Lab 2 - Description

(String processing in C)

Lab Overview:

For this lab, we will be learning how to process complex strings from the console input. Typically, user I/O for operating systems CLI's takes the form of processing complex string input by the user into a shell/terminal or console. These commands are then parsed and sent to a sub-process that executes them using a wide variety of system calls or predesigned routines. In this lab, we will be going over how to take complex input from the console on a line-by-line basis and parsing that into understandable tokens that can be used by the system.

Core Tasks:

1. Take input from the console using "getline()".
2. Tokenize the input string.
3. Display each individual token.
4. Implement the **"exit"** command.
5. File I/O.
6. Test your code with Valgrind for memory leaks.

Task Details:

1. Take input from the console using "getline()".
    a. Here is the man page for getline(3): http://man7.org/linux/man-pages/man3/getline.3.html
2. Tokenize the input string.
    a. Utilize strtok(2): https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm
3. Display each individual token. (see the following images for the format. Your program must match this **exactly**.) **Save this program as lab2-a.c.** Use the provided skeleton code for this file.

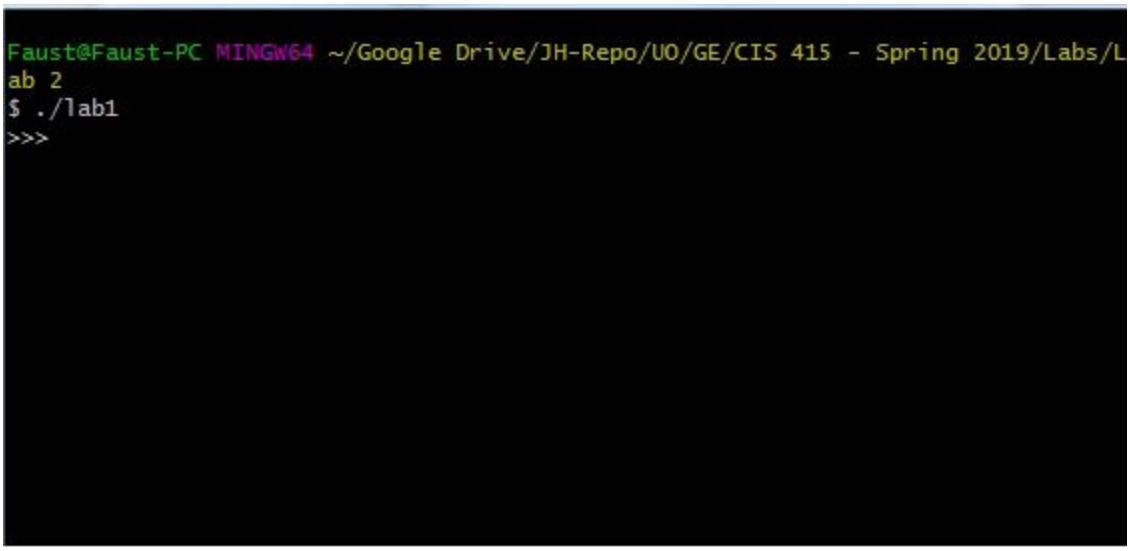

Fig. 1: Ready state

Fig 2. Displaying tokens gathered from string input

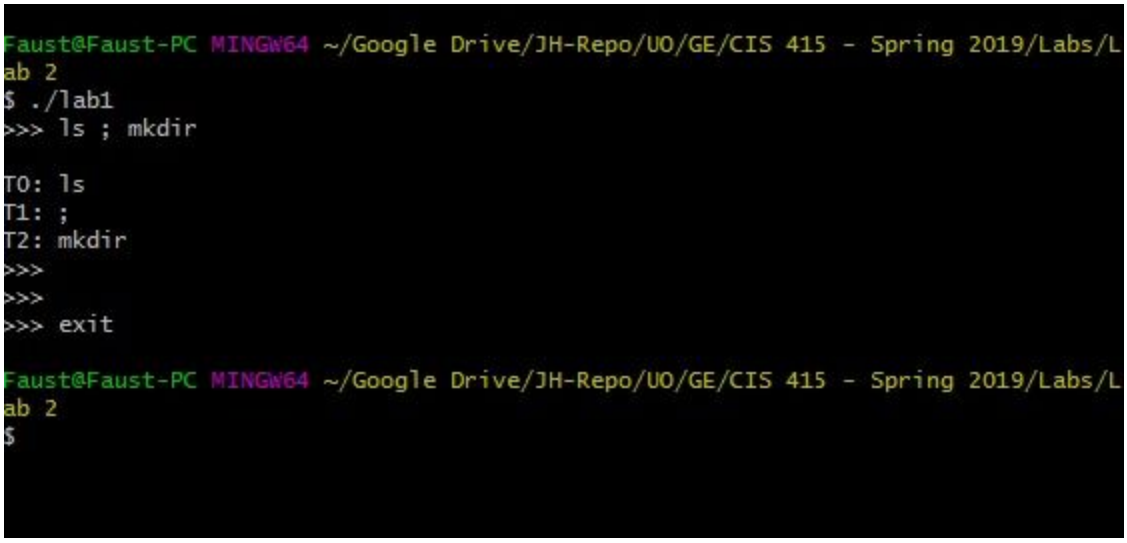

Fig. 3: Format for “exit” and <enter>

4. Include file I/O in your program.
    a. Edit the above code to take a filename from **argv,** open the file, and write each token in the same format as the images shown above. (See the attached sample input and output files). **Save this new program as lab2-b.c.**
5. Test your code with Valgrind for memory leaks.

Submission Requirements:

In order to receive any credit for a lab, completion of the labs' core tasks must be demonstrated to the TA’s. A file lab2-skeleton has been provided. **Use this file for both lab2-a and lab2-b.** In order to receive points for this lab the student must show 3 things:

1. Lab2-a.c
2. Lab2-b.c
3. Valgrind output with leak-check and mem-check showing no memory leaks.