CSCD 255
Lab 10

SPECIFICATIONS

The purpose of this lab is to use C system calls to write directly to the hardware. The system calls you will need are open, write, read.

I have provided a main.c, lab10.h and lab10.c

- readFileName(char * type) – asks the user to enter a filename based on the string held in the variable type and reads that filename from the keyboard. The function dynamically allocates the memory for that filename, string copies the filename entered on the keyboard into the dynamically allocated memory and returns that pointer for the dynamic memory.
- readString() – asks the user to enter a string and reads that string from the keyboard. The function dynamically allocates the memory for that string, string copies the string entered on the keyboard into the dynamically allocated memory and returns that pointer for the dynamic memory.
- textToBinary(char * theString, char * filename) – writes to the file, specified by filename, the string in binary format.
    - Uses the open command to open the file in the mode O_CREAT|O_WRONLY
    - Writes the length of the string into the file, in binary
    - Writes the string into the file, in binary
    - You will NOT write a carriage return after the string
    - Closes the file descriptor
- menu – provided by me. Carriage return is removed from the input buffer
- cleanUp(char * theString) – frees the parameter passed in and returns NULL;
- binaryToText(char * filename) – retrieves the string that was written to the file in binary and converts it back to ASCII text.
    - I have provided a small amount of code to fix a problem with repl.it
    - Uses the open command to open the file in the mode O_RDONLY
    - Reads the length of the string using the read command
    - Dynamically allocates a string of the length just read from the file + 1
    - Reads the string from the file into the dynamically allocated string
    - Closes the file descriptor
    - Returns the dynamically allocated string

NOTES

- You will need open, write and read system calls.
- The commands for the binary file must be low level system call commands. You are writing binary files so don’t write a carriage return.
- You will need to create your own files

- No line will be more than 100 characters.
- The output to the screen should be the exact same as when it was typed on the keyboard.

TO TURN IN:

A zip file containing:

- All your code.
    - NOTE: repl.it will not include your binary files that is fine
- An output run named cscd255lab10out.txt
- A valgrind run named cscd255lab10val.txt showing everything is leak free.

Your zip will be named your last name first letter of your first name lab10.zip (Example: steinerslab10.zip)