Asg0 Design Document

Alexander Soe cruzid: asoe

CSE 130, Fall 2019

1 Goal

The goal of this assignment is to design the dog command which is supposed to mimic the usage of the cat linux utility. The program should be able to print from stdin to stdout or print from files to stdout.

2 Assumptions

I'm assuming that we are going to have to check for an end of file or "ctrl-d" to check when to stop reading from stdin after a "-" is entered. Also there has to be a print to stderr when there a name is given that is not a file.

3 Design

The main parts of the program involve reading from stdin when a "-" is given and printing to stdout, reading from a file and printing to stdout, and printing to stderr when the argument given is not a file or "-". All of the parameters will be incapsulated by a for-loop that will go from one to argc which is the the number of arguments given. To print from stdin I'm going to have keep reading from stdin until a EOF is given or in other words when "ctrl-d" is typed by the user. To print from a file I'll have to read from the file with a buffer of size 32k and then write to stdout. If the file is larger than 32k I'll have a while loop that checks if the read returns 0 or if it has reached the EOF. If the file name given doesn't exist or a "-" is not given then I'm going to print out to stderr.

4 pseudocode

```
main(argc, argv)
    //Set buffer to 32k
    buffer[32k]
    //Loop through arguments given by dog call
    for(i to argc)
        //Check if the argument given was a ”-” and read from stdin and write
to stdout
        if(argv[i] == ”-”)
            set buffer = ” ”
            read(stdin, buffer)
            write(stdout, buffer)
            if(read != EOF)
                write(stdout, buffer)
            else
                break
        //Print an error if not a file
        else if (argv[i] != file  != ”-”)
            print error
        //Read from the file until eof and print to stdout
        else
            while(read != EOF)
                set buffer = ” ”
                read(argv[i], buffer)
                write(stdout, buffer)
            close(argv[i])
```