8.3 Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

Answer:

a. First-fit:
b. 212K is put in 500K partition
c. 417K is put in 600K partition
d. 112K is put in 288K partition (new partition 288K = 500K - 212K)
e. 426K must wait
f. Best-fit:
g. 212K is put in 300K partition
h. 417K is put in 500K partition
i. 112K is put in 200K partition
j. 426K is put in 600K partition
k. Worst-fit:
l. 212K is put in 600K partition
m. 417K is put in 500K partition
n. 112K is put in 388K partition
o. 426K must wait

In this example, Best-fit turns out to be the best.

8.9 Consider a paging system with the page table stored in memory.

a. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?

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

Answer:

a. 400 nanoseconds; 200 nanoseconds to access the page table and 200 nanoseconds to access the word in memory.

b. Effective access time = $0.75 \times (200 \text{ nanoseconds}) + 0.25 \times (400 \text{ nanoseconds}) = 250 \text{ nanoseconds}$.

第8章作业

- On a *simple* paging system with a page table containing 1024 entries of 13 bits (including one valid/invalid bit) each, and a page size of 4096 bytes
  - (a) what is the size of the logical address space?
  - (b) how many bits are there in a logical address?
  - (c) how many bits in the physical address specify the frame number?
  - (d) how many bits in the physical address specify the offset within the frame?
  - (e) what is the size of the physical address space ?
  - (f) how many bits are there in a physical address?

第8章作业

(a) logical address space: $2^{10} \times 2^{12} = 2^{22} B$

(b) bits of logical address: 22 bit

(c) bits of frame number: 页表项位数-1=13-1=12 bit

(d) bits of physical address specify the offset: $\log_2(4096) = 12\ bit$

(e) size of the physical address space: $2^{(12+12)} = 2^{24}\ B$

(f) bits of physical address: 24 bit

Chapter 8

Consider a system with physical memory of 2048 frames. The logical address space of a process is up to 512 pages, and the page size is 4KB.

(1) How many bits are there in the logical address referring to a page number?

(2) How many bits are there in the logical address?

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

(4) How many bits are there in the physical address referring to an offset?

(5) Given the logical address 2018117, figure out its page number and page offset.

Chapter 8

(1) 9 (2^9=512)

(2) 9+12=21 (2^9* 2^12=512*4K=512*4096)

(3) 11+12=23 (2^11*2^12=2048*4K =2048*4096)

(4) 12 (2^12=4096)

(5) **page number** 492 2018117/4096=492.7

page offset 2885 2018117%4096=2885

Chapter 9

9.5 Assume 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 page is available or 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 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Answer:

$$
\begin{aligned}
0.2\ \mu\text{sec} &= (1-P) \times 0.1\ \mu\text{sec} + (0.3P) \times 8\ \text{millisec} + (0.7P) \times 20\ \text{millisec} \\
0.1 &= -0.1P + 2400\ P + 14000\ P \\
0.1 &\simeq 16{,}400\ P \\
P &\simeq 0.000006
\end{aligned}
$$

9.18 Assume there is an initial 1024 KB segment where memory is allocated using the Buddy System. Using Figure 9.27 as a guide, draw the tree illustrating how the following memory requests are allocated:

Request A • request 240 bytes

Request B • request 120 bytes

Request C • request 60 bytes

Request D • request 130 bytes

Next, modify the tree for the following releases of memory. Perform coalescing whenever possible:

Release A • release 240 bytes

Release C • release 60 bytes

Release B • release 120 bytes

Answer: The following allocation is made by the Buddy system: The 240 byte request is assigned a 256 byte segment. The 120 byte request is assigned a 128 byte segement, the 60 byte request is assigned a 64 byte segment and the 130 byte request is assigned a 256 byte segment. After the allocation, the following segment sizes are available: 64 bytes, 256 bytes, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K, and 512K.
After the releases of memory, the only segment in use would be a 256 byte segment containing 130 bytes of data. The following segments will be free: 256 bytes, 512 bytes, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K, and 512K.

Draw the **kernel memory allocation diagram** to illustrate how the following memory requests are allocated:

	256	256			512		1k	2k	4k	8k	16k	32 k	64 k	128 k	256 k	512 k
Request A	A	256			512		1k									
Request B	A	B	128		512		1k									
Request C	A	B	C	64	512		1k									
Request D	A	B	C	64	D	256	1k									
Release A	256	B	C	64	D	256	1k									
Release C	256	B	128		D	256	1k									
Release B	512				D	256	1K									

第9章作业(网工+大数据)

- 作业3：Second Chance置换
  给定以下reference string

  3　2　3　1，4　3　5　4，2　3　4　3，5　2

可用frame数目为3，采用Second Chance algorithm，计算page fault和页置换数目

序列	3	2	3	1	4	3	5	4	2	3	4	3	5	2
Frame	3	3	3*	3*	1	1	3*	3*	3	3*	3*	3*	3	4
		2	2	2	3	3*	4	4*	4	4	4*	4*	4	5
				1	4	4	5	5	2	2	2	2	5	2
缺页?	T	T		T	T		T		T				T	T

缺页次数：8
置换次数：5

第9章作业(留学生)

- **reference string: 20 references**

  7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

  - **three frames available**
- FIFO and LRU algorithm are used for the reference string.
- Give the scheduling results and the number of page faults of the two algorithms, respectively.

第9章作业(留学生)

- FIFO
- #Page fault： 15

访问页面	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
物理块1	7	7	7	2		2	2	4	4	4	0			0	0			7	7	7
物理块2		0	0	0		3	3	3	2	2	2			1	1			1	0	0
物理块3			1	1		1	0	0	0	3	3			3	2			2	2	1
缺页否	√	√	√	√		√	√	√	√	√	√			√	√			√	√	√

第9章作业(留学生)

- LRU
- #Page fault： 12

访问页面	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
物理块1	7	7	7	2		2		4	4	4	0			1		1		1		
物理块2		0	0	0		0		0	0	3	3			3		0		0		
物理块3			1	1		3		3	2	2	2			2		2		7		
缺页否	√	√	√	√		√		√	√	√	√			√		√		√		

Chapter 11

11.6 Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:

a. How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)

b. If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

Chapter 11

Answer: Let Z be the starting file address (block number).

a. Contiguous. Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively.
   1. Add X to Z to obtain the physical block number. Y is the displacement into that block.
   2. 1

b. Linked. Divide the logical physical address by 511 with X and Y the resulting quotient and remainder respectively.
   1. Chase down the linked list (getting $X + 1$ blocks). $Y + 1$ is the displacement into the last physical block.
   2. 4

c. Indexed. Divide the logical address by 512 with X and Y the resulting quotient and remainder respectively.
   1. Get the index block into memory. Physical block address is contained in the index block at location X. Y is the displacement into the desired physical block.
   2. 2

There are two text files A and B in the file system. The size of the file A is 15MB, and the size of the file B is 300KB. When using the linked allocation scheme, each block's size is 1024B, and the block address in the block is of 4 bytes length. The directory entries are already in main memory.

1) What is the size of the maximum file in this file system?

2) When the file A will be revised, how many disk I/O operations are required if the information in the 15698^{th} byte in the file A is to be revised?

（1）由于块地址占 4 字节，即 32 位，所以能表示的最多块数为2^{32}=4G，而每个盘块中存放文件大小为1020B，故链接分配可管理的最大文件为：4G*1020B=4080 GB。

（2）读文件A的15698B处的信息，逻辑块号为16(15698B/1020B=15.39)，要先把该信息所在块读出，共花费16次磁盘 I/O 操作。修改还需要写回操作，由于写回时已获得该块的物理地址，只需要1次访问磁盘，故总共需要17次I/O 操作。

作业(3)

- Consider a file system on a disk. The size of disk blocks is 512B. The directory is organized into a tree structure, as shown in Fig.3, and the root directory resides in memory
- As shown in Fig.1, each directory entry has one bit **type** to define the entry as a subdirectory or as a file, and two 2-byte fields, one holds a file name and the other holds a pointer to a disk block
- For a directory file in Fig.1, the directory name and its first disk block address are recorded
- Directory files are organized into linked files, and common data files are organized into indexed files

name	address	type
A	Pointer to 1st block	directory
B	Pointer to 1st block	directory
C	Pointer to 1st block	directory
D	Pointer to FCB block	file

Figure 1 directory file

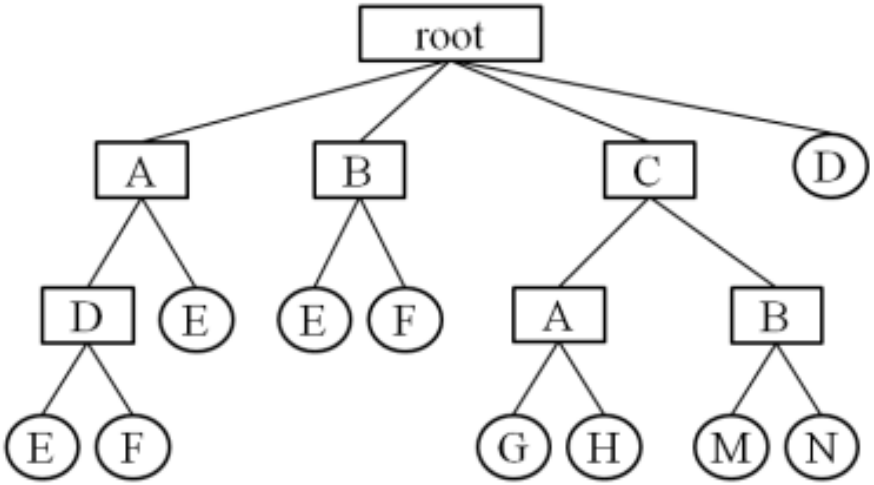


Figure 3 directory tree

	other attributes
1	pointer
2	pointer
...	
10	pointer
11	pointer
12	pointer
13	pointer

Figure 2 FCB structure

作业(3)

- For a data file, the file name and the disk address where its FCB resides are recorded, described as the file D in Fig.1
- The structure of the directory file is shown in Figure 1, and the structure of FCB is shown in Figure 2

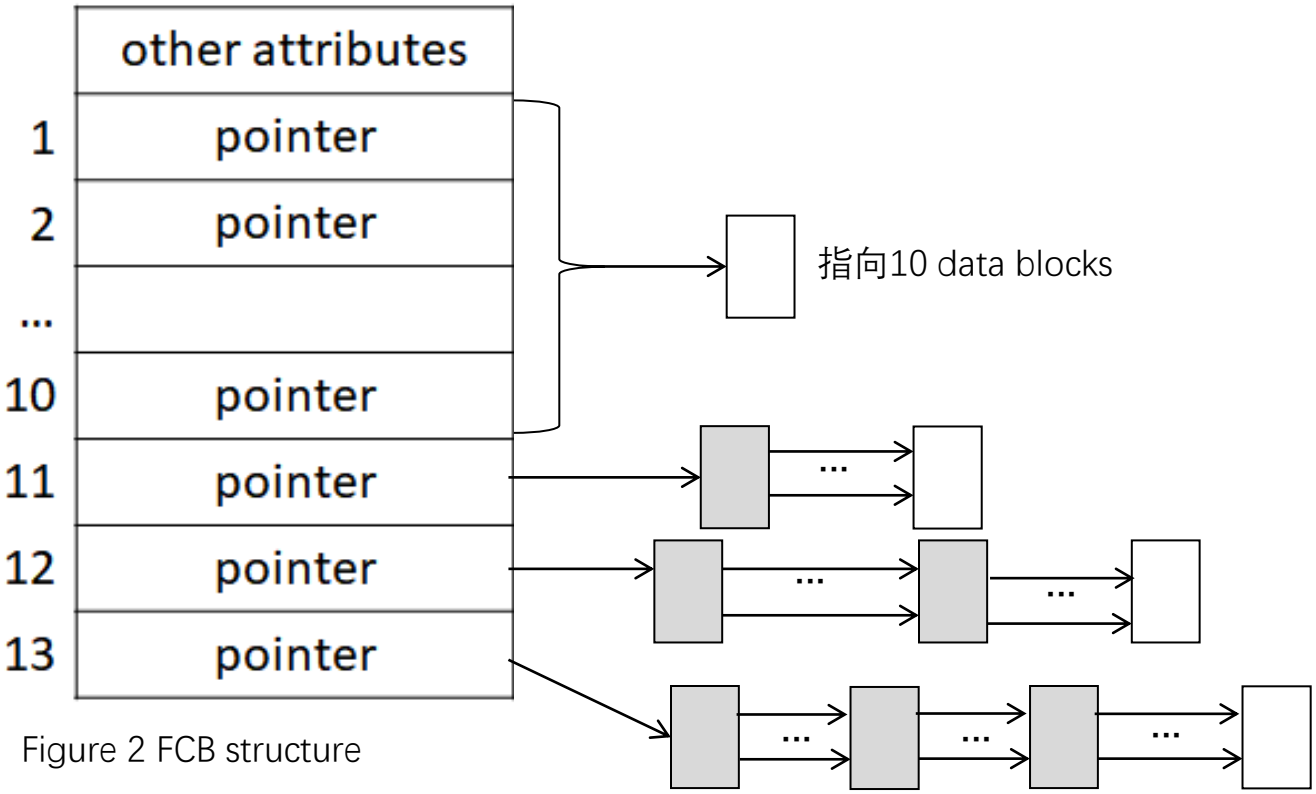
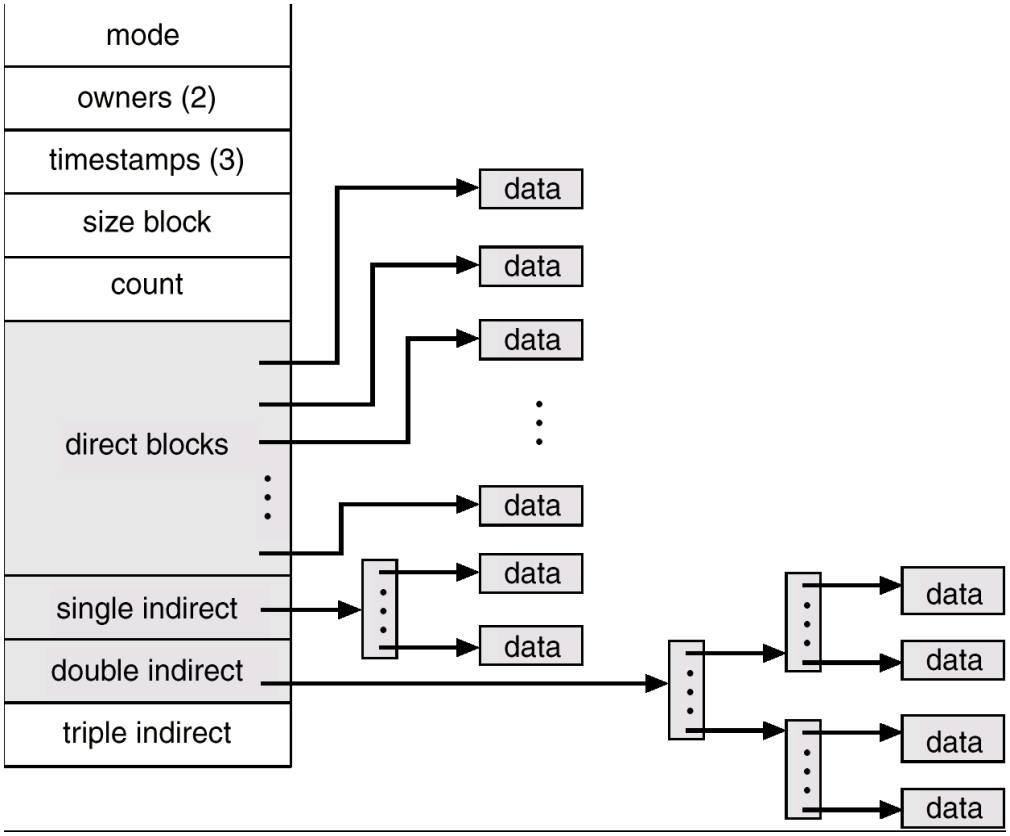


Figure 2 FCB structure

作业(3)


mode
owners (2)
timestamps (3)
size block
count
direct blocks
single indirect
double indirect
triple indirect
data
data
data
data
data
data
data
data
data
data

作业(3)

- There are 13 pointers in the file’ s FCB. Each pointer (disk address) requires 2 bytes.
- The first 10 of these pointers directly point to data blocks of files
- The next three pointers point to indirect index blocks, and among these three pointers,
  - the first points to a single indirect block, which is an index block containing the addresses of data blocks;
  - the second points to a double indirect block, which contains the address of index blocks, and these index blocks contain the addresses of data blocks;
  - the last pointer contains the address of a triple indirect block.

作业(3)

- Questions
  - (1) Calculate the maximum size of the file that can be accessed only through the direct index in FCB
  - (2) Calculate the maximum address space accessible through the FCB
  - (3) Suppose the directory tree of the file system is shown in Figure 3

    Now, we want to read the content at the 10240th byte in file M into memory, how many and which disk blocks must be read into memory?

Answers

- (1) 512B*10 = 5120B = 5 KB
- (2)

  原：(10+256 + 256*256 + 256*256*256) *512B

  改正：Each pointer requires 4B, (10 +128 + 128*128 + 128*128*128) *512B
- (3)
  - 10240th byte位于M的第[10240/512]+1=21个block中
  - 根据索引结构，第10240th bytes所在的第21 block由一级间接索引组织
  - 由于root目录已经在内存中，从根目录依次读取目录C、B和M所在block，再从M所在的block中取出文件 FCB，由FCB读取一级文件索引block，之后再读取第21个数据块

  1(读目录C) + 1(读目录B) + 1(读M的FCB)

  + 1(读一级index block) + 1(读data block)

Chapter 12

Several algorithms exist to schedule the servicing of disk I/O requests.

We illustrate them with a request queue (0-4999).

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

Head pointer 143

Illustrate how different algorithms (FCFS, SSTF, SCAN, C-SCAN, C-LOOK) schedule the servicing of disk I/O requests. Calculate the total head movement.

Answer:

a. The FCFS schedule is 143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. The total seek distance is 7081.

D=|86-143|+|1470-86|+|913-1470|+|1774-913|+|948-1774|
+|1509-948|+|1022-1509|+|1750-1022|+|130-1750|

b. The SSTF schedule is 143, 130, 86, 913, 948, 1022, 1470, 1509, 1750, 1774. The total seek distance is 1745.

D=|130-143|+|86-130|+|913-86|+|948-913|+|1022-948|
+|1470-1022|+|1509-1470|+|1750-1509|+|1774-1750|

c. The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 130, 86. The total seek distance is 9769.

D=|4999-143|+|130-4999|+|86-130|

D=|913-143|+|948-913|+|1022-948|+|1470-1022|+|1509-1470|
+|1750-1509|+|1774-1750|+|4999-1774|+|130-4999|+|86-130|

e. The C-SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 86, 130. The total seek distance is 9813.

D=|4999-143|+|86-4999|+|86-130|

f. (Bonus.) The C-LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 86, 130. The total seek distance is 3363.

D=|1774-143|+|86-1774|+|130-86|

Procedure of the arm moving of SCAN