7.6

(a)Increase new resource Available will be safe.

(b)Decrease Available resource by removing it from the system: will affect on the system and introduce the possibility of deadlock because of changing the number of resources.

(c)Increase Max for the one process (the process needs more resources than allowed,it may want more.):this will affect the system and introduce the possibility of deadlock.

(d)Decrease Max for one process (the process decides it does not need that many resources): will be safe.

(e)Increase the number of process: If new resources allocated to the new process(es) will not enter the system to an unsafe state.

(f)Decrease the number of processes: will be safe.

7.13

|  |  |  |  |
| --- | --- | --- | --- |
|  | Allocation A B C D | Max A B C D | Available  A B C D |
| P0 | 2 0 0 1 | 4 2 1 2 | 3 3 2 1 |
| P1 | 3 1 2 1 | 5 2 5 2 |  |
| P2 | 2 1 0 3 | 2 3 1 6 |  |
| P3 | 1 3 1 2 | 1 4 2 4 |  |
| P4 | 1 4 3 2 | 3 6 6 5 |  |

need=Max-Allocation

A B C D

P0 2 2 1 1

P1 2 1 3 1

P2 0 2 1 3

P3 0 1 1 2

P4 2 2 3 3

Step1找出P0的need符合available後先加入 safe sequence把他free調

Safe sequence:P0,

Step2:available=available+allocation

A:3+2=5

B:3+0=3

C:2+0=2

D:1+1=2

Step3找到P3的need符合available後先加入 safe sequence把他free調

Safe sequence:P0,P3

Step4:available=available+allocation

A:5+1=6

B:3+3=6

C:2+1=3

D:2+2=4

Step5找到P2的need符合available後先加入 safe sequence把他free調

Safe sequence:P0,P3,P2

Step6:available=available+allocation

A:6+2=8

B:6+1=7

C:3+0=3

D:4+3=7

Step7找到P4的need符合available後先加入 safe sequence把他free調

Safe sequence:P0,P3,P2,P4

Step8:available=available+allocation

A:8+1=9

B:7+4=11

C:3+3=6

D:7+2=9

Step9找到P1的need符合available後先加入 safe sequence把他free調

Safe sequence:P0,P3,P2,P4,P1

Step10:available=available+allocation

A:9+3=12

B:11+1=12

C:6+2=8

D:9+1=10

1. 因為有依序完全跑完P0,P3,P2,P4,P1所以是安全的
2. If request (P1)=(1,1,0,0) arrives:P1 allocation=4,2,2,1

Enter another safe state(Safe sequence:P0,P3,P2,P4,P1)

P1 need=1,0,3,1

It’s available=0,4,2,1

進去跑之後發現need 不符合available，所以會產生deadlock。

Ans:不同意

1. If request (P4)=(0,0,2,0):P4 allocation=1,4,5,2

P4 need=2,2,1,3

It’s available=3,3,0,1

進去跑之後發現need 不符合available，所以會產生deadlock。

Ans:不同意

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 occurs when memory is divided into fixed-sized partitions. External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.

8.9

Paging requires more memory overhead to maintain the translation structures. Segmentation requires just two registers per segment: one to maintain the base of the segment and the other to maintain the extent of the segment. Paging on the other hand requires one entry per page, and this entry provides the physical address in which the page is located.

8.16

(a)In a conventional single-level page table.

2^11<4000<2^12

So we need 12 out of 32-bit logical address for the offset. Then we have 32-12=20 bits left for the page number.

Therefore, 2^20=1048576 entries.

(b)

Size of physical address space=No. of frames X frame size

Where frame size=page size

In paging in general,(No. of frames =No. of pages)

In inverted paging only =No. of frames X 2^12

No. of frames = 2^29/2^12=2^17

Number of entries = 2^17=131072/1024=128K entries.

9.8

(a) LRU (18 page faults): 7; 7 2; 7 2 3; 1 2 3; 1 2 5; 3 2 5; 3 4 5; 3 4 6; 7 4 6; 7 1 6; 7 1 0; 5 1 0; 5 4 0; 5 4 6; 2 4 6; 2 3 6; 2 3 0; 1 3 0

(b) FIFO (17 page faults): 7; 7 2; 7 2 3; 1 2 3; 1 5 3; 1 5 4; 6 5 4; 6 7 4; 6 7 1; 0 7 1; 0 5 1; 0 5 4; 6 5 4; 6 2 4; 6 2 3; 0 2 3; 0 1 3

(c) OPT (13 page faults): 7; 7 2; 7 2 3; 1 2 3; 1 5 3; 1 5 4; 1 5 6; 1 5 7; 1 5 0; 1 4 0; 1 6 0; 1 2 0; 1 3 0

9.11

Consider the following sequence of memory accesses in a system that can hold four pages in memory: 1 1 2 3 4 5 1. When page 5 is accessed, the least frequently used page-replacement algorithm would replace a page other than 1, and therefore would not incur a page fault when page1 is accessed again. On the other hand, for the sequence “1 2 3 4 5 2,” the least recently used algorithm performs better

9.17

a. Define a page-replacement algorithm addressing the problems of:

i. Initial value of the counters-0.

ii. Counters are increased-whenever a new page is associated with that frame.

iii. Counters are decreased-whenever one of the pages associated with that frame is no longer required.

iv. How the page to replaced is selected—find a frame with the smallest counter. Use FIFO for breaking ties.

b. 14 page faults.

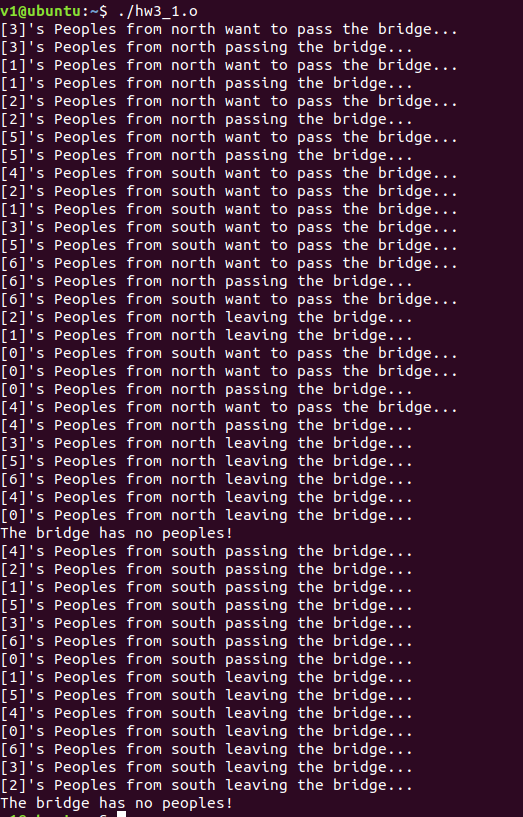
c. 11 page faults.

9.19

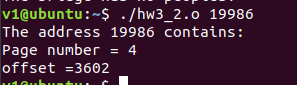
Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

Program

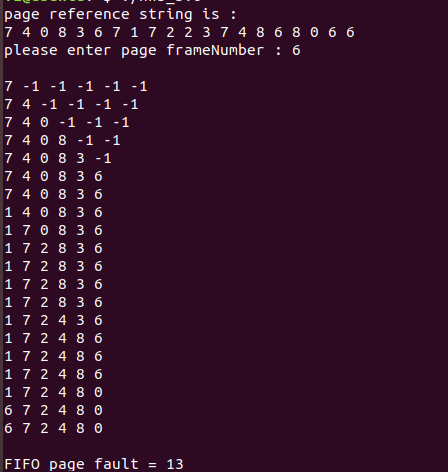
7.17

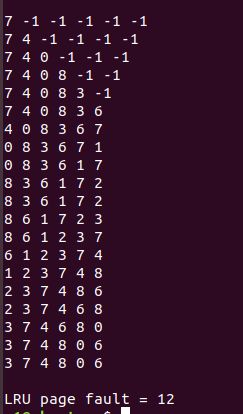


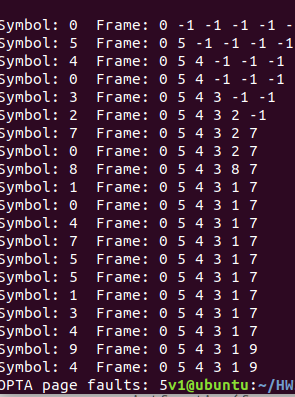
8.25



9.26







Teamproject

CH7

