CSCI 3323 (Principles of Operating Systems), Fall 2016

Homework 6

Credit: 25 points.

1 Reading

Be sure you have read, or at least skimmed, Chapter 4.

2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word "pledged", plus one or more of the following about collaboration and help (as many as apply).[1] Text *in italics* is explanatory or something for you to fill in. For written assignments, it should go right after your name and the assignment number; for programming assignments, it should go in comments at the start of your program.

- This assignment is entirely my own work.
- This assignment is entirely my own work, except for portions I got from the assignment itself *(some programming assignments include "starter code")* or sample programs for the course *(from which you can borrow freely — that's what they're for).*
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.*
- I got significant help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. ("Significant" here means more than just a little assistance with tools — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided significant help to *names of students* on this assignment. *("Significant" here means more than just a little assistance with tools — you don't need to tell me about helping other students decipher compiler error messages, but beyond that, do tell me.)*

3 Problems

Answer the following questions. You may write out your answers by hand or using a word processor or other program, but please submit hard copy, either in class or in one of my mailboxes (outside my office or in the ASO).

[1] Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM's Special Interest Group on CS Education.

1. (5 points) Consider a digital camera that records photographs in some non-volatile storage medium (e.g., flash memory). Photographs are recorded in sequence until the medium is full; at that point, the photographs are transferred to a hard disk and the camera's storage is cleared. If you were implementing a file system for the camera's storage, what strategy would you use for file allocation (contiguous, linked-list, etc.) and why? Notice that this camera does not have the ability to delete photographs from its storage one at a time, so your file system does not need to support that. (It's probably best to think of this as a somewhat hypothetical problem, using only the description supplied, rather than trying to extrapolate from your experience with actual cameras.)

2. (5 points) The textbook describes more than one strategy for keeping track of free blocks in a file system (free blocks, bitmaps, and FATs). All of these strategies rely on information that is kept both on disk and in memory, sometimes with the most-current information only in memory. What would happen if the copy on disk of whatever data structure is used to keep track of free blocks was lost or damaged because of a system crash — is there a way to recover, or do you have to just reformat the disk and hope you backed up any really important files? Answer separately for MS-DOS FAT-16 (which uses a FAT) and UNIX V7 filesystems (which use one of the other strategies).

3. (5 points) Linux includes code to access several types of Windows filesystems, including FAT-32. So on a system where one of the disk partitions holds a FAT-32 filesystem, one can configure Linux to access this filesystem through pathname `/windows/fat` for example. However, all the files in `/windows/fat` appear to be owned by user `root`, and attempts to change their ownership (with the `chown` command) fail with an error message "Operation not permitted". What's wrong?

4. (10 points) Consider a UNIX filesystem (as described in section 4.5.3) in which each i-node contains 10 direct entries, one single-indirect entry, one double-indirect entry, and one triple-indirect entry. If a block is 1KB (1024 bytes) and a disk addresses is 4 bytes, what is the maximum file size, in KB? (*Hint:* Use the blocksize and size of disk addresses to determine how many entries each indirect block contain.)