第四章课后作业参考答案

1. Consider the directory tree of **Fig. 4-8**. If */usr/jim* is the working directory, what is the absolute path name for the file whose relative path name is *../ast/x*?

   > /usr/ast/x

2. It has been suggested that efficiency could be improved and disk space saved by storing the data of a short file within the i-node. For the i-node of **Fig. 4-13**, how many bytes of data could be stored inside the i-node?

   > 如果属性中有剩余位，可以有 9 个指针来处理数据，如果每个指针都是 k 字节，那么可以存储 9k 个字节的数据；如果属性中没有剩余位，则第一个磁盘地址可以保存无效地址，那么可以存储 8k 个字节的数据。

3. Explain how hard links and soft links differ with respective to i-node allocations.

   > 硬链接情况下，多个文件名指向同一个 i-node 号码，inode 信息中的"链接数"属性 nlink 记录指向该 i-node 的文件名总数
   >
   > 在软链接情况下，被链接的文件指向不同 的 i-node，例如，当文件 A 被软链接到文件 B 上时，文件 A 和文件 B 的 inode 号码不 一样，文件 A 的内容是文件 B 的路径。读取文件 A 时，系统会自动将访问者导向文件 B。

4. Free disk space can be kept track of using a free list or a bitmap. Disk addresses re- quire *D* bits. For a disk with *B* blocks, *F* of which are free, state the condition under which the free list uses less space than the bitmap. For *D* having the value 16 bits, express your answer as a percentage of the disk space that must be free.

   > 位示图 (bitmap) : 每一字位对应一个物理块, 磁盘有 B 个块, 位示图需要 B 位.
   >
   > 空闲块链 (free list) : 各个空闲块地址是离散的, 需要存储空闲块的地址, 留出一个地址作为块间链接，每个地址为 D 位, 有 F 个空闲块, 需要 DF 位. 空闲块链占用空间小于位示图条件: DF < B 当 D = 16, F/B < 1/16 = 6.25%

5. The beginning of a free-space bitmap looks like this after the disk partition is first for- matted: 1000 0000 0000 0000 (the first block is used by the root directory). The system always searches for free blocks starting at the lowest-numbered block, so after writing file A, which uses six blocks, the bitmap looks like this: 1111 1110 0000 0000. Show the bitmap after each of the following additional actions: (a) File B is written, using five blocks. (b) File A is deleted. (c) File C is written, using eight blocks. (d) File B is deleted.

   > (a) File B is written, using five blocks.
   >
   > 1111 1111 1111 0000

> (b) File A is deleted.
>
> 1000 0001 1111 0000
>
> (c) File C is written, using eight blocks.
>
> 1111 1111 1111 1100
>
> (d) File B is deleted.
>
> 1111 111 0 0000 1100

6. What would happen if the bitmap or free list containing the information about free disk blocks was completely lost due to a crash? Is there any way to recover from this disaster, or is it bye-bye disk? Discuss your answers for UNIX and the FAT-16 file system separately.

> 计算机会对空闲链表或位示图进行恢复，恢复后不会产生任何后果。只需列出所有文件使用的所有块，再将未被列出的块挂在空闲链表的链尾中即可恢复空闲块表；只需 新建一个全为0的位示图，将所有文件使用的所有块对应的位变为1即可。
>
> 由于UNIX 使 用的是成组链表法，本质与空闲链表相同，只要恢复即可；而 FAT-16 文件系统使用 FAT 表记录未被分配的簇，因此即使空闲链表或位示图崩溃也不会产生任何影响。

7. Consider Fig. 4-27. Is it possible that for some particular block number the counters in both lists have the value 2? How should this problem be corrected?

> 在这种情况下，说明某个块会出现在两个文件中，并且在空闲块表中也出现两次。 解决方法是： 将空闲计数置0并将这个块在空闲块表种删除，接着获取一个空闲块，并将该块的内容复制到获取的空闲块中。最后将错误块所在的其中一个文件中的该块替换为这个新块，两个块的使用计数都置为1

8. Consider a disk that has 10 data blocks starting from block 14 through 23. Let there be 2 files on the disk: f1 and f2. The directory structure lists that the first data blocks of f1 and f2 are respectively 22 and 16. Given the FAT table entries as below, what are the data blocks allotted to f1 and f2? (14,18); (15,17); (16,23); (17,21); (18,20); (19,15); (20, −1); (21, −1); (22,19); (23,14). In the above notation, (*x*, *y*) indicates that the value stored in table entry *x* points to data block *y*.

> f1: 22, 19, 15, 17, 21；
>
> f2: 16, 23, 14, 18, 20。

9. A UNIX file system has 4-KB blocks and 4-byte disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and triple indirect entry each?

> 1KB=1024B，4-KB/4-byte=1024，则大小为 4KB 的磁盘块可以容纳的磁盘地址个数是 1024 个
>
> 10个直接索引有10个地址, 一个间接索引有1024个地址, 共有10+1024+1024^2+1024^3 = 1074791434 个地址，每个地址指向一个4KB的块，则文件的最大尺寸为 1074791434*4KB = 4299165736KB，约为 4100GB。

10. How many disk operations are needed to fetch the i-node for a file with the path name /usr/ast/courses/os/handout.t? Assume that the i-node for the root directory is in memory, but nothing else along the path is in memory. Also assume that all directories fit in one disk block.

> (1) 根据根目录的 inode 从磁盘读取根目录文件到内存. –directory for /
>
> (2) 检索根目录, 将 usr 目录的磁盘 inode 复制到活动 inode 表中. –i-node for /usr
>
> (3) 根据 usr 目录的 inode 从磁盘读取 usr 目录文件到内存. –directory for /usr
>
> (4) 检索 usr 目录, 将 ast 目录的磁盘 inode 复制到活动 inode 表中. –i-node for /usr/ast
>
> ...
>
> (9) 根据 os 目录的 inode 从磁盘读取 os 目录文件到内存. –directory for /usr/ast/courses/os
>
> (10) 检索 os 目录, 将 handout.t 文件的磁盘 inode 复制到活动 inode 表中. –i-node for /usr/ast/courses/os/handout.t
>
> 共10次磁盘操作.