

*2 problems. 40 points. 30 minutes Closed book. Closed notes. No electronic device. Write your name above.*

**1. [20 points]** A computer has an FFS filesystem in which each directory's data fits in one block. Starting from a state where the only block in memory is the superblock, give the sequence of blocks to read in order to read the last byte of the file at /x/y, in each of the following cases.

- a. The file has only 5 blocks of data.

**Solution [10 pt]**

Superblock has ptr to /.inode

- |                    |   |
|--------------------|---|
| 1. Read /.inode    | // has ptr to /.db0 (data block 0)              |
| 2. Read /.db0      | // has entry for x, including ptr to /x.inode   |
| 3. Read /x.inode   | // has ptr to /x.db0                            |
| 4. Read /x.db0     | // has entry for y, including ptr to /x/y.inode |
| 5. Read /x/y.inode | // has ptr to /x/y.db4                          |
| 6. Read /x/y.db4   | // has required data                            |

**Grading:** [2 pt] per entry approximately.

**End of solution**

- b. The file has the maximum amount of data allowed in FFS.

**Solution [10 pt]**

1. as in part a
2. as in part a
3. as in part a
4. as in part a
5. as in part a
6. Read /x/y.triple-indirect-block
7. Read last double-indirect-block in /x/y.triple-indirect-block
8. Read last single-indirect-block in above /x/y.double-indirect-block
9. Read last data-block in above /x/y.single-indirect-block

**Grading:** [4 pt] for entries 1-5; [6 pt] for entries 6-9.

**End of solution**

**2. [20 points]** Augment the FAT filesystem to provide hard-links (to files only). Do not modify the FAT or the data region (so a traditional FAT-filesystem OS can still use your augmented filesystem). Assume that the FAT region (which holds the FAT) has some free space which you can use.

Your answer should briefly state

- the new information to be stored in the FAT region
- how this information is updated when: creating a file, linking to an existing file, deleting a file

***Solution***

An unrestricted solution is not doable because that would require traditional OS's to update regarding hardlink state when files are created or deleted. [5 pt]

It is easily doable if traditional OS's do not create or delete files:

- Additional info: for each node (file or directory): [start block, # of hardlinks].  
Searchable by start block for efficiency. [6 pt]
- When a file is created: add new entry  $[x, 1]$ , where  $x$  is start block of the file. [3 pt]
- When a file is linked to: increment its hardlink count. [3 pt]
- When a file is deleted: decrement its hardlink count; remove file if zero. [3 pt]

It's also doable if the files with multiple (more than one) hardlinks are visible only to the augmented OS's.

***End of solution***