

This homework is due at the beginning of class on October 15, 2013 and is worth 1.5% of your grade.

Name: _____

CCIS Username: _____

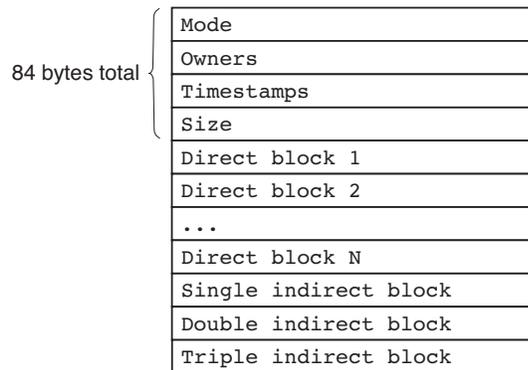
Problem	Possible	Score
1	10	
2	15	
3	20	
4	15	
5	25	
6	15	
Total	100	

1. In a *versioning* file system, each filename is augmented with a version number. For instance, `foo.txt;3` denotes the third version of file `foo.txt`. Whenever a file is modified and closed, its version number is automatically incremented. Previous versions of the file exist until they are explicitly deleted, and they can be accessed with their respective version number.

1a. Discuss the advantages and disadvantages of a versioning file system. Which applications would benefit from such a file system? Which applications would suffer? You must mention one advantage (and application that benefits) and one disadvantage (and application that suffers). (5 pts)

1b. Does a versioning file system make it easier or harder to deal with cache and/or metadata consistency in the event of a file system crash? Why? (5 pts)

2. Suppose that a UNIX file system uses the inode layout shown in the figure below, with all disk blocks (and inodes) being 4,096B in size and all disk block pointers being 4B in size.



Now suppose that a UNIX process has opened a file and the operating system has the inode cached in memory. If the program makes a read request for 4,096B starting at byte offset 8480000, how many total disk reads will the operating system have to make in order to get the data into main memory? Assume that the file inode is the only information in memory to begin with. You must explain your answer. (15 pts)

3a. Suppose that we are implementing a file system using a File Allocation Table (FAT). Disk blocks are 512B in size, and pointers to disk blocks are 4B in size. How many entries will there be in the File Allocation Table if the disk has a 8,256kB available for file storage and FAT storage? (ignore the space required for the volume control block and the directory entries) (15 pts)

3b. How many disk blocks will the FAT require? (5 pts)

4. User *A* creates a file *a* with permissions such that *A* can read and write the file and all other users can read but not write it. User *B* creates a hard link to *a* by the name of *b*. User *C* creates a soft link to *b* by the name of *c*. User *D* creates a soft link to *c* by the name of *d*.

4a. What happens if user *C* deletes *c*, then *D* tries to read *d*? (5 pts)

4b. Assume the same initial setup (i.e., *C* has not deleted *c*). What happens if *A* deletes *a* and *B* tries to read to *b*? (5 pts)

4c. Assume the same initial setup (i.e., *C* has not deleted *c* and *A* has not deleted *a*). What happens if *A* deletes *a* and *D* tries to write to *d*? (5 pts)

5. Suppose we have a UNIX directory with the following contents and permissions

```
drwxr-xr-x    6 amislove  staff      204 Feb 12 14:50 .
drwx-----   2 amislove  staff    10030 Feb  9 23:18 ..
-rw-r-xr-x    1 amislove  faculty   12 Feb 12 14:50 bar
-rw-r--r--    1 amislove  faculty  298 Feb 12 14:50 baz
-r--r-x---    1 amislove  staff    922 Feb 12 14:50 blah
-rwxr-----   1 amislove  staff    11 Feb 12 14:50 foo
-rwsr-sr-x    1 amislove  staff   2938 Feb 12 14:50 tmp
```

5a. Which of the files is user amislove able to execute? (5 pts)

5b. Suppose user marty is a member of faculty. Which of the files is marty able to read? (5 pts)

5c. Suppose user amy is a member of neither the group staff or faculty. Which of the files is amy able to write to? (5 pts)

5d. Suppose that when run, the program tmp attempts to create a file in the parent directory (. .). If user amy from above tries to execute tmp, what will happen? (5 pts)

5e. What command(s) should amislove run to make foo readable/writable by all faculty members and readable by all users? (5 pts)

6a. How could you simulate a multi-level directory structure with a single-level directory structure in which arbitrarily long names can be used? Explain how you can do so, and contrast this scheme with the multilevel directory scheme. (10 pts)

6b. How would your scheme perform compared to a traditional multi-level directory scheme? Are there file system operations that would be cheaper? More expensive? (5 pts)