Lab 12

Jarred Parr and Alexander Fountain

1. A file can be checked via the `ls -al` command which will give a string similar to: `lrwxrwxrwx` if there is a link and it will show: `-rwxrwxrwx` if not. Also, you can use the `file` command to have a printout of exactly what the file is and its type.
2. The link when a hard link is performed is tracked via the inode. When doing only a soft link we only track by the name, which does not increase the size of the link counts. Hard links increase the link count since two files are now pointing to the same inode location.
3. Hardjunk is 404 and softjunk is 11. This is because the hard link copies the entire file instead of only referencing the name. As a result, the entire contents of the file need to be added to the location of that file in order to have it be a true hard link, this is similar to the idea stated above.
4. Since the inode of the file persisted due to the multiple linkage of the data, `hardJunk` is still able to exist. The soft link, however, does not contain much more data than the location of the file and, as a result, does not persist when the source file is deleted. When you display `hardJunk` it works fine, but `softJunk` says `'no such file or directory'` which is to be expected.

Code Programming Assignment

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>

#define SIZE 30

int main (int argc, char *argv[]) {
    if (argc != 4) {
      printf("Usage:\n\t./a.out <link_type(h, s)> <target> <link_name>\n");
      exit(0);
    }

    if (argv[1][1] == 'h') {
      if ((link(argv[2], argv[3])) < 0) {
        perror("error creating hard link");
        exit(1);
      }
      printf("hard link created\n");
    } else if (argv[1][1] == 's') {
      if ((symlink(argv[2], argv[3])) < 0) {
        perror("error creating symbolic link");
        exit(1);
      }
      printf("symbolic link created\n");
    }
    return 0;
}
```