Design Document: Dog

Rosario Jauregui
CruzID: rmjaureg

1 Goal

The goal of this program is to implement an application that functions similar to the linux command cat. It will print out the information inputted to the command line, it will read out the information from files in the command line to the terminal.

2 Design

There are 3 things to consider. Number 1, reading and writing from and to the terminal. Number 2, reading from a file and writing to the terminal. Number 3, when to read from a file and when to read from the terminal.

2.1 Reading and writing to terminal

The data will be read from stdin and output to stdout one byte at a time until the function read returns a value of zero signaling there is nothing more that will be read. The function that will be used will be called SimpOut( ).

```
Void SimpOut(){
  while(1){
        Read_count = read(stdin, &Reader, 1);
        if(Read_count == 0){
                Break;
        }
        write(stdout,&Reader, 1);
  }
}
```

2.2 Reading from a file and writing to terminal

The data is read from the file that is past to the function. If the file could not be open for any reason an error is thrown. Else the file is read one byte at a time until the end of the file is reached.

```
        FileOut(file_name){
                File_check = open(file_name);
                if(File_check == -1){
                        fprintf(stderr, "%s\n", strerror(errno));
                }else{
                        while(1){
                                Read_count = read(file_check, &Reader, 1);
                                if(Read_count == 0){
                                        Break;
```

```
                        }
                        write(stdout,&Reader, 1);
                }
        }
```

2.3 Deciding input

It must be decided when it is time to read from the terminal and when to read from a file. We first check the number of arguments.

we know that no other argument other than ./dog is used so we can just read from the termina. We have to work backwards and take care of the arguments. We have to check if the argument is a dash or a file name. If it is a dash we read from terminal, else we read from a file

```
if(argc == 1){
        SimpOut()
}else{
        for(int iter = argc - 1; argc > 0; -- argc){
                if(srtcmp(argv[iter], '-') == 0){
                        SimpOut();
                }else{
                        FileOut(argv[iter]);
                }
}
```