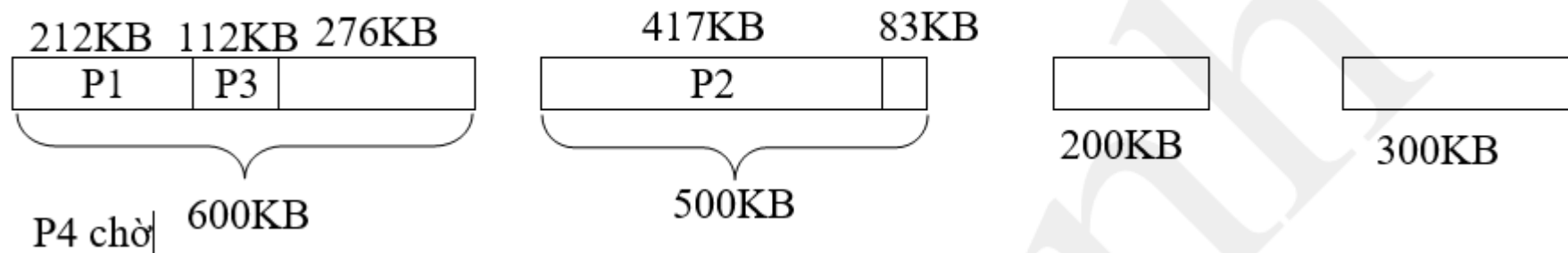


# Quản lý bộ nhớ

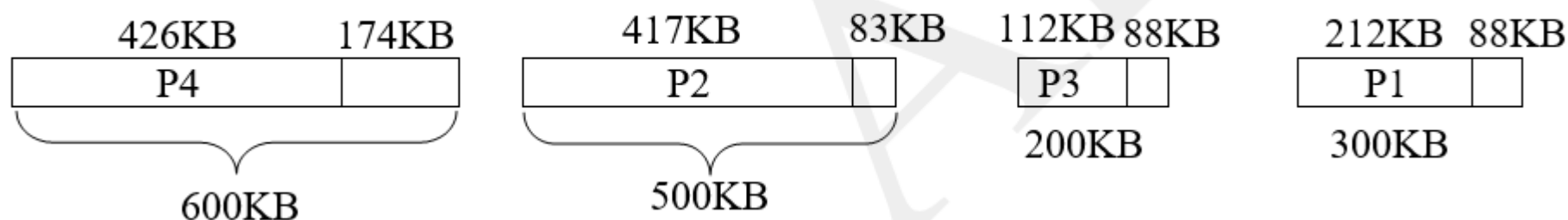
# Cấp phát liên tục

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.

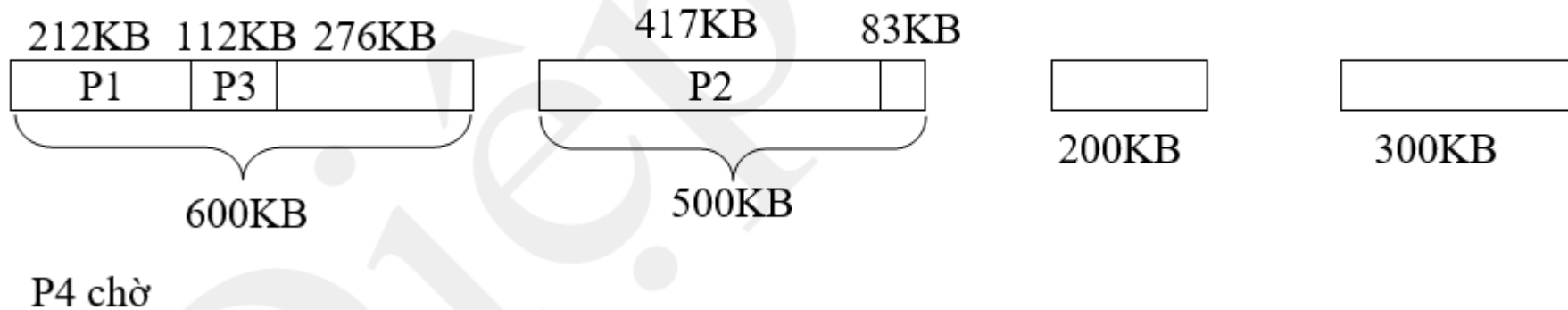
### First – fit



### Best – fit



### Worst – fit



# Cấp phát không liên tục – phân đoạn

Cho một bảng phân đoạn sau (segment table):

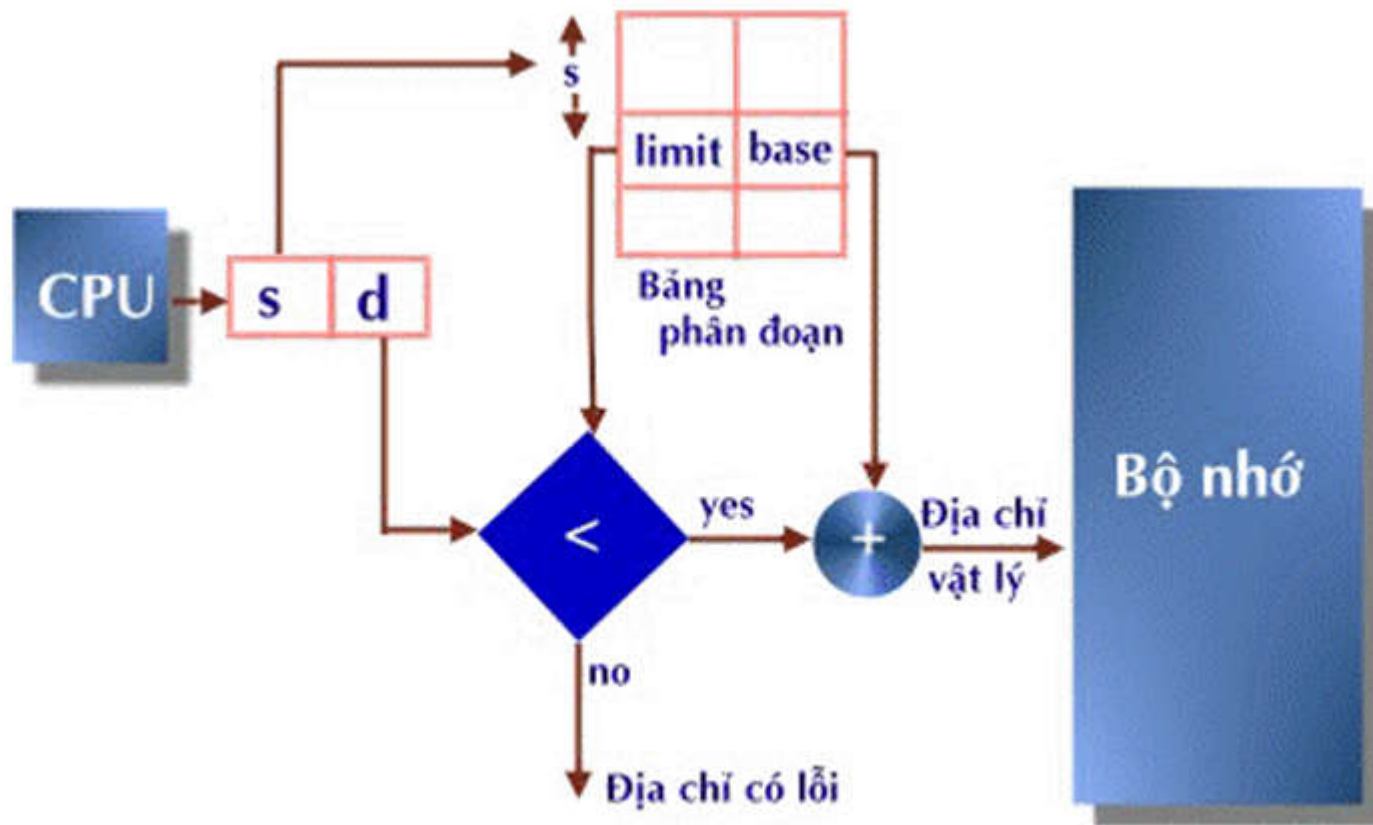
Segment	Limit	Base
0	500	215
1	25	2100
2	100	120

Xác định địa chỉ vật lý của các địa chỉ logic sau ?

0.410

1.12

2.300



**Hình 4.18** Cơ chế phần cứng hỗ trợ kỹ thuật phân đoạn

# Cấp phát không liên tục, phân trang

Một tiến trình được nạp vào bộ nhớ theo mô hình phân trang (paging) với kích thước trang là 1024 byte. Bảng trang như sau:

Hãy chuyển các địa chỉ logic sau thành địa chỉ vật lý:

a) 1521

b) 3249

1
5
3
6

a)

- $a = 1521$
- $p = 1521 \operatorname{div} 1024 = 1$
- $d = 1521 \bmod 1024 = 497$
- $f = 5$  (dựa vào bảng trang vì  $p = 1$ )
- $A = 5 * 1024 + 497 = 5617$

b)

- $a = 3249$
- $p = 3249 \operatorname{div} 1024 = 3$
- $d = 1521 \bmod 1024 = 177$
- $f = 6$  (dựa vào bảng trang vì  $p = 3$ )
- $A = 6 * 1024 + 177 = 6321$

# Paging

$$p = a \text{ div } ps$$

$$d = a \text{ mod } ps$$

Từ  $p$  và bảng trang để tìm  $f$

$$A = f * ps + d$$



# Bộ nhớ ảo

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 lý có 4 khung trang. Minh họa kết quả trình thay thế trang và số lỗi trang với các thuật toán thay thế sau.

- a) FIFO
- b) OPT
- c) LRU

### 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

# Quản lý tập tin

Một ổ đĩa C được định dạng dưới dạng FAT16 gồm có 15 cluster. Kích thước của mỗi cluster là 512 byte, giả sử có bảng FAT sau:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-1	0	5	6	8	7	-1	-1	-1	-1	12	-1	10	0

Thư mục gốc bắt đầu tại cluster 0, tại cluster 0 và cluster 9 xem được các entry như sau:

Filename	Ext	attrib	Start cluster	size
Hdh	Doc		11	800
HinhAnh		D	9	
pascal	Doc		4	1200

Filename	Ext	attrib	Start cluster	size
Hoguom	Jpg		3	1200
Halong	Jpg		13	700

Hãy vẽ cây thư mục và cho biết các số liệu cluster của từng file và thư mục

## Cluster

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
R	R		HG1	PC1	HG2	PC2	PC3	HG3	HA	HL2	HDH1	HDH2	HL1	

Các số hiệu cluster của từng file và thư mục:

- hdh: 11, 12
- HìnhAnh: 9
- Pascal: 4, 6, 7
- HG: 3, 5, 8
- HL: 13, 10

Cây thư mục:

