Design Document

Chris Sclipei

CruzID: csclipei

CSE 130, Fall 2019

1. **Goal**

   The goal of this program is to emulate Unix's *cat* command by reading in files specified by the user through standard input (*stdin*) and printing them back out to standard output (*stdout*); or if the user enters '-' it prints back what the user types in standard input back to standard output.

2. **Assumptions**

   I am assuming that all files that are used for *dog* are text files since that it is what *cat* is used. I am also assuming that I won't have to implement shell commands since that is already handled by Linux subsystem.

3. **Design**

   The approach I am taking with this program is using the basic Linux system calls which are: open(2), read(2), write(2), and close(2) to accomplish this rudimentary task. I will have to se a buffer limit too for reading in files to make sure there is a cap. Then I will use the number of bytes read from the system call to set the buffer for the write system call, this way there won't be extraneous buffer space when writing back to the standard output. This will be accomplished by using *file descriptors* to signify where the where the data will be read from, given either a file name or user input, and where the output will be printed, in this case, standard output. I will include error outputs to signal the user when there is an error with reading or writing a file. I would do this by using *fprintf()* and *perror()*, which would use *errno* to signal what error it ran into.

4. **Pseudocode**

Some pseudocode for the program is present below:

main arguments[]
Set buffer size for *read* and *write*
Declare *file descriptor* as an int
if *dog* is only argument **then**
read from stdin with a set *buffer*
write to stdout with a *buffer* set to value returned from *read*
end if
else
for *i* < *argument count* **do**
if argument *contains* '-' **then**
same as above if only dog is present
end if
open file
if fd is valid **then**
read file from stdin with a set *buffer*
write file to stdout with *buffer* size set form *read*
end if
else
print error msg for issues reading and writing files
end for
exit