OS 第四章作业

191220008 陈南疃

问答与计算

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*?

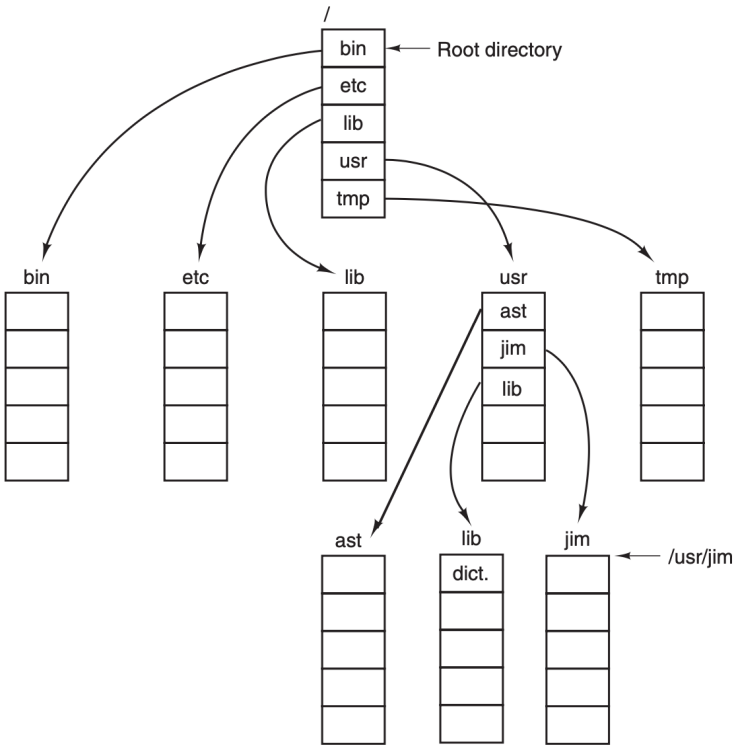

Figure 4-8. A UNIX directory tree.

`../` 代表上一级目录，所以

绝对路径的文件名为：`/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?

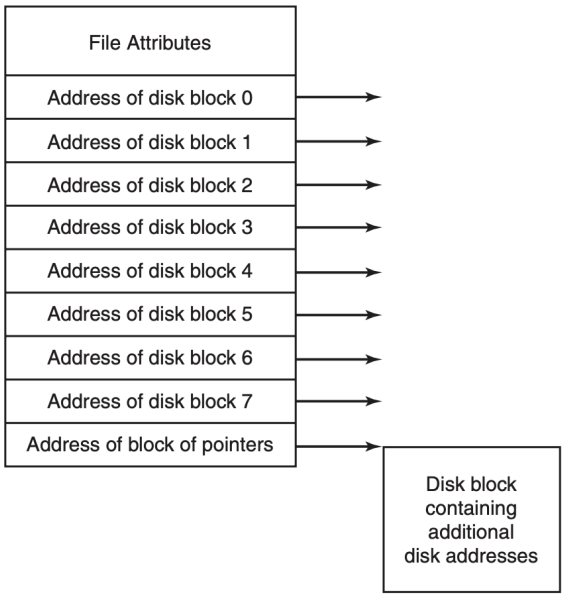

Figure 4-13. An example i-node.

需要有一种方法来表示地址块指针保存的是数据，而不是指针。 如果属性中有空余的空间，则可以使用它。

将设每个指针都是 k 个字节。

① 如果属性中有空余的空间，则可以用于说明保存的是数据。此时所有的九个指针都可以存储数据。则存储的文件最长可达 9k 字节。

② 如果属性中没有空余的空间，则第一个磁盘地址用于说明后面保存的是数据。则存储的文件最长可达 8k 字节。

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

硬链接：对于给定的硬链接的所有目录项指向单个 inode 文件。

软链接：会为软链接创建一个新的 inode，该 inode 本质上指向被链接的原始文件。

4、Free disk space can be kept track of using a free list or a bitmap. Disk addresses require *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.

bitmap 需要 B bits， free list 需要 DF bits。

如果 DF < B，free list 占用空间更小。

对于 16 bits 的磁盘地址，如果磁盘空闲空间 <= 6%，则 free list 会更短。

5、The beginning of a free-space bitmap looks like this after the disk partition is first formatted: 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) 1111 1111 1111 0000

(b) 1000 0001 1111 0000

(c) After writing file C: 1111 1111 1111 1100

(d) 1111 1110 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.

这不是一个严重的问题。恢复算法是将所有文件中的所有块做一个列表，取补码作为新的空闲列表。

UNIX：可以通过扫描所有 inode 来完成。

FAT：由于没有free list，因此不会出现此问题。 即使有，恢复它所要做的就是扫描 FAT 以寻找空闲项。

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?

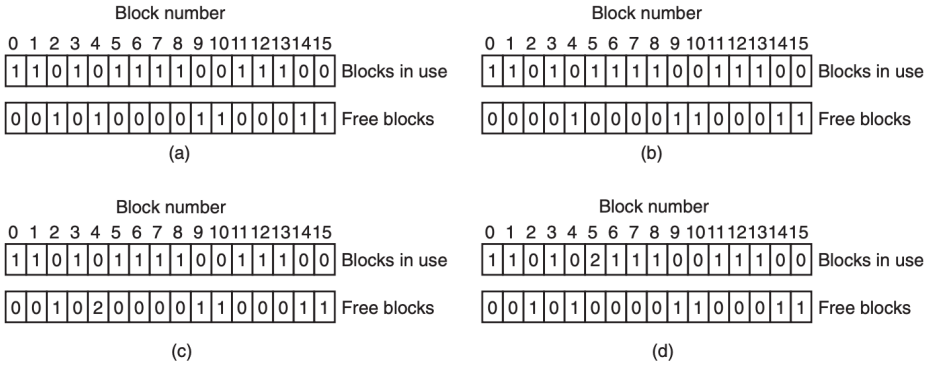

Figure 4-27. File-system states. (a) Consistent. (b) Missing block. (c) Duplicate block in free list. (d) Duplicate data block.

这种情况不应该发生，但如果出错则可能会发生。 这意味着某个块出现在两个文件中，并且在 free list 中出现两次。

修复的第一步是从空闲列表中删除两个副本。 接下来必须获得一个空闲块，并将出现问题的块的内容复制到那里。 最后，应更改其中一个文件的块以引用新获取的块副本。 此时系统又变得一致。

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?

i-node 有 10 个直接索引指针。 有 2^10 个一级间接索引指针，有 2^20 个二级间接索引指针，有 2^30 个三级间接索引指针。

所以，最大文件大小为 10+2^10+2^20+2^30 = 1074791434 个文件控制块，约为 1074791434 * 16 / 2^30 = 16.02 GB。

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. / 的目录文件
2. /usr 的 inode
3. /usr 的目录文件
4. /usr/ast 的 inode
5. /usr/ast 的目录文件
6. /usr/ast/courses 的 inode
7. /usr/ast/courses 的目录文件
8. /usr/ast/courses/os 的 inode
9. /usr/ast/courses/os 的目录文件
10. /usr/ast/courses/os/handout.t 的 inode

总共需要10个操作。