



COMP 201 - Spring 2021
Assignment 2 - Strings in C
Assigned: 11 March 2021 23:59, Due: 24 March 2021 23:59

Contact TA for this Assignment: Ahmed Imam Shah (ashah20@ku.edu.tr)

1 Introduction

The purpose of this assignment is to become more familiar with Strings in C language and analyzing some text. You will have to deal with text input and output modules. You'll do this by completing the some tasks in this assignment. You can be creative in implementing your algorithms and designing the I/O interface.

2 Logistics

This is an individual project. All hand-ins are electronic. Clarifications and corrections will be announced on Blackboard. If you face any problem, please post on the discussion board on the Blackboard.

3 Handout Instructions

3.1 How to start

- I Accept the GitHub Classroom assignment using the link: <https://classroom.github.com/a/9JBIiArc>
- II Clone the GitHub repository created for you to a Linux machine in which you plan to do your work (We advice you to do your work on our Linux servers [linuxpool.ku.edu.tr]. See Section 7 for details.)
:

```
$ git clone https://github.com/COMP201-Spring2021/assignment-2-USER.git
```

(Replace USER with your GitHub username that you use to accept the assignment)
- III In this assignment, you are expected to use the command line arguments as the use cases in 4.1 and 4.2 suggests. A template for this purpose is provided in the main.c file that needs to be extended for your usage.
- IV Commit changes to repository from time to time.

3.2 Dataset

You are provided with a text file *shakespeare.txt*. This file contains 17 plays written by William Shakespeare (1564-1616). Here are some details about the dataset that you may need during the assignment:

- Each play begins with a particular String - [NEW PLAY].
- An act or a scene starts by the stings <ACT *Number*> and <SCENE *Number*> respectively. And an act or a scene ends by the stings </ACT *Number*> and </SCENE *Number*> respectively.
- A dialogue of a character starts by the string <NAME >. For example, Antonio's dialogue start with a string <ANTONIO> And a dialogue of a character ends by the string </NAME>. For example, Antonio's dialogue ends with a string </ANTONIO>

You are also provided with another text file *characters.txt*. This file contains names of 20 characters in the plays. Here are some details about the dataset that you may need during the assignment:

- Each line has a different character.
- Each line ends with an end-of-line character.

Please take a look at the data set before you start implementing your assignment. Please note that you are not allowed to make any changes to the original text files that are provided to you.

4 Tasks

This section there are two tasks that you need to implement in order to complete this assignment.

4.1 Dialogue Counter

In this task you will have a character name as an input, then you will check if the character is there in *characters.txt*, if the character is there in the file then count the number of dialogues of the character given by user.

- First, you should ask for actor name as an input from the user. Hint: check out *fgets* function.
- Check for this input character in the *characters.txt*.
- Convert this input string to uppercase.
- Add '<' and '>' characters as first and last character of this uppercase input string.
- Search the entire *shakespeare.txt* to look for this exact string. Count the number of occurrence and report the count by using *printf* function.

You can find some examples of the output below and command line parameter passing below. This is just to make sure that your algorithm works fine and it should work fine for other characters as well. Note that students are expected to follow the same approach in getting the inputs.

```
./main dialogueCount Antonio  
Antonio has 167 dialogues.
```

```
./main dialogueCount Julia
Julia has 107 dialogues.
```

```
./main dialogueCount Siri
There is no such character as Siri in shakespeare.txt.
```

```
./main dialogueCount Ariel
Ariel has 46 dialogues.
```

4.2 Counting the Mentions

For this part of the assignment, you will get the file name as an input. The input file will contain some names of characters. You will count the number of time the name of each character is mentioned in the dialogues by the other characters or by themselves. This means you only have to consider the dialogues and not the stage direction. Here are some important notes that will help you do the task:

- The number of characters in input file may vary. Therefore, your program should be dynamic that deals with any number of characters in the file.
- Dialogues do not start with '<' character.
- You do not need to convert the character names to uppercase because in the dialogues the names are written as proper noun convention (Only first letter capital).

Some file (*example.txt*) that has only 3 names in it is used as an input. In Table 1, you can find the number of mentions of character names.

Name of the Play Character	Number of their Mentions
Celia	3
Mopsa	2
Panthino	3

Table 1: Example of Number of Mentions of some Characters.

An example of your output can be as following:

```
./main mentionCount example.txt
Celia is mentioned 3 times
Mopsa is mentioned 2 times
Panthino is mention 3 times
```

Another example is here is the file here *julius.txt* has no characters in it:

```
./main mentionCount julius.txt
```

5 Evaluation

Your score will be computed out of a maximum of 50 points based on the following distribution:

35 Correctness points. (15 for Task 1 and 20 for Task 2)

8 Performance points.

2 Effective use of version control points.

5 Style points.

Correctness points. The 2 tasks you must solve have different difficulty levels. Their points are 15 for Task 1 and 20 for Task 2, giving the sum 35. We will evaluate your first Task by giving different input of character names and checking for the output. For task two we will look at the entire list that is printed. You will get full credit for the tasks if it passes all of the tests performed and there might be some partial credit otherwise depending on the output and effort.

Performance points. Our main concern at this point in the course is that you can get the right answer. However, we want to instill in you a sense of keeping things as short and simple as you can. Furthermore, some of the tasks can be achieved by brute force, but we want you to be more clever.

Effective use of version control points. You are required to push your changes to the repository frequently. If you only push the final version, even if it is implemented 100% correctly, you will lose a fraction of the grade because you are expected to learn to use Version Control Systems effectively. You do not have to push every small piece of change to Github but every meaningful change should be pushed. For example, each of the functions coded and tested can be one commit. **For each function, there should be at least one commit (with proper commit message) that includes just modifications on that function.**

Style points. Finally, we've reserved 5 points for a subjective evaluation of the style of your solutions and your commenting. Your solutions should be as clean and straightforward as possible. Your comments should be informative, but they need not be extensive. You can visit the link [here](#) for more information.

Important Note: We use automated plagiarism detection to compare your assignment submission with others and also the code repositories on GitHub and similar sites. Moreover, we plan to ask randomly selected 10% of students to explain their code verbally after the assignments are graded. And one may lose full credit if he or she fails from this oral part.

6 Hand-in Instructions

As with Assignment 0, we use GitHub for the submissions as follows. Note that we want you to get used to using a version management system (Git) in terms of writing good commit messages and frequently committing your work so that you can get most out of Git.

I Commit all the changes you make: `$ git commit -a -m "commit message"`

Note: please use meaningful commit messages because

II Push your work to GitHub servers: `$ git push origin main`

7 How to use linuxpool.ku.edu.tr linux servers

I Connect to KU VPN (If you are connected to the KU network, you can skip this step.)

See for details: <https://confluence.ku.edu.tr/kuhelp/ithelp/it-services/network-and-wireless/vpn-access>

II Connect to linuxpool.ku.edu.tr server using SSH (Replace USER with your Koç University username):

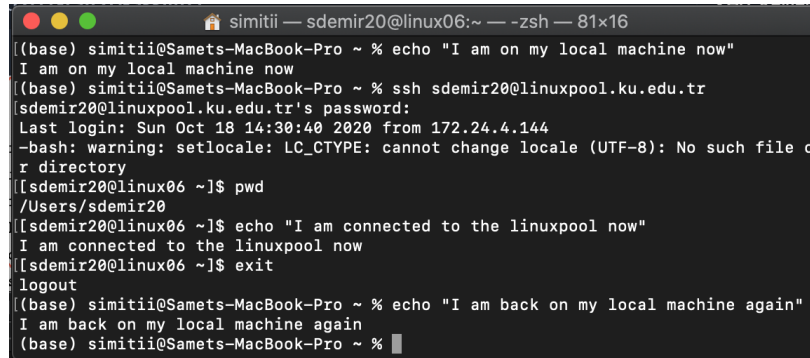
```
$ ssh USER@linuxpool.ku.edu.tr
```

(It will ask your password, type your Koç University password.)

III When you are finished with your work, you can disconnect by typing: `$ exit`

Your connection to the server may drop sometimes. In that case, you need to reconnect.

We advice you to watch the following video about the usage of SSH, which is used to connect remote servers, and SCP, which is used to transfer files between remote servers and your local machine: <https://www.youtube.com/watch?v=rm6pewTcSro>



```
simitii — sdemir20@linux06:~ — zsh — 81x16
((base) simitii@Samets-MacBook-Pro ~ % echo "I am on my local machine now"
I am on my local machine now
((base) simitii@Samets-MacBook-Pro ~ % ssh sdemir20@linuxpool.ku.edu.tr
sdemir20@linuxpool.ku.edu.tr's password:
Last login: Sun Oct 18 14:30:40 2020 from 172.24.4.144
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or
directory
[sdemir20@linux06 ~]$ pwd
/Users/sdemir20
[sdemir20@linux06 ~]$ echo "I am connected to the linuxpool now"
I am connected to the linuxpool now
[sdemir20@linux06 ~]$ exit
logout
((base) simitii@Samets-MacBook-Pro ~ % echo "I am back on my local machine again"
I am back on my local machine again
((base) simitii@Samets-MacBook-Pro ~ %
```

Figure 1: How to connect and disconnect using SSH

8 Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. See [Koç University - Student Code of Conduct](#).

9 Late Submission Policy

You may use up to 7 grace days (in total) over the course of the semester for the assignments. That is you can submit your solutions without any penalty if you have free grace days left. Any additional unapproved late submission will be punished (1 day late: 20% off, 2 days late: 40% off) and **no submission after 2 days will be accepted.**

Acknowledgement

The dataset used in this assignment is from <https://www.kaggle.com/asimzahid/shakespeare-plays>.