```
//          Course: CS3820-01 Operating System
//            Name: Sajan, Kefin
//      Assignment: Programming Assignment 2
//   Date assigned: 9/05/17
//        Date due: 9/14/18
//  Date handed in: 9/13/18
//          Remark: This is a system program that copies the contents of an existing ASCII
//                  text file (name it source.txt) to a newly created empty file
//                  (name it target.txt). The program is compiled and linked into
//                  an executable file name copy. So when the user executes the command
//                  copy source.txt target.txt at command line, the contents of the
//                  source file are copied to the target file.

// Simple utility program: copies the contents of a source file to a target file
// Objectives: to learn UNIX system calls open(), create(), read(), write(), and exit()
// ***************************************
// * To compile: g++ 2.cpp -o copy           *
// *     To run: ./copy file2 file1          *
// ***************************************

#include <iostream>
#include <fcntl.h>
//fcntl is C header used for file management
//      Ex. opening, closing, changing permissions, etc.

void copy(int, int);
// What are passed?
// The file descriptors are passed into the function

char buffer[2048];
// What is the buffer used for? What does 2048 mean?
//     This Buffer is used to limit the amount of character stored inside the program
//     2048 stands for 2 kilobytes of storage to store 2048 characters

int main(int argc, char *argv[])
{
       int fd_source, fd_target;
       // file descriptors; What are they used for?
       //     A file descriptor is an integer value returned by the open() call

       if (argc != 3) {
              // To make sure program has two file to
              printf("Need two arguments!\n");
              return 1;
       }
       fd_source = open(argv[1], O_RDONLY);
       // What is argv[1]? What is "O_RDONLY"
       //     "argv[1]" is the file name
       //     O_RDONLY means that the file contents is to be read only
       //       and is not allowed to be modified.

       if (fd_source == -1) {
              // make sure "open" call is successful
              printf("Cannot open %s file!", argv[1]);
              return 1;
       }

       fd_target = creat(argv[2], 0666);
       // What does 0666 mean?
       //     "0666" is specific permission that is set to the file
       //       File is only read and write for user, group and other
```

```
	if (fd_target == -1) {
		// make sure "create" call is successful
		printf("Cannot create %s file!", argv[2]);
		// What is argv[2]?
		//  The second file name passed into the program
		return 1;
	}

	copy(fd_source, fd_target);
	// Copy function

	return 0;
}
void copy(int source, int target)
{
	int count;

	while ((count = read(source, buffer, sizeof(buffer))) > 0)
		write(target, buffer, count);

}

// Output:
```