Khoa Le
CS 390 - Project 2 - File Sharing with Links in Linux
Dr. Matthews

3. ls - l - - inode file0.txt
2250581 -rw------- 1 kdl2483 student 65 May 1 15:32 file0.txt

4. Created a hard link between file0.txt and new file1.txt
ls -l —inode file0.txt file1.txt
2250581 -rw------- 2 kdl2483 student 65 May 1 15:32 file0.txt
2250581 -rw------- 2 kdl2483 student 65 May 1 15:32 file1.txt

The inode numbers of both files are the same. This is because both files are linked to one inode table. The information in the third column changes the value from 1 to 2 after creating a hard link from file1.txt to file0.txt. The information in this column represents the number of files that are linked to the same inode

5. Changed the first 0 to 'X' in file1.txt
The first character 0 in file0.txt also changed to 'X' after I made the change in file1.txt

6. Remove file0.txt
The file1.txt still exists with unchanged content after removing file0.txt

7. Which system call is used to remove a file from a directory?
Using strace to remove file1.txt shows that the file system uses many calls to remove file1.txt. But the important call to remove the file is **unlinkat()**

8. Created a symbolic link between file2.txt and file3.txt
4564510 -rw------- 1 kdl2483 student 65 May 6 13:11 file2.txt
2106992 lrwxrwxrwx 1 kdl2483 student 9 May 6 13:11 **file3.txt** -> file2.txt

The inode numbers of file2.txt and file3.txt are different

9. File2.txt and file3.txt
The first character '2' from file3.txt also changes to character 'X' as the change is made in file2.txt

10. Remove file2.txt
After removing the file2.txt, the content of file3.txt is also deleted.
The error message: file3.txt: No such file or directory

11.
The error message takes place because as I delete the original file, the symbolic link still remains but it points to a file that does not exist. That is why when I use the more command to check the content of the file3.txt is says that it has no such file. This is called dangling link.

12.
The content of file3.txt has the new text that is added to the new file2.txt

13.
Error message
ln: '.': hard link not allowed for directory

14. Why do you think Linux does not allow the creation of hard links to directories?
Linux does not allow the creation of hard links to directories because allowing hard links to directories would break the directed acyclic graph structure of the file system. If hard links were allowed to be created to the directories then two different directories in different points can point to the same location. In fact, the subdirectories can point to the parent directories and results in a cycle which there is no way to detect.

15.
I was able to create a symbolic link to the directory. Linux system treats the symbolic links as pointers that redirect to another file or directory. Therefore, when the Linux kernel encounters the symbolic links when it traverses the file system, it would follow the pointer to the next file or directory. If the next link is also a symbolic link it will continue to be dereferenced until an object that is not a symbolic link is found or a loop is detected.
The kernel cannot use the same technique when it encounters a hard link because of the difference in nature between a hard link and symbolic link. According to Linux documentation, a symbolic link is "a special type of file whose contents are a string" of the file path. The kernel system can dereference the symbolic until it reaches the target object. On the other hand, the hard link to a file is "indistinguishable from the original file" (Linux documentation) as both files reference to the same underlying object. Therefore, the kernel gets confused and trapped in the loop when it follows hard links.