```
//Jessica Elkins
//CS332
//Lab 4
//2/13/20
//This program takes 2 files as command line aarguments and concatenates file argument
two on to the
// end of file argument one.

// To compile: gcc lab4.c -o lab4
// To run: ./lab4 file1 file2

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char **argv) {

        // read and write file descriptors
        int rfd, wfd;
        //buffer
        char buffer[BUFSIZ];
        //stores number of bytes
        long int n;

        // if there are not 3 command line arguments
        if(argc != 3) {
                // prints error message
                printf("Usage: %s <destination> <source> \n", argv[0]);
                // terminates program
                exit (-1);
        }

        // if input and output file are the same
        if(strcmp(argv[1], argv[2]) == 0) {
                // prints error message
                printf("Input and output file names cannot be the same. Exiting. \n");
                // terminates program
                exit (-1);
        }

        // opening the file given as the third arrgument as read only
        rfd = open(argv[2], O_RDONLY);

        // opening the file given as the second argument as write only,
        // creates new file if the file name does not exist, and
        // appends the file if it already exists
        wfd = open(argv[1], O_CREAT|O_WRONLY|O_APPEND);


        // if rfd or wfd return -1
        if(rfd == -1 || wfd == -1) {
                //print error message
                printf("Error with opening the file. Exiting. \n");
                //terminates program
                exit (-1);
        }

        // read function takes the file descriptor, input buffer, and the max
        // buffer size as parameters and returns the number of bytes read
        while((n = read(rfd, buffer, BUFSIZ)) > 0){
                // write function takes the file descriptor, buffer, and number
                // of bytes to write as parameters and returns number of bytes written
```

```
                if(write(wfd, buffer,n) != n){
                        // prints error message if there was an error writing to the fi
le
                        printf("Error writing to %s. Exiting. \n", argv[1]);
                        exit (-1);
                }
        }

        // closing the files that were opened
        close(rfd);
        close(wfd);

        return 0;

}
```