Steps for Exercise 12.21

1. Obtain the inode of file1.txt

    16963 -rw-r--r-- 1 os os 33 Jan 8 12:47 file1.txt

2. Enter the command ln file1.txt file2.txt

The inodes for file1.txt and file2.txt are

    16963 -rw-r--r-- 2 os os 33 Jan 8 12:47 file1.txt
    16963 -rw-r--r-- 2 os os 33 Jan 8 12:47 file2.txt

So we can see they are the same from creating a hard link. Examining the files file1.txt and file2.txt we also see they have the same contents.

3. Edit file2.txt and changes its contents. Afterwards, examine the contents of file1.txt . Are the contents of file1.txt and file2.txt the same of are they different?

We see file1.txt and file2.txt still have the same contents. They have the same contents because they are in fact the same file! Remember, an inode is a file. Names such as file1.txt and file2.txt are names - references - to the actual files.

4. Remove file1.txt by entering rm file1.txt . Does file2.txt still exist as well?

Yes, after removing file1.txt, we see that file2.txt still exists. This is because that the rm command only decrements the count of links to an inode. The next step examines this.

5. Remove file2.txt by entering the command strace rm file2.txt

Examine the trace of system calls, we see that the rm command in fact invokes the unlink command which decrements the number of links to an inode. A file is only deleted when the number of links goes to zero.

6. Enter the command ln -s file3.txt file4.txt to create a soft link. What are the inode numbers of file3.txt and file4.txt?

We can see they have different inode numbers:

    16967 -rw-r--r-- 1 os os 37 Jan 8 13:10 **file3.txt**
    16963 lrwxrwxrwx 1 os os 9 Jan 8 13:10 **file4.txt -> file3.txt**

Also notice how file4.txt is a link to file3.txt . file4.txt is its own file, but it is a special type of file that is merely a link (a pointer/reference) to another file.

7. Edit the contents of file4.txt . Have the contents of file3.txt been altered as well?

Yes, when we edit file4.txt, we see that file3.txt is edited as well. This is because file4.txt is really just a link to file3.txt .

8. Explain what happens when you try to edit file4.txt .

Some rather interesting results. Because file4.txt is a link to file3.txt, when we edit file4.txt, it treats it as a new file. Once we save the edits, we see in fact that file3.txt has returned:

> **16965** -rw-r--r-- 1 os os 16 Jan 8 13:25 **file3.txt**
> **16963** lrwxrwxrwx 1 os os 9 Jan 8 13:10 file4.txt -> file3.txt

albeit with a different inode number than it previously had.