Design Document: Dog

Zhihang Zhou
CruzID: zzhou37

1. Goals

The goal of the program is to write a program can do something similar with cat does in Unix. Basically, what it does is reading the files name follow by the dog and print the file into the stout, then keep doing the same thing for the next file. If there is a file name called "-", the program will print what the user typed in through the keyboard.

One thing I need to ensure is none of the File* in the C library is allowed. So, try to avoid functions such as fread() or printf(), instead use the function fprintf(), perror() and warn(). The only function we can use is read(), write(), open() an close().

2. Design

Using a for loop to handle each of the argument typed by the user. In each of the for loop, the first step is to open the file and handling the error, like the file doesn't exist. The read the file into a buffer(32KB) and use write to print the file to stdout, if the file is bigger than 32KB, what the program does is printing the first 32KB and read the next 32KB in the same file and keep following the same step until the file is finished. Then the next step is close the file and the for loop will do the same thing for the next file in the command line until all the files or input been handled.

2.1 handling the argument and print the files/input

Using for loop to handle the argument (field 0 is stdin, 1 is stdout, 2 is stderro) followed by the ./dog.

First, the program will handle the argument "-" with function strcmp(). If the argument is – then witch, the input souse to stdin by give variable fild 0.

If the argument is a file name, the program will try to find if the file exists, if it doesn't exist print the error to stdout with perror(). Hint, open will return -1 if the file doesn't exist. Then give the field function open returned.

Finally, if the file is bigger than 32KB, use a while loop with the condition read returned is not equal 0, to print 32KB of the file each time until everything in the data have be printed.

The following is the Sudo code describe the program.

```
Nbyte of buffer
Int field
If (there is no argument){
        //Do the same thing as dash does;
        Read input from keyboard
        While (the file is not empty){
                Read 32kb and write it;
        }
}

For(the first argument to the last argument){
        Nbyte = size of the buffer
        If(argument is -){
                Set the input source to stdin;
        }
        Else{
                Open the file and give the return value to field
                If(the fild is -1 which means the file doesn't exist){
                        Print error message about file doesn't exist
                        Break the loop
                }

        Set the initial value of bufferSize to zero

        While(the read still return a positive number){
                read 32KB from the input source and give the return value to bufferSize
                write 32KB to the stdout
        }
```