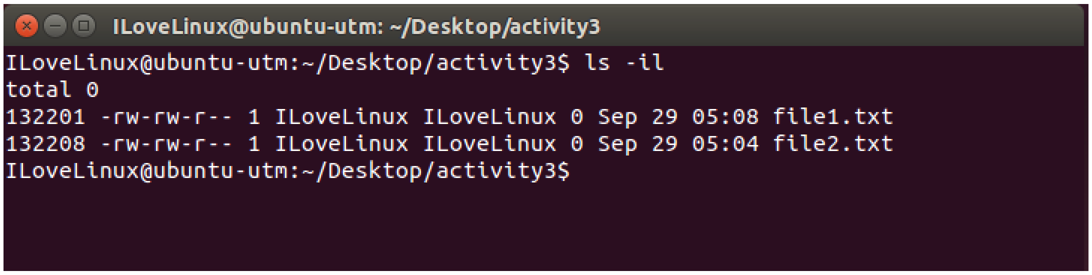
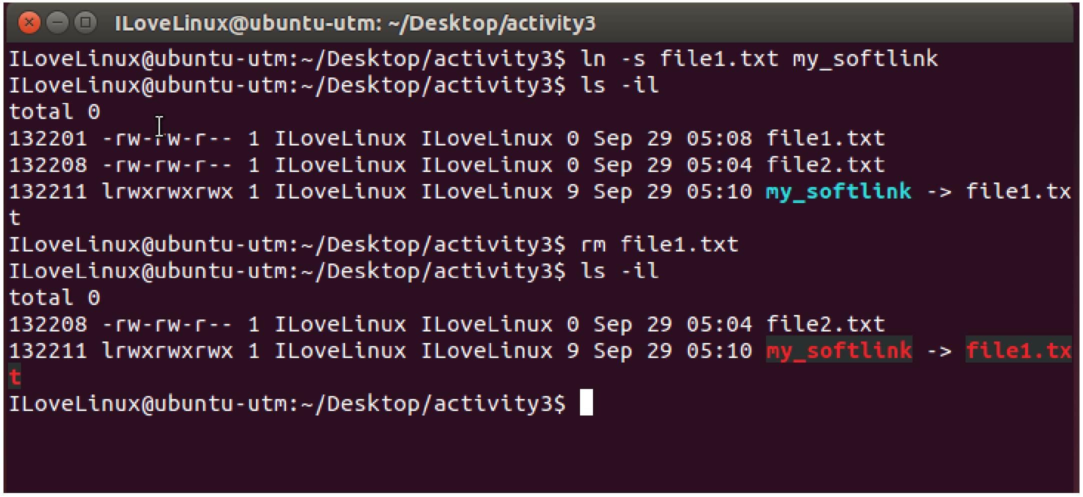
Ashlyn Cooper
CPSC 4240
Activity 3

1. After running that command, we can see some information about the two files we have created. For each file we can see the inode number, the file's permissions, the reference count, the owner/creator of the file, as well as the date and time when the file was last edited/interacted with.



```
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132201 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:08 file1.txt
132208 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:04 file2.txt
ILoveLinux@ubuntu-utm:~/Desktop/activity3$
```

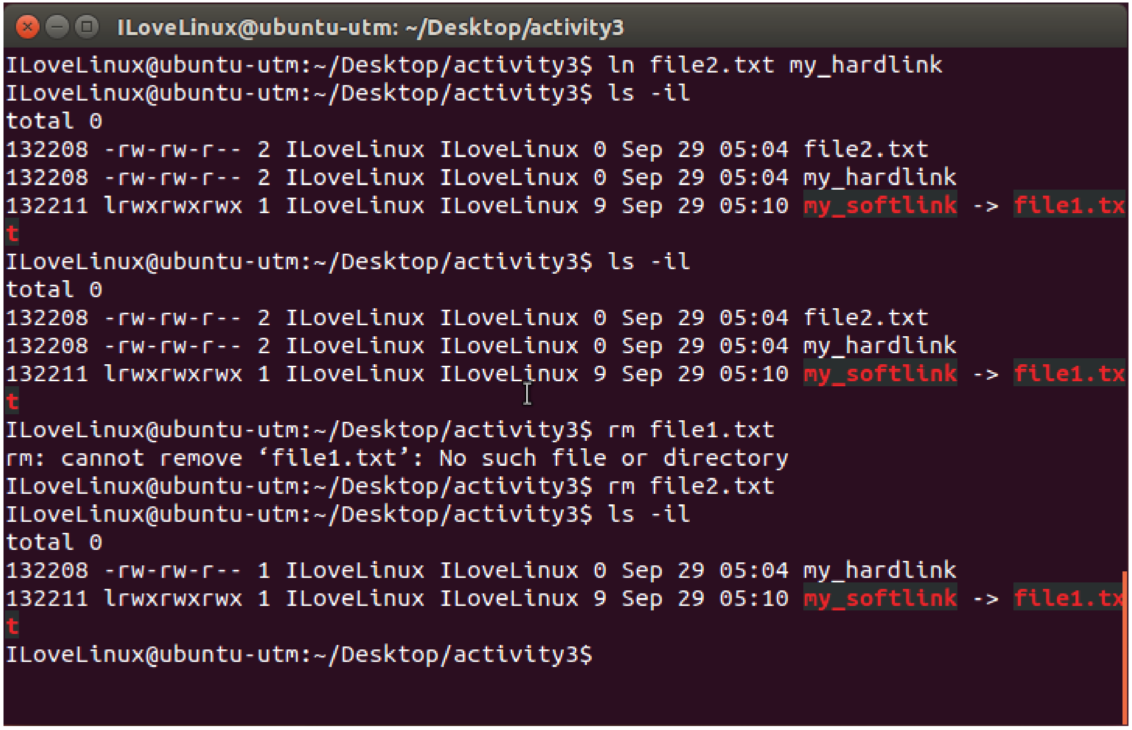
2. After creating the soft link and running the ls command, we can now see the soft link listed. We see similar information for the soft link to what we saw before for the files. However, the soft link has a unique inode number compared to the inode for file1.txt. After deleting, file1.txt the soft link no longer works. When listing the files in the directory the soft link now appears in red to indicate that it no longer is valid because it now is pointing to a nonexistent file.



```
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ln -s file1.txt my_softlink
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132201 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:08 file1.txt
132208 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:04 file2.txt
132211 lrwxrwxrwx 1 ILoveLinux ILoveLinux 9 Sep 29 05:10 my_softlink -> file1.tx
t
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ rm file1.txt
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132208 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:04 file2.txt
132211 lrwxrwxrwx 1 ILoveLinux ILoveLinux 9 Sep 29 05:10 my_softlink -> file1.tx
t
ILoveLinux@ubuntu-utm:~/Desktop/activity3$
```

3. After creating a hard link to file2.txt, the inode number of the hard link and file2.txt are the same. They also both have the same permissions and reference numbers. After removing the original file, file2.txt, the hard link will still function, as it contains the contents of the original file and was assigned to the same inode value as the original file,

so it references the same location as the original file.



```
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ln file2.txt my_hardlink
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132208 -rw-rw-r-- 2 ILoveLinux ILoveLinux 0 Sep 29 05:04 file2.txt
132208 -rw-rw-r-- 2 ILoveLinux ILoveLinux 0 Sep 29 05:04 my_hardlink
132211 lrwxrwxrwx 1 ILoveLinux ILoveLinux 9 Sep 29 05:10 my_softlink -> file1.tx
t
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132208 -rw-rw-r-- 2 ILoveLinux ILoveLinux 0 Sep 29 05:04 file2.txt
132208 -rw-rw-r-- 2 ILoveLinux ILoveLinux 0 Sep 29 05:04 my_hardlink
132211 lrwxrwxrwx 1 ILoveLinux ILoveLinux 9 Sep 29 05:10 my_softlink -> file1.tx
t
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ rm file1.txt
rm: cannot remove ‘file1.txt’: No such file or directory
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ rm file2.txt
ILoveLinux@ubuntu-utm:~/Desktop/activity3$ ls -il
total 0
132208 -rw-rw-r-- 1 ILoveLinux ILoveLinux 0 Sep 29 05:04 my_hardlink
132211 lrwxrwxrwx 1 ILoveLinux ILoveLinux 9 Sep 29 05:10 my_softlink -> file1.tx
t
ILoveLinux@ubuntu-utm:~/Desktop/activity3$
```

4. The symbolic links stdin, stdout, and stderr are linked to directories in the proc directory. More speicifcally stdin, stdout, and stderr are linked to /proc/self/fd/2, /proc/self/fd/0, and /proc/self/fd/I respectively. /proc/self is a symbolic link as well. This directory holds the processes that are currently running. Stdin, stdout, and stderr are standard streams. Even though they aren’t “devices,” by being in the /dev directory any processes can open, read from, and write to those files, which is useful as a standard stream file.
5. Initrd and initramfs are different methods used to load a temporary root file system to the RAM or system memory for successful booting. Initrd is used for Linux kernels 2.4 and lower, while initramfs is used for kernels 2.6 and above. When compiling initrd into the kernel, at least one filesystem driver is required. On the other hand, with initramfs does not need a driver and it is always on-kernel with no device or additional driver needed.