Homework 2 - Copy File

Due by 5:00 p.m. Tuesday, 9/16/14

Write a copy program similar to `cp` in Bash. An execution of your program will take two command-line arguments. It should then try to open the file named by the first command-line argument. If successful, it should try to create a file with the name given in the second command-line argument, but without clobbering (deleting) any existing file by that name. If again successful, it should loop through the contents of the first file, writing them to the second file. Finally, it should close both files and print out the total number of bytes copied. Be sure to check for any errors in opening, creating, reading and writing the files, and print an appropriate error message if one is encountered. An execution of your program should look like this:

```
~$ ./mycp file1 file2
copied 512 bytes
```

You will implement this program using both the POSIX API and the Win32 API. Your code should use a buffer size of 256 bytes. For the POSIX implementation, use the low level file I/O API for reading/writing the files (i.e. `open`, `close`, `read` and `write`). This API is also referred to as the OPEN family of Linux system calls. Use the stdio formatted I/O API (i.e. the `printf` family) for writing the number of bytes copied. Use the man page entries for function prototypes, details on how to use each system call, as well as all of the necessary `#include` files. Remember that most system calls are in section 2 or 3 of the man pages (i.e. use 'man 2 write' not 'man write' to get information about the `write` system call). Use the Linux `diff` utility to compare the input and output files of your program.

For the Win32 implementation, follow the instructions above but use the Win32 File Management API routines for low level file I/O (`OpenFile`, `CreateFile`, `ReadFile`, `WriteFile`, `CloseFile`). See the MSDN File Management Document for details (note that you ONLY need to worry about simple File I/O, not all of the various extensions supported by the Win32 API): http://msdn.microsoft.com/en-us/library/windows/desktop/aa363858%28v=vs.85%29.aspx. Use the Windows `comp` utility to compare the input and output files of your program.

You should submit your source code files (one for each platform) and a short writeup in pdf format that includes a description of what you did and the compilation and execution output from each of your programs. Test each version of your program with both a binary file and a text file and both a file under 256 bytes and one over 256 bytes. (These may overlap, so that you only need two total tests for each version.) Submit everything to the regular submission link on iLearn, and then submit just the writeup to the TurnItIn link to generate an originality report.