小测1 概念+process schedule

实在没考古到，就是简单的process schedule

小测2 synchronize+deadlock

There are 4 processes sharing 18 resources. For dynamic avoid deadlock, the number of resources which every process could request at most is ()

A）4　　B）5　　C）6　　D）7

> B
>
> 4(M-1)+1 ≤ 18
> M < 5.25

When using a counting semaphore to control the usage of 5 printers shared among many processes, which value could not occur for the counting semaphore?

A. -6　　B.-5　　C. 5　　D. 6

> D
>
> 要小于等于设定的usage值

Assume there are 3 kinds of resources (A B C) and 5 processes (P1, P2, P3, P4, P5) in a system. The amounts of resource A are 17, resource B are 5, resource C are 20. The system state is as follows table at time T0. Please avoid deadlock by using Bank's algorithm.

process	Max used	allocated	left
	A B C	A B C	A B C
P_1	5 5 9	2 1 2	2 3 3
P_2	5 3 6	4 0 2	
P_3	4 0 11	4 0 5	
P_4	4 2 5	2 0 4	
P_5	4 2 4	3 1 4	

Please Answer:

(a) Is it safety when it is in time T_0? If yes, write down the safety order; if not, write down why?

(b) If P_1 now requests resource (1, 1, 2) at time T_0, could you allocate them? If yes, write down the new safety order; if not, write down why?

> 银行家算法
> max为总共需要，allocated为已经分配的，left为现有资源，先求need矩阵，为max-allocated
> 然后看如果need小于等于available就可以分配

a.

	Max	Allocation	Need	left
P1	5 7 9	2 1 2	3 4 7	2 3 3
P2	5 3 6	4 0 2	1 3 4	
P3	4 0 11	4 0 5	0 0 6	
P4	4 2 5	2 0 4	2 2 1	
P5	4 2 4	3 1 4	1 1 0	

	work	need	Allocation	work+allocation
P4	2 3 3	2 2 1	2 0 4	4 3 ~~4~~7
P2	4 3 ~~8~~7	1 3 4	4 0 2	8 3 ~~8~~9
P3	8 3 ~~8~~9	0 0 6	4 0 5	12 3 14
P5	12 3 14	1 1 0	3 1 4	15 4 18
P1	15 4 18	3 4 7	2 1 2	17 5 20

safety order 为p4-p2-p3-p5-p1

b.同理a也是可以的

↵

4. There is an empty plate on the table. Assume there are two kinds of fruits, the apple and the orange. Every time, only one fruit could be put on the plate. There are three persons – a waiter, a male customer and a female customer – who share the plate. The waiter can put an apple or an orange on the plate one at a time. The male customer is waiting to eat an apple, while the female customer is waiting to eat an orange. The following program is a solution to the problem, please fill the

```
blank.
    begin
    empty, full1, full2:semaphore;
    empty:=_(_1__)__;
    full1=__0__;
    full2:=__0__;;
    cobegin
        processor waiter
        begin
            wait_(_empty_)_;
            if put an apple
                signal_(full1)_;
            else
                signal (full2);
        end
        processor male customer
        begin
            wait (   full1   )   ;
            get an apple to eat;
```

> semophore简单填空即可

小测3 memory

1. Which memory partition cannot cause internal fragmentation （C）
   A）fixed partitions　　B）paging
   C）segmentation　　D）segmentation with paging.

2. Which page replacement algorithm can cause Belady's Anomaly. （B）
   A）LRU.　　B）FIFO.
   C）Working Set.　　D）Optimal

3. Here we use the hybrid scheme to support virtual memory. The segment number is no more than 128, and each segment can contain at most 512 pages. If the size of the page is 4KB, how many bits should be used to indicate the position of an instruction (1 byte) in the program space? B
   A. 26　　B. 28　　C. 30　　D. 32
   7 + 9 + 12

> segement不会造成内碎片
>
> | 分配方式 | 内碎片 | 外碎片 | 跨进程共享代码 |
> | --- | --- | --- | --- |
> | 固定分区分配 fixed-size blocks | 是 | 否 | 否 |
> | 可变分区分配 variable partition allocation | 否 | 是 | 否 |
> | 页式分配 page | 是 | 否 | 是 |
> | 段式分配 segment | 否 | 是 | 是 |
> | 段页式分配 segment-page | 是 | 否 | 是 |
>
> FIFO会造成Belady's Anomaly
>
> 2^7 = 128, 2^9=512, 2^12 = 4K

4. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1952	96

What are the physical addresses for the following logical addresses [(x,y): "x" is the segment, and "y" corresponds to the relative address]?
a. 0,430
b. 1,10
c. 2,500
d. 3,112

> 看对应segment的length和base推出physical addresses

> a 219+430 = 649
> b 2300+10 = 2310
> c 500>length illegal
> d 96>length illegal

An OS uses demand paging system in memory management. Assume the capacity of the main memory which a process could be allocated to is 300 byte, which is divided into 3 frames. The process will access the following Logical address byte series: 115, 228, 120, 88, 446, 102, 321, 432, 260, 167（Attention: 115B is only equivalent to 1 page）.

Please Answer：

(a) Writing down the page-reference string.

(b) Analyze the page replacement situation and calculate the page fault frequency when LRU and FIFO algorithm is used

> a. 1 - 2 - 1 - 0 - 4 - 1 - 3 - 4 - 2 - 1
>
> b. LRU page fault = 7
>
> | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|
> | A | 1 | | * | | | * | | | 2 | |
> | B | | 2 | | | 4 | | | * | | |
> | C | | | | 0 | | | 3 | | | 1 |
>
> FIFO page fault = 7
>
> | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|
> | A | 1 | | * | | 4 | | | * | 2 | |
> | B | | 2 | | | | 1 | | | | * |
> | C | | | | 0 | | | 3 | | | |

1. Hard disk is a popular permanent storage media used in modern computer systems. Its basic storage unit is ⑯_**sector**_, whose default size is usually 512 bytes. So the hard disk could be seen as a collection of addressed ⑯s, whose ID is indicated traditionally by 3 numbers of ⑰_**cylinder** __, head and ⑯. However, except for some reserved ⑯s like MBR (Master Boot Record, the first ⑯), the entire disk is usually reorganized into ⑱__**partition**__ first, and then the ⑯s in each ⑱ are further reorganized as ⑲__**blocks**__. ⑲ is the basic unit for the file system when storing the data of a file into hard disk. To store a file, the information of available ⑲s should be known first. ⑳ _**bitmap** is the simple one of the data structures used to represent the usage status of ⑲s, in which 1 is defined for a free ⑲, and 0 is for an occupied ⑲.

   ⑯⑰⑱⑲⑳: data block, sector, track, data node, cell, frame, block, partition, segment, graph, cylinder, head, MBR, CHS, linked list, node, bit map, tree structure, matrix, page, frame

> 就是完形填空

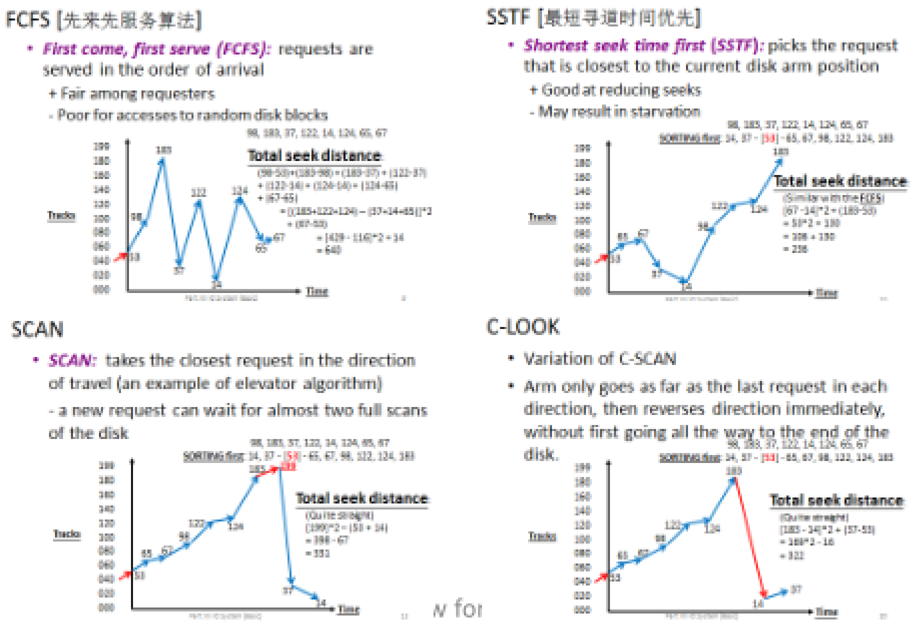
小测4 I/O + File System

- A hard disk has 40G, Its each block size is 1K，and each table entry of FAT needs 20 bits， then Its FAT (File Allocation Table) need （ ） memory space
  - A） 100M B） 120M C） 140M D） 160M
  - How about using bit map?

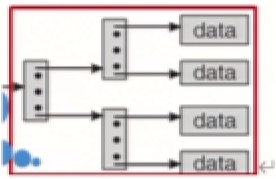
> A
>
> 40G/1K = 40*2^20
> 40*2^20*20bits = 800M bits = 100 M Bytes
>
> bit map = 40K bits = 8 K Bytes

- We illustrate them with a request queue (0-199).
  - 98, 183, 37, 122, 14, 124, 65, 67
  - After visiting 40, current Head pointer is at 53
- FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

> 很简单直接按照步骤算就行了



4. Here is a file system, which adopts multi-level index structure to support the search some records in a file. The block size is 512 bytes, and 4 bytes are used to represent the block number. Please figure out the largest size of a file when using 16 entries for 2-level index structures.

> From "The block size is 512 bytes, and 4 bytes are used to represent the block number ", we could conclude each block could contain 128 entries.
> "When suing 2-level", which means there are 2 level of internal nodes for entries (while the data nodes are finally connected with the 2nd level nodes.) as following:
>
> 
>
>
> **# of entries of a block: 512/4 = 128**
> **# of blocks by 2-level index structures: 128*128**
> **So, max size by using 16 entries for 2-level index structures: 16*128*128*512 B= 128MB**
>
> 每个block能算出可以存128个block的entry
> 第二层的block就是128个
> 第二层的entry就是128*128个
> 最终计算就是16*128*128*512 bytes = 2^4*2^7*2^7*2^9 = 2^27 bytes = 2^7 MB = 128 MB