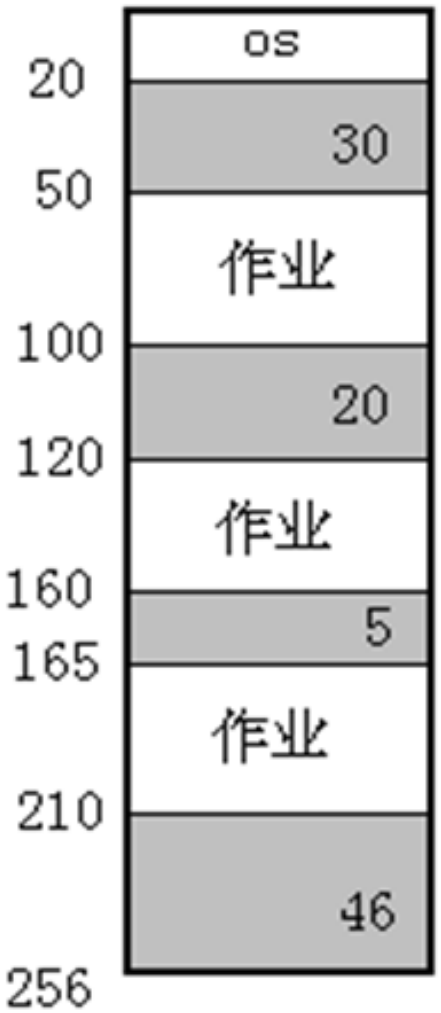
* Exercise 1

某系统，内存状态如右所示。

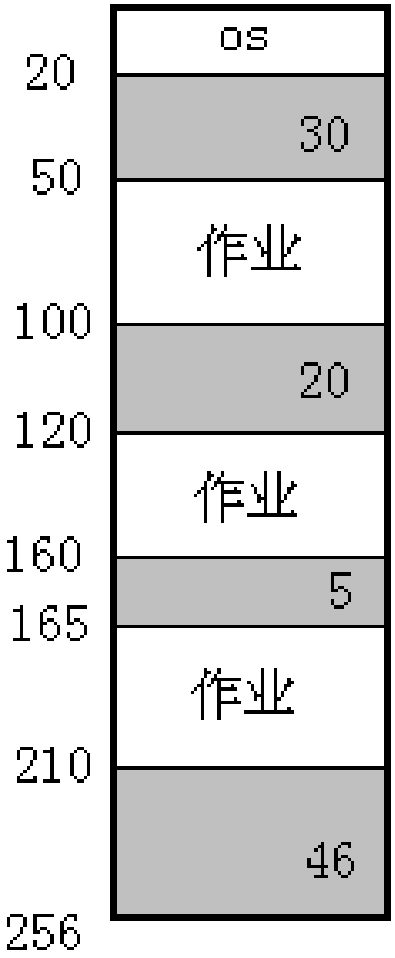
有如下作业序列：

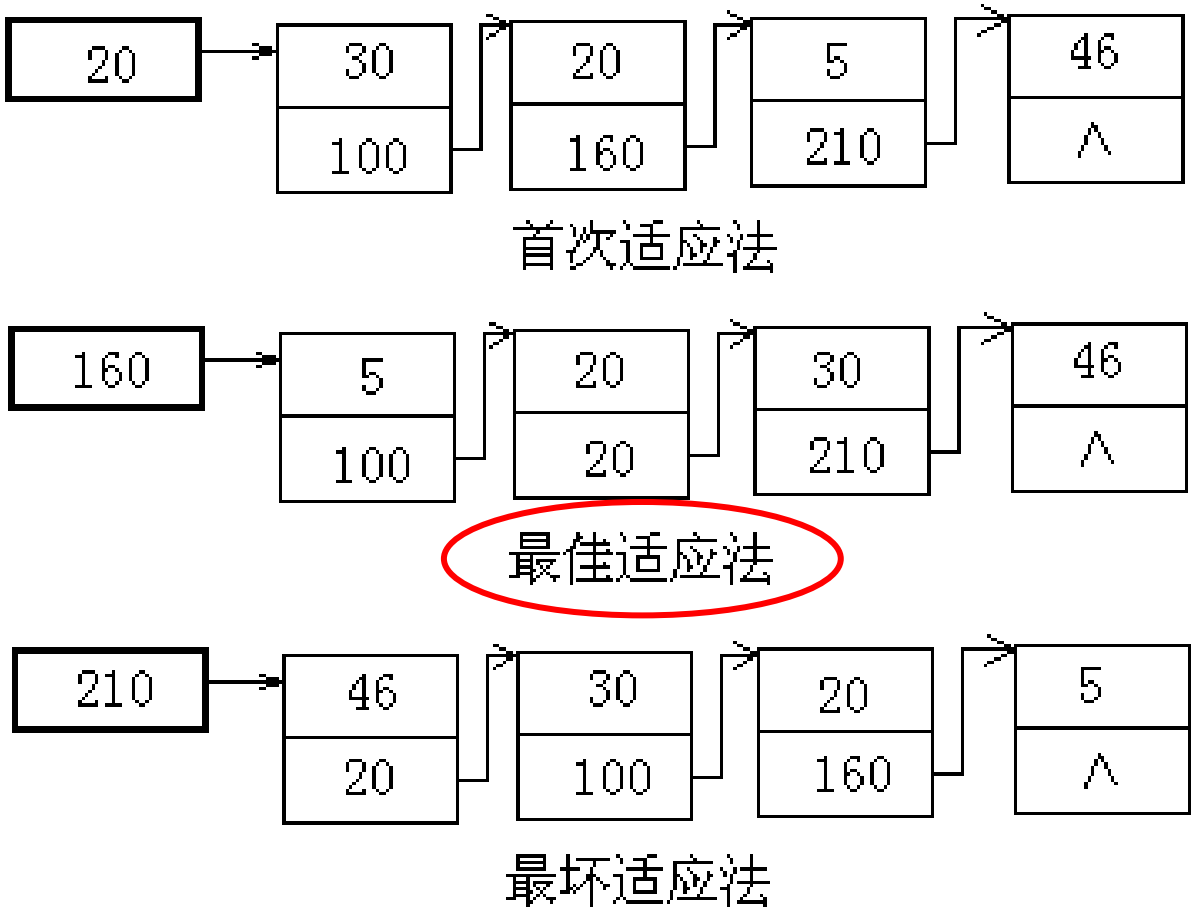
- A需要内存18M，B需要25M，C需要30M。
- 根据分析结果回答：哪种分配算法对此作业序列是合适的？



* Answer to Exercise 1

- A需要18M，B需要25M，C需要30M。





* Exercise 2

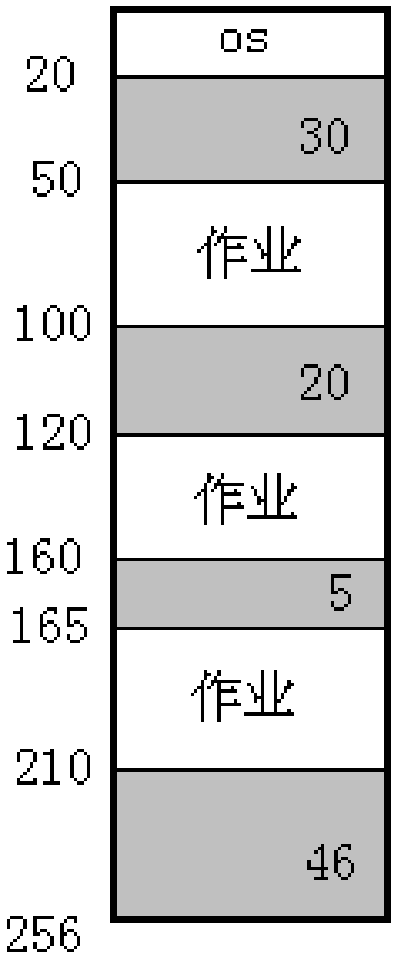
某系统，内存状态如右所示。

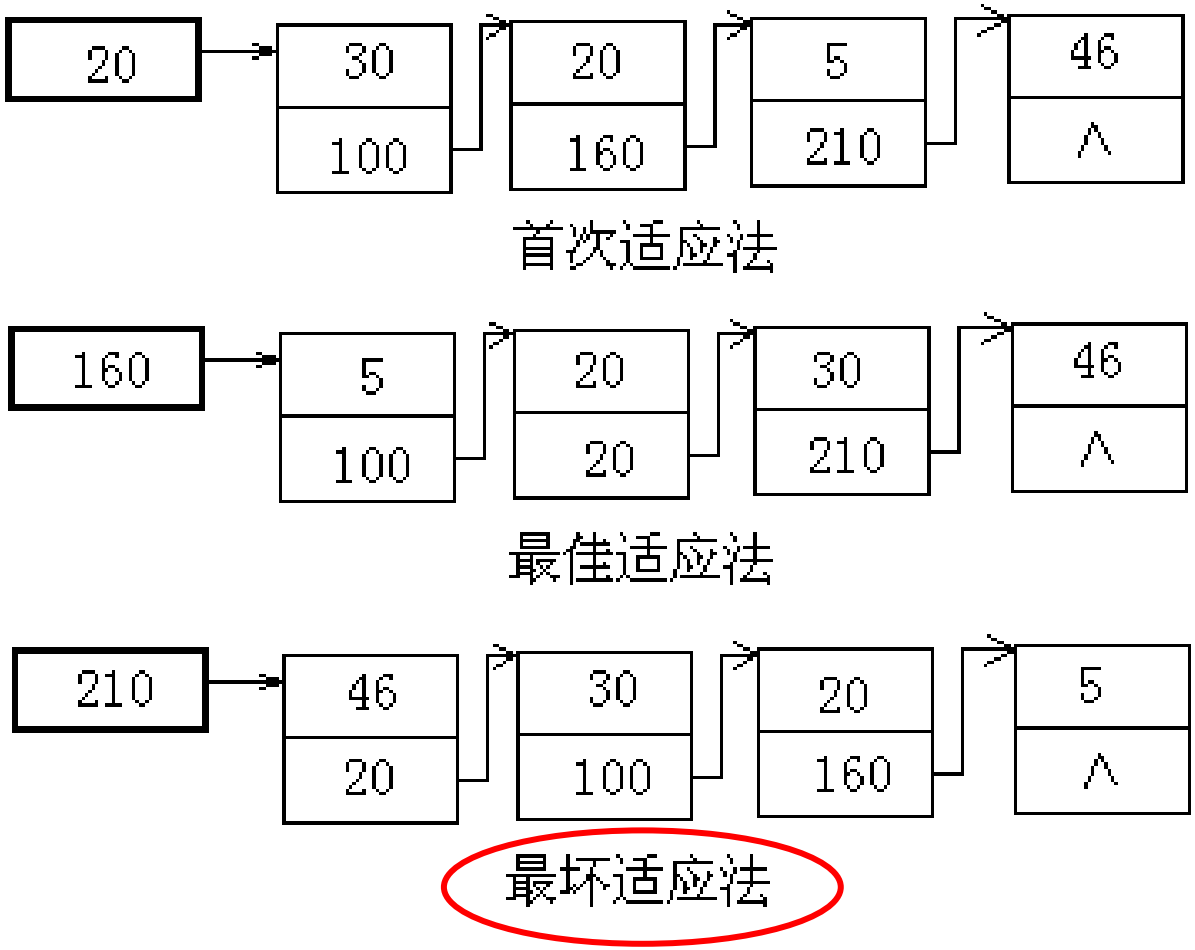
有如下作业序列：

- A需要内存21M，B需要30M，C需要25M。
- 根据分析结果回答：哪种分配算法对此作业序列是合适的？

* Answer to Exercise 2

- A需要21M，B需要30M，C需要25M。





* Exercise 3

- 用可变分区(动态重定位)方式管理内存时，假定内存中按地址顺序依次有5个空闲区，空闲区的大小依次为：32M、10M、5M、228M和100M。
  现在有5个作业A、B、C、D、E。它们各需主存大小为：1M、10M、108M、28M和115M。
  问：
  采用哪种算法能把这5个作业按顺序全部装入内存?

* Answer to Exercise 3

A: 1M
B: 10M
C: 108M
D: 28M
E: 115M

- **First fit:**
  - **Hole list: 32M、 10M、5M、228M、100M**
  - **Allocate: A(1) C(108)**
    B(10) D(28)
    21M 92M
  - **E(115M)?**
- **Best fit:**
  - **Hole list: 5M、10M、32M、100M、228M**
  - **Allocate: A(1) B(10) C(108)**
    4 120
    D(28) E(115)
    4 5
  - **Hole list: 4M、4M、5M、100M**

* Answer to Exercise 3 (Cont.)

A: 1M
B: 10M
C: 108M
D: 28M
E: 115M

- **Worst fit:**
  - **Hole list: 228M、 100M、32M、10M、5M**
  - **Allocate: A(1)**
    B(10)
    C(108)
    D(28)
  - **Hole list: 100M、81M、32M、10M、5M**
  - **E(115M)?**

2048个块.

$2^{11} \times 2^{11} = 2^{22}$

32页，每页2048B

32×2048B
$= 2^5 \times 2^{11} = 2^{16}$ B(64K)

1111234516

Exercise 4:

- **Consider a system with physical memory of 2048 frames, the logical address space of a process is up to 32 pages, the page size is 2048 bytes.**

(1) How many bits are there in the logical address? **16**

(2) How many bits in the logical address refer to virtual page number? **5**

(3) How many bits are there in the physical address? **22**

(4) How many bits in the physical address refer to frame number? **11**

(5) How many bits in the physical address refer to offset in a frame? **11**

Exercise 5:

- A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4,096 bytes.

  (1) How many bits are there in the logical address? **32**

  (2) How many bits in the logical address refer to virtual page number? **20**

  (3) How many bits are there in the physical address? **18**

  (4) How many bits in the physical address refer to frame number? **6**

  (5) How many bits in the physical address refer to offset in a frame? **12**

  (6) A user process generates the virtual address 11123456, figure out its page number and page offset. **2715 2816**

  Explain how the system establishes the corresponding physical location.

Exercise 6:

- Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

a. 3085 — P: 3, d: 13

b. 42095 — P: 41, d: 111

c. 215201

d. 650000

e. 2000001 — P: 1953, d: 129

3085 % 1024 = 3 ··· 13

42095 / 1024. 42095 % 1024

Logical address / page size → **Quotient ⇒ page number**; **Remainder ⇒ offset**

Exercise 7

- Consider a paging system with the page table stored in memory.
  - if a memory reference takes 200 nanoseconds, how long does a paged memory reference take?

    400

  - If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time?
    (Assume that finding a page-table entry in the TLBs takes 4 time, if the entry is there.)

    (4+200)*0.75+(4+200+200)*0.25=254

这是课上习题，好好复习！

Exercise 8

- **Segment table:**

Segment no.	Limit	base
0	660	2219
1	140	3300
2	100	90
3	580	1237
4	960	1959

2219+432

- **mapping logical address to physical address:**
  [0, 432], [1, 10], [2, 500], [3, 400]

  2651 **3310** **invalid** **1637**

- **Summarize the procedure that mapping a logical address to physical address.**

Exercise 1

- On a system using paging, references to a swapped-in locations accessible through an entry in an associative table take 150ns.
  If the main memory page table must be used, the reference takes 400ns.
  References that result in page faults require 8ms if the page to be replaced has been modified, 3ms otherwise.
  If the page fault rate is 2%, the associative table(TLB) hit rate is 70%, and 50% of replaced pages have been modified, what is the effective access time?
  Assume the system is running only a single process and the CPU is idle during page swaps.

Answer for exercise 1

1s=1000ms
1ms=1000μs
1μs=1000ns
1ns=1000ps

EAT=

(150ns*70% + 400ns*30%)* 98% +
(8ms*50% + 3ms*50%)*2%

= 100220.5ns

EAT = [150×0.7 + 400×0.3] ×0.98
+ (8ms×0.5 + 3ms×0.5) ×0.02.

Exercise 2

- Assume that we have a demand-paged memory. The page table is held in registers.
  It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified.
  Memory-access time is 100 nanoseconds.
  Assume that the page to be replaced is modified 30 percent of the time.
  What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Answer for exercise 2

1s=1000ms
1ms=1000μs
1μs=1000ns
1ns=1000ps

$(1-p)*100ns + (0.7p)*8*10^6\ ns + (0.3p)*20*10^6\ ns \leq 200ns$

$-100p+5600000p+6000000p \leq 100$

$11599900p \leq 100$

$P \leq 8.62*10^{-6}$

page-fault p

$(1-p)\times 100ns + p\times 0.7\times 8\times 10^6 + 0.3\times p\times 20\times 10^6 \leq 200ns$

$\Rightarrow p \leq 8.62\times 10^{-6}$

Exercise 3

- Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.)
  - TLB miss with no page fault ----- √
  - TLB miss and page fault ----- √
  - TLB hit and no page fault ----- √
  - TLB hit and page fault ----- X 不可能

Exercise 4

Consider the following page reference string:

1, 2, 4, 5, 3, 4, 1, 6, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

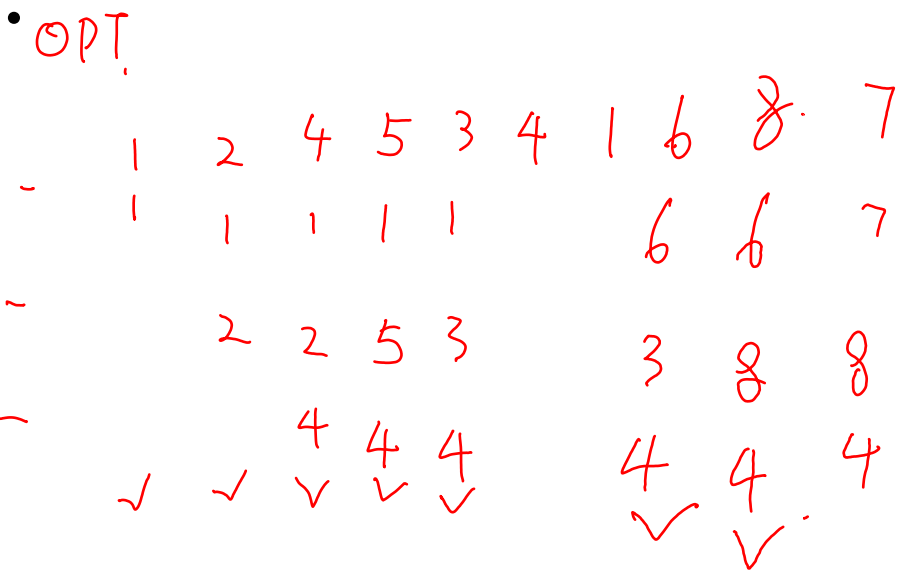
How many page faults would occur for the following replacement algorithms.

Assume there are three frames available, and all frames are initially empty.

(1) Optimal replacement

(2) LRU replacement

(3) FIFO replacement



Answer for exercise 4

OPT: **Page fault: 12 times,** **page fault rate: 12/20=0.6**

	1	2	4	5	3	4	1	6	8	7	8	9	7	8	9	5	4	5	4	2
-	1	1	1	1	1			6	6	7		7				7	4			4
-	-	2	2	5	3			3	8	8		8				5	5			5
-	-	-	4	4	4			4	4	4		9				9	9			2
	F	F	F	F	F			F	F	F		F				F	F			F

LRU: **Page fault: 13 times,** **page fault rate: 13/20=0.65**

	1	2	4	5	3	4	1	6	8	7	8	9	7	8	9	5	4	5	4	2
-	1	1	1	5	5		1	1	1	7		7				5	5			5
-	-	2	2	2	3		3	6	6	6		9				9	9			2
-	-	-	4	4	4		4	4	8	8		8				8	4			4
	F	F	F	F	F		F	F	F	F		F				F	F			F

Answer for exercise 4 (Cont.)

FIFO: **Page fault: 13 times, page fault rate: 13/20=0.65**

	1	2	4	5	3	4	1	6	8	7	8	9	7	8	9	5	4	5	4	2
-	1	1	1	5	5		5	6	6	6		9				9	9			2
-	-	2	2	2	3		3	3	8	8		8				5	5			5
-	-	-	4	4	4		1	1	1	7		7				7	4			4
	F	F	F	F	F		F	F	F	F		F				F	F			F

Exercise 5

Page	Frame number	Valid/ invalid	Page in time	Reference time	Reference bit	Modify bit
0	60	1	225	326	1	1
1	35	0	100	105	1	
2	50	0	138	245	1	1
3	35	1	289	321	1	1
4		0				
5	50	1	312	387	1	0
6		0				
7	35	0	268	280	1	0

- A process has 8 pages, its page table is shown as above, assume **three** frames are allocated to this process.
- Now page 6 is needed.
- **FIFO** replacement algorithm is used, which **frame** will the page be paged into?

  Frame 60 (now used by page 0)
- **LRU** replacement algorithm is used, which **frame** will the page be paged into?

  Frame 35 (now used by page 3)

Exercise 6

- **Consider a machine in which all memory-reference instructions have two memory addresses, and one-level indirect addressing is allowed; if an instruction is assumed to be stored in only one frame, then the minimum number of frames per process is _____. And briefly why?**

Answer to exercise 6

- **The minimum number of frames per process is 5.**
  - **1 page for instruction.**
  - **1 page for source address, which is an indirect reference to the source operand.**
  - **1 page for destination address, which is an indirect reference to the destination operand.**
  - **1 page for source operand.**
  - **1 page for destination operand.**

Supplement 1

- **Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.**
- **Answer:**

  A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid.
  If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

Supplement 2

- **Consider the following page reference string:**
  1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.
  How many page faults would occur for the following replacement algorithms, assuming three, or four frames?
  Remember all frames are initially empty, so your first unique pages will all cost one fault each.
  1. **LRU replacement**
  2. **FIFO replacement**
  3. **Optimal replacement**

Answer for 3 frames

- **FIFO**

16 page faults page fault rate: 16/20=80%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	4		4	4	6	6	6		3	3	3		2	2		2	6
-	-	2	2	2		1	1	1	2	2		2	7	7		7	1		1	1
-	-	-	3	3		3	5	5	5	1		1	1	6		6	6		3	3
	F	F	F	F		F	F	F	F	F		F	F	F		F	F		F	F

Answer for 3 frames

- **LRU**

15 page faults page fault rate: 15/20=75%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	4		4	5	5	5	1		1	7	7		2	2			2
-	-	2	2	2		2	2	6	6	6		3	3	3		3	3			3
-	-	-	3	3		1	1	1	2	2		2	2	6		6	1			6
	F	F	F	F		F	F	F	F	F		F	F	F		F	F			F

Answer for 3 frames

- **OPT**

11 page faults page fault rate: 11/20=55%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	1			1	1				3	3			3	3			3
-	-	2	2	2			2	2				2	7			2	2			2
-	-	-	3	4			5	6				6	6			6	1			6
	F	F	F	F			F	F				F	F			F	F			F

Answer for 4 frames

- **LRU**

10 page faults, page fault rate: 10/20=50%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	1			1	1				1	1	6			6			
-	-	2	2	2			2	2				2	2	2			2			
-	-	-	3	3			5	5				3	3	3			3			
-	-	-	-	4			4	6				6	7	7			1			
	F	F	F	F			F	F				F	F	F			F			

Answer for 4 frames

- **FIFO**

14 page faults, page fault rate: 14/20=70%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	1			5	5	5	5		3	3	3		3	1		1	
-	-	2	2	2			2	6	6	6		6	7	7		7	7		3	
-	-	-	3	3			3	3	2	2		2	2	6		6	6		6	
-	-	-	-	4			4	4	4	1		1	1	1		2	2		2	
	F	F	F	F			F	F	F	F		F	F	F		F	F		F	

Answer for 4 frames

- **OPT**

8 page faults, page fault rate: 8/20=40%

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
-	1	1	1	1			1	1					7				1			
-	-	2	2	2			2	2					2				2			
-	-	-	3	3			3	3					3				3			
-	-	-	-	4			5	6					6				6			
	F	F	F	F			F	F					F				F			

* 连续文件举例

例如：某文件系统，磁盘块大小为 512 B

LA=1248
PA=?

- 字符流文件 A，长度为 1980 B

  $\lceil \frac{1980}{512} \rceil$

  1248/512　2..224

  32块, 224th字节

  - 文件A需要占用 4 个物理块。
  - 假如分配到30、31 、32和33四个相邻的物理块中。
  - 第33块中实际使用了444字节，剩余的68字节形成“内部碎片”。

- 记录文件 B，逻辑记录长 100 B，有23个记录

  LA=19
  PA=?

  $\lfloor \frac{512}{100} \rfloor$

  19/5　3..4

  - 假设：逻辑记录不能跨物理块存放。
  - 每个物理块中可以存放5个逻辑记录，需要5个物理块。

  $\lceil \frac{23}{5} \rceil$

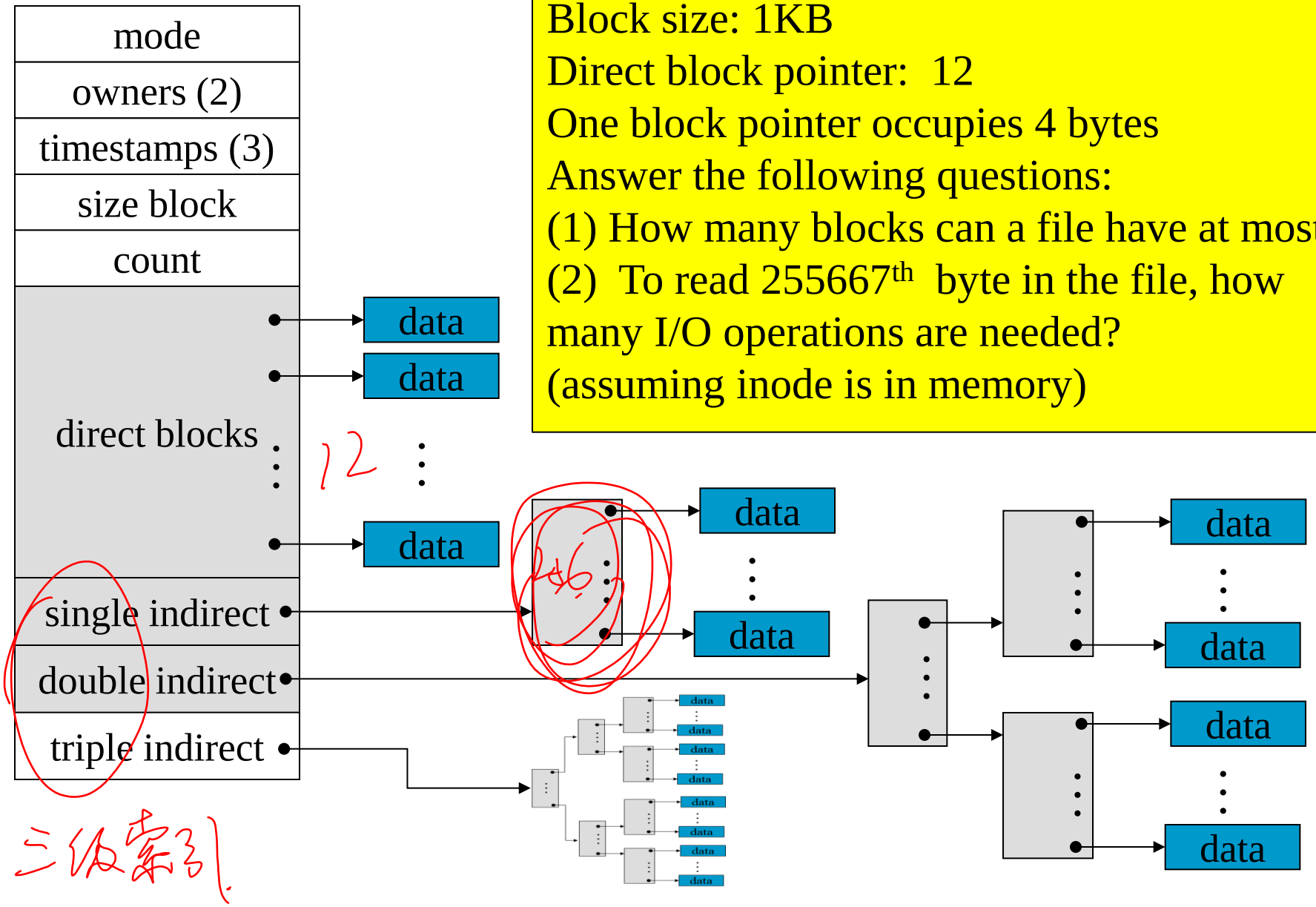
  9块, 400th字节

  - 假设分配到第 6、7、8、9、10 块
  - 前4块各有内部碎片12B，最后一块有212B没有使用。

* 连续文件的地址转换

- 在文件说明信息中有关于存放位置的描述
  - 开始块号、总块数（物理文件的长度）
- 字符流文件中逻辑地址 L
  - L/物理块大小，商s：逻辑块号，余数d：块内地址
  - 物理地址：块号：开始块号+s，块内地址：d
- 记录文件中的逻辑记录号 n (记录不可跨块存放时)
  - 每个物理块中可以存放逻辑记录数 m
    - m=物理块大小/逻辑记录大小
  - n/m，商s：逻辑块号，余数w：块内记录号
  - 物理地址
    - 块号：开始块号+s，块内地址d=逻辑记录长度*w

Combined Scheme: UNIX (4K bytes per block)



Answer

每个索引块可以存放1024/4=256个块号。

(1) How many blocks can a file have at most?

12+256+256*256+256*256*256=16843020

(2) To read 255667th byte in the file, how many I/O operations are needed?

255667/1024=249 … 691

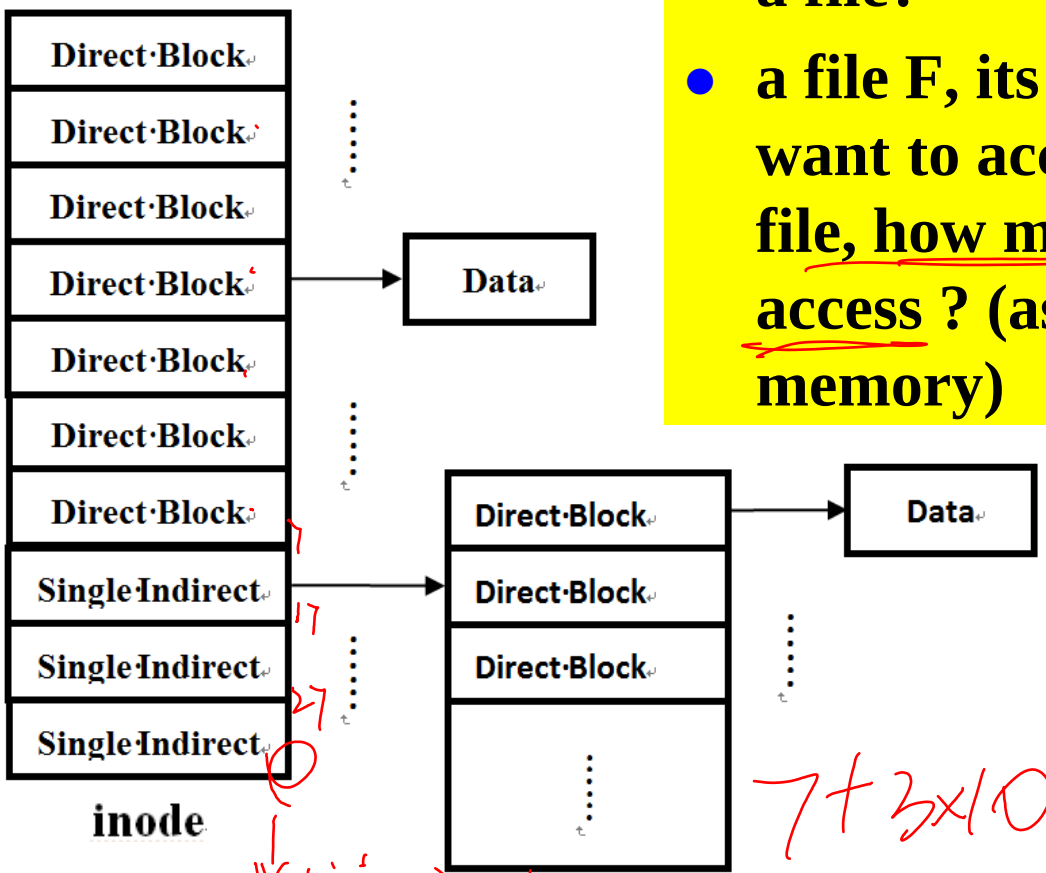
文件的第249块的块号存放在一级间接指针指向的索引块中。

所以，需要2次I/O操作，第一次I/O将索引块读进内存，找到相应的块号，第二次I/O，将字节255667所在的物理块读入内存。

exercise

- A file system, its allocation scheme is as follows:
- each file, only one block can be used as inode, in which, 10 block numbers can be hold at most, 7 point to data blocks and 3 Single indirect blocks.

- How many blocks is the largest size of a file?
- a file F, its size is 30 Blocks. If we want to access the third Block in this file, how many Blocks we need to access ? (assuming all Blocks is not in memory)



(1) 7+3*10=37
(2) 2

Supplement 1

- **Consider a file currently consisting of 100 blocks. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous-allocation case, assume that there is no room to grow in the beginning, but there is room to grow in the end. Assume that the block information to be added is stored in memory.**

  a. The block is added at the beginning.

  b. The block is added in the middle.

  c. The block is added at the end.

  d. The block is removed from the beginning.

  e. The block is removed from the middle.

  f. The block is removed from the end.

Answer

	contiguous	Linked	indexed
a	201	1	1
B	101	50+1+1=52	1
C	1	1+1+1=3	1
D	99*2=198	1	0
E	49*2=98	50+1+1=52	0
f	0	99+1=100	0

Exercise 1

- **Suppose that a disk drive has 5000 cylinders, number 0 to 4999. The drive is current serving a request at cylinder 143, and the previous request was at cylinder 94. The queue of pending requests, in FIFO order, is:**

  86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

- **Starting from current position. If the total distance (in cylinders) that the disk arm moves to satisfy all the pending request is 9769, so what disk-scheduling algorithm may the system use? And why?**

Answer to exercise 1

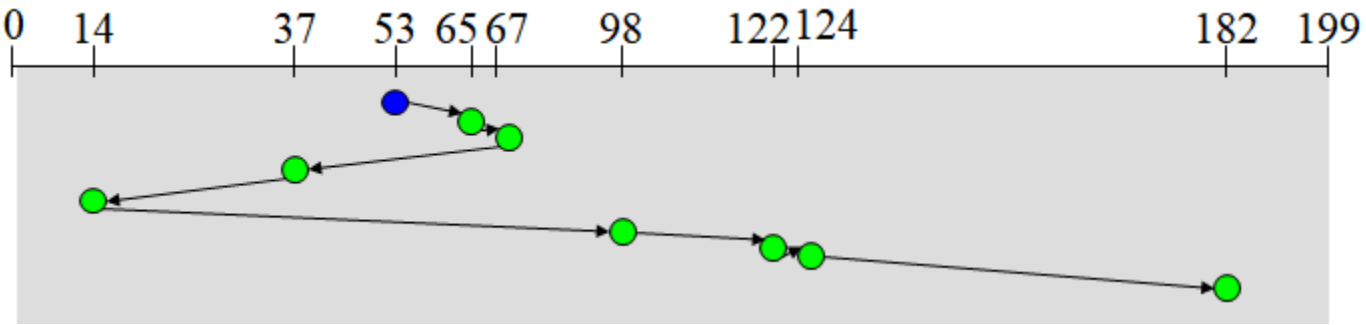
- **FCFS：143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.**
  寻道距离：7081
- **SSTF：143, 130, 86, 913, 948, 1022, 1470, 1509, 1750, 1774.**
  寻道距离：1745
- **SCAN：143, 913, 948, 1022, 1470, 1509, 1750, 1774, 130, 86.**
  寻道距离：9769
- **C-SCAN：143, 913, 948, 1022, 1470, 1509, 1750, 1774, 86, 130.**
  寻道距离：9985
- **LOOK：143, 913, 948, 1022, 1470, 1509, 1750, 1774, 130, 86.**
  寻道距离：3319

Exercise 2

- **There is a disk drive which has 200 cylinders 0 through 199. The SCAN algorithm and SSTF algorithms are used to schedule the request queue. Consider a disk queue with requests for I/O to blocks on cylinders,**
  98,182,37,122,14,65, 124,67.
  and the disk head is initially at cylinder 53.
  Suppose these two algorithms result in a same total head movement of cylinders.
- **Question:**
  Please describe the movement direction of the disk head when using the SCAN algorithms, which movement direction?

Answer to exercise 2

- When SSTF algorithm is used
  - Scheduling order: 65, 67, 37, 14, 98, 122, 124, 182



  - Head movement of 235 cylinders.
- When SCAN algorithm is used
  - If head is moving toward 199, Head movement of 331 cylinders
  - If head is moving toward 0, Head movement of 235 cylinders
- So, When SCAN algorithm is used, head is moving toward 0。

Exercise 3

- **There is a disk drive which has 200 cylinders 0 through 199. The SCAN algorithm and SSTF algorithms are used to schedule the request queue. Consider a disk queue with requests for I/O to blocks on cylinders,**
  98,182,37,122,14,65, 124,67.
  Staring from the current head position, the SCAN algorithm is used to schedule the request queue, no matter which direction the disk head moves, the schedule algorithm results in a same total distance (in cylinders).
- **Which cylinder is the current head on? Why?**

Answer to exercise 3

- **The current head is on cylinder 101.**
- **The reasons are as follows,**
  - **The SCAN algorithm is used, suppose the head is on cylinder x**
    - **If head is moving toward 199, Head movement of cylinders is: (199-x)+(199-14)**
    - **If head is moving toward 0, Head movement of cylinders is: x+182**
  - **So, there is following equation: (199-x)+(199-14)=x+182**
  - **Solve the equation and get: x=101.**
  - **So, the current head is on cylinder 101.**