OS第四章课后作业

姓名	学号	院系	邮箱
张洋彬	191220169	计算机科学与技术系	1016466918@qq.com

一、问答题

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*?**

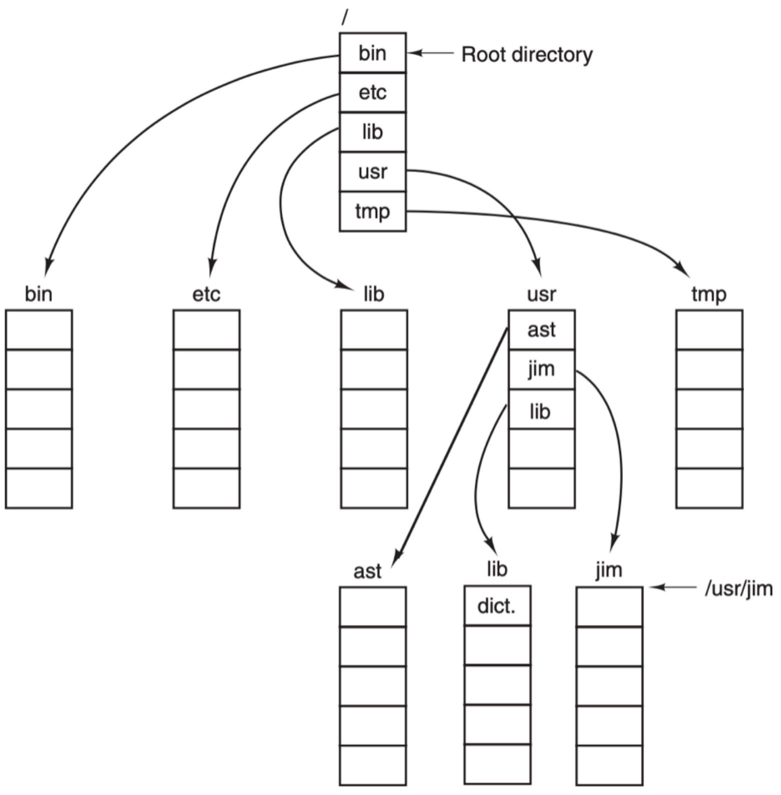

Figure 4-8. A UNIX directory tree.

Fig. 4-8

因为../意味着回到jim的上级目录usr中，所以 *../ast/x*（相对路径）的绝对路径是*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?**

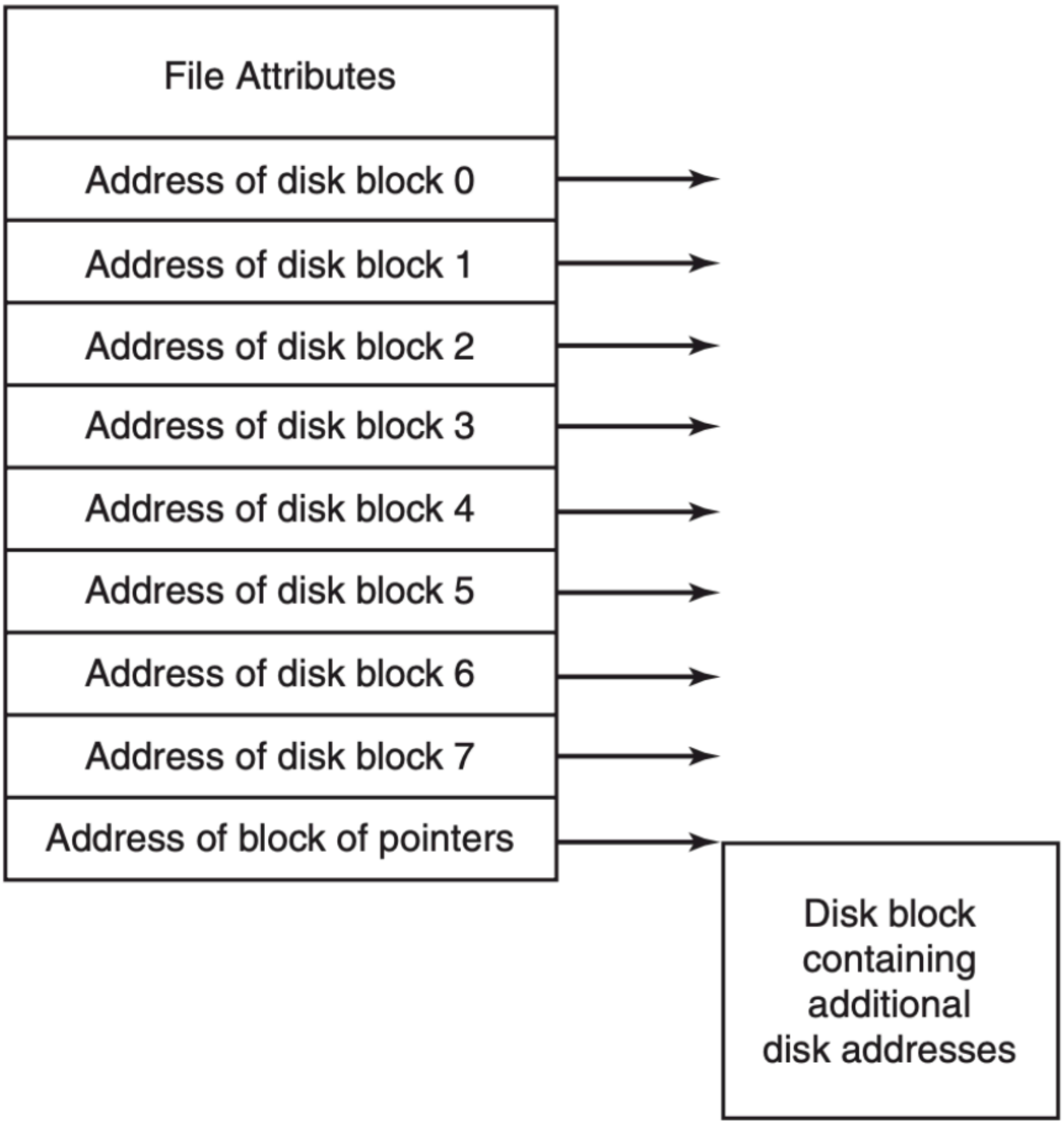

Figure 4-13. An example i-node.

必须有一种方法来表示地址块保存数据，而不是指针。 如果attribute中的某处有一点剩余，则可以使用它。九个指针都可供数据存储， 如果每个指针都是 k 字节，则存储的文件最长可达 9k 字节。 如果attributes 中没有剩余位，则第一个磁盘地址可以保存一个无效地址，以将后面的字节标记为数据而不是指针。 在这种情况下，最大文件为 8k 字节。

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

硬链接是直接饮用文件，将文件名和自身的inode链接起来；而软链接是将文件名指向文件，不指向inode，通过名称来引用文件。因此，硬链接只能用于单个文件系统，不能跨越文件系统；软链接正好相反。

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.**

   假设块大小为1K，free list和bitmap的使用空间如下：

   Free List Size (in bits): (F*1024)*D
   Bit Map Size (in bits): B*1024

   如果要free list比bitmap使用更少的空间的话，因为D的值是16，所以F/B应该<=1/16；所以必须有6.25%或更少的磁盘可用空间。

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） 1111 1111 1111 0000

   b) 1000 0001 1111 0000

   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.**

   当崩溃发生时，对于UNIX和FAT-16都不是问题。恢复算法是将所有块标记为空闲，然后从根目录开始递归遍历文件系统，并从空闲列表中删除发现与文件相关联的块。在UNIX文件系统中，可以在所有的I-node节点上扫描；在FAT文件系统中，因为没有free list，所以这个问题不会发生。即使有free list的话，恢复也只需要扫描FAT寻找空闲的表项。

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?**

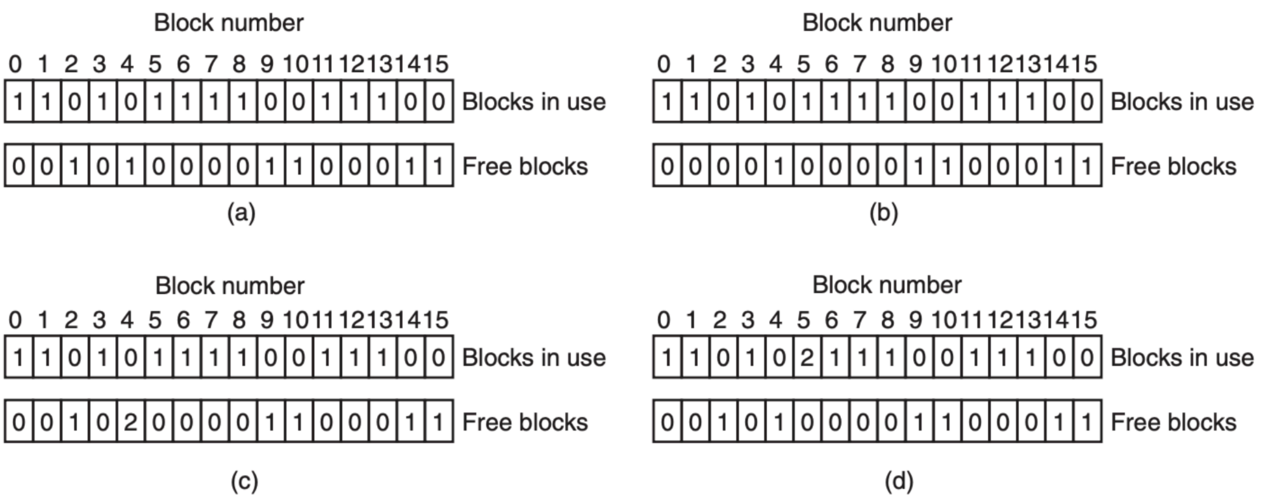

Figure 4-27. File-system states. (a) Consistent. (b) Missing block. (c) Duplicate block in free list. (d) Duplicate data block.

Fig. 4-27

因为某些地方存在错误，所以value=2发生了，它意味着一些块出现在了两个文件中并在free list中出现了两次。解决问题应该首先将空闲列表中的两个都给移除，然后添加一个新的free块（保持原有的内容）。最后，这个块在一个文件中的出现会成为最新引用块的副本，此时就是稳定的了。

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*.**

   分配给 f1 的块是：22、19、15、17、21。
   分配给 f2 的块是：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有十个指针，单个间接块有1024个指针，double有1024^2个指针，triple有1024^3个指针，加起来有1,074,791,434 个块，大概是4096 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.**

    需要的磁盘操作：
    directory for /
    i-node for /usr
    directory for /usr
    i-node for /usr/ast
    directory for /usr/ast
    i-node for /usr/ast/courses
    directory for /usr/ast/courses
    i-node for /usr/ast/courses/os
    directory for /usr/ast/courses/os

i-node for /usr/ast/courses/os/handout.t
总共需要十个磁盘操作