

Your project deliverables are to be posted to CUNY Blackboard using the Project 1 link.

1) Only one source code file as follows with filename as in the example where you will put your own first initial and lastname and your

own section number. I show the example for my own name with an example section 23:  
JSvadlenka\_prj1\_sect23\_src.c (or .cpp)

a) At top of the file: your name in comments, for example: // First\_Name Last\_Name

b) Next, the gcc or g++ code compilation and linkage command, for example: // gcc  
JSvadlenka\_prj1\_sect23\_src.c -o JSvadlenka\_prj1\_sect23\_src.exe

c) In-line code comments no fewer than every 5 lines of code which describe your algorithmic steps. Syntax transcriptions of the code are  
not considered in-line code comments! (For example, I do not want to see the comment for i++ which says "Increment i.")

2) An MS Word or PDF document with a write-up of your software design and an explanation of the issues encountered and resolved. (1 page)

Filename should be according to this example of my own name:  
JSvadlenka\_prj1\_sect23\_wrt.doc (or pdf)

## Project Description

-----

In your first project you will create a program that, when executed, has both a parent process and a child process. You use the fork() system call to create a child process.

As in the cpfile.c program, your program takes two command line arguments for a source text file name and a target filename with the same semantics (source file exists in the same directory as the executable and the target file does not already exist in this same directory).

The parent process will read the source file contents in 128-character chunks (as in cpfile.c) and write it to a pipe (instead of an output file). The child process will read the contents of the pipe in 128-character chunks and write the contents to both the target file and to the display screen in 128-character chunks. Obviously, the last chunk may be less than 128 characters based on the size of the source file.

1) Your program accepts two command line arguments which are the name of the input file to be read and the output file to be created.

2) Your program will make use of the POSIX system I/O calls discussed in lectures.

3) You are NOT permitted to use any C++ nor C standard I/O library functions nor cin nor cout. Furthermore, you cannot use  
any C++ nor C standard library functions for string operations.

4) The parent and child processes may each use their own character buffer array of size 128.

5) I will test your code with source text files having 1 or more lines of text and each line may have more than 128 characters.

6) This project essentially utilizes the code of cpfile.c for file copying and the code of Fig 3.2.2 for pipes in the textbook.