操作系统(2)
第七、第八章 内存管理 课程作业

严昕宇 20121802

上海大学 计算机工程与科学学院

1 假定一个盘组共有100个柱面，每个柱面上有16个磁道，每个磁道分成4个扇区。

1.1 整个磁盘空间共有多少个存储块?

由题意知，盘组共有100个柱面，每个柱面上有16个磁道，每个磁道分成4个扇区，则$100\times16\times4=6400$，即整个磁盘空间共有6400个存储块。

1.2 若用字长32位的单元来构造位示图，共需要多少个字?

由于用字长为 32 位的单元来构造位示图，$6400/32=200$，则共需200个字。

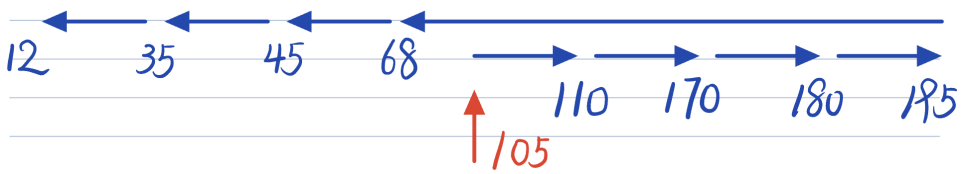
1.3 位示图中第18个字的第16位对应的块号是多少?

位示图法共分四种情况，即：行列号从0/1开始，盘块号从0/1开始

此处认为行列号从0开始，盘块号从0开始（与PPT上的例题相同），其他情况与之相似 。

则位示图中第18个字的第16位对应的存储块号为$(18-1)\times32+(16-1)=559$。

2 假设磁头当前位于第105道，正在向磁道序号增加的方向移动。现有一个磁道访问请求序列为35，45，12，68，110，180，170，195，采用SCAN调度(电梯调度)算法得到的磁道访问序列和平均寻道长度是多少?



采用SCAN调度(电梯调度)算法得到的磁道访问序列：

$$110\ \ 170\ \ 180\ \ 195\ \ 68\ \ 45\ \ 35\ \ 12$$

平均寻道长度：磁头在此过程中共移动过$(195-105)+(195-12)=273$ 个磁道，则平均寻道长度为$273/8=34.125$

3　某文件系统采用索引结点存放文件的属性和地址信息，簇大小为4KB。每个文件索引结点占64B，有11个地址项，其中直接地址项8个，一级、二级和三级间接地址项各1个，每个地址项长度为4B，请回答下列问题：

3.1　该文件系统能支持的最大文件长度是多少？(给出计算表达式)

簇(物理块)大小为4KB，地址项大小为4B，则每个物理块可存储的地址数为4KB/4B = 1024。

由题意知，每个文件索引结点占64B，有11个地址项，其中直接地址项8个，一级、二级和三级间接地址项各1个，则最大文件的物理块数可达$8+1024+1024^2+1024^3$，且每个物理块大小为4KB，因此总长度为：

$$(8+1024+1024^2+1024^3)\times 4KB = 32KB+4MB+4GB+4TB$$

该文件系统能支持的最大文件长度即为上述结果。

3.2　文件系统用1M ($1M=2^{20}$)个簇存放文件索引结点，用512M个簇存放文件数据。若一个图像文件的大小为5600B，则该文件系统最多能存放多少个这样的图像文件？

由题意知：

- 文件系统用1M ($1M=2^{20}$)个簇存放文件索引结点，则文件索引结点共有1M×4KB/64B = 64M个。
- 用512M个簇存放文件数据，且一个图像文件的大小为5600B，则一个文件占两个簇，且512M个簇可存放512M/2 = 256M个文件。

取min{64M, 256M}（木桶原理，存放个数受限于最小的），则该文件系统最多能存64M个大小为5600B的图像文件。

3.3　若文件F1的大小为6KB，文件F2的大小为40KB，则该文件系统获取F1和F2最后一个簇的簇号需要的时间是否相同？为什么？

不相同。原因如下：

- 文件F1的大小为6KB，由于6KB<4KB×8 (直接地址项8个，4KB×8为只使用直接地址索引的最大文件长度)，因此，获取文件F1的最后一个簇的簇号，只需要访问索引结点的直接地址项即可。
- 文件F2的大小为40KB，由于4KB×8<40KB<4KB×8+4KB×1024 (直接地址项8个，一级索引1个，且每个地址项长度为4B，4KB×8+4KB×1024为使用直接地址和一级索引索引的最大文件长度)。因此，获取F2的最后一个簇的簇号，还需要访问一级间接地址索引表才能得到。

  因此，文件系统获取F1和F2最后一个簇的簇号需要的时间是不相同的

4　在某个文件系统中，每个盘块为512个字节，文件控制块占64个字节，其中文件名占8个字节。如果索引结点编号占2个字节，对一个存放在磁盘上的、256个目录项的目录，试比较引入索引结点前后，为找到其中一个文件的FCB，平均启动磁盘的次数。

- 引入索引结点前

  在引入索引结点前，每个目录项中存放的是对应文件的FCB，故256个目录项的目录总共需要占用256×64/512=32个盘块。因此，在该目录中检索到一个文件，平均启动磁盘的次数为(1+32)/2=16.5次。

- 引入索引结点后

  在引入索引节点之后，每个目录项中只、需存放文件名和索引节点的编号，因此256个目录项的目录总共需要占用256× (8+2)/512=5个盘块。因此，找到匹配的目录项平均需要启动(1+5)/2，即3次磁盘，而得到索引节点编号后，还需启动磁盘将对应文件的索引结点读入内存，故平均需要启动磁盘4次（找 inode所在块 + 找文件数据所在块）。

5　在经典UNIX类操作系统中，采用混合索引结构组织文件数据块，采用成组链接法组织空闲数据块。

5.1　请论述根据文件读写指针，如何确定文件数据块在块设备上的位置。

系统为每个进程打开的每个文件维护一个读写指针，即相对于文件开头的偏移地址。读写指针指向每个文件读写的开始位置，在每次读写完成后，读写指针按照读写的数据量自动后移相应数值。

数据在块设备上的位置由块索引和块内偏移确定。块索引（sector indices）在32或64 位系统上的变量类型为sector_t。

与此同时，在Unix系统中，文件的物理结构采用索引方式。定义有一个索引节点字符数组（一张索引表），该字符数组最多可以放下13个地址项（理解成13个盘块或13个物理块），并且如下规定：

- 地址项0-9采用直接寻址方法
- 地址项10采用一级间接寻址
- 地址项11采用二级间接寻址
- 地址项12采用三级间接寻址

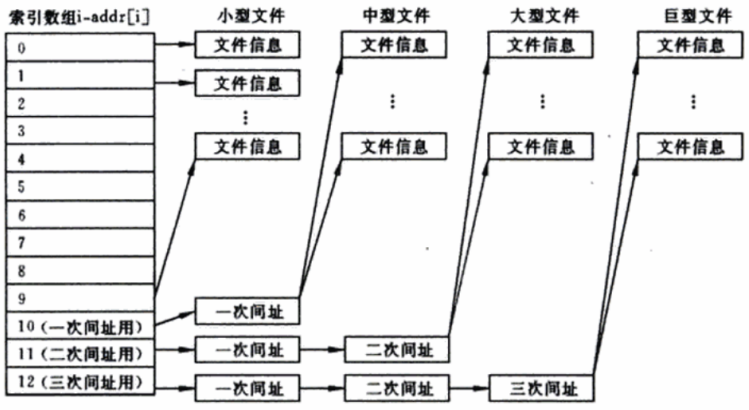


图 3-1　UNIX 成组链接文件示意图

文件系统会每个文件设置一个文件控制块 (FCB)。FCB 中的文件物理地址，以直接或间接的方式给出数据文件所在盘块的编号。

如果是直接寻址，则可直接从 FCB 中即可获得文件的盘块地址。

如果是一次间接寻址，则需要先从 FCB 中找到该文件的索引表，再从索引表中获得盘块地址。

如果是两次间接寻址或更多级的索引组织方式，则需依次读入每一级的索引表，直到查询到所需的盘块地址。

5.2 请论述混合索引结构优缺点。

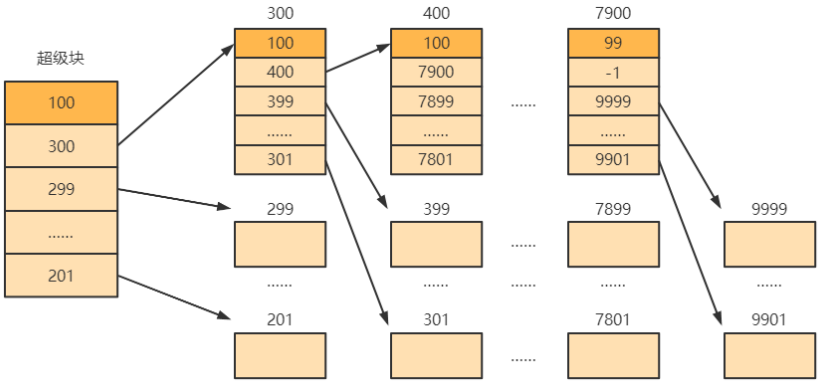
优点：能较全面照顾到小型（直接寻址）、中型（一次间址）、大型 （二级间址）和特大型文件（三级间址），提高数量众多的小型作业的访问速度（因为相比多级索引，可以采取直接寻址方法，混合索引访问磁盘的次数下降了）。对于小文件来说，访问一个数据块所需的读磁盘次数更少。

缺点：索引表需要占用一定的存储空间，存储开销变大。访问数据块前往往需要先读入索引块。若采用链接方案，查找索引块时可能需要多次读磁盘操作。而且索引的维护方式也变得不唯一，增加了复杂程度。

5.3 请论述当向文件末尾追加数据时,如何申请空闲数据块；删除文件时，如何回收空闲数据块。

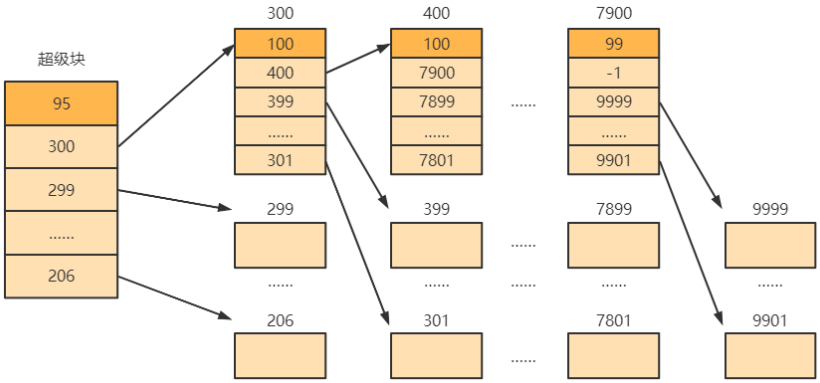
空闲表法、空闲链表法不适用于大型文件系统，因为空闲表或空闲链表可能过大，Unix系统中采用了成组链接法对磁盘空闲块进行管理。这是将上述两种方法相结合而形成的一种空闲盘块管理方法，它兼备了上述两种方法的优点而克服了两种方法均有的表太长的缺点。

文件卷的目录区中专门用一个磁盘块作为“超级块”，当系统启动时需要将超级块读入内存，并且要保证内存与外存中的“超级块”数据一致。超级块的第一个元素是下一组空闲盘块数 n，接下来跟着的是 n 个空闲块号。其中第一个空闲块指向保存下一个超级块的信息，仍然是空闲盘块数 n 和 n 个空闲块号。直到最后一个超级块的信息没有后续的空闲块号，它的第一个空闲块号为特殊值例如 -1。

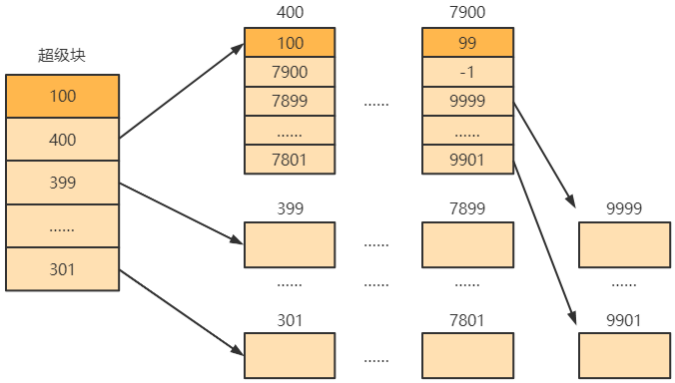


申请空闲数据块：

当需要 i 个空闲块时，检查第一个分组的块数是否足够，如果 i< n 是足够的，就分配第一个分组中的 i 个空闲块，并修改相应数据。例如上图的成组链接，若需要 5 个空闲块，可以分配 201 ~ 205 号空闲块，并修改超级块的空闲块数量为 95。



如果 i ⩾ n 刚好或不足够，则需要分配第一个分组中的全部空闲块，由于第一个空闲块存放了再下一组的信息，因此号块的数据需要复制到超级块中。换句话说，若将第一个分组全部予以分配，则需要将再下一个分组的信息复制到超级块中，否则链接就会被破坏。例如上图的成组链接，若需要 95 个空闲块，可以分配 206 ~ 300 号空闲块，由于 300 号块内存放了再下一组的信息，因此 300 号块的数据需要复制到超级块中。



回收空闲数据块:

如果需要回收空闲块，则需要把空闲块号加入超级块中，修改空闲盘块数，并且进行相应的链接。如果回收空闲块后达到了超级块空闲盘块数的上限，需要将超级块中的数据复制到新回收的块中，并修改超级块的内容，让新回收的块成为第一个分组。例如上图的成组链接，若回收 1 个空闲块 300 号，进行修改后的状态如下。