Design Document: dog

Anthony Ho
CruzID: Ankho

1) Goals

The goal of this program is to implement dog which should be a simpler version of the cat command without support for flags. The command will take in file(s) as arguments and copy all the data from the file(s) to standard output in order. If the - is given as a file name, then the program will use standard input for that file.

2) Design

The design will include two parts. The program initializes, then processes the arguments and send them to respective functions based on the argument type.

2.1) Handling arguments

The arguments provided for the dog program will either be a file name or a - (dash). Otherwise, the input is invalid and we print an error. In order to handle these arguments, the program will take in the array of arguments and process them one by one. It will then check if the argument is either a file, a dash, or invalid and jump to the appropriate function. Initialization and argument handling will be shown in the following algorithm:

```
Input: Argument count: argc
Input: Array of arguments: argv
if(argc == 0):
        Pass to function to print from standard input
For int i = 1 to argc:
        If argv[i] is a file:
                Pass to file function to print data
        If argv[] == “-”:
                Pass to function to print from standard input
        Else:
                Print a warning (file/directory does not exist)
```

2.2) Printing

Printing will be handled by the different functions of the program depending on the input type.

2.2.1) Files

If the current input enters this function, then the input was a file and we need to take the data from that file and print it out. The algorithm will be like so:

Input: current file: file
Open file
Create buffer of size 32kb
Create boolean for loop condition
Read from the file and store the file size
If file size == 0 (Reached EOF):
    Set loop condition to true
If the file size i== -1:
    Print a warning (argument is a directory)
    Set loop condition to true
Write data to STDOUT
Free the buffer space
Close the opened file

2.2.2) Standard input

If the current input enters this function, then the input was a dash or there were not arguments and we need to take input from standard input and print out anything entered into it. The function will be like so:

Input: None
Create buffer of size 32kb
Read from the file and store the file size
While file size != 0: (Not at the end of STDIN/Ctrl+D not pressed)
    Write content from STDIN to STDOUT
    Reset the buffer to empty
    Set read file size to the next input in STDIN
Free the buffer space
