

Since this program was able to be done without many functions, I mainly did system-testing. After completing my code I just tested the functionality of it by running dog with .txt and .bin file types to ensure that it worked like cat. I also tested for when there were no arguments or if a “-” was found to check that the output matched cat. To make sure that they provided exact outputs, I run unix redirection commands to save the output of when dog and cat were ran on the same files. From there, I used diff/cmp/hexdump to compare the files.

The code for handling a file and for handling standard input are quite similar besides a few key differences. For handling a file, I made sure to use the open(2) function, to open a file from standard input. From there, I would pass the int file descriptor returned by open() to read(3). The return value of read() was used in write(3) to determine the max buffer size to write up to. This makes it so that no extra garbage out from the buffer would be written to standard output. For handling standard input I only needed to utilize 2 functions, read() and write(). There is no need to use open(), as the file descriptor used in read() would be 0 which is standard input. That’s the key difference between the two. Handling a file requires opening a file to use in read, while handling standard input only requires reading from standard input.