

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_r(3): <http://man7.org/linux/man-pages/man3/strtok.3.html>
3. Display each individual token. (see the following images for the format. Your program must match this **exactly**.) **Save this program as main.c.** Use the provided skeleton code for this file. **Note:** If the user enters the keyword **exit** then your program should break from the loop, free all allocated memory and exit gracefully (i.e. no errors or leaks). **Note2:** Pay attention to figure 3, it shows how your program should behave when the user presses the enter key and when it exits.

```
Faust@Faust-PC MINGW64 ~/Google Drive/JH-Repo/U0/GE/CIS 415 - Spring 2019/Labs/L
ab 2
$ ./lab1
>>>
```

Fig. 1: Ready state

```
Faust@Faust-PC MINGW64 ~/Google Drive/JH-Repo/U0/GE/CIS 415 - Spring 2019/Labs/L
ab 2
$ ./lab1
>>> ls ; mkdir

T0: ls
T1: ;
T2: mkdir
>>> |
```

Fig 2. Displaying tokens gathered from string input

```
Faust@Faust-PC MINGW64 ~/Google Drive/JH-Repo/U0/GE/CIS 415 - Spring 2019/Labs/L
ab 2
$ ./lab1
>>> ls ; mkdir

T0: ls
T1: ;
T2: mkdir
>>>
>>>
>>> exit

Faust@Faust-PC MINGW64 ~/Google Drive/JH-Repo/U0/GE/CIS 415 - Spring 2019/Labs/L
ab 2
$
```

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 main2.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 main.c and main2.c.**

In order to receive points for this lab the student must show 3 things:

1. That both files successfully compile.
2. That both programs successfully run with the correct console output.
3. The Valgrind output with leak-check and mem-check showing no memory leaks. (See lab 1 for the cmd to do this)