Julia Sales
ID: jesales
Asgn0
CMPS130-02

Write Up for Asgn0

Testing:

For dog.c I tested both on my local machine, on the schools ssh account, and on Ubuntu. On my local machine, I tested my program in small portions. I first tested standard input when "-" was entered and then when nothing was entered at all. I would go back and fourth between my design document and code because I constantly had to make changes because there were things that I did not get right the first time. I tested on the ssh account in a similar way. Created test files and made sure the program ran correctly. I would also pull my code from my git repository to Ubuntu to test it on there as well.

Questions:

How does this code for handling a file differ from that for handling standard input? What concept is this an example of?

Answer: My code for handling a file differs from standard input because of the values of the parameters are different in the read and write function. For handling files, I would open the file. If there was a file detected then read took in the file descriptor from the open function and used a separate buffer from handling standard inputs. Write in handling files took in the file buffer instead of the standard input buffer and used the value of file_in instead of the length of the buffer. Standard input could not take in a file and had to take input from the console so it would read the input from the console by the user and then print it back out to terminal. So read's parameters were standard input, the buffer for standard input, and the size of standard input. Write for standard out was similar to read for standard input but the only thing that was changed was the int value for the file descriptor because 1 means write to stdout. This concept is an example of modularity because we were able to separate the functionalities of the program.