Danny Hong and Ravindra Bisram
ECE 357 Operating Systems Assignment 1 Problem 3

Source Code:

```
// Danny Hong and Ravindra Bisram
// ECE 357 Problem Set 1 Question 3
// Kitty Source Code written in C

#include <string.h>
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <unistd.h>
#include <ctype.h>

int main(int argc, char* argv[]){
    int bufferSize = 4096, systemReadCalls = 0, systemWriteCalls = 0,
totalWritten = 0, writeCount = 0, readCount, outputFd = STDOUT_FILENO,
inputFd, flag;
    char *outfile = "";
    char *buffer = malloc((sizeof(char)) * bufferSize);

    /*Checks for the possibility of a malloc error when defining the
buffer.*/
    if(buffer == NULL) {
        fprintf(stderr, "Error! Malloc error. \n%s", strerror(errno));
    }

    /*Uses getopt to search for the [-o outfile] and [-b ###]
options.*/
    while((flag = getopt(argc, argv, "b:o:")) != -1){
        switch(flag){
            case 'o':
                outfile = optarg;
                break;
            case '?':
                fprintf(stderr, "Error! Unrecognized argument
detected.");
                return -1;
            default:
                fprintf(stderr, "Error! Incorrect format: No argument
following the -o and -b flags. \n%s\n", argv[0]);
                return -1;
        }
    }

    /*Checks to see if there is a specified outfile. If there is, it
is then opened.*/
    if(strcmp(outfile, "")){
        outputFd = open(outfile, O_WRONLY | O_CREAT | O_TRUNC, 0666);
        /*Returns an error statement if output file can't be opened
for writing.*/
        if(outputFd < 0){
            fprintf(stderr, "Error! Cannot open output file: %s for
writing. \n%s\n", outfile, strerror(errno));
            return -1;
        }
    }
```

```
    /*Checks for zero input. If there is, it is treated as a "-" and
"-" is appended to argv.*/
    if (optind == argc){
        optind = argc = 0;
        argv[argc++] = "-";
    }

    /*Iterates through the arguments beginning at optind + 1 and
ending at the final argument.*/
    for(; optind < argc; optind++){
        /*Returns an error statement if input file can't be opened for
writing.*/
        if(!(strcmp(argv[optind], "-"))){
            inputFd = STDIN_FILENO;
            argv[optind] = "standard input";
            }
        else if((inputFd = open(argv[optind], O_RDONLY)) < 0){
            fprintf(stderr, "Error! Cannot open input file: %s for
reading. \n%s\n", argv[optind], strerror(errno));
            return -1;
            }
        /*Implementing the read and write operations with correction
to deal with partial writes.*/
        while ((readCount = read(inputFd, buffer, (sizeof(char)) *
bufferSize)) != 0){
            if(readCount > 0){
                /*Returns an error warning if an input file happens to
be binary file. */
                for(int index = 0; index < readCount; index++){
                    if(!(isspace(buffer[index]) ||
isprint(buffer[index]))){
                        fprintf(stderr, "Warning! Trying to
concatenate the binary file: %s\n", argv[optind]);
                        return -1;
                    }
                }
                        systemReadCalls = systemReadCalls + 1;
                while(readCount > writeCount){
                    if((writeCount = write(outputFd, buffer,
readCount)) < 0){
                        fprintf(stderr, "Error! Issues writing to
output file: %s. \n%s\n", outfile, strerror(errno));
                                            return -1;
                    }
                    systemWriteCalls = systemWriteCalls + 1;
                    totalWritten = totalWritten + writeCount;
                    readCount = readCount - writeCount;
                    writeCount = 0;
                    buffer = buffer + readCount;
                }
            }
            else{
                fprintf(stderr, "Error! Issues reading input file: %s.
\n%s\n", argv[optind], strerror(errno));
                return -1;
            }
```

```
        }
        if(inputFd != STDIN_FILENO){
            fprintf(stderr, "%d bytes transferred to output file: %s
from input file: %s. Number of system read calls = %d. Number of
system write calls = %d\n", totalWritten, outfile, argv[optind],
systemReadCalls, systemWriteCalls);
            /*Returns an error statement if there are issues closing
an input file that is not standard input.*/
            if(close(inputFd) < 0){
                fprintf(stderr, "Error! Cannot close input file: %s
%s\n", argv[optind], strerror(errno));
                                return -1;
            }
                }
        else {
                        fprintf(stderr, "%d bytes transferred to
output file: <standard output> from input file: <standard input>.
Number of system read calls = %d. Number of system write calls =
%d\n", totalWritten, systemReadCalls, systemWriteCalls);
                }
    }

    /*Returns an error statement if there are issues closing an output
file that is not standard output.*/
    if (close(outputFd) < 0 && outputFd != STDOUT_FILENO){
        fprintf(stderr, "Error! Cannot close output file: %s. \n%s\n",
outfile, strerror(errno));
        return -1;
    }
    return 0;
}
```

Concatenate With and Without -o Flag Test

```
ravin@DESKTOP-1H47C32 ~/OS
$ cat input1.txt
Ravindra Bisram test 1

ravin@DESKTOP-1H47C32 ~/OS
$ cat input2.txt
Danny Hong test input 2

ravin@DESKTOP-1H47C32 ~/OS
$ cat input3.txt
OS Fall 2020 Prof Hakner test input 3

ravin@DESKTOP-1H47C32 ~/OS
$ gcc kitty.c -o kitty

ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty input1.txt input2.txt input3.txt
Ravindra Bisram test 1
23 bytes transferred to output file:  from input file: input1.txt. Number of system read calls = 1. Number of system write calls = 1
Danny Hong test input 2
47 bytes transferred to output file:  from input file: input2.txt. Number of system read calls = 2. Number of system write calls = 2
OS Fall 2020 Prof Hakner test input 3
85 bytes transferred to output file:  from input file: input3.txt. Number of system read calls = 3. Number of system write calls = 3

ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -o output.txt input1.txt - input2.txt - input3.txt
23 bytes transferred to output file: output.txt from input file: input1.txt. Number of system read calls = 1. Number of system write calls = 1
Standard input test 1
45 bytes transferred to output file: output.txt from input file: <standard input>. Number of system read calls = 2. Number of system write calls = 2
69 bytes transferred to output file: output.txt from input file: input2.txt. Number of system read calls = 3. Number of system write calls = 3
Standard input test 2
91 bytes transferred to output file: output.txt from input file: <standard input>. Number of system read calls = 4. Number of system write calls = 4
129 bytes transferred to output file: output.txt from input file: input3.txt. Number of system read calls = 5. Number of system write calls = 5

ravin@DESKTOP-1H47C32 ~/OS
$ cat output.txt
Ravindra Bisram test 1
Standard input test 1
Danny Hong test input 2
Standard input test 2
OS Fall 2020 Prof Hakner test input 3
```

*When printing to standard output, it should read “x bytes transferred to output file: <standard output>” instead of an empty output file name. This was corrected in the submitted code.

Testing standard input to standard output (No input arguments)

```
ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -
HI
HI
How are you?
How are you?
This is just a test :D
This is just a test :D
40 bytes transferred to output file:  from input file: <standard input>. Number of system read calls = 3. Number of system write calls = 3
```

Testing Error Cases for Input

```
ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty thisFileDoesNotExist.txt
Error! Cannot open input file: thisFileDoesNotExist.txt for reading.
No such file or directory

ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -o
kitty: option requires an argument -- o
Error! Unrecognized argument detected.
ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -w
kitty: unknown option -- w
Error! Unrecognized argument detected.
```

Testing with Multiple Hyphens on Command Line

```
ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -o output3.txt input1.txt - - input2.txt
23 bytes transferred to <output3.txt> from <input1.txt>. # of read sys call = 1. # of write sys call = 1
Consecutive std input 1
24 bytes transferred to <output3.txt> from <standard input>. # of read sys call = 1. # of write sys call = 1
Consecutive std input 2
24 bytes transferred to <output3.txt> from <standard input>. # of read sys call = 1. # of write sys call = 1
24 bytes transferred to <output3.txt> from <input2.txt>. # of read sys call = 1. # of write sys call = 1

ravin@DESKTOP-1H47C32 ~/OS
$ cat output3.txt
Ravindra Bisram test 1
Consecutive std input 1
Consecutive std input 2
Danny Hong test input 2
```

Concatenate Binary File Test

```
ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -o output2.txt dummybinaryfile.txt
Warning! Trying to concatenate the binary file: dummybinaryfile.txt

ravin@DESKTOP-1H47C32 ~/OS
$ ./kitty -o output2.txt input1.txt dummybinaryfile.txt
23 bytes transferred to output file: output2.txt from input file: input1.txt. Number of system read calls = 1. Number of system write calls = 1
Warning! Trying to concatenate the binary file: dummybinaryfile.txt

ravin@DESKTOP-1H47C32 ~/OS
$ cat output2.txt
Ravindra Bisram test 1
```