

# Ng lý hệ Diên hành

ĐỀ 2014-2015

## Câu 1:

A, Định nghĩa tiến trình và các thành phần của tiến trình?

B, Giải thích sự khác biệt giữa hiện tượng phân mảnh nội vi và phân mảnh ngoại vi quản lý bộ nhớ. Lấy một ví dụ

A, - Tiến trình là một bộ phận của chương trình đang thực hiện. Tiến trình là đơn vị làm việc cơ bản của hệ thống, trong hệ thống có thể tồn tại nhiều tiến trình cùng hoạt động, trong đó có cả tiến trình của hệ điều hành và tiến trình của chương trình người sử dụng. Các tiến trình này có thể hoạt động đồng thời với nhau.

- Khi Process đang thực hiện, nó gồm có thành phần thông tin sau:

+ ID (Identifier): là đặc điểm nhận dạng từng process với nhau vì có nhiều process tồn tại đồng thời.

+ State (trạng thái): Là trạng thái hoạt động của Process, nếu đang được CPU thực thi thì state là Running.

+ Priority (độ ưu tiên): Mức ưu tiên của Process so với các Process khác đang được thực hiện đồng thời.

+ Program Counter (bộ đếm chỉ thị): Mang địa chỉ của lệnh tiếp theo sẽ được thực thi.

+ Memory Pointer: Gồm có con trỏ đến địa chỉ của Program Code và data.

+ Context Data: dữ liệu lấy từ thanh ghi của CPU khi process đang ở trạng thái running

+ IO Status Information: danh sách các thiết bị IO cấp phát cho process, danh mục các file được Process này sử dụng.

+ Accounting Information: Thông tin này là lượng CPU được phép sử dụng. Giới hạn thời gian thực hiện,...

B, Phân mảnh ngoại (external fragmentation): là hiện tượng khi kích thước không gian nhớ còn trống đủ để thỏa mãn yêu cầu cấp phát nhưng không gian nhớ này lại không liên tục. Hiện tượng phân mảnh ngoại xảy ra khi bạn thường xuyên cấp phát vùng nhớ mới, sau đó xóa đi những phần vùng nhớ đã cấp phát một cách không thứ tự.

Phân mảnh nội (internal fragmentation): là hiện tượng sẽ có vùng nhớ dư thừa khi ta cấp phát một vùng nhớ hơi lớn hơn kích thước yêu cầu.

Ví dụ: Cấp một khoảng trống 18,464 bytes cho một process yêu cầu 18,462 bytes. Hiện tượng phân mảnh nội thường xảy ra khi bộ nhớ thực được chia thành các khối kích thước cố định (fixed-sized block) và các process được cấp phát theo đv khối

## Câu 2: (14-15)

a) giải sử rằng thời điểm thứ 5 ko có tài nguyên nào đang được sd ngoại trừ vxl và bộ nhớ chính, hãy theo dõi các sự kiện sau :

- tại thời điểm thứ 5: tiến trình p1 thực hiện 1 yêu cầu đọc từ unit 3 của đĩa

- tại thời điểm thứ 15: thời gian sử dụng vxl dành cho p5 kết thúc

- tại thời điểm thứ 18 : p7 thực hiện 1 yêu cầu ghi vào unit 3 của đĩa

- tại thời điểm thứ 20 : p3 thực hiện 1 yêu cầu ghi vào unit 2 của đĩa

- tại thời điểm thứ 24: p5 thực hiện 1 yêu cầu ghi vào unit 3 của đĩa

- tại thời điểm thứ 28 : tiến trình p5 bị đưa ra ngoài bộ nhớ phụ
- tại thời điểm thứ 33: một ngắt xuất hiện từ unit 2 của đĩa : quá trình đọc của p3 được chấp nhận (p3 hoàn thành quá trình đọc đĩa \)
- tại thời điểm thứ 36: một ngắt xuất hiện từ unit 3 của đĩa : quá trình đọc của p1 được chấp nhận (p3 hoàn thành quá trình đọc đĩa \)
- tại thời điểm thứ 38 : p8 kết thúc
- tại thời điểm thứ 40: một ngắt xuất hiện từ unit 3 : quá trình ghi của p5 được chấp nhận
- tại thời điểm thứ 44: tiến trình p5 được đưa vào lại bộ nhớ chính .
- tại thời điểm thứ 48: một ngắt xuất hiện từ unit 3 của đĩa : quá trình đọc của p7 được chấp nhận

#### Câu hỏi:

1. Tại mỗi thời điểm thứ 22, 37, 47 hãy cho biết trạng thái của mỗi tiến trình ?

b) xét tiến trình p1, p2 hoạt động trong hệ thống đơn vị xử lý với các tiến trình hợp tác tuần tự, thực hiện bất đồng bộ và cùng chia sẻ 1 đoạn code (và dữ liệu) sau:

P1:	P2:
{	{
Share int x;	Share int x;
X=10;	X=10;
While(1){	While(1){
X=x-1;	X=x-1;
X=x+1;	X=x+1;
If(x!=10){	If(x!=10){
Printf("x is %d",x) }	Printf("x is %d",x) }
}	}

- chỉ ra chuỗi tuần tự (xen kẽ) quá trình thực hiện các lệnh trong đoạn code trên của 2 tiến trình p1 và p2 sao cho thông báo "x is 10" được in ra.
- sử dụng phương pháp semaphore để đồng bộ hóa 2 tiến trình p1 và p2 với đoạn code trên để đảm bảo câu lệnh printf("x is %d",x) không bao giờ thực hiện được.



a,  
 Thời điểm 22: P1: blocked for I/O  
 P7: blocked for I/O  
 Thời điểm 37: P1: ready/running  
 P7: blocked for I/O  
 Thời điểm 47: P1: ready/running  
 P7: blocked for I/O

P3: blocked for I/O P5: ready/running  
 P8: ready/running  
 P3: ready/running P5: blocked suspend  
 P8: ready/running  
 P3: ready/running P5: ready suspend  
 P8: exit

B, i, P1:  $x = x - 1$ ;  
 P1:  $x = x + 1$ ;  
 P2:  $x = x - 1$ ;  
 P1: if( $x \neq 10$ )  
 P2:  $x = x + 1$ ;  
 P1: printf("x is %d", x);

M(x)  
 9  
 10  
 9  
 9  
 10  
 10

"x is 10" được in ra.

ii, s: semaphore;

```
parbegin P1: {
    shared int x; x = 10;
    for (; ;) { semWait(s); x = x - 1; x = x + 1; if (x != 10)
        printf("x is %d", x); semSignal(s);
    }
}
```

```
P2: {
    shared int x; x = 10;
    for (; ;) { semWait(s); x = x - 1; x = x + 1; if(x != 10)
        printf("x is %d", x); semSignal(s);
    }
}
```

Parend

Câu 3:

a) xét 1 không gian địa chỉ có 8 trang, mỗi trang có kích thước 1K, ánh xạ vào bộ nhớ vật lý có 32 khung trang

i) địa chỉ logic bao gồm bao nhiêu bit?

ii) địa chỉ vật lý bao gồm bao nhiêu bit?

b) xét hệ thống bộ nhớ ảo sử dụng bảng trang 1 cấp có TLB. Cho biết các thông số sau:

- thời gian truy xuất trong TLB là:  $t_c = 20ns$
- tỉ lệ tìm thấy 1 số hiệu trang trong TLB là: 80%
- thời gian truy cập và bộ nhớ chính là:  $t_m = 75ns$
- thời gian chuyển trang khi trang bị lỗi trang là:  $t_d = 500000 ns$
- tỷ lệ trang bị thay thế khi bị lỗi trang là: 50%

Yêu cầu:

- i. tính thời gian truy cập thực tế nếu số trang bị lỗi là 0%
- ii. tính thời gian truy cập thực tế nếu số trang bị lỗi là 10%

với cả 2 trường hợp, giả thiết mọi chi phí để truy cập TLB, bảng trang và bảng khung trang đều được bỏ qua.

Câu 3: a. Địa chỉ logic =  $8.1024 \Rightarrow$  có 13 bit cho địa chỉ logic

Địa chỉ vật lý =  $32.1024 \Rightarrow$  có 15 bit cho địa chỉ vật lý

B, i.  $EAT = 0.8 (TLB + tm) + 0.2 (TLB + tm + tm)$

$$EAT = 0.8 (20 + 75) + 0.2 (20 + 75 + 75)$$

$$EAT = 76 + 34 = 110 \text{ ns}$$

ii,  $EAT = 0.8 (TLB + tm) + 0.2 (0.9(TLB + tm + tm) + 0.1 (TLB + tm + 0.5 (td) + 0.5 (2td + tm)))$

$$EAT = 0.8 (20 + 75) + 0.2 (0.9 (20 + 75 + 75) + 0.1 (20 + 75 + 0.5 (500000) + 0.5 (1000000) + 75))$$

$$EAT = 76 + 0.2 (153 + .1 (750170)) = 76 + 15034 = 15110 \text{ ns}$$