Design Document: Dog

Jake Hsueh-Yu Liao | hsliao

1. Goals

The goal of program ‘dog’ is to replicate the program ‘cat’ without certain functionalities. The program does not utilize any FILE* functions such as fopen(). The buffer size allocated by malloc with no more than 32kiB at a time.

2. Design

There are three parts to the design. The first is the outer loop that loops through all the arguments give or continue to take stdin when no arguments are given. The second part is a function that will take a file descriptor and copy the content of the file descriptor to stdout. A third function is used to find the path of the file.

2.1 Main

```
int main()
{
        if ( no input file) {
                While (!EOF) {
                        copyFile(stdin)
                }
        } else {
                for ( all files ) {
                        if (strcmp (“-”, arg[])) {
                                copyFile(stdin)
                        } else {
                                fd = getPath()
                                copyFile(fd)
                        }
                }
        }
}
```

2.2 CopyFile Function

```
copyFile(fd)
{
        Filesize = lseek(stdin, 0, seekend)
        lseek(fd, 0, SEEK_SET);
```

```
        do {
                if(residualBytes > BUFMAX) {
                        buf = malloc(BUFMAX);
                        read(fd, buf, BUFMAX);
                        write(STDOUT_FILENO, buf, BUFMAX);
                } else {
                        buf = malloc(residualBytes);
                        read(fd, buf, residualBytes);
                        write(STDOUT_FILENO, buf, residualBytes);
                }
                residualBytes = fileSize - (uint16_t)lseek(fd, 0, SEEK_CUR);
           } while(residualBytes > 0);
        return;
}
```

CopyFile finds out the file size using lseek, then reset file offset to the front. In a do-while loop, the buffer size is decided based on how many bytes still need to be read. The do-while loop ends when the remaining byte size is zero.

2.3 getPath Function

```
getPath(filename)
{
        path = malloc(sizeof(filename) +2;
        strcpy(path, "./")
        strcat(path, filename)
        return path
}
```

getPath is a simple function that appends a filename to a string "./" in order to get the filepath used for open().