Operating Systems Homework #3

電資三 胡綸宇
108200008

7.6 (a) This could safely be changed without any problems.

(b) This could have an effect on the system and introduce the possibility of deadlock as the safety of the system assumed there were a certain number of available resources.

(c) This could have an effect on the system and introduce the possibility of deadlock.

(d) This could safely be changed without any problems.

(e) This could be allowed assuming that resources were allocated to new processes such that the system does not enter an unsafe state.

(f) This could safely be changed without any problems.

7.13

	Allocation				Max				Available				Need			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_0	2	0	0	1	4	2	1	2	3	3	2	1	2	2	1	1
P_1	3	1	2	1	5	2	5	2					2	1	3	1
P_2	2	1	0	3	2	3	1	6					0	2	1	3
P_3	1	3	1	2	1	4	2	4					0	1	1	2
P_4	1	4	3	2	3	6	6	5					2	2	3	3

(a) P_0 $(2,2,1,1) \leq (3,3,2,1)$, work $= (3,3,2,1) + (2,0,0,1) = (5,3,2,2)$

P_1 $(2,1,3,1) > (5,3,2,2)$

P_2 $(0,2,1,3) > (5,3,2,2)$

P_3 $(0,1,1,2) \leq (5,3,2,2)$, work $= (5,3,2,2) + (1,3,1,2) = (6,6,3,4)$

P_4 $(2,2,3,3) \leq (6,6,3,4)$, work $= (6,6,3,4) + (1,4,3,2) = (7,10,6,6)$

P_1 $(2,1,3,1) \leq (7,10,6,6)$, work $= (7,10,6,6) + (3,1,2,1) = (10,11,8,7)$

P_2 $(0,2,1,3) \leq (10,11,8,7)$, work $= (10,11,8,7) + (2,1,0,3) = (12,12,8,10)$

System is in safe state, $\langle P_0, P_3, P_4, P_1, P_2 \rangle$

(b) $(1,1,0,0) \leq (3,3,2,1)$, request can be granted immediately.

(c) $(0,0,2,0) \leq (3,3,2,1)$, request can be granted immediately.

7.15

```
semaphore ok_to_cross = 1

void enter_bridge() {
      P(ok_to_cross);
}

void exit_bridge() {
     V(ok_to_cross);
}
```

8.1

Internal fragmentation: In the case where memory is split into mount-sized blocks, when the allocated memory is larger than requested memory, the difference is called internal fragmentation.

External fragmentation: When you have a sufficient quantity of area within the memory, external fragmentation occurs when you apply a first-fit or best-fit memory allocation strategy.

8.9 Paging requires more memory overhead to maintain the translation structures. Paging also requires one entry per page, which provides the physical address of the page.

Segmentation requires a register to maintain the base of the segment and another to maintain the extent of the segment.

8.16 (a) 2^{20} entries

(b) $\frac{512MB}{4KB} = 128K$ entries

9.8

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

(a) LRU → 18 PF

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

(b) FIFO → 19 PF

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

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

(c) Optimal → 13 PF

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

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

9.11 Assume demanding paging with 3 frames:

1, 1, 2, 3, 4, 1 → LFU has less page faults than LRU

1, 1, 2, 3, 4, 2 → LRU has less page faults than LFU

9.17

(a) i. Initial value is 0

ii. When a new page is associated with a frame

iii. When pages associated with that frame is not needed

iv. The frame with the smallest counter is replaced, and the first entered is replace with same counters.

(b) 14 page faults

(c) 11 page faults

9.19

A1. Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault.

A2. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming.

A3. It can be eliminated by reducing the level of multiprogramming.