

1 Code Listing

Code Listing 1: Minicat.c

```
1  /* Minicat_basic.c
2  *   Created on: Sep 15, 2017
3  *   Author: scott Jin
4  */
5  #include <errno.h>
6  #include <fcntl.h>
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <string.h>
10 #include <unistd.h>
11 #define EXIT_FAIL -1
12 void Read_Write(int fd_in,int fd_out,long buffersize, char* buffer, char* argu, char*
    write_to_file){
13     int writeout=0,readin=0;
14     while ((readin= read (fd_in, buffer, buffersize))){
15         writeout=write(fd_out, buffer, readin);
16         if(writeout<0) {                                //writing error
17             fprintf(stderr, "ERROR-->WRITE_FAILURE:_Could_not_write_to_file_(%s):_%s\n",
                write_to_file, strerror(errno));
18             exit(EXIT_FAIL);
19         }
20     } else if (readin!=writeout){
21         readin-=writeout;
22         buffer+=writeout;
23         write(fd_out, buffer, readin);
24         if(readin!=writeout){                            //try again with buffer offset
25             fprintf(stderr, "ERROR-->WRITE_FAILURE:_Could_not_write_to_file_(%s):_%s\n",    //partial writing error
                write_to_file, strerror(errno));
26             exit(EXIT_FAIL);
27         }
28     }
29 }
30 if (readin<0) {
31     fprintf(stderr, "ERROR-->READ_FAILURE:_Could_not_read_from_file_(%s):_%s\n", argu,
        strerror(errno)); //reading error
32 }
33 }
34 int main (int argc, char ** argv){
35     int bflags=0, oflags=0,fd_in=-2,fd_out=STDOUT_FILENO,opt=0;           //initialization
36     char* write_to_file=0;
37     long buffersize=512;
38     while ((opt = getopt(argc, argv, "b:o:")) != -1) {
39         switch (opt) {
40             case 'b':
41                 if(!strcmp ("-b", optarg)|| bflags++>0){
42                     fprintf(stderr, "ERROR->Invalid_argument:_only_one_-'b'_flag_can_be_accepted
                        .\n");
43                     exit(EXIT_FAIL);
44                 } else if(!strcmp ("-", optarg) || !strcmp ("-o", optarg)){
45                     fprintf(stderr, "ERROR->Void_argument:_no_argument_provided!\n");
46                     exit(EXIT_FAIL);
47                 } else if((buffersize= strtol(optarg, NULL, 0)) < 1){
48                     fprintf(stderr, "ERROR->Invalid_argument:_buffer_size_must_be_greater_than_0\n");
49                     exit(EXIT_FAIL);
50                 } break;
51             case 'o':
52                 if( !strcmp ("-o", optarg) || oflags++>0){
53                     fprintf(stderr, "ERROR->Invalid_argument:_only_one_-'o'_flag_can_be_accepted
                        .\n");
54                     exit(EXIT_FAIL);
55                 } else if (!strcmp ("-", optarg)||(!strcmp ("-b", optarg))){
56                     fprintf(stderr, "ERROR->Void_argument:_no/invalid_argument_provided!\n");
57                     exit(EXIT_FAIL);
```

```

58         }else{
59             fd_out = open(optarg, O_RDWR|O_CREAT|O_TRUNC, 0666);
60             write_to_file=optarg;
61             if(fd_out<0){
62                 fprintf(stderr, "ERROR->Failed to open file (%s) for writing: %s\n",
63                     optarg, strerror(errno));
64                 exit(EXIT_FAIL);
65             }
66         }break;
67     case '?':
68         fprintf(stderr, "ERROR-->Invalid option (or missing argument): %c\nUsage: Minicat [-b #####] [-o outfile] [infile1 [infile2 [...]]]\n", optopt);
69         exit(EXIT_FAIL);
70         break;
71     default:
72         fprintf(stderr, "ERROR-->Incorrect format: %s\nUsage: Minicat [-b #####] [-o outfile] [infile1 [infile2 [...]]]\n", argv[0]);
73         exit(EXIT_FAIL);
74     }
75 }
76 char *buffer=malloc(buffer_size);
77 if (write_to_file==0) write_to_file = "STDOUT_FILENO"; //in case writing error
78 for (; optind < argc; ++optind) { //loop through the option argument
79     //fprintf(stdout, "Processing Argument: %s\n", argv[optind]);
80     if(!strcmp("-", argv[optind])) {
81         fd_in = STDIN_FILENO;
82         argv[optind]="STDIN_FILENO";
83     }else if((fd_in = open(argv[optind], O_RDONLY)) < 0) {
84         if(!strcmp("-b", argv[optind])||!strcmp("-o", argv[optind])){
85             fprintf(stderr, "ERROR->Possibly Wrong place of option (%s): %s\nUsage: Minicat [-b #####] [-o outfile] [infile1 [infile2 [...]]]\n:", argv[optind], strerror(errno));
86         }else{
87             fprintf(stderr, "ERROR->Failed to open file (%s) for reading: %s\n", argv[optind], strerror(errno));
88         }
89         exit(EXIT_FAIL);
90     }
91     Read_Write(fd_in, fd_out, buffer_size, buffer, argv[optind], write_to_file);
92     if(fd_in!=STDIN_FILENO){
93         int status =close (fd_in);
94         if(status<0) {
95             fprintf(stderr, "ERROR->Failed to close Inputfile (%s): %s\n", argv[optind], strerror(errno));
96             exit(EXIT_FAIL);
97         }
98     }
99     if(fd_in==-2){
100         fd_in = STDIN_FILENO;
101         Read_Write(fd_in, fd_out, buffer_size, buffer, "STDIN_FILENO", write_to_file);
102     }
103     if(fd_out!=STDOUT_FILENO){
104         if ((close (fd_out))<0){
105             fprintf(stderr, "ERROR->Failed to close Outputfile (%s): %s\n", write_to_file, strerror(errno));
106             exit(EXIT_FAIL);
107         }
108     }
109     free(buffer);
110     return(EXIT_SUCCESS);
111 }

```
