第八章　磁盘存储器的管理

8.1 为文件分配磁盘块的方法
8.2 磁盘空闲空间的管理
8.3 提高磁盘I/O速度的途径
8.4 提高磁盘可靠性的技术
8.5 数据一致性控制

韩明峰　　QQ 2022446359

8.1　为文件分配磁盘块的方法

为文件分配磁盘块的常用方法有：

(1)连续分配

(2)链接分配

(3)索引分配

8.1.1 连续分配方式

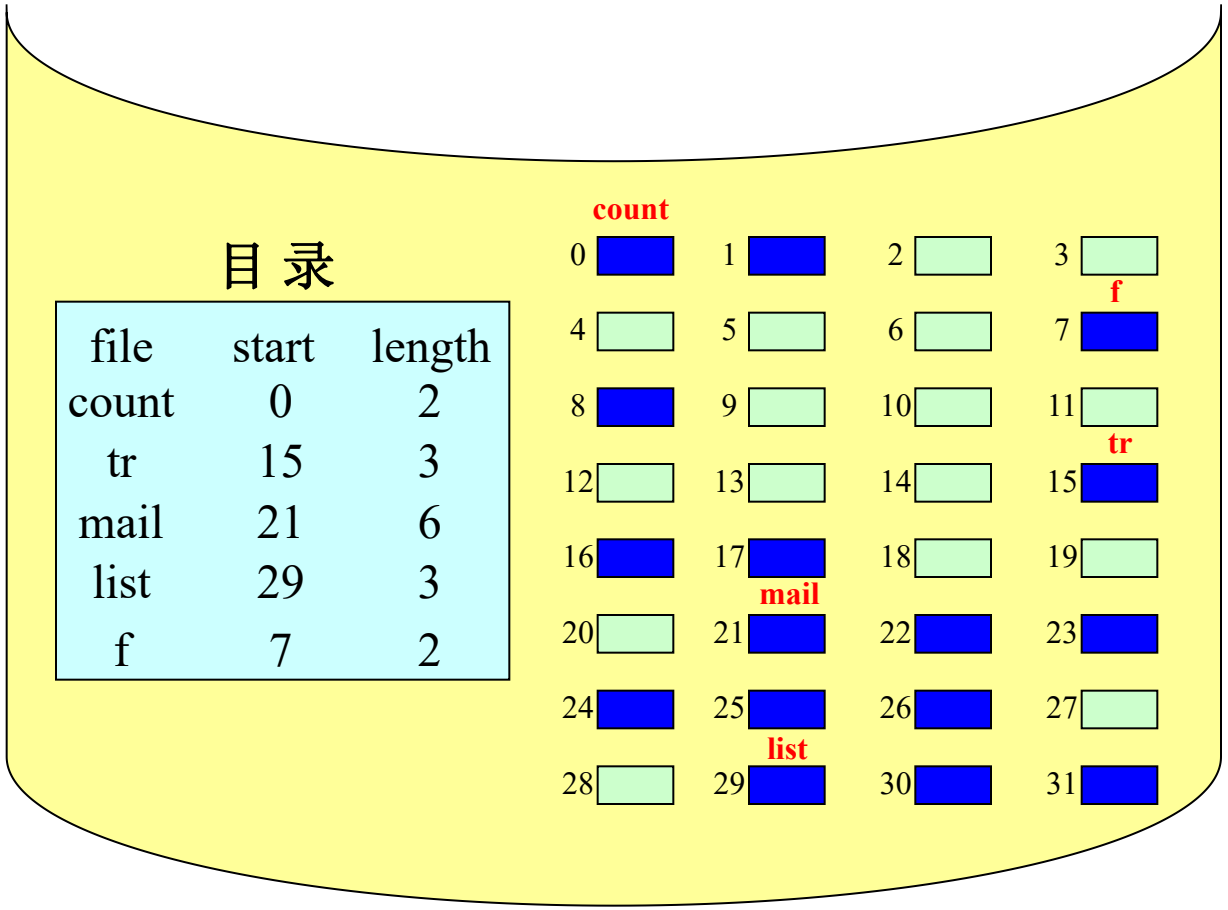


图8-1 磁盘空间的连续分配方式

1. 连续分配方式的主要优点有：

(1)顺序访问和直接访问容易。

(2)顺序访问速度快。

2. 连续分配方式的主要缺点如下：

(1)要求为一个文件分配连续的存储空间。

(2)必须事先知道文件的长度。

(3)不能灵活地删除和插入记录。

(4)对于动态增长的文件，很难为其分配空间。

8.1.2 链接分配方式

1. 隐式链接

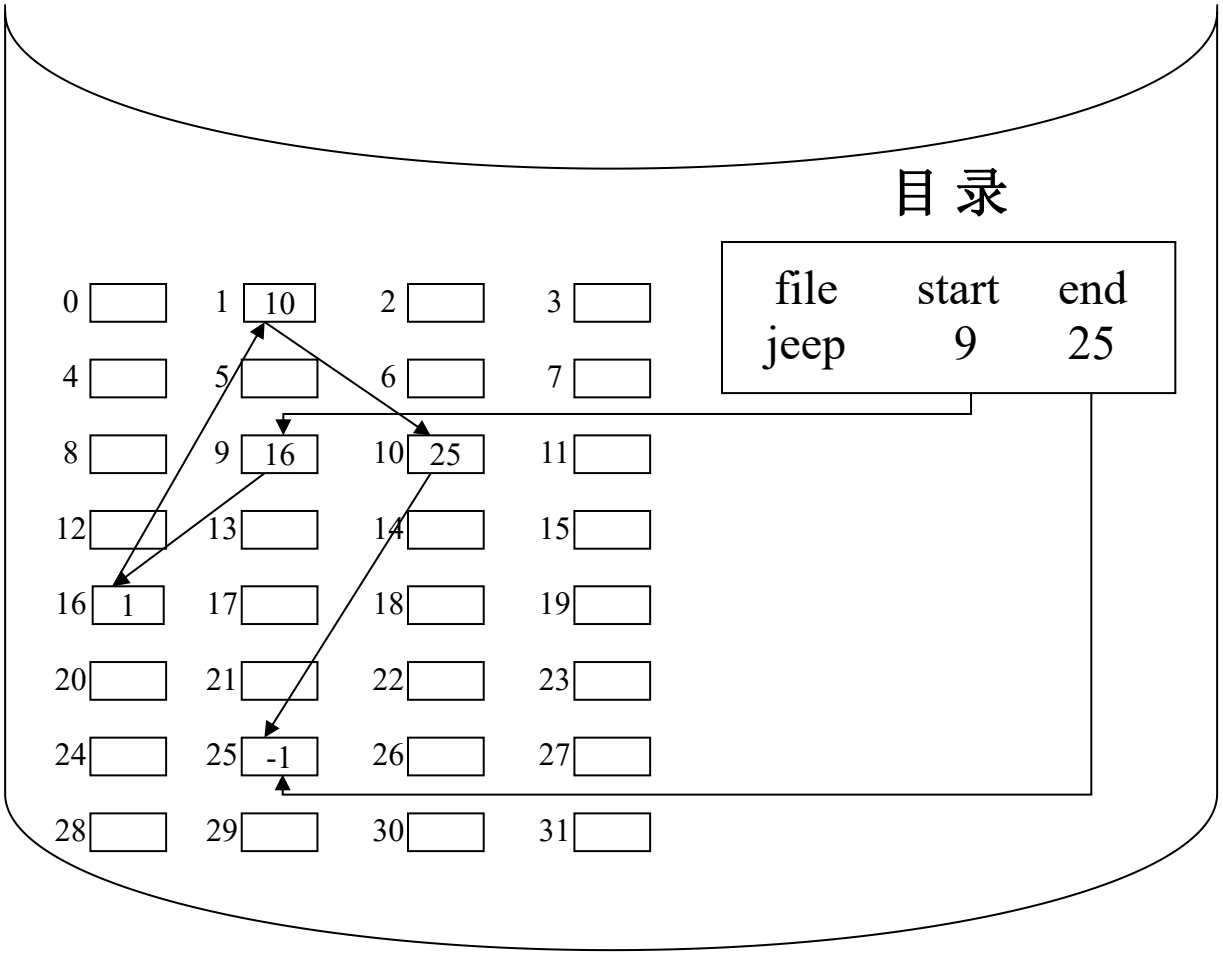


图8-2 磁盘空间的链接式分配

8.1.2 链接分配方式

1. 隐式链接

缺点：只适合顺序访问，随机访问要从头查找极低效；可靠性差，盘块的指针出现问题会导致链断开。

改进：将几个盘块组成一个簇，以簇为单位分配盘块。

2. 显式链接

把用于链接文件各物理块的指针显式地存放在一个链接表中。

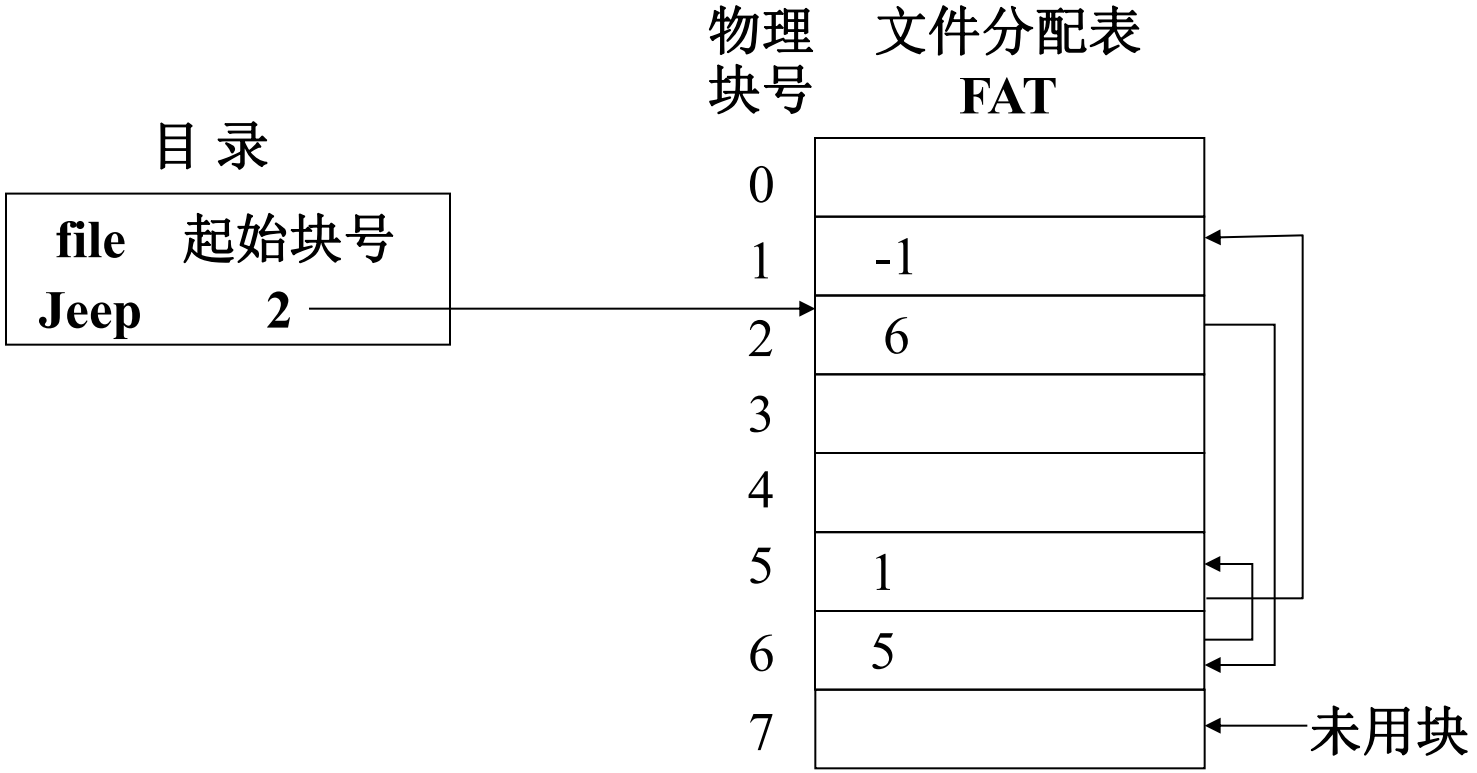


图8-3 显式链接结构

8.1.3 索引分配方式

链接方式存在问题：

（1）不能支持高效直接存取；

（2）一个文件所占盘块号随机分布在FAT中，因此需将整个FAT调入内存，占用了较大的内存空间。

1. 单级索引分配方式

为每个文件分配一个集中存放的索引块(表),包含文件的所有物理块号，因而索引块实质就是磁盘块地址数组。

在该文件的目录项中存储了指向该索引块的指针。

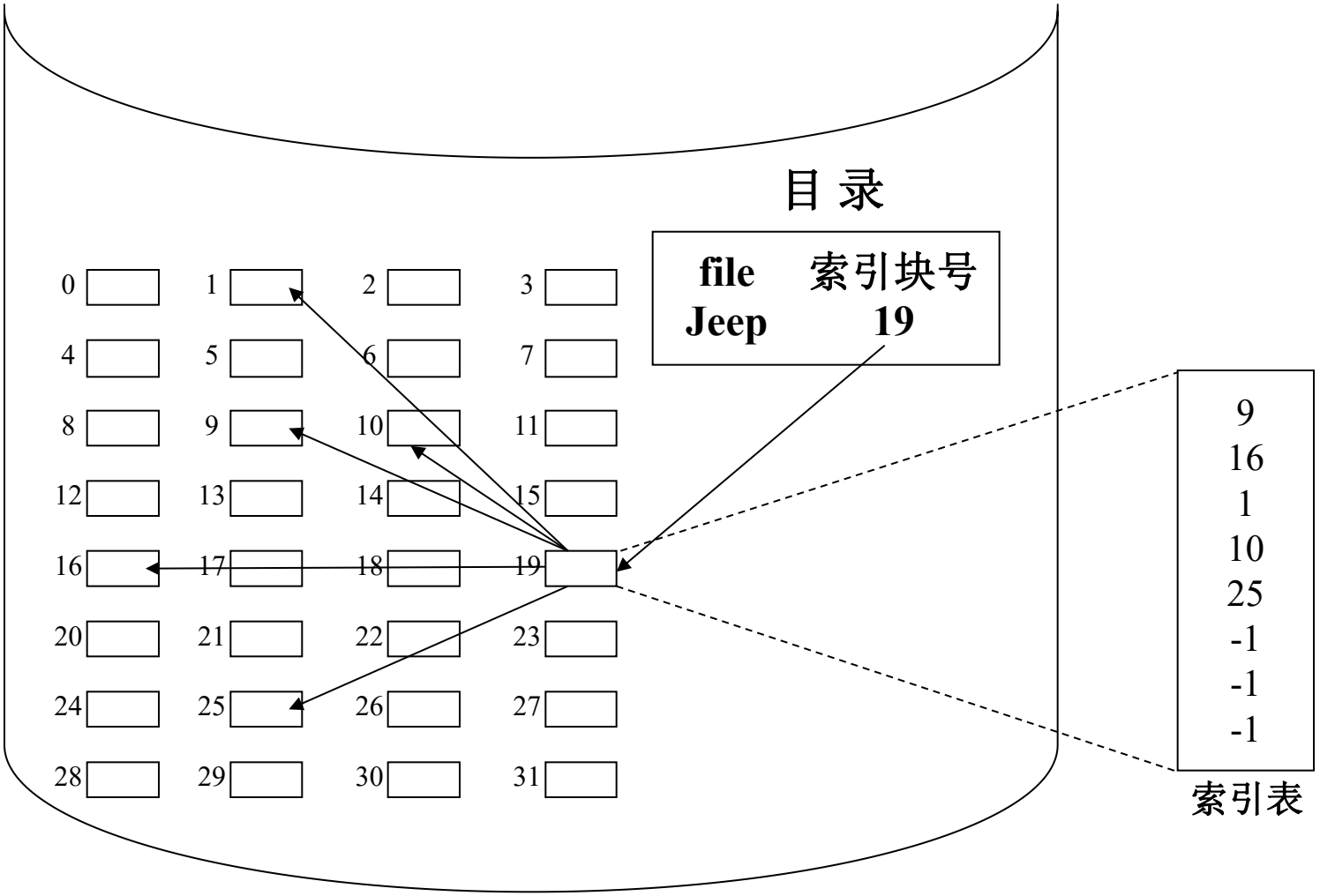

目 录
file 索引块号
Jeep 19
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
16 17 18 19
20 21 22 23
24 25 26 27
28 29 30 31
9
16
1
10
25
-1
-1
-1
索引表

图8-6 索引分配方式

2. 多级索引分配方式

在为一个大文件分配磁盘空间时，如果所分配出去的盘块的盘块号已经装满一个索引块时，OS须再为该文件分配另一个索引块，用于将以后继续为之分配的盘块号记录于其中。

依此类推，再通过链指针将各索引块按序链接起来。

当文件太大，索引块太多，这种方法是低效的。为此引入多级索引。

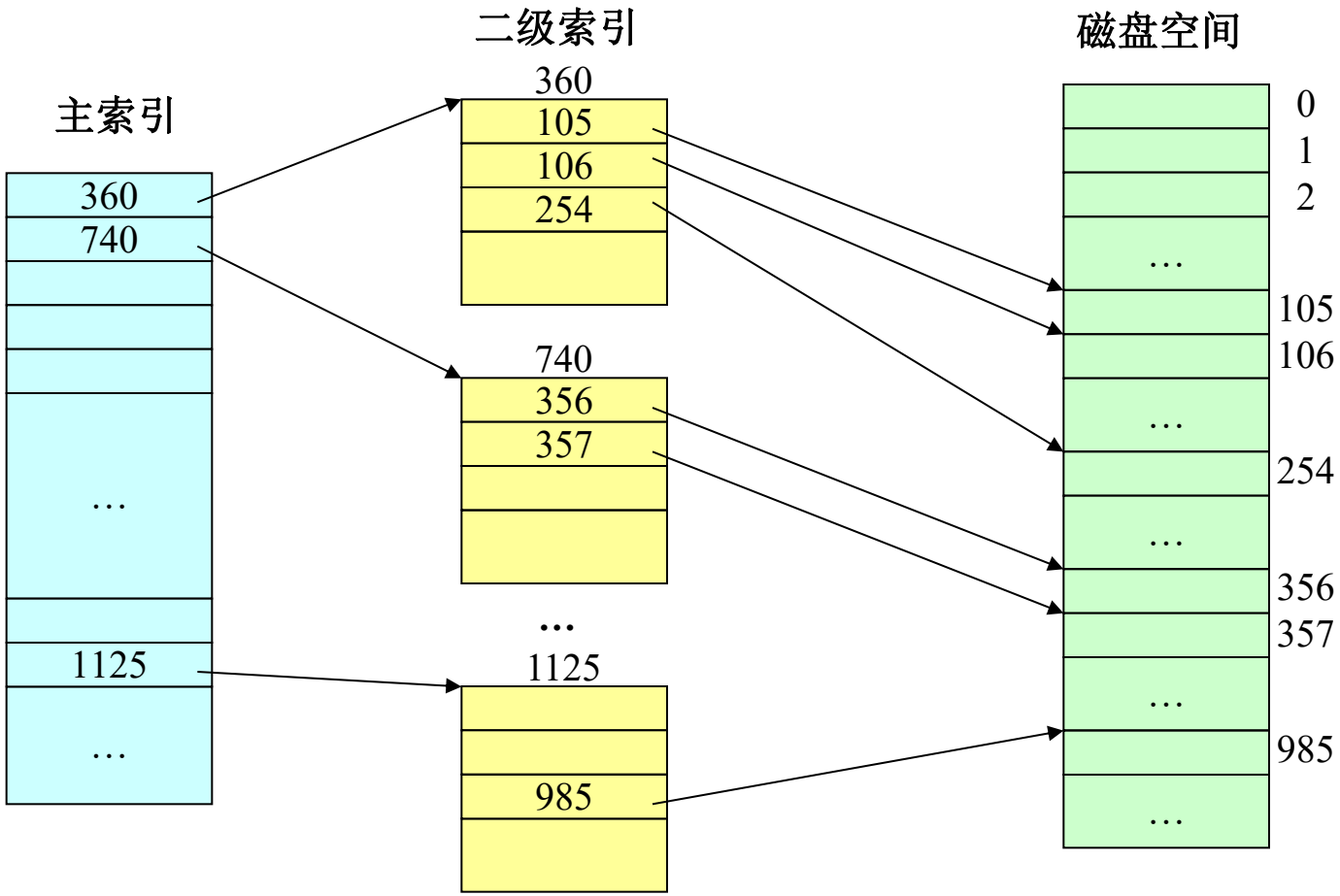

二级索引
磁盘空间
主索引
360
740
…
1125
…
360
105
106
254
740
356
357
…
1125
985
0
1
2
…
105
106
…
254
…
356
357
…
985
…

图8-7 两级索引分配

3. 增量式索引分配方式

为了能较全面地照顾到小、中、大及特大型文件，可以采取增量方式来构成文件的物理结构。

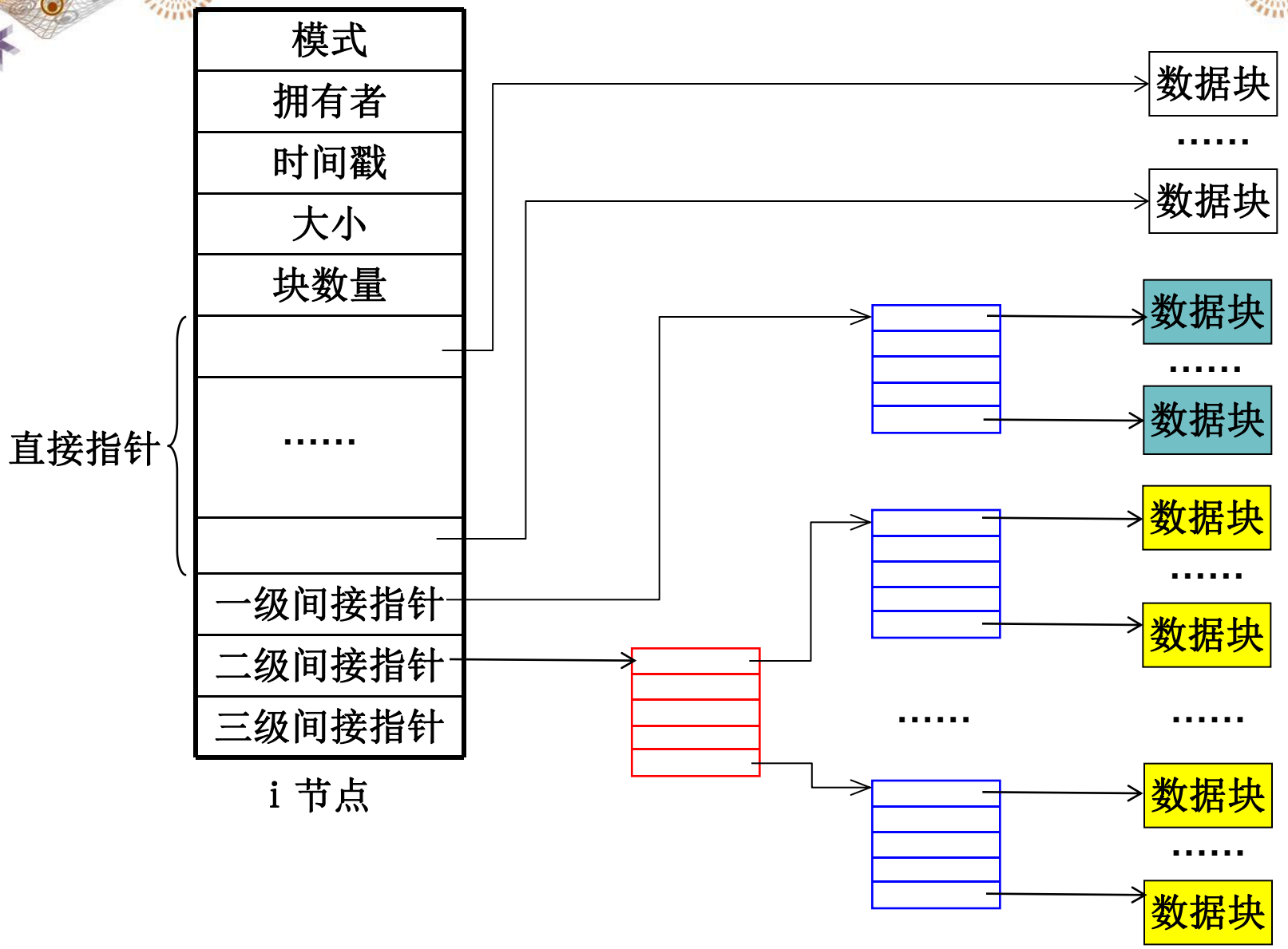

模式
拥有者
时间戳
大小
块数量
直接指针
……
一级间接指针
二级间接指针
三级间接指针
i 节点
数据块
……
数据块
数据块
……
数据块
数据块
……
数据块
……
……
数据块
……
数据块

图8-8 增量索引方式

4. 混合索引（增量式索引）题型分析

（1）给定i节点，计算最大文件（数据部分）的大小。

（2）给定文件（数据部分）的大小，计算其实际占用的磁盘块数（数据块数+索引块数）。

（3）将文件逻辑地址映射为物理地址。

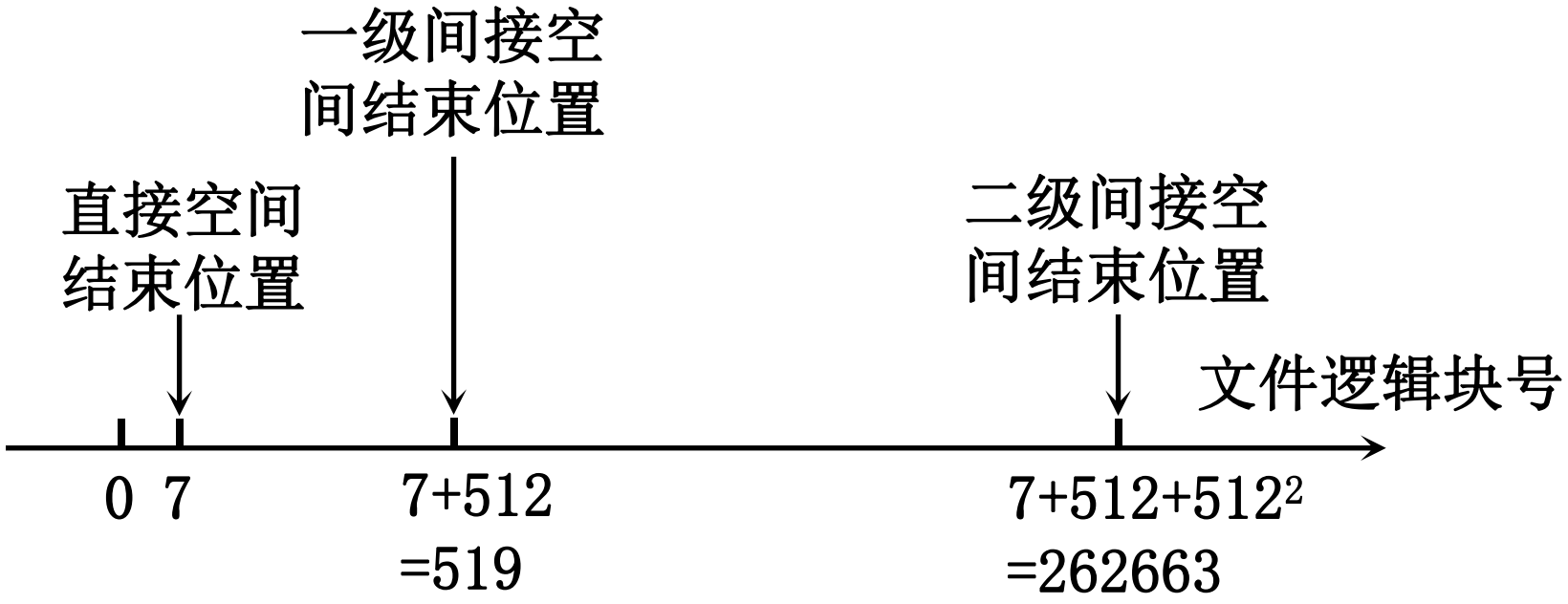
【例8.1】某操作系统的文件管理采用直接索引和多级索引的混合方式，文件索引表共有10项，其中前8项是直接索引项，第9项是一级间接索引项，第10项是二级间接索引项，假定物理块的大小是2K，每个索引项占用4个字节，试问：

（1）该文件系统中最大的文件可以达到多大？

（2）假定一个文件的实际大小是128M字节，该文件实际占用磁盘块数是多少（包括间接索引块数）？

解：本题中混合索引包括8个直接索引、一个一级间接索引项和一个二级间接索引项。

每块可以存储的索引项个数m=⌊2*1024/4⌋=512。



（1）最大文件大小为262664*2K。

（2）

① 计算文件数据部分所需块数n，

n=⌈ 128*1024*1024/（2*1024）⌉ =65536 块；

② 经判断，该文件的块数覆盖了直接空间和一级间接空间，部分覆盖了二级间接空间。

③ 直接索引不产生索引块。

④ 一级间接空间使用1个索引块。

⑤ 二级间接空间有数据块65536-8-512=65016 块，使用一级间接块⌈ 65016/512⌉ =127块，还有1个二级间接块。

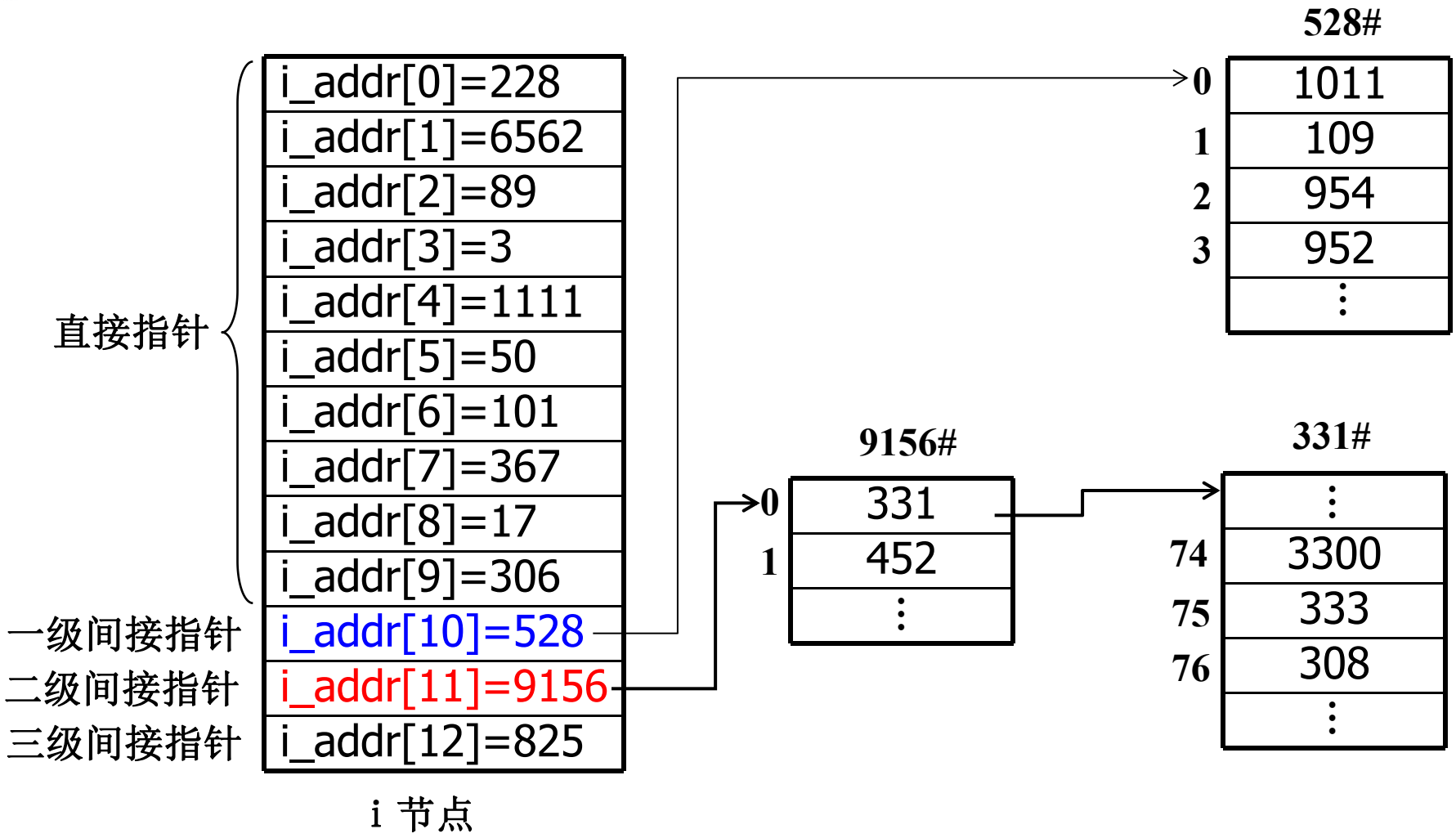
⑥ 文件实际占用磁盘块数：65536+1+127+1=65665块。

【例8.2】在UNIX文件系统中，假定磁盘块大小为1KB，每个盘块号占4B，某文件inode中的磁盘地址如下图所示，请将下列文件的字节偏移量转换为物理地址（盘块号、块内字节号）。

（1）8000

（2）13000

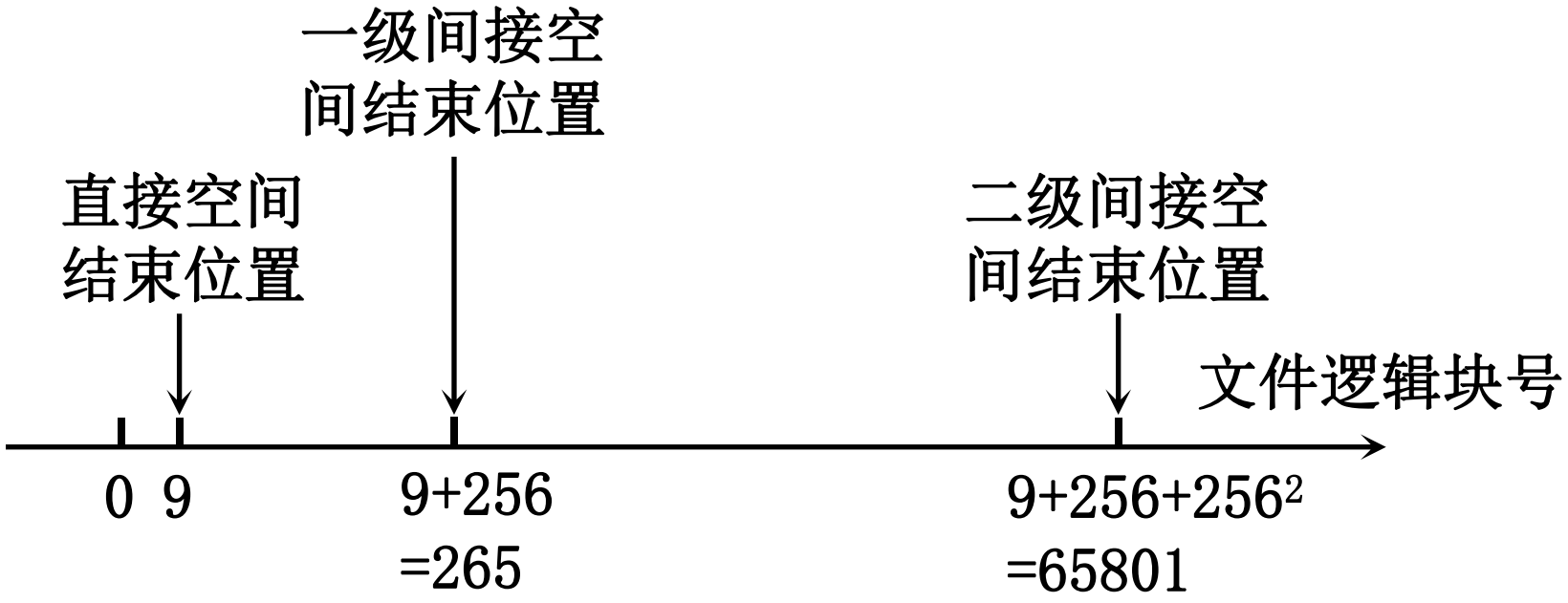
（3）350000



inode中的磁盘地址示意图

解：

每块可以存储的索引项个数m=⌊1024/4⌋=256。



（1）8000÷1024=7……832，

查表知在367块上，第832字节。

（2）13000÷1024=12……712，

12-10=2，

查表知在954块上，第712字节。

（3）350000÷1024=341……816，

341-256-10=75，

75÷256=0……75

查表知在333块上，第816字节。

8.2 磁盘空闲空间的管理

8.2.1 位示图法

1. 位示图

位示图是利用二进制的一位来表示磁盘中一个盘块的使用情况。当其值为“0”时表示对应的盘块已分配，为“1”时，表示空闲。也可反过来定义。

磁盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为位示图。

位 →

字 ↓

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	1
1	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1	1
2	0	1	0	0	1	1	0	1	0	0	1	0	1	0	1	1
3																
…																
15																

图8-10 位示图

2. 盘块的分配

(1)假设map[i, j]是首个为1的位，则其相应的盘块号b = i*n + j，n代表每个字的位数。

有些CPU有计算前导0数量的指令，可以加快j的计算。

(2)修改位示图，令map[i, j] = 0。

3. 盘块的回收

(1)将回收盘块的盘块号转换成位示图中的行号和列号。转换公式为：

i = b /n ;

j = b %n ;

(2)修改位示图，令map[i, j] = 1。

8.2.2　空闲链表法

1. 空闲盘块链

将磁盘上的所有空闲空间以盘块为单位拉成一条链，其中的每一个盘块都有指向后继盘块的指针。

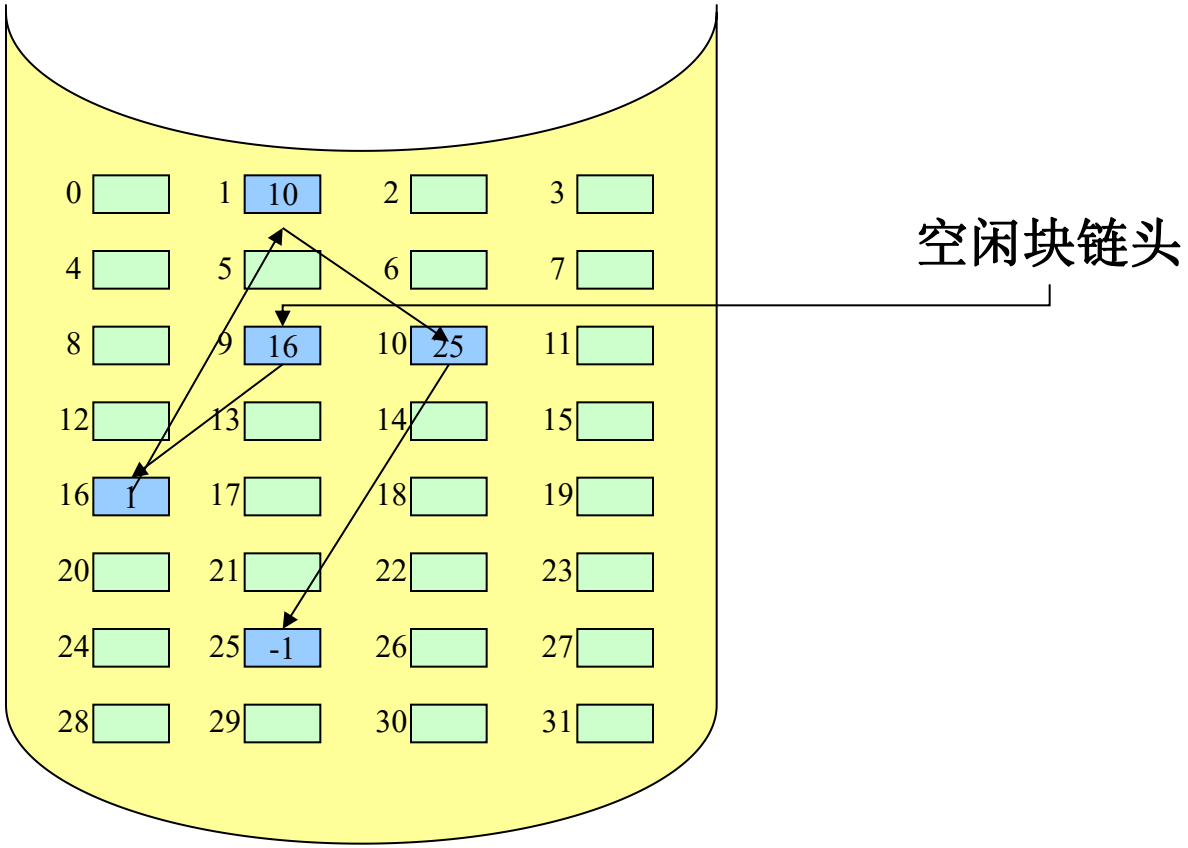

0
1
10
2
3
4
5
6
7
8
9
16
10
25
11
12
13
14
15
16
1
17
18
19
20
21
22
23
24
25
-1
26
27
28
29
30
31
空闲块链头

图8-9 空闲盘块链

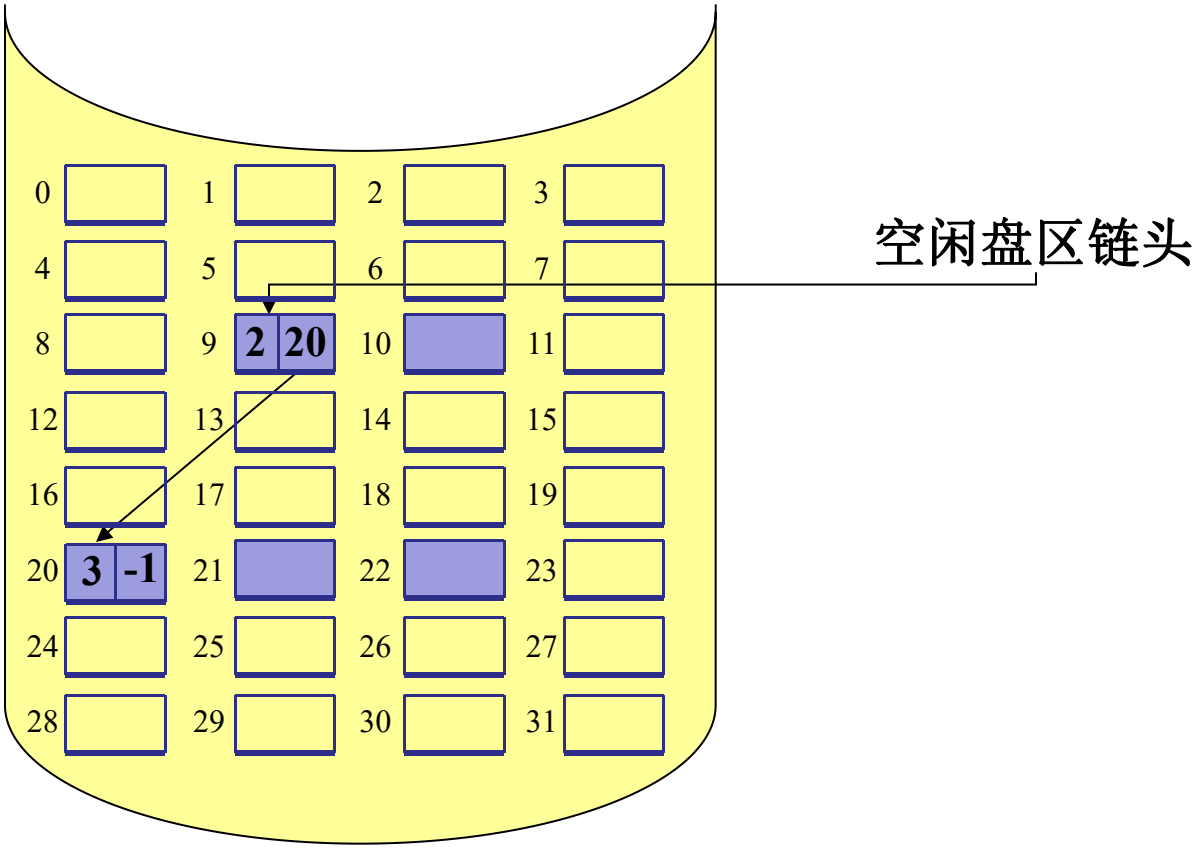
2. 空闲盘区链

将磁盘上的所有空闲盘区（每个盘区可包含若干个连续的盘块）拉成一条链。

在每个盘区上除含有用于指示下一个空闲盘区的指针外，还应有能指明本盘区大小（盘块数）的信息。

盘区的分配与回收类似于内存的动态分区分配与回收。

相比于空闲盘块链，空闲盘区链较短。


空闲盘区链头
0
1
2
3
4
5
6
7
8
9
2
20
10
11
12
13
14
15
16
17
18
19
20
3
-1
21
22
23
24
25
26
27
28
29
30
31

空闲盘区链

8.2.3 成组链接法

1. 空闲盘块的组织

空闲盘块号栈用来存放当前可用的一组空闲盘块的盘块号（最多含100个号），以及栈中尚有的空闲盘块（号）数N。

N有栈顶指针的作用。

空闲盘块号栈是专用块，系统初始时复制到内存。

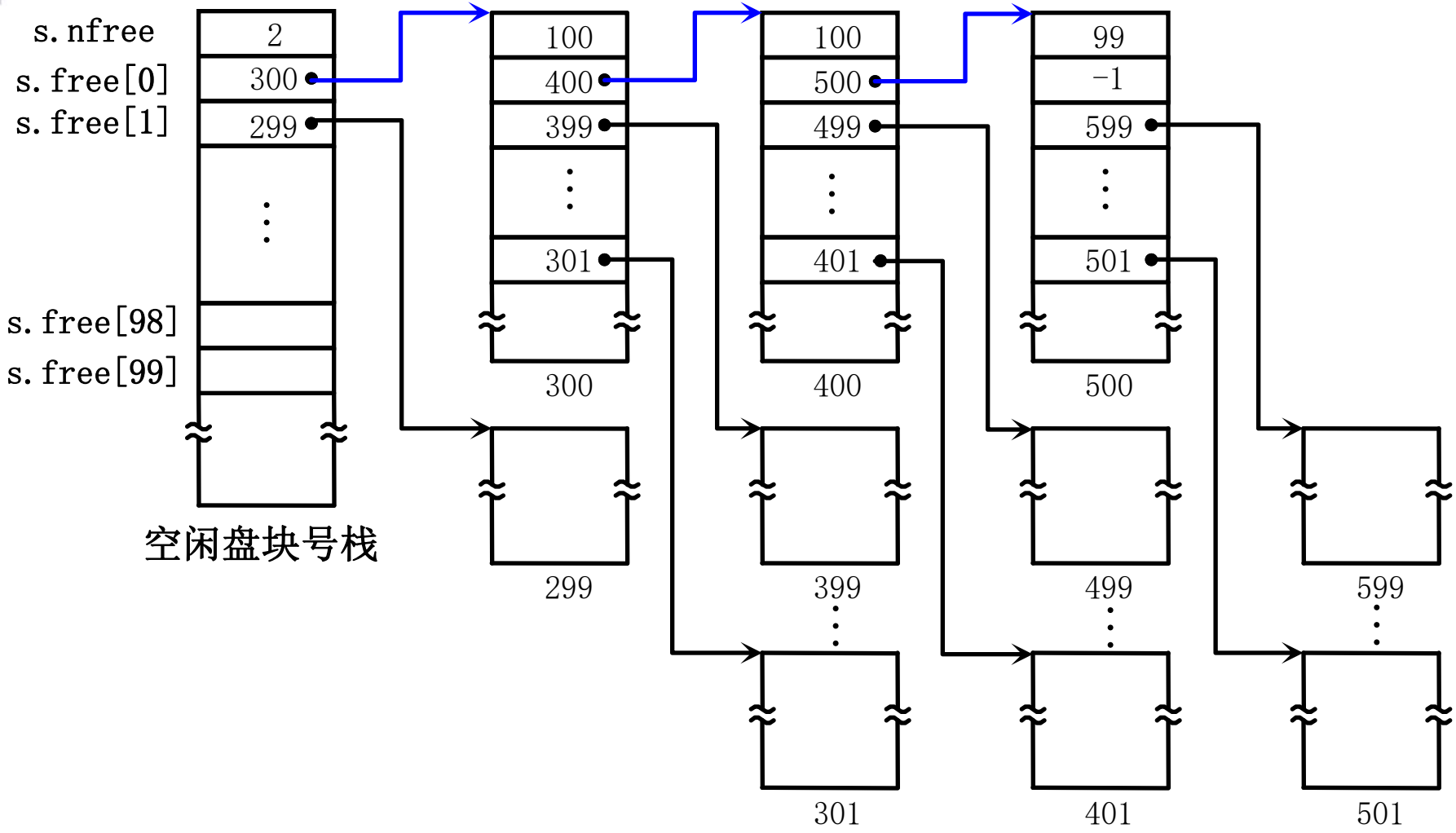


图8-11 空闲盘块的成组链接法

2. 空闲盘块的分配与回收

1）分配盘块：

（1）出栈，将栈顶盘块号分配出去；

（2）当分配到栈底盘块时，先将栈底盘块出栈，将该盘块数据读到内存空闲盘块号栈中，再将原栈底盘块分配出去。

2）回收盘块：

（1）进栈，依次将回收盘块号压入栈中；

（2）当栈满时，若再回收一个盘块，先将栈内数据写到该新回收盘块中，将栈清空，再将该回收盘块进栈。

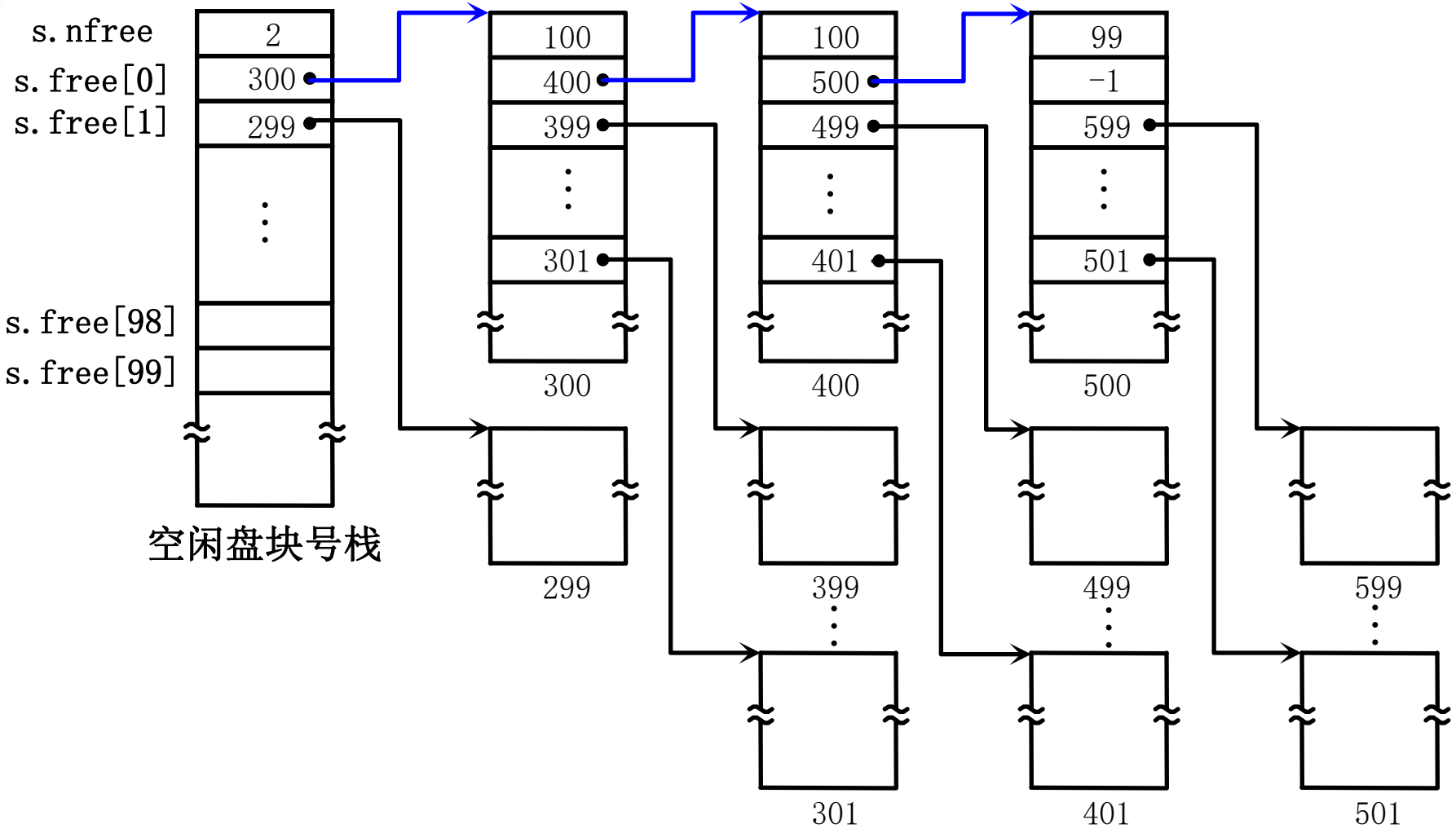
3. 成组链接法的特点

利用空闲盘块存放（暂时）链接数据，只需为空闲盘块号栈分配辅助存储空间（内存及外存）即可，且为分配、回收操作进行的读、写盘次数相对较少。

【例8.3】某系统采用成组链接法来管理磁盘的空闲空间，目前磁盘的状态如下图所示：

（1）该磁盘中目前还有多少个空闲盘块?

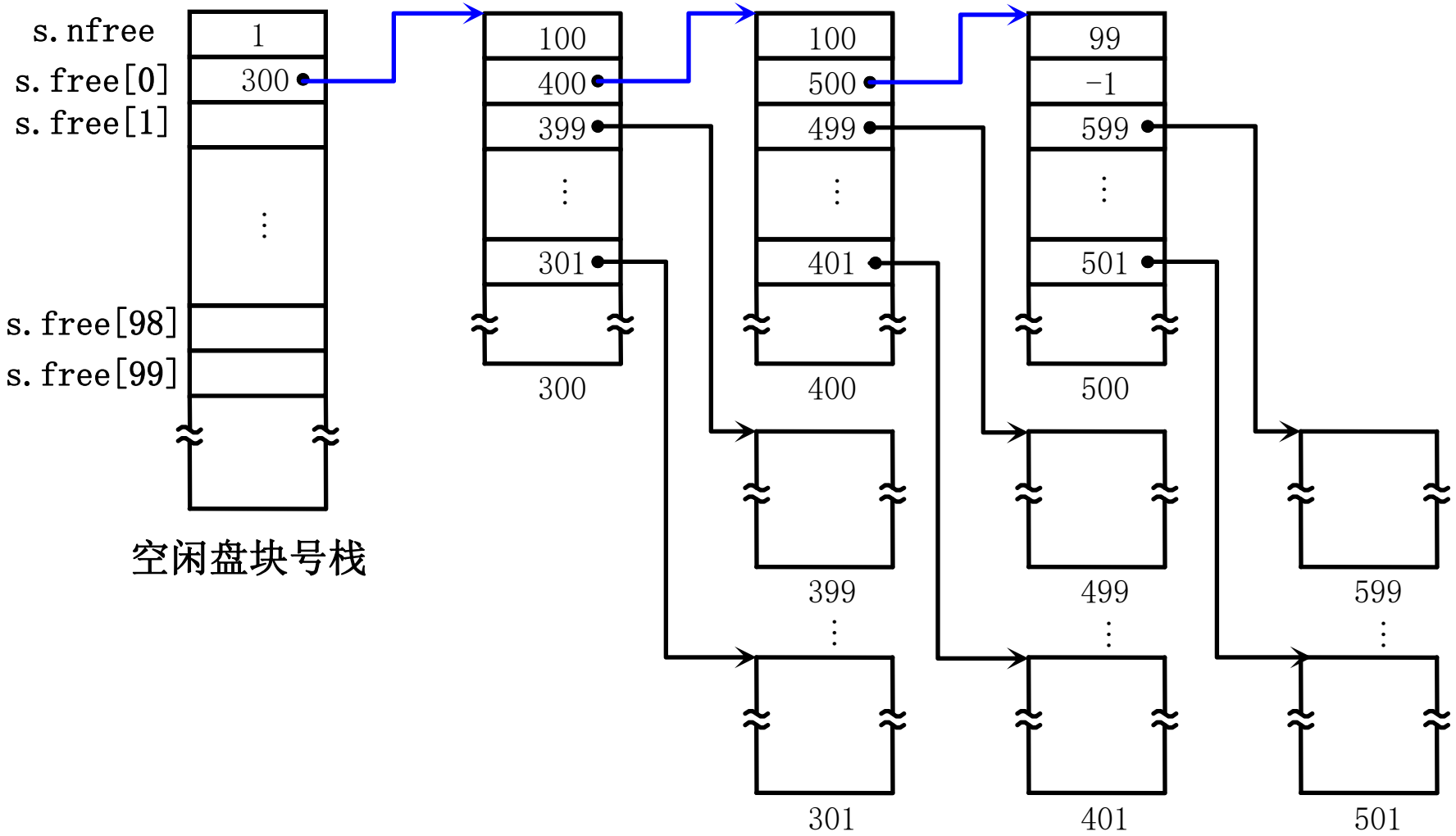
（2）在为某个文件分配3个盘块后，系统要删除另一文件，并回收它所占的5个盘块，它们的盘块号依次为700、711、703、788、701，请画出回收后的盘块链接情况。


s.nfree
s.free[0]
s.free[1]
s.free[98]
s.free[99]
2
300
299
100
400
399
301
100
500
499
401
99
-1
599
501
300
400
500
299
399
499
599
301
401
501
空闲盘块号栈

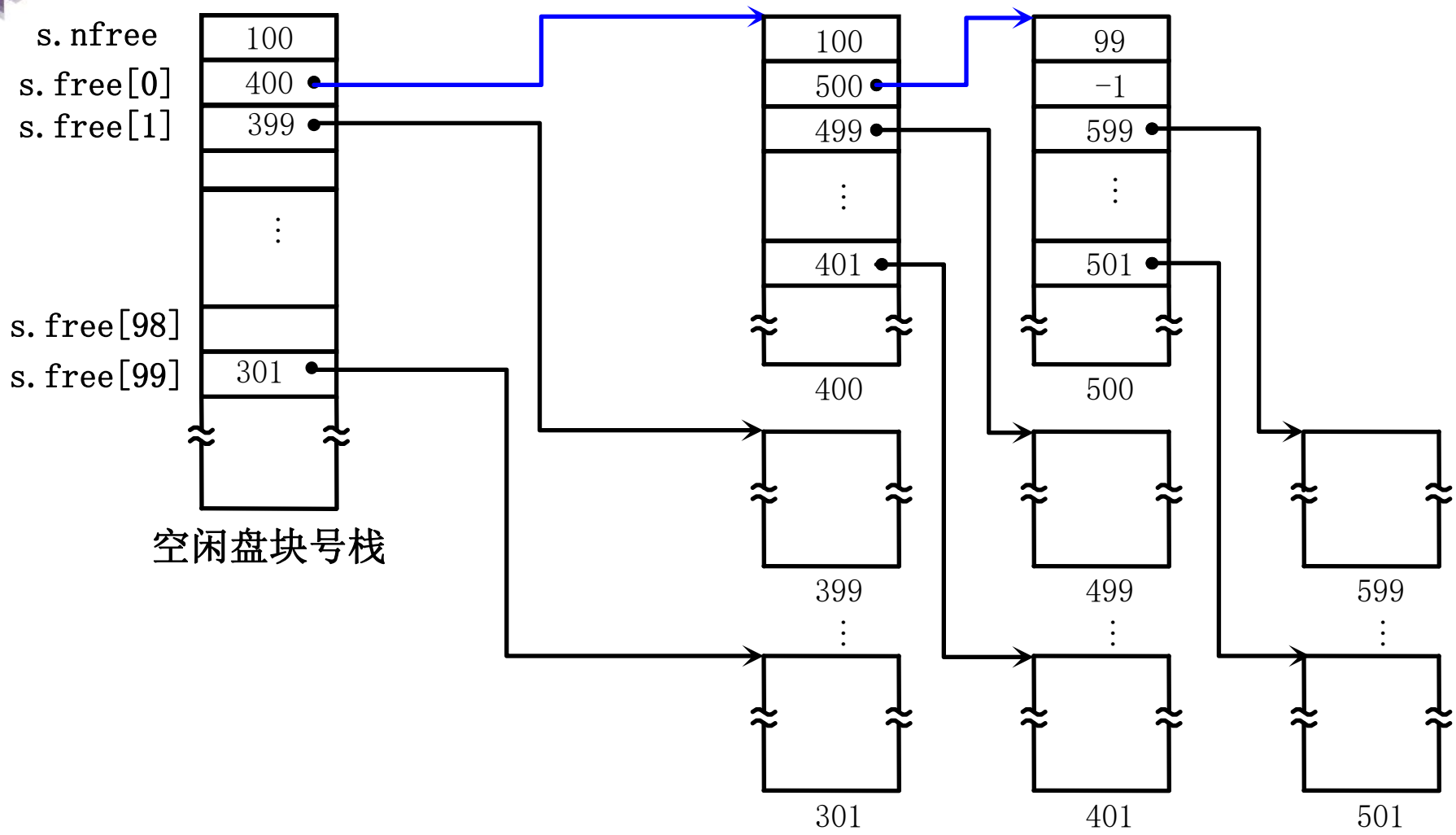
当前空闲块的情况图

解：（1）空闲块的个数：99+100×2+2=301（块）

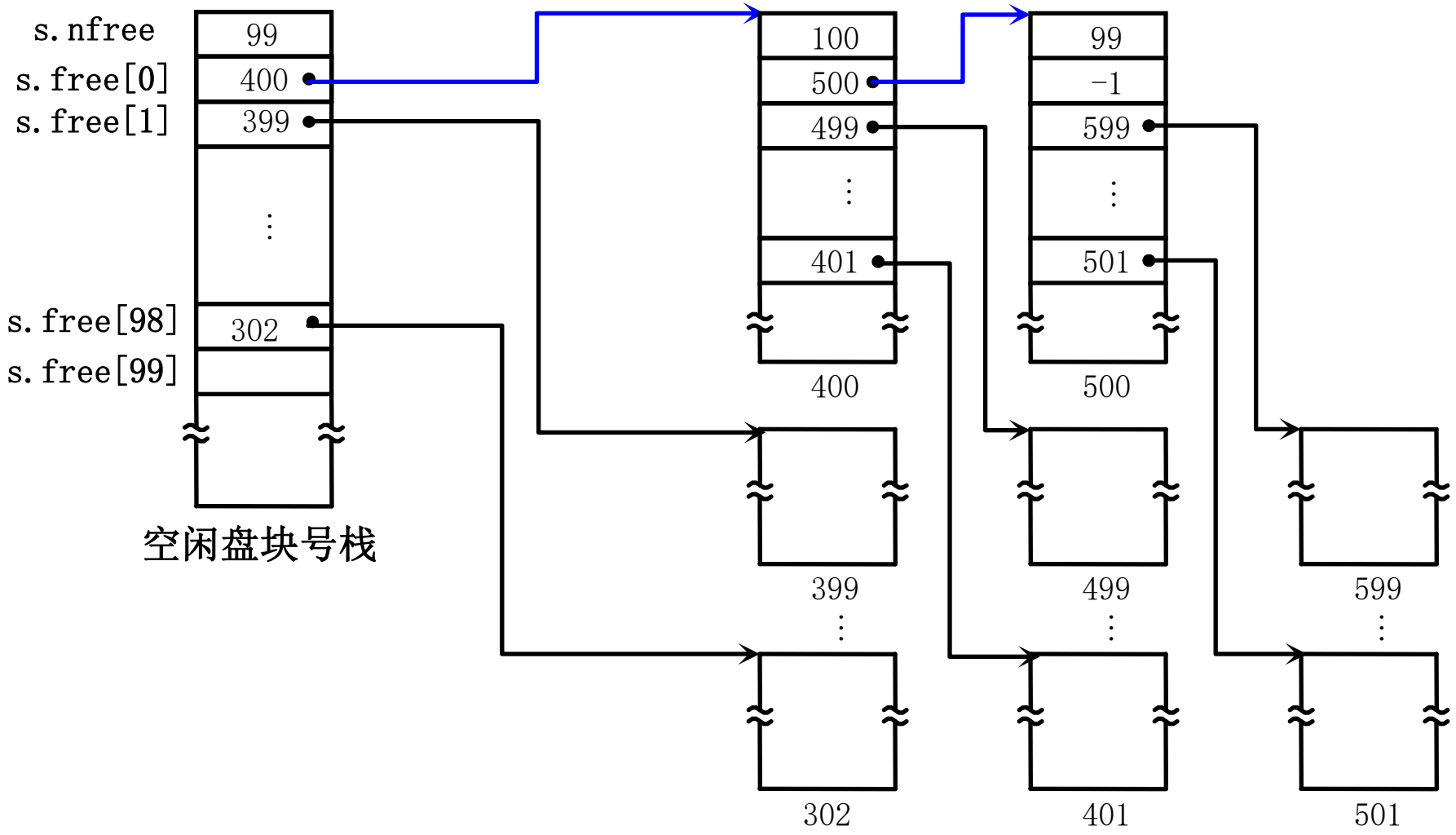
（2）



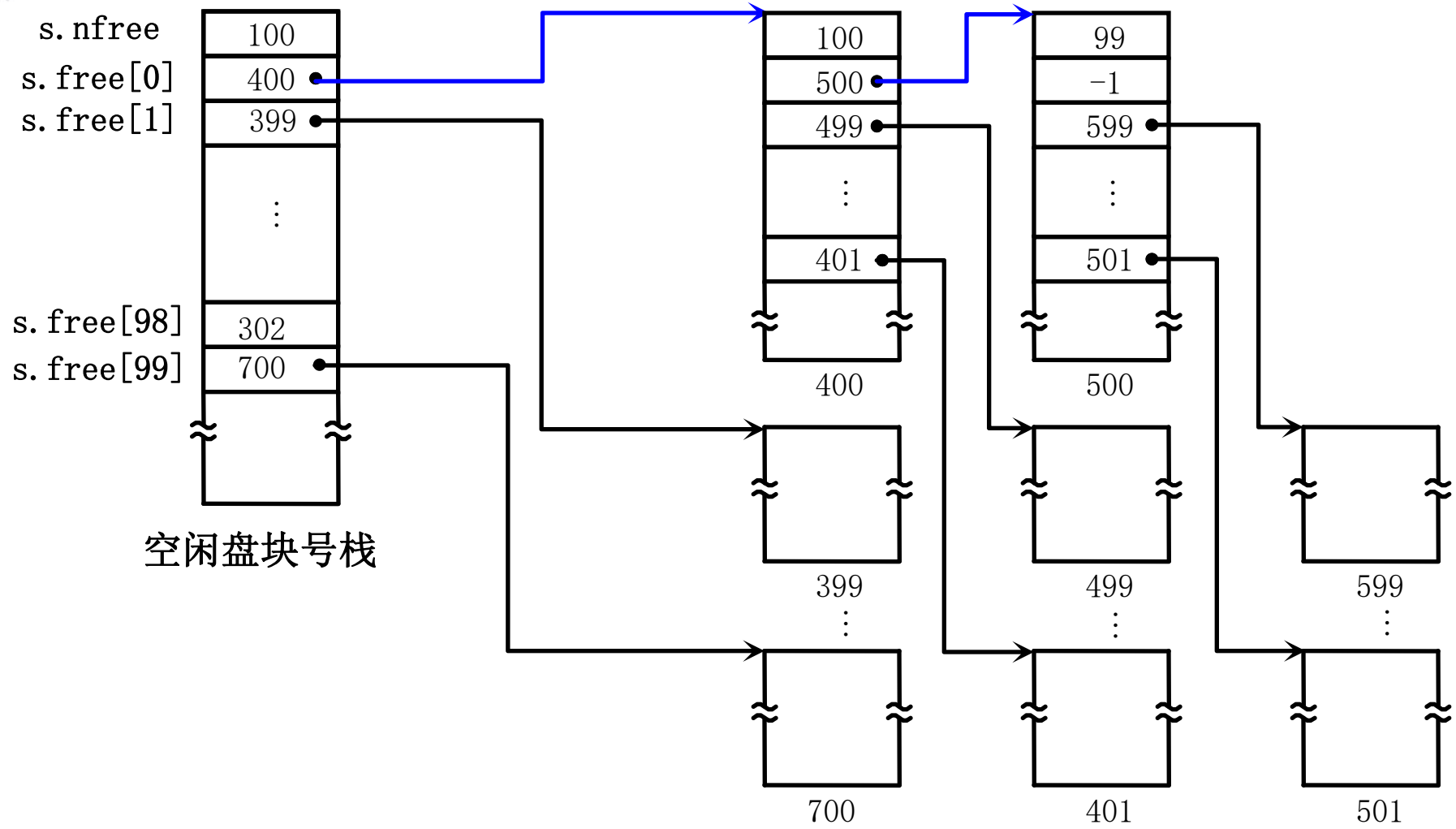
分配第1个盘块299后的成组链接图



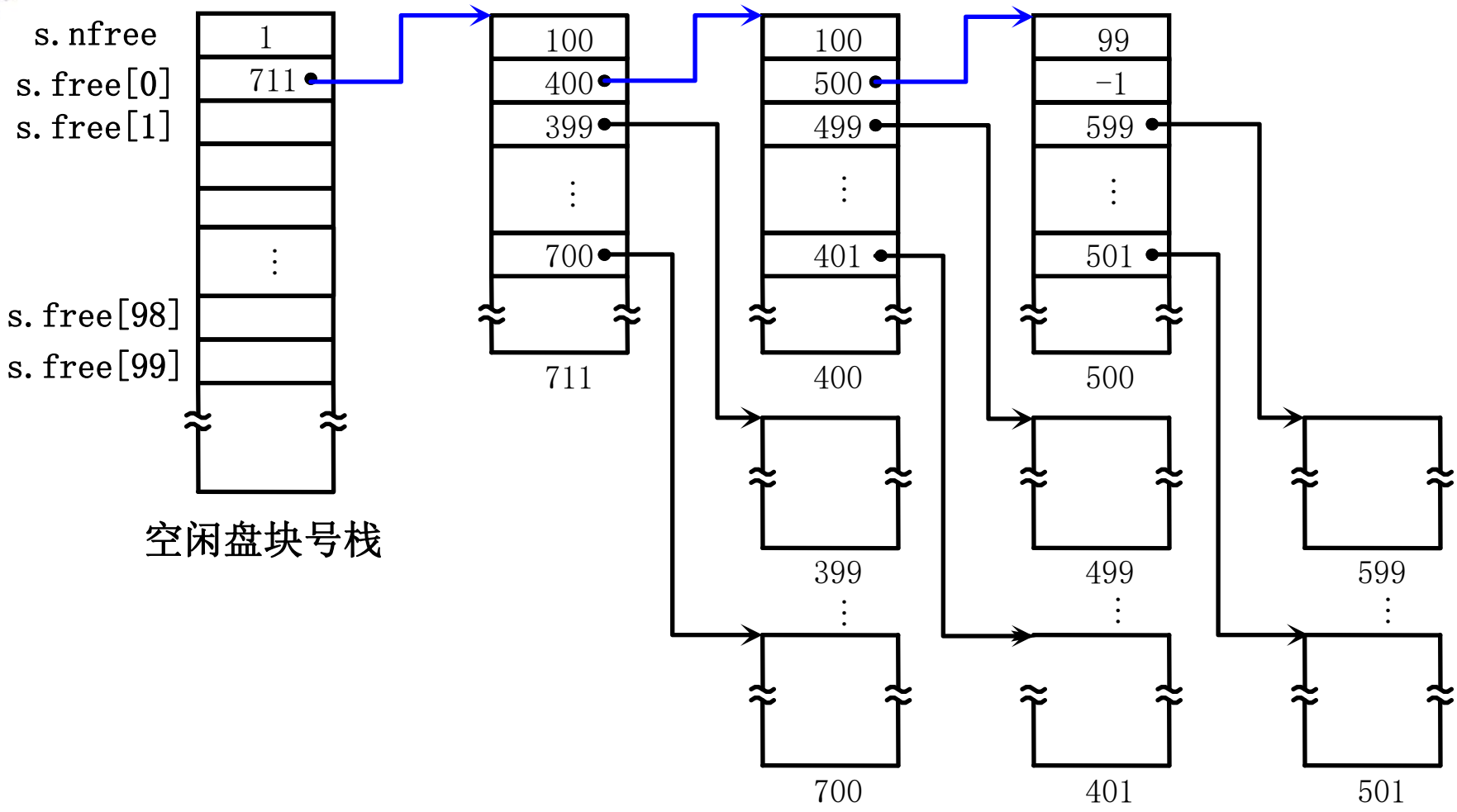
分配第2个盘块300后的成组链接图



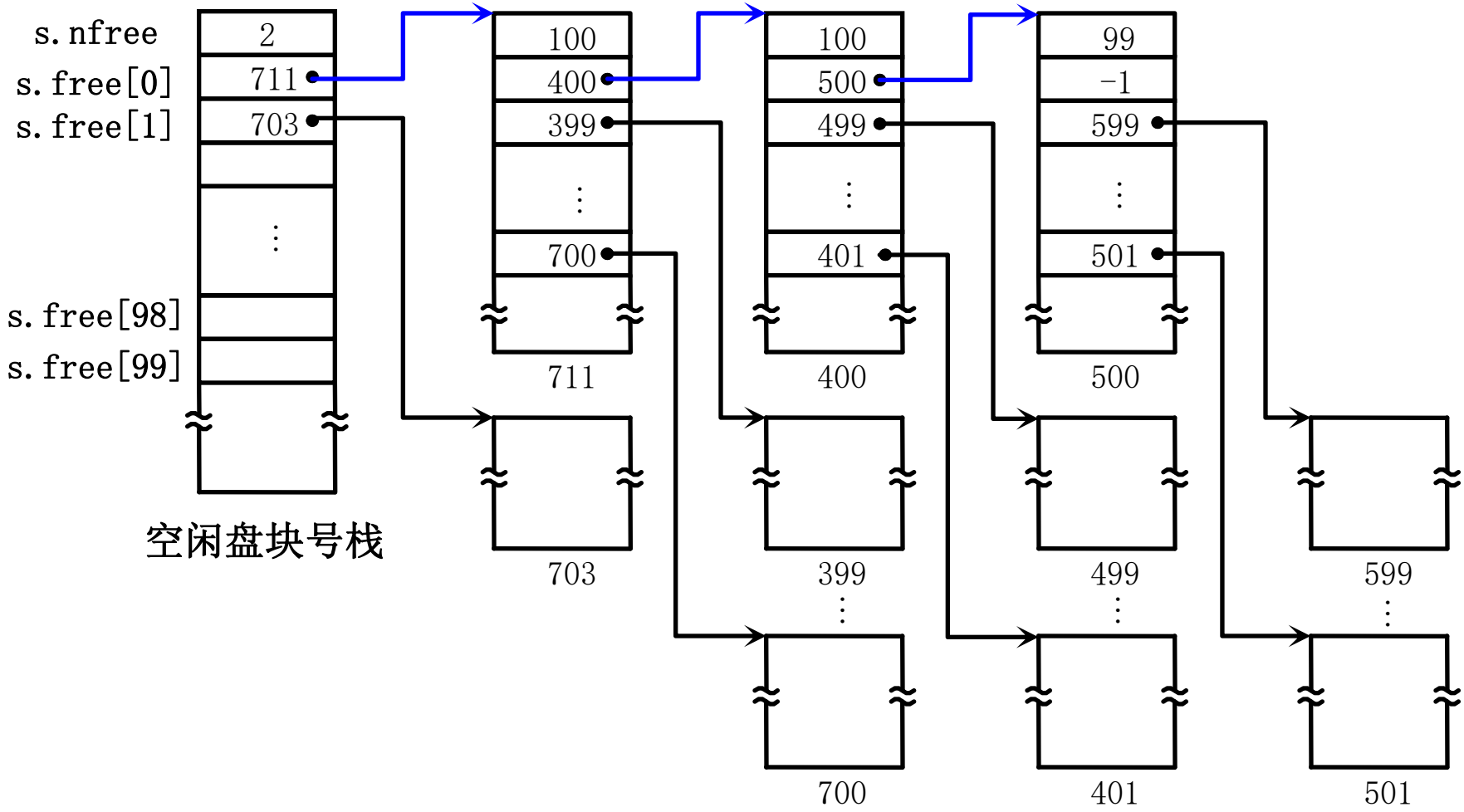
分配第3个盘块301后的成组链接图



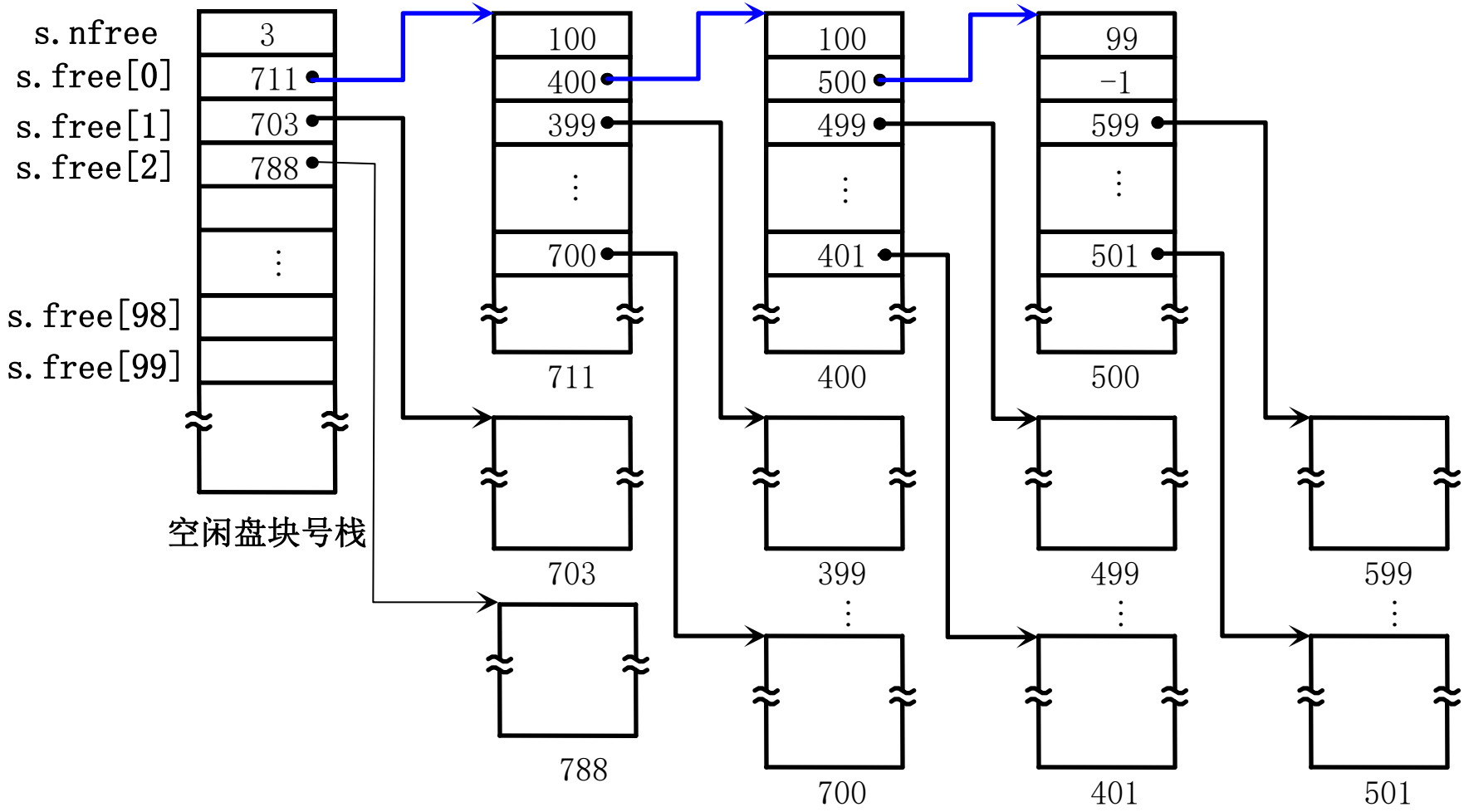
释放盘块700后的成组链接图



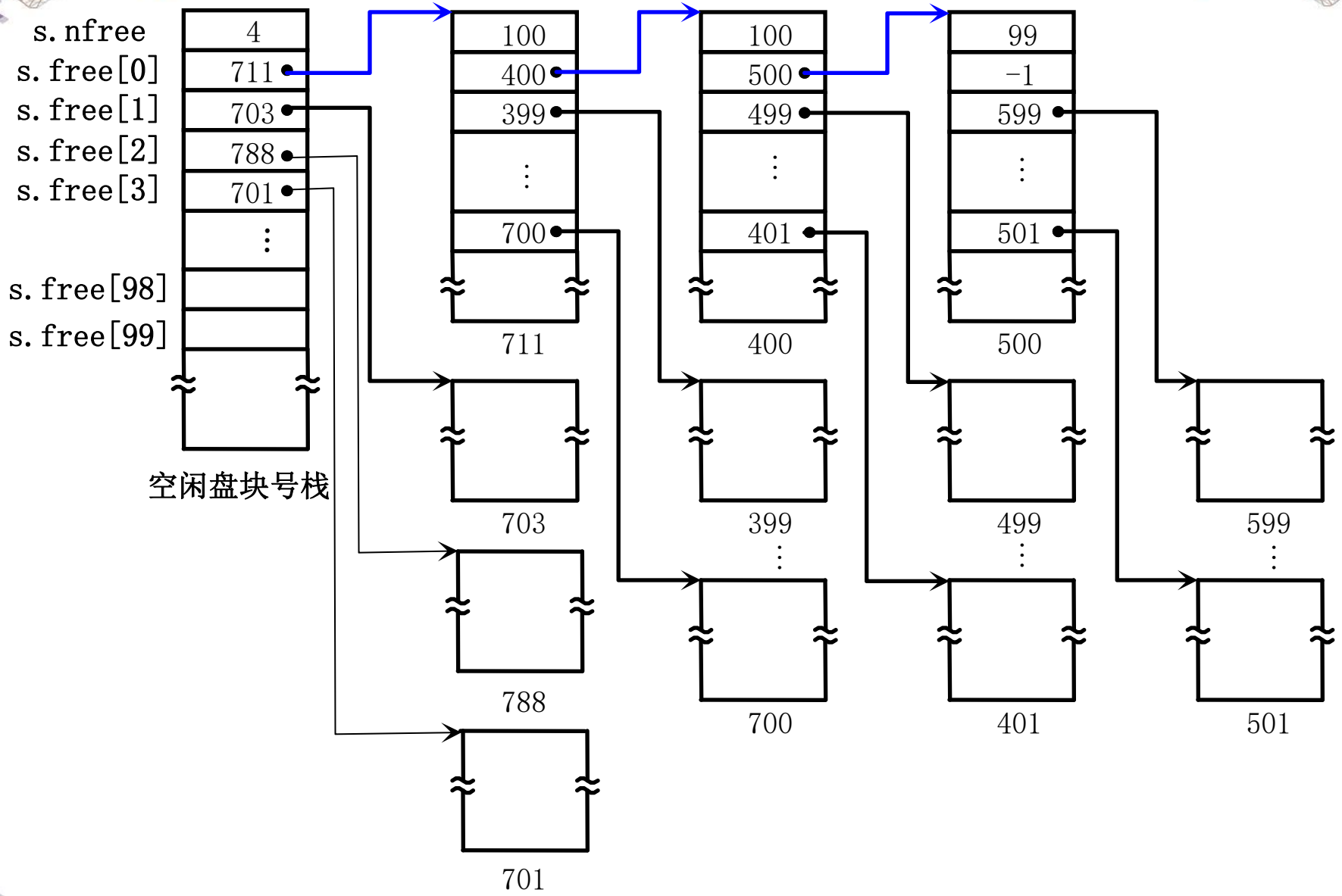
释放盘块711后的成组链接图



释放盘块703后的成组链接图



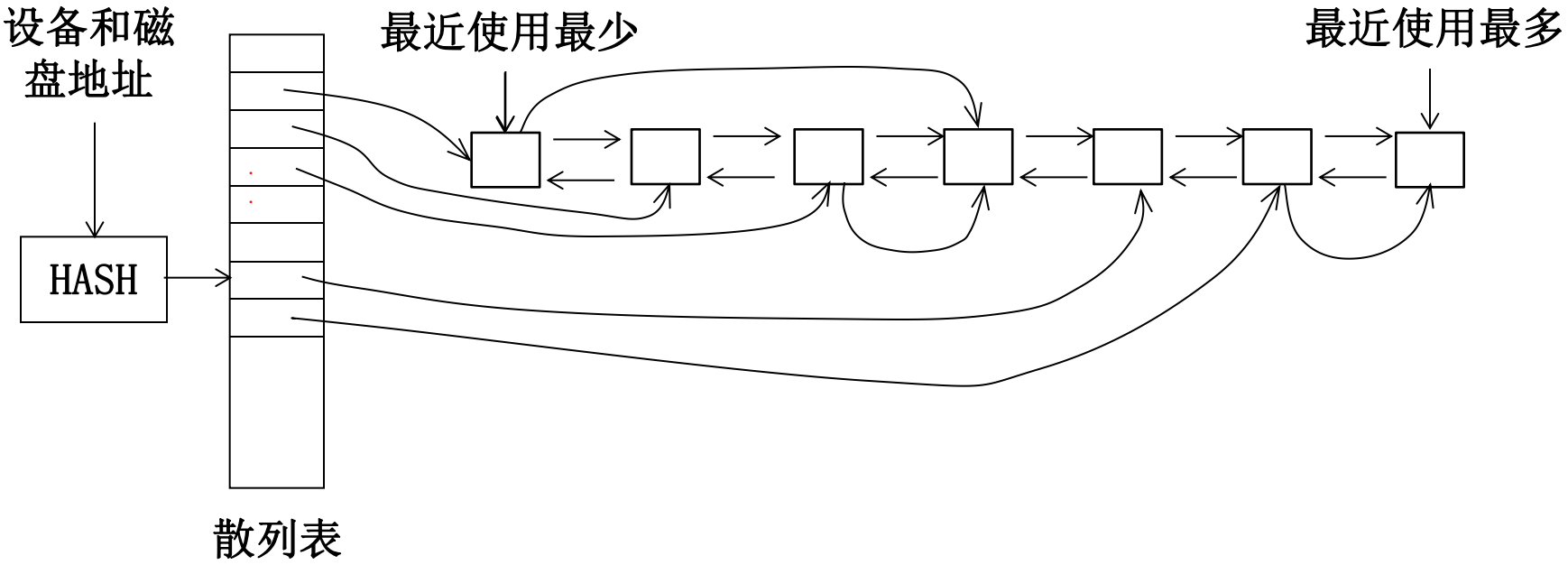
释放盘块788后的成组链接图



释放盘块701后的成组链接图

8.3 提高磁盘I/O速度的途径

1. 磁盘高速缓存(Disk Cache)



磁盘高速缓存数据结构示意图

磁盘高速缓存LRU算法需考虑以下因素：

(1)访问频率。与分页相比，对磁盘高速缓存的访问频率不高，所以可按精确的**LRU**顺序在链表中记录全部的块。

(2)可预见性。如对**i**节点块，极少可能在短时间内被访问两次，所以放在**LRU**链表的最近使用最少一端。

(3)文件系统的一致性。如**i**节点块、目录块的数据已被修改，但尚未拷回磁盘，此时发生故障，可能造成数据的不一致性，所以应立即写回磁盘。

（4）周期性地写回磁盘。根据LRU算法，那些经常要被访问的盘块数据可能会一直保留在高速缓存中，长期不会被写回磁盘，此时发生故障，可能造成用户数据的丢失，因此需要周期性地写回磁盘。

2. 提前读

预先将下一盘块数据一次读入减少读盘次数。

3. 优化物理块的分布，减少磁盘臂运动

（1）分配物理块时，把有可能顺序存取的块放在一起，尽量分布在同一柱面或相邻柱面上，从而减少磁盘臂的移动次数。

（2）减少i节点和它指向的块之间的距离。

对于固态磁盘，则不存在磁臂移动问题。

4. 虚拟盘

该盘的设备驱动程序也可以接受所有标准的磁盘操作，但这些操作的执行不是在磁盘上而是在内存中。这对用户都是透明的。