

Homework 4

Hand in your *typed* answers at the start of class on Monday, June 23rd. Figures and equations can be done by hand. Please hand in at most 3 pages of answers.

You are permitted to discuss homework questions with other students, but you must write out the answers yourself and give the names of the students you discussed the homework with. We will not look favorably on answers that are copied from online sources.

1. Silberschatz 14.2
2. Suppose you open an existing file (call it `my_file`) using a text editor. When you save the file, the editor first makes a backup of the old version under a different name (for instance, `my_file.1`), then replaces the old file with the new version (i.e. that's what it looks like to the user). Give an efficient sequence of operations, including systems calls and block writes to the disk, which must be done to make the backup and save the new version of the file.
3. Some operating systems (e.g. FreeBSD) have an additional set of system calls which allow a process to specify the name of a file and get back the OS unique file ID, and open a file using the file ID, rather than the file name. For instance, rather than just the `fd = open(filename, ...)` system call, they have the following as well:
 - `getfh(filename, &filehandle)`, where `filehandle` is a small structure which is filled in to contain the file's unique ID (i.e. inode number).
 - `fd = fhopen(filehandle)`, does a "direct open", using the file's unique ID.

What would be the advantage of this mechanism? (Hint: What inefficiency is it intended to overcome?). What drawback does it have, which might lead to its use by arbitrary processes being disallowed?

4. Silberschatz 12.1
5. Silberschatz 12.2
6. Silberschatz 12.6
7. Silberschatz 12.9; treat the question as referring to an operating system *disk block* cache.
8. The Amoeba distributed operating system is a microkernel developed at the Free University of Amsterdam (microkernels are briefly described in section 3.5.3 of Silberschatz, but the precise details of microkernels are unimportant to this question). One of the features of Amoeba is that specialised file systems for specific tasks can be run at user level, and given raw access to a disk partition which they can organise according to their own allocation scheme. One example of a specialised (and unorthodox) file system is the Amoeba Bullet Server, which works as follows: files are immutable. When a process accesses a file, the entire file is cached by the OS. When the access by the process finishes (i.e. the file is closed), the file is written to the disk in one go, and the old version of the file is deleted (i.e. the old version is not overwritten). Assume that directories are handled specially.
 - (a) How could this design simplify the implementation of the file system?
 - (b) How might this design improve the performance of the file system?
 - (c) Give two disadvantages of the design.