

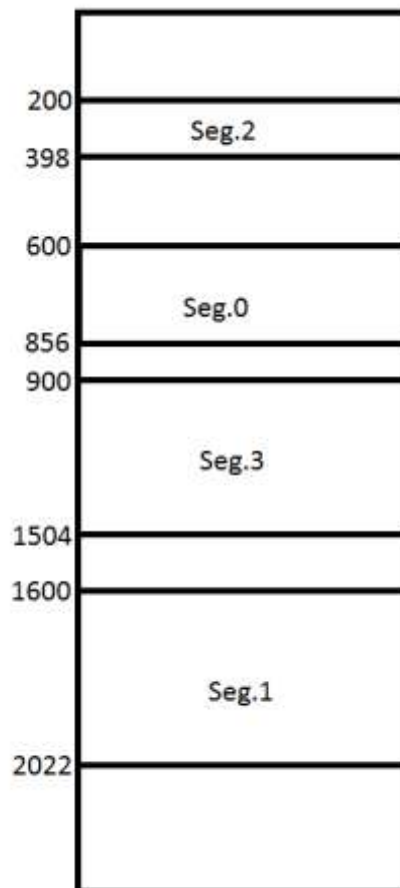
Ôn tập Hệ điều hành

1 Quản lý Bộ nhớ - 2 điểm (dạng 1)

Cho hệ thống phân đoạn có bảng đoạn như sau:

Segmentation	Base (Starting Address)	Limit (Length) (bytes)
0	600	256
1	1600	422
2	200	198
3	900	604

a) Hãy vẽ vùng nhớ vật lý ứng với bảng đoạn trên.



b) Hãy cho biết các địa chỉ logic dưới đây sẽ chuyển thành địa chỉ vật lý nào và lỗi bảo vệ có xảy ra không: (0, 198), (2, 156), (1, 530), (3, 444), (0, 222).

- (0, 198) $\rightarrow 600 + 198 = 798$
- (2, 156) $\rightarrow 200 + 156 = 356$
- (1, 530) \rightarrow lỗi truy cập ngoài phân đoạn vì $530 > 422$

- (3, 444) → 900 + 444 = 1344
- (0, 222) → 600 + 256 = 856

2 Quản lý Bộ nhớ - 2 điểm (dạng 2)

Một hệ thống máy tính với bộ nhớ chính có kích thước 800MB. Hệ thống sử dụng địa chỉ logic 32 bit. Kích thước trang được sử dụng là 16KB. Yêu cầu xác định các thông số sau:

- Cho biết số bit dùng cho địa chỉ offset.
- Số khung trang vật lý.
- Số trang logic trong không gian tiến trình.
- Cho địa chỉ logic 23032, yêu cầu đổi sang dạng <p, d>.

ý 1	Số bit dùng cho địa chỉ offset: Là số bit cần dùng để mô tả tất cả các địa chỉ trong một trang. Kích thước một trang: $16K = 2^4 \cdot 2^{10} B = 2^{14} B \rightarrow$ Số bit cần dùng là 14 bit.	0.5
ý 2	Số khung trang vật lý: $\frac{\text{Kích thước bộ nhớ vật lý}}{\text{Kích thước trang}} = \frac{800 \text{ MB}}{16 \text{ KB/trang}} = \frac{800 * 2^{20} B}{16 * 2^{10} B/\text{trang}} = 50 * 2^{10} = 51200 \text{ trang}$	0.5
ý 3	Số trang logic trong không gian tiến trình: $\frac{\text{Kích thước không gian tiến trình}}{\text{Kích thước trang}} = \frac{2^{32} B}{16 \text{ KB/trang}} = \frac{2^{32} B}{2^{14} B/\text{trang}} = 2^{18} \text{ trang}$	0.5
ý 4	Do kích thước trang là $2^{14} = 16384$, lấy 23032 chia cho 16384 được 1 dư 6648. 23032 được đổi thành <p = 1, d = 6648>.	0.5

3 Quản lý Bộ nhớ ảo – 2 điểm

Hãy thực hiện các giải thuật thay thế trang LRU (trang lâu nhất chưa sử dụng bị thay ra), giải thuật tối ưu (trang ít sử dụng nhất trong tương lai bị thay ra) cho chuỗi truy xuất trang sau (vẽ lại và điền kết quả vào bảng sau):

Trang	2	5	2	7	3	6	4	6	5	4	2	3	1
Khung trang (k = 3)													
Lỗi													

Giải:

ý 1	giải thuật thay thế trang LRU													
	Trang	2	5	2	7	3	6	4	6	5	4	2	3	1
	Khung trang (k = 3)	2	2	2	2	2	6	6	6	6	6	2	2	2
			5	5	5	3	3	3	3	5	5	5	3	3
					7	7	7	4	4	4	4	4	4	1
	Lỗi	*	*		*	*	*	*		*		*	*	*
	Số trang bị lỗi: 10													
ý 2	giải thuật tối ưu													
	Trang	2	5	2	7	3	6	4	6	5	4	2	3	1

Khung trang (k = 3)	2	2	2	2	2	2	2	6	6	6	2	3	1
	5	5	5	5	5	5	5	5	5	5	5	5	5
				7	3	4	4	4	4	4	4	4	4
Lỗi	*	*		*	*	*		*			*	*	*

Số trang bị lỗi: 9

4 Điều phối CPU – 2 điểm

Hãy vẽ sơ đồ điều phối CPU (sử dụng chiến lược SJF không độc quyền), và tài nguyên R1, R2 (sử dụng chiến lược FIFO) cho các tiến trình.

Bài tập 1:

Tiến trình	Thời điểm vào Ready list	Thời gian CPU lần 1	IO lần 1		Thời gian CPU lần 2	IO lần 2	
			Thời gian	Thiết bị		Thời gian	Thiết bị
P1	0	6	5	R1	2	2	R1
P2	2	1	7	R2	3	4	R1
P3	8	4	3	R1	2	3	R2
P4	11	2	7	R2	1	4	R2

Giải

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	P1	P1	P2	P1	P1	P1	P1		P3	P3	P3	P3	P4	P4	P1	P1	P3	P3	P2
R1								P1	P1	P1	P1	P1	P3	P3	P3		P1	P1	
R2				P2	P2	P2	P2	P2	P2	P2					P4	P4	P4	P4	P4

19	20	21	22	23	24	25	26	27
P2	P2	P4						
		P2	P2	P2	P2			
P4	P4	P3	P3	P3	P4	P4	P4	P4

$T=0$: p1(6) → P1 vào

$T=2$: p1(4), p2(1) → P2 vào

$T=7$: null

$T=8$: p3(4) → p3 vào

$T=10$: p3(2), p2(3) → p3 giữ nguyên

$T=11$: p3(1), p2(3), p4(2) → p3 giữ nguyên

$T=12$: p2(3), p4(2), p1(2) → p4 vào (do vào danh sách ready list trước)

$T=14$: p2(3), p1(2) → p1 vào

$T=15$: p2(3), p1(1), p3(2) → p1 giữ nguyên

$T=16$: p2(3), p3(2) → p3 vào

$T=18$: p2(3) → vào, p1 kết thúc

$T=21$: p4(1) → vào

$T=22$: null.

$T=24$: P3 kết thúc; $T=25$: P2 kết thúc; $T=28$: P4 kết thúc.

Bài tập 2:

Tiến trình	Thời điểm vào Ready list	Thời gian CPU lần 1	IO lần 1		Thời gian CPU lần 2	IO lần 2	
			Thời gian	Thiết bị		Thời gian	Thiết bị
P1	0	9	5	R1	2	2	R1
P2	2	1	8	R2	3	0	null
P3	6	5	4	R1	2	3	R1
P4	12	2	7	R2	1	4	R2

5 Deadlock – 2 điểm

Một hệ thống có 3 loại tài nguyên (A, B, C) và 5 tiến trình (P1, P2, P3, P4, P5) kèm theo các thông số được mô tả trong bảng sau.

5.1 Bài tập 1:

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P1	3	0	1	9	7	4	6	2	2
P2	3	2	1	8	5	3			
P3	2	1	3	6	3	4			
P4	0	3	2	9	6	3			
P5	1	1	2	7	4	5			

Giả sử Tiến trình P2 yêu cầu thêm tài nguyên là (1, 0, 1). Sử dụng giải thuật Banker, cho biết có thể thực hiện yêu cầu cấp phát tài nguyên này hay không?

Giải:

Bước 1: Kiểm tra Request (P2) \leq Available
 $(1, 0, 1) \leq (6, 2, 2)$ True

Yêu cầu là hợp lệ.

Thử kiểm tra việc cấp phát có an toàn không

Bước 2: Work = Available – Request(P2) = (6, 2, 2) – (1, 0, 1) = (5, 2, 1)
 Cập nhật Allocation cho P2 = (3, 2, 1) + (1, 0, 1) = (4, 2, 2)

Bước 3: Tính Need = Max - Allocation

P1: (9, 7, 4) – (3, 0, 1) = (6, 7, 3)

P2: (8, 5, 3) – (4, 2, 2) = (4, 3, 1)

P3: (6, 3, 4) – (2, 1, 3) = (4, 2, 1)

P4: (9, 6, 3) – (0, 3, 2) = (9, 3, 1)

P5: (7, 4, 5) – (1, 1, 2) = (6, 3, 3)

Bước 4: Xác định Need (i) $<$ Work

Với P1: (6, 7, 3) \leq (5, 2, 1) -> False

Với P2: (4, 3, 1) \leq (5, 2, 1) -> False

Với P3: (4, 2, 1) \leq (5, 2, 1) -> True

Work = Work + Allocation (P3) = (5, 2, 1) + (2, 1, 3) = (7, 3, 4)

Với P4: (9, 3, 1) \leq (7, 3, 4) -> False

Với P5: (6, 3, 3) \leq (7, 3, 4) -> True

Work = Work + Allocation (P5) = (7, 3, 4) + (1, 1, 2) = (8, 4, 6)

Với P1: (6, 7, 3) \leq (8, 4, 6) -> False

Với P2: (4, 3, 1) <= (8, 4, 6) -> True

Work = Work + Allocation (P2) = (8, 4, 6) + (4, 2, 2) = (12, 6, 8)

Với P4: (9, 3, 1) <= (12, 6, 8) -> True

Work = Work + Allocation (P4) = (12, 6, 8) + (0, 3, 2) = (12, 9, 10)

Với P1: (6, 7, 3) <= (12, 9, 10) -> True

Kết thúc tất cả process.

Vậy ta có chuỗi cấp phát an toàn là <P3, P5, P2, P4, P1>. Suy ra có thể thực hiện yêu cầu cấp phát tài nguyên cho P2.

5.2 Bài tập 2:

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P1	2	1	0	7	5	3	3	3	2
P2	3	0	2	9	0	2			
P3	0	0	2	4	3	3			
P4	1	0	0	3	2	2			
P5	1	1	1	2	2	2			

Tiến trình P3 yêu cầu tài nguyên là (A:3, B:1, C:0). Sử dụng giải thuật Banker, cho biết có thể thực hiện yêu cầu cấp phát tài nguyên này cho P3 hay không? Tại sao?

6 Đồng bộ hóa tiến trình – Semaphore

6.1 Bài 1

Đồng bộ hoạt động của một xưởng sản xuất ô tô sử dụng nhiều tiểu trình đồng thời như sau:

TaoBanhXe() // Tiến trình TaoBanhXe tạo 1 bánh xe

```
{  
    printf("Banh xe \n");  
}
```

TaoKhungXe() // Tiến trình TaoKhungXe tạo 1 khung xe

```
{  
    printf("Khung xe \n");  
}
```

LapRap() // Tiến trình LapRap tạo 1 chiếc xe khi chờ đủ 4 bánh xe và 1 khung xe

```
{  
    printf("Chiec xe \n");  
}
```

Giải

a	Khởi tạo Semaphore Semaphore mutexBanhXe, mutexKhungXe, mutexLapRap;	0.5
	Đồng bộ tiến trình TaoBanhXe TaoBanhXe() <pre> { while(1) { down(mutexBanhXe); printf("Banh xe\n"); up(mutexLapRap); } } </pre>	0.5
	Đồng bộ tiến trình TaoKhungXe TaoKhungXe() <pre> { while(1) { down(mutexKhungXe); printf("Khung xe\n"); up(mutexLapRap); } } </pre>	0.5
b	Đồng bộ tiến trình LapRap LapRap() <pre> { while(1) { down(mutexLapRap); down (mutexLapRap); down (mutexLapRap); down (mutexLapRap); down (mutexLapRap); printf("Chiec xe\n"); up(mutexBanhXe); up(mutexBanhXe); up (mutexBanhXe); up (mutexBanhXe); up (mutexKhungXe); } } </pre>	0.5

6.2 Bài 2

Một chiếc thuyền chỉ đưa khách qua sông khi có đủ 4 người trên thuyền. Hãy đồng bộ hoạt động đưa khách qua sông dựa vào nhiều tiến trình khách lên thuyền :

```
KhachLenThuyen() // Tiến trình một khách lên thuyền
```

```
{
```

```
    printf("LenThuyen \n");
```

```

}
QuaSông() // Tiến trình thuyền chuyển động đưa khách qua sông
{
    printf("Đưa khách qua sông \n");
}

```

Giải

Khởi tạo Semaphore Semaphore mutexLT, mutexQS;	0.5
Đồng bộ tiến trình lên thuyền KháchLenThuyen KháchLenThuyen () <pre> { while(1) { down(mutexLT); printf("LenThuyen \n"); up(mutexQS); } } </pre>	0.75
Đồng bộ tiến trình QuaSông QuaSông () <pre> { while(1) { down(mutexQS); down(mutexQS); down(mutexQS); down(mutexQS); printf("Đưa khách qua sông \n"); up(mutexLT); up(mutexLT); up(mutexLT); up(mutexLT); } } </pre>	0.75

Lưu ý:

- **Đề thi 90 phút, KHÔNG được sử dụng tài liệu.**

Chúc các bạn thi tốt

---HẾT---