Theo Song
Prof. Hakner
ECE 357
09/18/2019

Problem 3 -- Use of system calls in a simple concatentation program

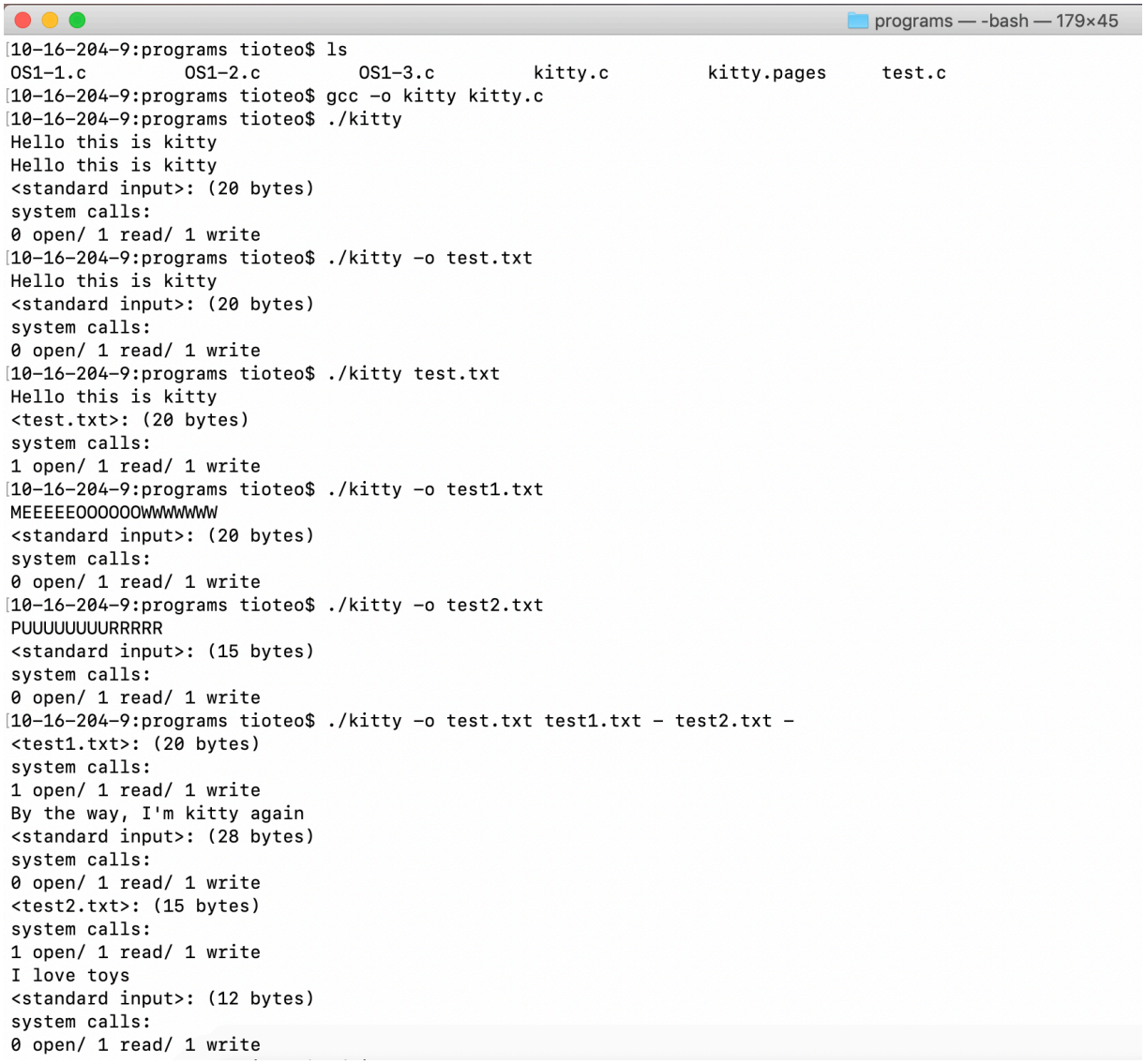
<Source Code>

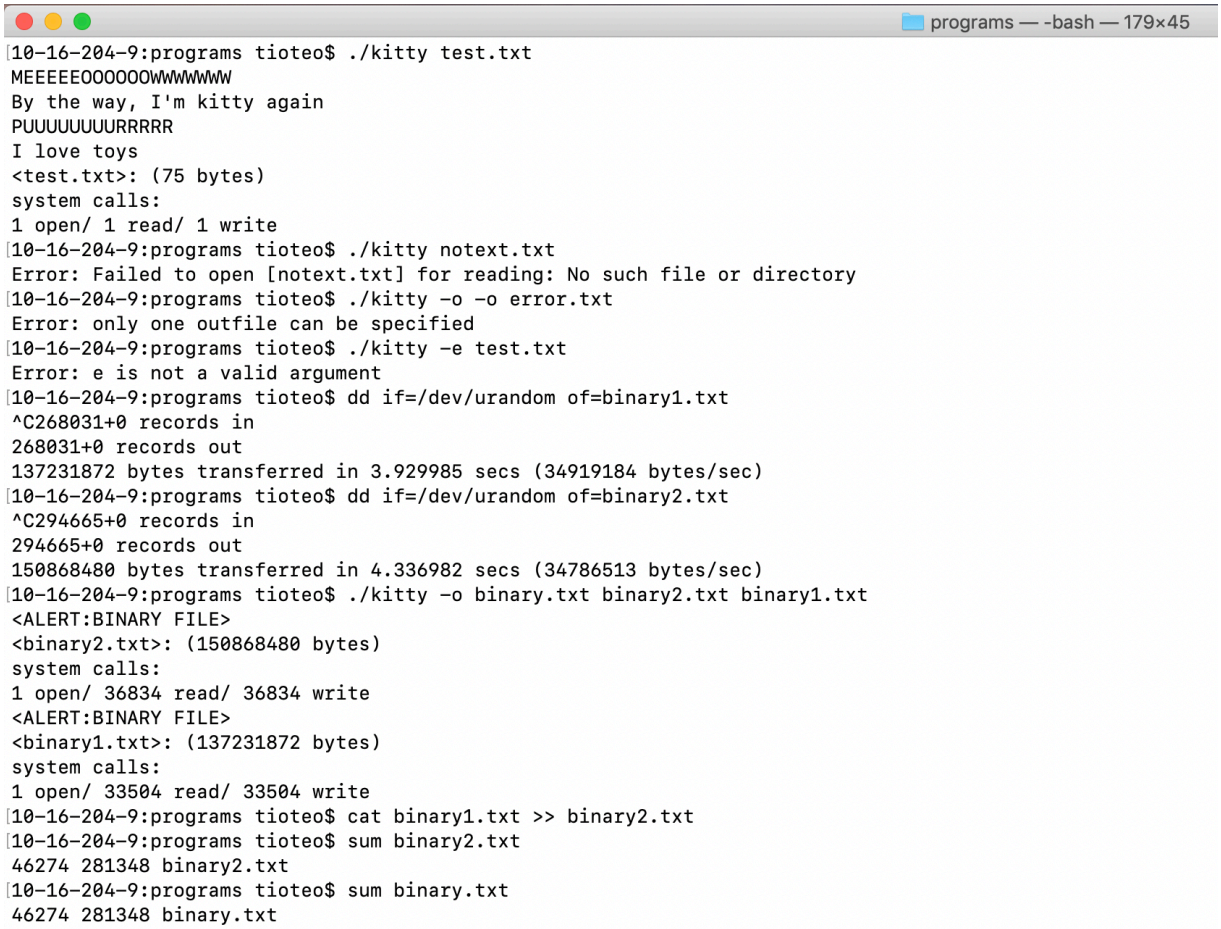
```
 1  #include <stdio.h>
 2  #include <string.h>
 3  #include <stdlib.h>
 4  #include <fcntl.h>
 5  #include <errno.h>
 6  #include <unistd.h>
 7  #include <ctype.h>
 8
 9  void report(int bytes, int o, int r, int w, char* filename, char* buf){ //reporting to stderr
10      int binary;
11      for(int n=0;n<bytes;n++){
12          binary = buf[n];
13          if(!(isprint(binary)||isspace(binary))){
14              fprintf(stderr,"<ALERT:BINARY FILE>\n");
15              break;
16          }
17      }
18      fprintf(stderr, "<%s>: (%d bytes)\nsystem calls:\n",filename,bytes);
19      fprintf(stderr,"%d open/ %d read/ %d write\n",o,r,w);
20  }
21
22  int perr(char* op, char* file, char* para, char*err){//printing error
23      if(errno != 0){
24          fprintf(stderr, "Error: Failed to %s [%s] %s: %s\n", op, file, para, err);
25          exit(errno);
26          return -1;
27      }
28      return 0;
29  }
30
31  void rnw(int c, char argv[], int fd){
32      int bytes=0, o=0, r=0, w=0;
33      int lim = 4096;
34      int tempfd, i, j;//i = read // j = write
```

```
35    char *buf = (char*)malloc(4096*sizeof(char));
36    switch (c)
37    {
38    case 1: //reading from infile writing to stdout
39       fd = open(argv,O_RDONLY,0666);
40       o++;
41       if(fd<0) perr("open",argv,"for reading",strerror(errno));
42       while((i=read(fd,buf,lim))>0){
43          r++;
44          perr("read",argv,"",strerror(errno));
45          while((j=write(1,buf,i))<i) {write(1,&buf[j],i-j); w++;} //partial write
46          bytes = bytes+j;
47          w++;
48          perr("write",argv,"to stdout",strerror(errno));
49       }
50       close(fd);
51       perr("close",argv,"",strerror(errno));
52       report(bytes,o,r,w,argv,buf);
53       break;
54    case 2: //reading from stdin writing to stdout
55       while((i=read(0,buf,lim))>0){
56          r++;
57          perr("read","stdin","",strerror(errno));
58          while((j=write(1,buf,i))<i) {write(1,&buf[j],i-j); w++;}
59          bytes = bytes+j;
60          w++;
61          perr("write","stdin","to stdout",strerror(errno));
62       }
63       report(bytes,o,r,w,"standard input",buf);
64       break;
65    case 3: //reading from infile writing to outfile
66       tempfd = open(argv,O_RDONLY,0666);
67       o++;
68       if(tempfd<0) perr("open",argv,"for reading",strerror(errno));
69       while((i=read(tempfd,buf,lim))>0){
70          r++;
71          perr("read",argv,NULL,strerror(errno));
72          while((j=write(fd,buf,i))<i) {write(fd,&buf[j],i-j); w++;}
73          bytes = bytes+j;
74          w++;
75          perr("write",argv,"to outfile",strerror(errno));
76       }
77       close(tempfd);
```

```
78        perr("close","infile",NULL,strerror(errno));
79        report(bytes,o,r,w,argv,buf);
80        break;
81     case 4: //reading from stdin writing to outfile
82        while((i=read(0,buf,lim))>0){
83           r++;
84           perr("read","stdin",NULL,strerror(errno));
85           while((j=write(fd,buf,i))<i) {write(fd,&buf[j],i-j); w++;}
86           bytes = bytes+j;
87           w++;
88           perr("write","stdin","to outfile",strerror(errno));
89        }
90        report(bytes,o,r,w,"standard input",buf);
91        break;
92     }
93     free(buf);
94  }
95
96  int main(int argc, char *argv[]){
97     int fd, option;
98     int o = 0;
99     while((option = getopt(argc,argv,":o")) != -1){ //argument checking
100       switch (option)
101       {
102       case 'o':
103          if(o>0) {fprintf(stderr,"Error: only one outfile can be specified\n"); exit(errno);}
104          o++;
105          break;
106       case '?':
107          fprintf(stderr, "Error: %c is not a valid argument\n", optopt);
108          exit(errno);
109          break;
110       }
111    }
112    if(argc==1) rnw(2,NULL,0); // outfile and infile not specified
113    else if(argc>1){
114       if(strcmp(argv[1],"-o")==0){ // outfile specified
115          fd = open(argv[2], O_RDWR|O_APPEND|O_CREAT|O_TRUNC,0666);
116          if(argc==3) rnw(4,NULL,fd); //infile not specified
117          if(argc>3){
118             for(int n=3;n<(argc);n++){
119                if(strcmp(argv[n],"-")==0) rnw(4,NULL,fd); // infile specified(-)
120                else rnw(3,argv[n],fd); // infile specified
```

```
121             }
122         }
123         close(fd);
124         perr("close","outfile",NULL,strerror(errno));
125     }
126     else{ //outfile not specified
127         for(int n=1;n<(argc);n++){
128             if(strcmp(argv[n],"-")==0) rnw(2,NULL,0); // infile specified(-)
129             else rnw(1,argv[n],0); // infile specified
130         }
131     }
132 }
133 return 0;
134}
```


programs — -bash — 179×45
10-16-204-9:programs tioteo$ ls
OS1-1.c OS1-2.c OS1-3.c kitty.c kitty.pages test.c
10-16-204-9:programs tioteo$ gcc -o kitty kitty.c
10-16-204-9:programs tioteo$ ./kitty
Hello this is kitty
Hello this is kitty
<standard input>: (20 bytes)
system calls:
0 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty -o test.txt
Hello this is kitty
<standard input>: (20 bytes)
system calls:
0 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty test.txt
Hello this is kitty
<test.txt>: (20 bytes)
system calls:
1 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty -o test1.txt
MEEEEEOOOOOOWWWWWWW
<standard input>: (20 bytes)
system calls:
0 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty -o test2.txt
PUUUUUUUURRRRR
<standard input>: (15 bytes)
system calls:
0 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty -o test.txt test1.txt - test2.txt -
<test1.txt>: (20 bytes)
system calls:
1 open/ 1 read/ 1 write
By the way, I'm kitty again
<standard input>: (28 bytes)
system calls:
0 open/ 1 read/ 1 write
<test2.txt>: (15 bytes)
system calls:
1 open/ 1 read/ 1 write
I love toys
<standard input>: (12 bytes)
system calls:
0 open/ 1 read/ 1 write


programs — -bash — 179×45
10-16-204-9:programs tioteo$ ./kitty test.txt
MEEEEEOOOOOOWWWWWW
By the way, I'm kitty again
PUUUUUUUURRRRR
I love toys
<test.txt>: (75 bytes)
system calls:
1 open/ 1 read/ 1 write
10-16-204-9:programs tioteo$ ./kitty notext.txt
Error: Failed to open [notext.txt] for reading: No such file or directory
10-16-204-9:programs tioteo$ ./kitty -o -o error.txt
Error: only one outfile can be specified
10-16-204-9:programs tioteo$ ./kitty -e test.txt
Error: e is not a valid argument
10-16-204-9:programs tioteo$ dd if=/dev/urandom of=binary1.txt
^C268031+0 records in
268031+0 records out
137231872 bytes transferred in 3.929985 secs (34919184 bytes/sec)
10-16-204-9:programs tioteo$ dd if=/dev/urandom of=binary2.txt
^C294665+0 records in
294665+0 records out
150868480 bytes transferred in 4.336982 secs (34786513 bytes/sec)
10-16-204-9:programs tioteo$ ./kitty -o binary.txt binary2.txt binary1.txt
<ALERT:BINARY FILE>
<binary2.txt>: (150868480 bytes)
system calls:
1 open/ 36834 read/ 36834 write
<ALERT:BINARY FILE>
<binary1.txt>: (137231872 bytes)
system calls:
1 open/ 33504 read/ 33504 write
10-16-204-9:programs tioteo$ cat binary1.txt >> binary2.txt
10-16-204-9:programs tioteo$ sum binary2.txt
46274 281348 binary2.txt
10-16-204-9:programs tioteo$ sum binary.txt
46274 281348 binary.txt