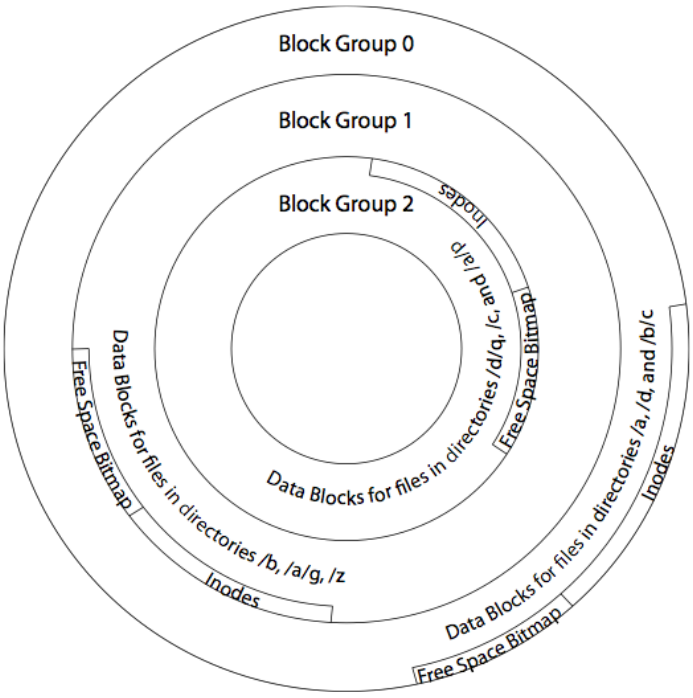
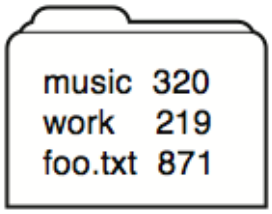
Topic 17: File Naming -- Directories

Reading: 13-13.2
Next reading: 12.2-12.3

- Naming: how do users refer to their files? How does OS find a file, given its name?
- The file record must be stored on disk, so it will stay around even when the OS doesn't.
    - In Unix, all the records (inodes) are stored in fixed- size array on disk. An inode's index in the array is called its *i-number*, and is used by the OS to refer to the file.
- Special areas of disk are used for this
    - Originally: inode array at one side of disk.
    - Then: inode array mid-way across disk.
    - Today: inode array in each cylinder group. (Figure 13.13)

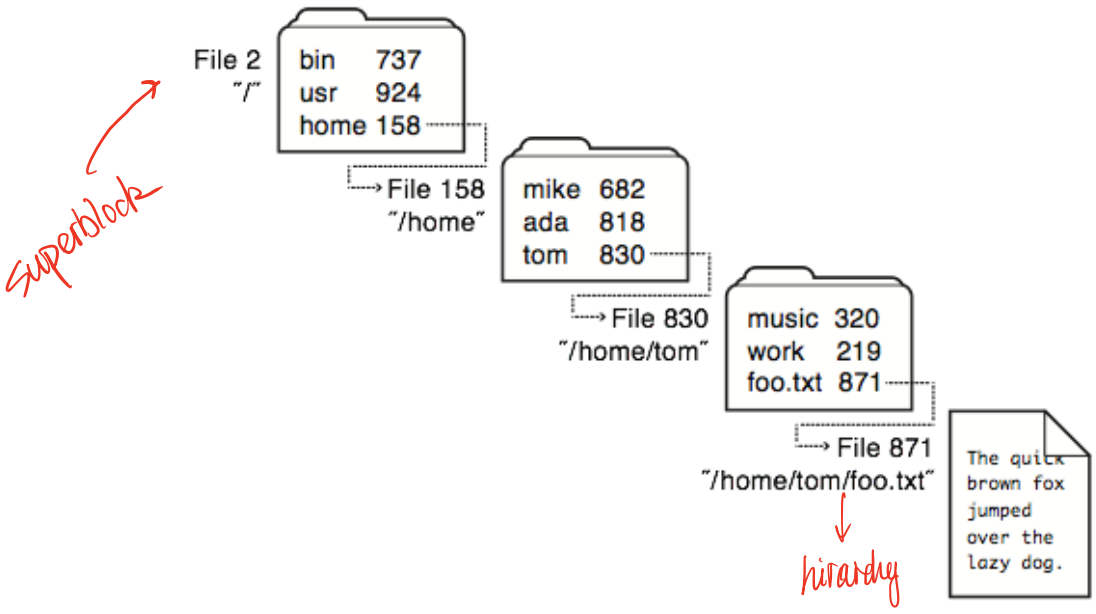


- The sizes of the inode arrays are determined when the disk is initialized, and can't be changed.
- When a file is open, its inode is kept in main memory. When the file is closed, the inode is stored back to disk.
- Users need a way of getting back to files that they leave around on disk. One approach is just to have users remember i-numbers. This is the Social Security approach.
- Of course, users want to use text names to refer to files. Special files called *directories* are used to tell what i-numbers correspond to what names. The conversion of a file name into the i- number for the file is known as *name lookup*.



inode → datablock

- Approach #1: have a single directory for the whole disk. Use a special area of disk to hold the directory.
    - Directory contains <name, index> pairs.
    - If one user uses a name, no-one else can.
- Approach #2: have a separate directory for each user. This is still clumsy: names from a user's different projects get confused.
- Approach #3 (Unix): generalize the directory structure to a tree.



- Directories are stored on disk just like regular files (i.e. inode with 15 pointers, etc.) except inode has special flag bit set. User programs can read directories just like any other file (try it!). Only special system calls may write directories (e.g. creat, open, link, unlink).

- Each directory contains <name, i-number> pairs in no particular order. Each entry is variable-size, and contains the name length, the name (up to 255 characters), and the i-number. Entries do not cross block boundaries, however. (Figure 13.4)