

1. /usr/ast/x
2. 必须采用某种方式来表示 i-node 直接存储短文件而非文件地址。若 File attribute 有空余空间可以用作标记位，则九个地址段全部用于存储数据，总共可存储 9KB。否则用第一个地址段来标记（如填入非法地址），剩余八个存储数据，共 8KB。
3. 硬链接和原文件共享一个 inode 号，直接建立在节点表上，通过更新节点表的计数来实现。软链接则会分配新的 inode 号，通过在磁盘对应空间内存放文件的绝对路径来实现。若删除原文件，则使用硬链接仍可访问（仅对应的节点表计数减一），软链接则不可（对应路径为空）。此外，硬链接不可以跨文件系统，也不可以针对不存在的文件或目录创建，而软链接可以，只是在文件创建之前不能正常访问。
4. 采用空闲块表需要的空间：F * D 位
采用位图需要的空间：B 位
要使空闲块表所需的空间少于位图，则有 $F * D < B$
若 D 为 16， $F * 16 < B$ ， $F/B < 1/16$ ，即至少有 6.25% 的磁盘空间为空。
5. a) 1111 1111 1111 0000
b) 1000 0001 1111 0000
c) 1111 1111 1111 1100
d) 1111 1110 0000 1100
6. 对于 UNIX 系统可通过扫描所有的 inode 依次对文件中的所有块表进行扫描，得到新的空闲块表或位图。FAT 文件系统不采用空闲块表或位图，不存在这样的问题。
7. 正常工作时不可能，如果发生了，则一定是 bug。若对某一个块号，两个计数器的值都是 2，则此时该块在两个文件中都出现，且在空闲块表中也出现两次。要想修复这个 bug，应该取一个新的空闲块，把错误块的内容复制到其中，对其中一个文件的对应块进行替换，并从空闲块表中删除该重复出现的块号。
8. f1: 22、19、15、17、21
f2: 16、23、14、18、20
9. 每个一次间接块指向 2^{10} 个磁盘块，二次间接块指向 2^{20} 个，三次间接块指向 2^{30} 个，10 个直接表项指向 10 个，则文件大小最大可以为
 $(10 + 2^{10} + 2^{20} + 2^{30}) = 1074791434 \text{ KB} \approx 4 \text{ TB}$
10. 1) 获取根目录的地址
2) 获取/usr 的 inode
3) 获取/usr 的地址
4) 获取/usr/ast 的 inode
5) 获取/usr/ast 的地址
.....
8) 获取/usr/ast/course/os 的 inode
9) 获取/usr/ast/courses/os 的地址
10) 获取 usr/ast/course/os/handout.t 的 inode，共需要 10 个磁盘操作。