CS231 Laboratory Assignment 3

Spring 2019

Overview

For this 40 point lab, you will implement the `cat` command. This means, among other things, that you will not call `cat` from your program by using `system()` or `exec()` or other similar means.

Specifics

You will write a program which will run correctly on the linux operating system. Your program will output the same results as running the command `cat`. For example, if you name your executable file `kat`, then there will be no difference in output from running `cat` and running your program, `kat`.

Your program will also perform correctly with command line arguments b, E, and n. The argument 'E' means write a '$' character at the end of each output line; 'n' means write the line number before each line of output; and 'b' is the same as 'n' except that empty lines (lines containing just the newline character) are not labelled with a number, nor do they increase the line number. Zero, one, two, or all three arguments may be applied to a single run of cat. Whenever 'b' and 'n' are used on the same run of cat, the effects are those of the argument 'b'. The command `cat` with no file name arguments takes its input from standard input. There can be any number of file names for command line arguments, which `cat` will concatenate and print on stdout.

Note that a `"-"` anywhere in the list of files stands for input from stdin, so you can mix stdin with file input. Note also that there can be several `"-"` characters in the list of files. To send an end of file from the keyboard, use the combination of keys CTRL and D.

Input

Test your program with multiple files. Be sure that you have lines containing only blanks, lines containing no characters (other than the newline character to end the line) and lines that end in blanks, as well as lines that end in printable characters.

Output

The output will be the same as running the command `cat` with the appropriate command line arguments.

Notes

1. Actually run cat on a linux system before writing your code to see how it behaves. If you create test files file1, file2, and file3 which contain multiple lines, some lines with only white space in them, some lines which are completely empty, then run cat in these ways (and others) to see how it behaves:
   - cat file1 - -b file2 - file3
   - cat file1 -bEn file2 -
   - cat -n file1 - file3 file2 -
2. the arguments -b,-n, and -E can be given in any order in the command line, and can be combined in any order. For example, all the following give the same result:
   - cat -b file -E -n
   - cat file -E -b -n
   - cat -bn file -E
   - cat file -bEn

3. Make sure your program behaves appropriately when supplied with a file which does not exist. (Note in particular which values go to stdout and which go to stderr, you can tell this by redirecting stdout with the '>' character, for example: cat a > outFile. //stdout goes to the file outFile, stderr goes to the screen).

4. Remember that your command line arguments accessed by parameter argv of main are strings, not just arrays of char.

Deliverables

Submit your source code via Blackboard by 11:59:59p.m. on Tuesday April 23. You should comment your program similar to the example prime.c, but do not have your program print information which the linux cat program does not output. (For example, do not print title/author information).