

CMSI 387
OPERATING SYSTEMS
Spring 2009

File System Implementation Supplementary Exercise

Due to a combination of time constraints and the need to have *sudo* capability on a Linux machine, this exercise is not an official assignment. Get together with a classmate who has *sudo* access to a machine and walk through this, in order to get some hands-dirty quality time with the *ext2* file system.

The Short Version

Make an *ext2* disk image, mount it, put some files on it, print the user view of the file system (i.e., a series of *ls* invocations), dump the disk image to hex, and identify, at the hex level, the various sections listed in Step 6 of The Long Version.

The Long Version

1. To create the disk image, you'll need to learn how to use the *dd* ("disk dump") command. The following example creates a new file called *image* consisting of 1024 default-size blocks, and initializes its contents with zeroes:

```
dd if=/dev/zero of=image count=1024
```

2. You should now have a file that is equivalent to a brand-new, unformatted disk. "Format" it by installing an empty *ext2* file system on it:

```
mke2fs image
```

3. Mount the disk image — this is what requires *sudo* access:

```
mount -o loop -o nosuid -o nodev image mountpoint
```

...where *mountpoint* is the directory under which you'd like to mount *image*. You can use *df* to verify that your command worked. To unmount the disk image, do:

```
umount mountpoint
```

Again, *df* will tell you if all went well.

4. Create the following items within that mounted file system:
 - a. A non-empty text file at the top-level directory of the file system
 - b. A directory at the top-level directory of the file system
 - c. A second non-empty text file inside that subdirectory (give it different content so you can differentiate the two files)

- d. A symbolic link inside that subdirectory to the text file in the top-level directory
 - e. A hard link from the top-level directory to the text file in the subdirectory
5. Run a series of *ls* commands on the now-populated file system, and note the output. Feel free to use various *ls* switches (e.g., *-F*, *-l*, *-a*, *-i*, etc.) to see as much interesting information as possible.
 6. Dump the disk image file to hex using *hexdump -C*, then try to identify these items:
 - a. The disk image's superblock
 - b. The directory entries for the files, links, and directories that you created
 - c. Where applicable, the inodes for the items that you created
 - d. Where applicable, the data blocks occupied by these items

Get a feel for that and enjoy the hacker buzz :)