

Chọn câu trả lời đúng bằng điền đáp án vào bảng

Câu 1	<b>D</b>	Câu 4	<b>C</b>	Câu 7	<b>B</b>	Câu 10	<b>B</b>	Câu 13	<b>D</b>
Câu 2	<b>D</b>	Câu 5	<b>B</b>	Câu 8	<b>C</b>	Câu 11	<b>A</b>	Câu 14	<b>C</b>
Câu 3	<b>B</b>	Câu 6	<b>C</b>	Câu 9	<b>B</b>	Câu 12		Câu 15	<b>A</b>

Xét một hệ thống sử dụng kỹ thuật phân trang, với bảng trang được lưu trữ trong bộ nhớ chính. Thời gian cho một lần truy xuất bộ nhớ bình thường là 250 nanoseconds. Nếu sử dụng TLBs với hit-ratio (tỉ lệ tìm thấy) là 85%, thời gian để tìm trong TLBs xem như bằng 0, thời gian truy xuất bộ nhớ trong hệ thống là khoảng bao nhiêu? (effective memory reference time)

- Cho process P1 có kích thước là 240 KByte cấp phát trong bộ nhớ theo phân hoạch có thứ tự như sau: 1 (250 KByte), 2 (100 KByte), 3 (200 KByte), 4 (400 KByte). Biết con trỏ đang nằm ở vùng nhớ thứ 2, vùng nhớ thứ 2 đã được cấp phát và vùng nhớ 1, 3 và 4 vẫn còn trống. Hỏi process P1 sẽ được cấp phát trong vùng nhớ nào, nếu dùng giải thuật next-fit

- Xét 1 máy tính có không gian địa chỉ luận lý 32 bit, và kích thước 1 trang là 2 KByte. Hỏi bảng trang (page table) có bao nhiêu mục (entry)?

- Xét bài toán Reader-Writers: có một database hoặc file với nhiều Readers (đề đọc) và nhiều Writers (đề ghi) dữ liệu vào database/file. Khi một Writer đang truy cập database/file thì không một quá trình nào khác được truy cập, trong khi đó nhiều Readers có thể cùng lúc đọc database/file.

Writer Process	Reader Process
<pre> <b>wait(wrt);</b> ... writing is performed ... <b>signal(wrt);</b> </pre>	<pre> <b>wait(mutex);</b> readcount++; if (readcount == 1)     <b>wait(wrt);</b> <b>signal(mutex);</b> ... reading is performed ... <b>wait(mutex);</b> readcount--; if (readcount == 0)     <b>signal(wrt);</b> <b>signal(mutex);</b> </pre>

Biết Semaphore dùng trong đoạn code trên có cấu trúc dữ liệu và hai hàm wait và signal hoạt động như trong phần phụ lục. Biết **mutex** và **wrt** là hai biến Semaphore được khởi tạo với value bằng 1;

**readcount** là một biến nguyên được khởi tạo bằng 0; và ba biến này chia sẻ chung giữa Writer Process và Reader Process.

Giả sử rằng một Writer Process đang truy xuất vào database/file trong khi đang có một Writer Process khác (đặt tên là  $W_1$ ) và  $n$  Reader Process (đặt tên là  $R_0, R_1, \dots, R_{n-1}$  theo thứ tự chờ đợi) đang chờ đợi bên ngoài để truy xuất database/file này. Chọn phát biểu đúng:

- |   |   |
|---|---|
| a. $W_1$ đang chờ trên hàng đợi của Semaphore “wrt”; $R_0$ đang chờ trên hàng đợi của Semaphore “mutex”; và $R_1, R_2, \dots, R_{n-1}$ đang chờ trên hàng đợi của “wrt” | b. $W_1$ đang chờ trên hàng đợi của Semaphore “wrt”; và $R_0, R_1, \dots, R_{n-1}$ đang chờ trên hàng đợi của “mutex” |
| c. $W_1$ đang chờ trên hàng đợi của Semaphore “wrt”; $R_0$ đang chờ trên hàng đợi của Semaphore “wrt”; và $R_1, R_2, \dots, R_{n-1}$ đang chờ trên hàng đợi của “mutex” | d. $W_1$ đang chờ trên hàng đợi của Semaphore “mutex”; và $R_0, R_1, \dots, R_{n-1}$ đang chờ trên hàng đợi của “wrt” |

Câu 5.

Khi dùng bộ nhớ ảo và cần thay thế trang, thuật toán nào dưới đây dùng thời điểm trang đã được sử dụng để xem xét việc chọn trang thay thế?

- |            |                    |
|------------|--------------------|
| a. FIFO    | b. LRU             |
| c. Optimal | d. Tất cả đều đúng |

Câu 6.

Xét một không gian địa chỉ luận lý có 32 trang, mỗi trang có kích thước 1KByte. Ánh xạ vào bộ nhớ vật lý có 16 khung trang. Địa chỉ luận lý và địa chỉ vật lý gồm bao nhiêu bit?

- |  |  |
|--|--|
| a. Địa chỉ luận lý cần 14 bits, địa chỉ vật lý cần 15 bits | b. Địa chỉ luận lý cần 24 bits, địa chỉ vật lý cần 25 bits |
| c. Địa chỉ luận lý cần 15 bits, địa chỉ vật lý cần 14 bits | d. Địa chỉ luận lý cần 25 bits, địa chỉ vật lý cần 24 bits |

Câu 7.

Cho 1 hệ thống có 4 tiến trình  $P_1, P_2, P_3, P_4$  và 3 loại tài nguyên  $R_1$  (có 3 thực thể),  $R_2$  (có 2 thực thể)  $R_3$  (có 2 thực thể).  $P_1$  giữ 1 thực thể  $R_1$  và yêu cầu 1 thực thể  $R_2$ ;  $P_2$  giữ 2 thực thể  $R_2$  và yêu cầu 1 thực thể  $R_1$  và 1 thực thể  $R_3$ ;  $P_3$  giữ 1 thực thể  $R_1$  và yêu cầu 1 thực thể  $R_2$ ;  $P_4$  giữ 2 thực thể  $R_3$  và yêu cầu 1 thực thể  $R_1$ . Có bao nhiêu chuỗi an toàn cho hệ thống trên?

- |      |      |
|------|------|
| a. 1 | b. 2 |
| c. 3 | d. 4 |

Câu 8.

Cho biết kích thước của page và frame là 1KB. Hỏi địa chỉ ảo là bao nhiêu để có được địa chỉ vật lý là 4100 sau khi ánh xạ (bảng ánh xạ như hình vẽ)

- |         |         |
|---------|---------|
| a. 4100 | b. 1024 |
|---------|---------|

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

Page Table

- |         |         |
|---------|---------|
| c. 1028 | d. 5124 |
|---------|---------|

Câu 9.

Mô hình quản lý bộ nhớ nào trong các mô hình sau đây gây ra hiện tượng phân mảnh ngoại?

- |   |                       |
|---|-----------------------|
| a. Phân vùng cố định với kích thước từng vùng bằng nhau | b. Phân vùng động     |
| c. Phân vùng cố định với kích thước từng vùng khác nhau | d. Cả a và c đều đúng |

Câu 10.

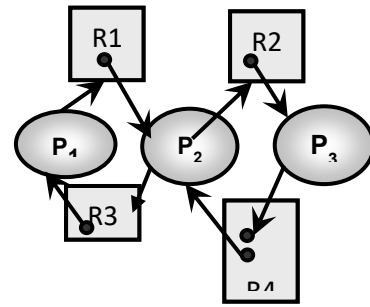
Ưu điểm nào sau đây **KHÔNG** phải là ưu điểm của bộ nhớ ảo?

- |  |  |
|--|--|
| a. Số lượng process trong bộ nhớ nhiều hơn                                       | b. Giúp hệ điều hành quản lý tiến trình dễ dàng hơn            |
| c. Một process có thể thực thi ngay cả khi kích thước của nó lớn hơn bộ nhớ thực | d. Giảm nhẹ việc quản lý về giới hạn memory cho lập trình viên |

Câu 11.

Với đồ thị cấp phát tài nguyên như bên dưới (RAG), hỏi có deadlock xảy ra hay không? (với điều kiện mutual exclusion và no preemption được đảm bảo).

- a. Có
- b. Không
- c. Không chắc



Câu 12.

Xét 2 process P1 và P2 thực thi đồng thời, với P1 {A1, B1, C1}; P2 {A2, B2, C2} (tức P1 phải hoàn tất lần lượt 3 lệnh A1, B1, C1; P2 phải hoàn tất lần lượt 3 lệnh A2, B2, C2; và các lệnh này đều là đơn nguyên (atomic)).

Các lệnh trong P1 và P2 thực thi với điều kiện: A2 chỉ được thực thi sau khi A1 hoàn tất (điều kiện 1), B2 chỉ được thực thi sau khi B1 hoàn tất (điều kiện 2), và C2 chỉ được thực thi sau khi C1 hoàn tất (điều kiện 3).

Giả sử sử dụng 3 biến Semaphore (S1, S2, S3) dùng chung cho P1 và P2 để đồng bộ yêu cầu trên (Semaphore có cấu trúc dữ liệu và hai hàm wait và signal hoạt động như trong phần phụ lục). Biết giá trị value của các Semaphore được khởi tạo bằng 0; và S1, S2, S3 lần lượt dùng tương ứng cho các điều kiện 1, 2 và 3.

Đặt các lệnh wait(), signal() của các Semaphore vào dấu “.....” bên dưới sao cho hợp lý để đồng bộ theo yêu cầu đề bài (chỗ nào không cần điền sinh viên có thể bỏ trống)

**Process P1**

```
{
    .....
    A1
    signal (S1)
    B1
    signal (S2)
    C1
    signal (S3)
}
```

**Process P2**

```
{
    wait (S1)
    A2
    wait (S2)
    B2
    wait (S3)
    C2
    .....
}
```

Câu 13.

Địa chỉ lệnh và dữ liệu được chuyển đổi thành địa chỉ thực có thể xảy ra tại thời điểm nào?

- a. Compile time
- b. Execution time
- c. Load time
- d. Đáp án a, b, c đều đúng

Câu 14.

Một chương trình liên kết với một số thư viện tại thời điểm thực thi của chương trình được gọi là?

- a. Static linking
- b. Đáp án a và c đều đúng
- c. Dynamic linking
- d. Đáp án a và c đều sai

Câu 15.

Khi phát hiện hệ thống có deadlock xảy ra, thì giải pháp nào trong các giải pháp sau **KHÔNG** được hệ điều hành chọn để phục hồi hệ thống?

- a. Thực hiện giải thuật Banker
- b. Lấy lại tài nguyên từ một hay nhiều tiến trình
- c. Chấm dứt một hay nhiều tiến trình
- d. Báo cho người vận hành

## II - Tự Luận (4 điểm):

1. (2 điểm) Xét hệ thống tại thời điểm  $t_0$  có 5 tiến trình: P1, P2, P3, P4, P5; và 4 loại tài nguyên: R1, R2, R3, R4. Xét trạng thái hệ thống như sau:

Process	Allocation				Max			
	R1	R2	R3	R4	R1	R2	R3	R4
P1	1	1	2	3	1	1	4	3
P2	3	1	1	1	3	8	6	1
P3	1	1	4	5	7	7	6	7
P4	3	4	6	5	5	4	6	7
P5	1	4	4	3	1	7	6	3

Available			
R1	R2	R3	R4
3	2	3	1

Tại thời điểm  $t_1$ , biết hệ điều hành giải quyết deadlock theo kiểu tránh deadlock (Deadlock Avoidance) và dùng giải thuật Banker để kiểm tra độ an toàn của hệ thống. Hỏi nếu tiến trình P3 yêu cầu thêm tài nguyên (2, 1, 2, 1), hệ thống có đáp ứng không và giải thích tại sao?

### Gợi ý giải:

Tìm ma trận Need:

Process	Allocation				Max				Need				Available			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	1	1	2	3	1	1	4	3	0	0	2	0	3	2	3	1
P2	3	1	1	1	3	8	6	1	0	7	5	0				
P3	1	1	4	5	7	7	6	7	6	6	2	2				
P4	3	4	6	5	5	4	6	7	2	0	0	2				
P5	1	4	4	3	1	7	6	3	0	3	2	0				

- B1: Request (2,1,2,1) < Need P3 (6,6,2,2)
- B2: Request (2,1,2,1) < Available (3,2,3,1)
- Cập nhật lại hệ thống ta có:

Process	Allocation				Max				Need				Available				Finish
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	
P1	1	1	2	3	1	1	4	3	0	0	2	0	1	1	1	0	
P2	3	1	1	1	3	8	6	1	0	7	5	0					
P3	3	2	6	6	7	7	6	7	4	5	0	1					
P4	3	4	6	5	5	4	6	7	2	0	0	2					
P5	1	4	4	3	1	7	6	3	0	3	2	0					

Vì ma trận Available mới không thể cấp phát đủ cho 1 tiến trình nào trong 5 tiến trình để thực thi vì vậy hệ thống không an toàn khi P3 yêu cầu thêm tài nguyên

2. (2 điểm) Giả sử một tiến trình được phát 4 khung trang (frame) trong bộ nhớ vật lý và 8 trang (page) trong bộ nhớ ảo. Biết ban đầu, khi nạp tiến trình vào, 4 frame trên bộ nhớ vật lý này đang trống. Process truy xuất 8 trang (1, 2, 3, 4, 5, 6, 7, 8) trong bộ nhớ ảo theo thứ tự như sau:

1 2 4 6 8 4 3 6 4 4 3 2 5 6 7 3 2 5 6 7 1 2 3 4 5

Vẽ bảng minh họa thuật toán và tính số lỗi trang (page fault) khi:

- Tiến trình truy xuất chuỗi bộ nhớ trên và hệ điều hành thay trang theo giải thuật OPT.
- Tiến trình truy xuất chuỗi bộ nhớ trên và hệ điều hành thay trang theo giải thuật LRU.

**Gợi ý giải:**

- a. Giải thuật OPT

1	2	4	6	8	4	3	6	4	4	3	2	5	6	7	3	2	5	6	7	1	2	3	4	5
1	1	1	1	8	8	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	5
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	6	6	1	1	1	1	1
			6	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7
*	*	*	*	*		*						*		*				*		*			*	*

Tổng số lỗi trang là: 12

- b. Giải thuật LRU

1	2	4	6	8	4	3	6	4	4	3	2	5	6	7	3	2	5	6	7	1	2	3	4	5
1	1	1	1	8	8	8	8	8	8	8	2	2	2	2	3	3	3	3	7	7	7	7	4	4
	2	2	2	2	2	3	3	3	3	3	3	3	3	7	7	7	7	6	6	6	6	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	6	6	6	6	5	5	5	5	2	2	2	2
			6	6	6	6	6	6	6	6	6	5	5	5	5	2	2	2	2	1	1	1	1	5
*	*	*	*	*		*					*	*	*	*	*	*	*	*	*	*	*	*	*	*

Tổng số lỗi trang là: 20

---Hết---

Duyệt đề của Trưởng Khoa/Bộ môn

Giảng viên ra đề

**Phụ lục:**

Cấu trúc dữ liệu và hai hàm wait và signal của Semaphore sử dụng trong đề thi:

```
typedef struct {  
    int value;  
    struct process *L; /* process queue */  
} semaphore;
```

<pre><b>void wait(semaphore *S) {</b>     <i>S.value--;</i>     <i>if (S.value &lt; 0) {</i>         <i>add this process to S.L;</i>         <i>block();</i>     <i>}</i> <b>}</b></pre>	<pre><b>void signal(semaphore *S) {</b>     <i>S.value++;</i>     <i>if (S.value &lt;= 0) {</i>         <i>remove a process P from S.L;</i>         <i>wakeup(P);</i>     <i>}</i> <b>}</b></pre>
--	---