



## # exercise 习题课

### 程序运行时间



## Exercise



One system has one CPU (single core), two independent processes, each process need 5 minutes of CPU time and 5 minutes for I/O operations. Moreover, the two I/O operations cannot run parallel. If these two processes start simultaneously (并发), what's the minimum time it will need to complete the two processes? What about average time? What about maximum time?

Gao Haichang, Software School, Xidian University 2

Minimum time=5+5=10 (1CPU while 2 I/O, 2CPU while 1 I/O), maximum time=10+10, average time=13.33

求解平均运行时间的时候需要用到CPU利用率,

$Average\ time = minimum\ time / CPU\ utilization$

$CPU\ utilization = 1 - q^n$  ( $n$ 是程序个数,  $p$ 是I/O占比)

### 虚拟内存物理内存地址关系



## Exercise

Consider a system where the virtual memory page size is 2K (2048 bytes), and main memory consists of 4 page frames. Now consider a process which requires 8 pages of storage. At some point during its execution, the page table is as shown below:

Answer the following questions:

- (1) List the virtual address ranges for each virtual page.
- (2) List the virtual address ranges that will result in a page fault.
- (3) Give the main memory (physical) addresses for each of the following virtual addresses (all numbers decimal十进制): (i) 8500, (ii) 14000, (iii) 5000, (iv) 2100.

Virtual page	Valid	Physical page
0	No	
1	No	
2	Yes	1
3	No	
4	Yes	3
5	No	
6	Yes	0
7	Yes	2

(1) 虚拟地址范围,  $page\ size = page\ frame\ size$ , page frame: the corresponding units in physical memory (via, MOS). 进程中的块, 称为页 (page), 内存中的块称为页框 (page frame或页帧, 物理页)。

virtual page	address range
0	0~2047
1	2048~4095
2	4096~6143
3	6144~8191
4	8192~10239
5	10240~12287
6	12288~14335
7	14336~15359

Virtual page	Valid	Physical page
0	No	
1	No	
2	Yes	1
3	No	
4	Yes	3
5	No	
6	Yes	0
7	Yes	2

在0, 1, 3, 5范围内的均会发生缺页中

断 (page fault)

(3)注意题目要求我们由虚拟地址求实际的物理地址。首先我们需要确定在哪一个虚拟页（virtual page），然后确定偏移量（offset），根据虚拟页和物理页的关系得出实际的物理地址。


(i)  $8500 = 2048 * 4 + 308$ , 对应虚拟页4, offset 308, 而虚拟页4对应物理页3, 故实际物理地址为  $6144 + 308 = 6452$

$14000 = 6 * 2048 + 1712$ , 对应虚拟页6, offset 1712, 而虚拟页6对应物理页0, 故实际物理地址为  $0 + 1712 = 1712$

$5000 = 2048 * 2 + 904$ , 对应虚拟页2, offset 904, 而虚拟页2, 对应物理页1, 故实际物理地址为  $2048 + 904 = 2952$

$2100 = 2048 * 1 + 52$ , 对应虚拟页1, 无效（valid显示no），没有对应的物理地址。

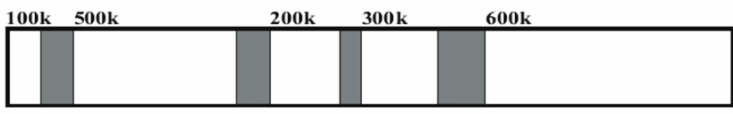
## Managing free memory



### Exercise

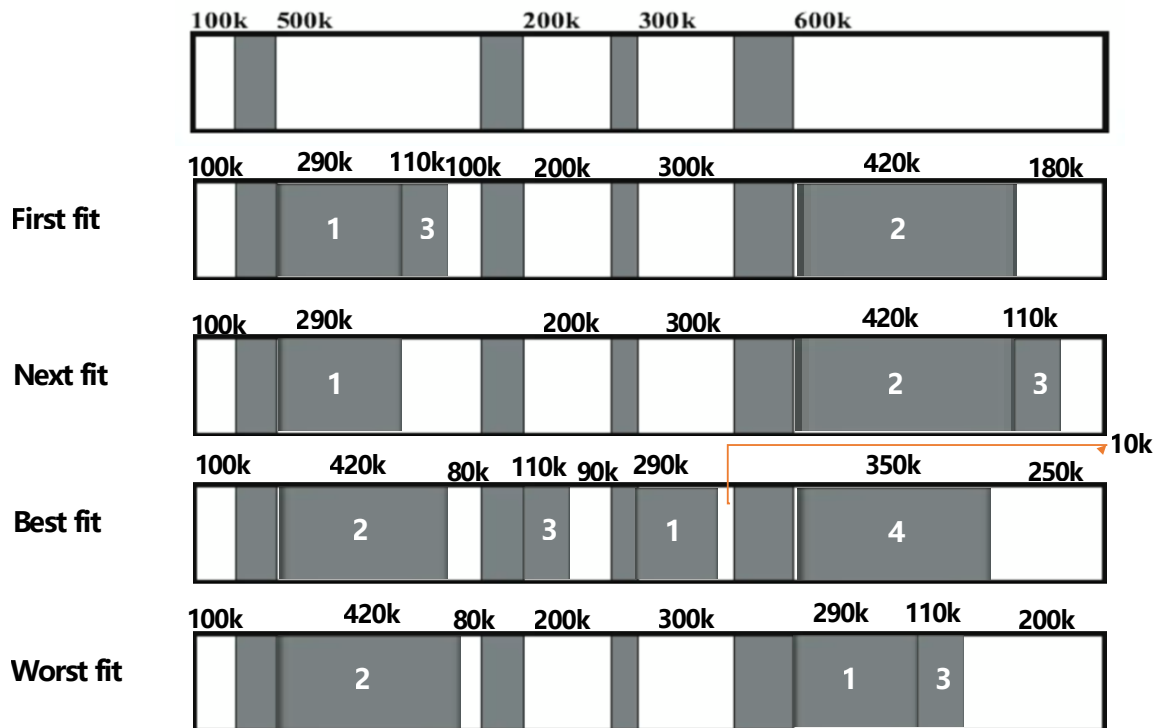
Operating Systems  
精品课程

Given memory holes (i.e., unused memory blocks) of 100K, 500K, 200K, 300K and 600K (in address order) as shown below, how would each of the first-fit, next-fit, best-fit, and worst-fit algorithms allocate memory requests of 290K, 420K, 110K and 350K (in this order)? The shaded areas are used allocated regions that are not available. Write your answer into the following figures. Use shaded areas to indicate unused memory blocks. You should write down the size of each allocated and unused memory block. (15 points)



Gao Haichang, Software School, Xidian University 4

答案如下图



## scheduling 调度



### Exercise

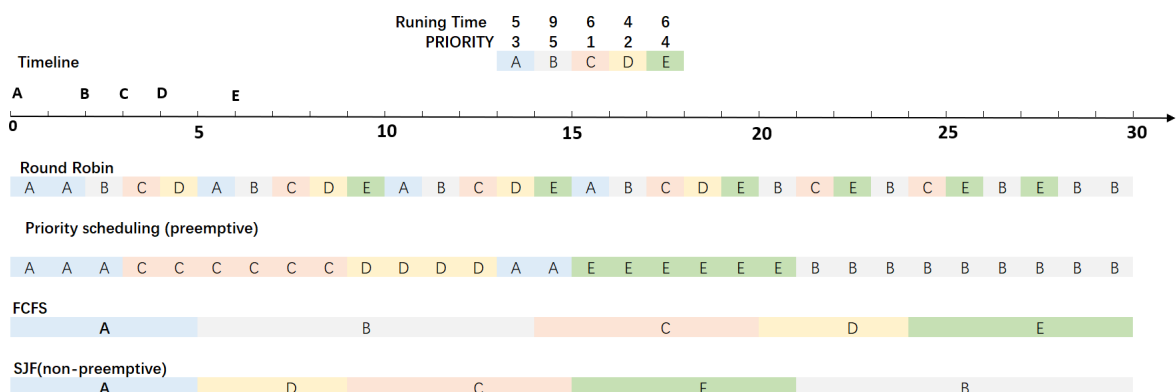
Operating Systems  
精品课程

Five batch jobs A through E, arrive at a computer center at 0, 2, 3, 4, 6 second. They have estimated running time of 5, 9, 6, 4, and 6 seconds. Their priorities are 3, 5, 1, 2, and 4, respectively, with 1 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead.

- Round robin (quantum=1s).
- Priority scheduling (Preemptive).
- FCFS.
- Shortest job first (Nonpreemptive).

Jobs	Arrive time	Running time	Priority
A	0	5	3
B	2	9	5
C	3	6	1
D	4	4	2
E	6	6	4

Gao Haichang, Software School, Xidian University 5



周转时间=作业完成时间-作业提交时间，平均周转时间= $\frac{\sum_{i=1}^n \text{作业}i\text{的周转时间}}{n}$

平均周转时间表

算法	A	B	C	D	E	平均周转时间
RR	16	28	22	15	20	20.2
priority	15	28	6	9	15	14.6
FCFS	5	12	17	20	24	15.6
SJF	5	28	12	5	15	13

## Deadlock死锁



### Exercise

Operating Systems  
操作系统

Suppose we have four resources: R1,R2,R3, and R4, total: 3,8,3 and 5. Four processes: P1,P2,P3,P4 competing for them. We could have the following situation:

Resources process	Max needs				Has allocation				Available			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	1	2	3	4	1	2	1	0	1	1	0	1
P2	1	1	2	2	0	1	2	2				
P3	1	3	1	1	1	3	0	1				
P4	1	1	2	3	0	1	0	1				

- (1) Is the current state a safe state?
- (2) P1 request one resource R4, can system allocate R4 to P1? Why?

Gao Haichang, Software School, Xidian University

6

- (1) The current state is a safe state.

Process	Max need	Already have	Still nedd	Available
P1	1 2 3 4	1 2 1 0	0 0 2 4	1 1 0 1
P2	1 1 2 2	0 1 2 2	1 0 0 0	
P3	1 3 1 1	1 3 0 1	0 0 1 0	
P4	1 1 2 3	0 1 0 1	1 0 2 2	

P2->P3->P4->P1, Available:1101->1223->2524->2625->3835.

(2) P1 request one resource R4, can system allocate R4 to P1? Why?

意思就是current available 变成 1 1 0 0 (将R4立即分配给了P1)，状态是否还安全。

Process	Max need	Already have	Still nedd	Available
P1	1 2 3 4	1 2 1 1	0 0 2 3	1 1 0 0
P2	1 1 2 2	0 1 2 2	1 0 0 0	
P3	1 3 1 1	1 3 0 1	0 0 1 0	
P4	1 1 2 3	0 1 0 1	1 0 2 2	

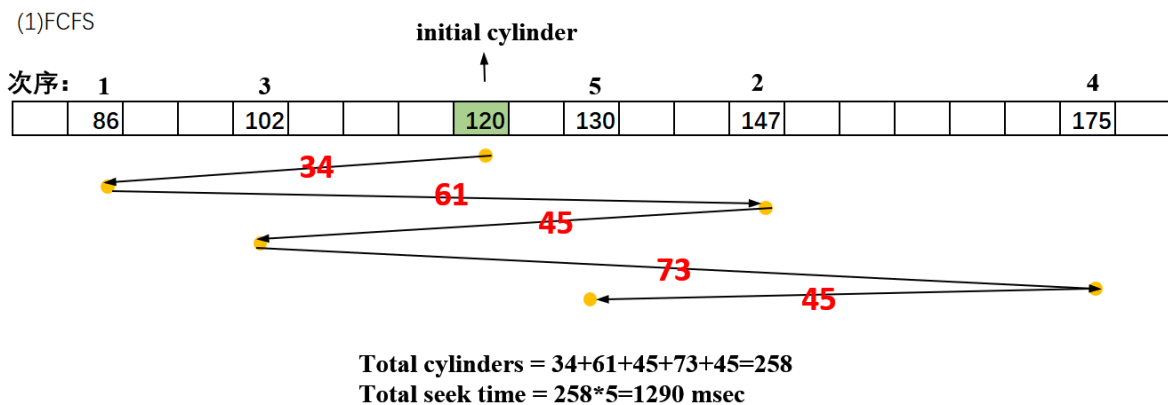
P2(Available:1222), P3(Available:2523), P4(Available:2624), P1(Available:3835), Still safe.

## Disk Arm Schelduling Algorithm

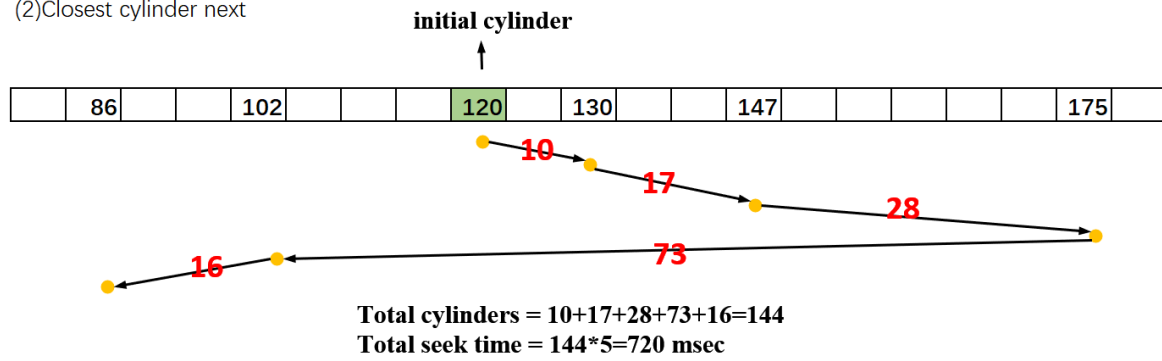
☞ Disk requests come in to the disk driver for cylinders 86, 147, 102, 175, and 130, in that order. A seek takes 5 msec per cylinder moved. How much seek time is needed for

- (a) First-come, first served.
- (b) Closest cylinder next.
- (c) Elevator algorithm (initially moving upward).

In all cases, the arm is initially at cylinder 120.



(2)Closest cylinder next



(3)同(2)，数据巧合。

## PRA & page fault

考虑下面的页访问串: 1,2,3,4,2,1,5,6,2,1,2,3。假定物理块数为3，若应用下面的页面替换算法，分别出现多少次缺页？

1)LRU

2)FIFO

3)Optimal

LRU算法 缺页次数：10

访问页	1	2	3	4	2	1	5	6	2	1	2	3
物理块1	1	1	1	4	4	4	5	5	5	2	2	2
物理块2		2	2	2	2	2	2	6	6	6	6	3
物理块3			3	3	3	1	1	1	2	1	1	1
是否缺页	√	√	√	√		√	√	√	√	√		√

LRU算法，剔除当前物理块中最长时间未被使用的页面

FIFO算法 缺页次数：10

访问页	1	2	3	4	2	1	5	6	2	1	2	3
物理块1	1	1	1	4	4	4	4	6	6	6	6	3
物理块2		2	2	2	2	1	1	1	2	2	2	2
物理块3			3	3	3	3	5	5	5	1	1	1
是否缺页	√	√	√	√		√	√	√	√	√		√

FIFO算法，剔除当前物理块中最先进入的页面（如果有多次访问，按第一次算）

OPT算法 缺页次数：7

访问页	1	2	3	4	2	1	5	6	2	1	2	3
物理块1	1	1	1	1	1	1	1	1	1	1	1	1
物理块2		2	2	2	2	2	2	2	2	2	2	2
物理块3			3	4	4	4	5	6	6	6	6	3
是否缺页	√	√	√	√			√	√				√

## semaphore: sincronization and mutal exclusion

若有一个铁笼子，最多能装3只兔子。猎人每次能向笼子里放一只兔子，如果笼子满了，猎人等待;饭店老板每次从笼子里取一只兔子，如果笼子空了，饭店老板等待。假设笼子一开始是空的，请使用信号量PV操作，模拟猎人和店老板之间的同步和互斥。