Read Tanenbaum, Bos: Chapter 4, 5.1, 5.4, 10.6, 11.8

Exercises

1. Suppose your current directory is `/Users/Bob/Documents`.
   (a) What is the absolute pathname for `Biz/Taxes`?
   (b) What is the absolute pathname for `../Music/Radiohead`?
   (c) What is a relative pathname for `/Users/Ann/Dropbox`?

2. When the file `/usr/lib/ruby/1.9.1/time.rb` is opened, several disk accesses are needed to read inode and directory blocks. Assume the inode for the root directory is in memory, and all directories are one block long. How many disk accesses are required to open the file?

3. CD-ROM filesystems and Windows NTFS keep directories sorted alphabetically by name. Unix and Windows FAT filesystems don't, because it requires a lot of disk I/O and would be slow.
   (a) What's good about keeping directories sorted?
   (b) Why can CD-ROM filesystems use sorted directories?
   (c) What does NTFS do to keep directories sorted without giving up too much speed?

4. Some Linux backup methods create a periodic 'snapshot' of the filesystem (or some subdirectory). See your `Backup` folder on turing for an example, where there are daily snapshots for the last year or so.

   Instead of copying every file for each snapshot, the snapshot copies only the files that have changed, and uses links to older snapshots for files that have not changed.

   Should these be hard links or symbolic links? What happens when old snapshots are removed?

5. Discuss how disk storage is subject to both internal and external fragmentation.

6. Suppose the beginning of the free space bitmap for a newly formatted disk looks like this:

   `11111100 00000000 00000000 00000000`

   (the first few blocks are in use for the superblock and other file system structures)

   If the system allocates files in contiguous blocks when possible, show the free space bitmap as the following operations occur:
   (a) File A is written, using 6 blocks.
   (b) File B is written, using 8 blocks.
   (c) File A is deleted.
   (d) File C is written, using 8 blocks.
   (e) File B is deleted.
   (f) File D is written, using 4 blocks.

7. In the Linux ext4 filesystem, there is a bitmap indicating which inodes are free and which are in use. Suppose this bitmap is destroyed (by a disk error, say). Is it possible to recreate the inode information?

8. Suppose a 40 cluster file in NTFS has the the following runs (each run given as start:length)
   `30:5 100:25 50:10`

   What physical block on disk is the file's logical block 1? Logical block 15? Logical block 32?

9. When a file is deleted from a disk, its directory entry is removed and the file's data blocks are marked as free, but the actual data blocks are not erased. Since approximately 2010, modern OSes support a "trim" command, which allows the OS to notify a storage device that these blocks are no longer in use. Why would this be helpful for the operation of a solid state drive (SSD)?