

COMP 201 - Fall 2022

Assignment 2 - Strings in C

Assigned: 27 October 2022 23:59, Due: 9 October 2022 23:59

Burak Can Biner (bbiner21@ku.edu.tr) is the lead person for this assignment.

1. Introduction:

This assignment aims to advance your skills in manipulating strings and learn to use string functions efficiently on C. Every device connected to the internet or a local network has a unique identifier address called Internet Protocol (IP). The Domain Name System (DNS) matches these IP addresses with specific domain names. DNS servers store IP addresses and their corresponding domain names. In other words, when we try to fetch the data of a website, we enter a hostname URL, then DNS servers translate this URL to the exact IP address we need. In this homework, you will work on IP addresses stored in a DNS server. These IP addresses are stored in IPv4 format. You are provided with an IP address list and a query list. You will check if IP addresses are valid, convert them to hexadecimal, search for a query number, keep the statistics of the data according to queries and finally modify IP addresses under given conditions. You will not work on the corresponding domain names in the server. You will only work on IP addresses.

2. Logistics

This is an individual project. All handins are electronic. Clarifications and corrections will be announced on Blackboard.

3. Handout Instructions:

Accept the GitHub Classroom assignment using the link: <https://classroom.github.com/a/2SnUb9jo>. 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 8 for details.) :

```
git clone https://github.com/COMP201-Fall122/assignment-2-USER.git
```

(Replace USER with your GitHub username that you use to accept the assignment) For this assignment, you need only modify the main.c file, and at the end, submit main.c file.

- **main.c:** The main file of the assignment that should be modified and submitted to be graded.
- **sample.txt:** Contains IP addresses in the DNS server.
- **queries.txt:** List of queries to search and keep statistics.

4. Tasks:

This assignment contains five main parts. Make sure that you have sample.txt and queries.txt in the same folder with main.c . These files are read inside the given code.

Please create a makefile with the following commands.

```
1 gcc main.c -o main
```

```
1 ./main
```

4.1 isValidIP:

Our DNS server contains some invalid IP addresses due to some previous errors. First, you will find the valid IP addresses. The code already provided for you reads Ip addresses and stores them in "input_lines" for you. You need to check if these addresses are valid IP addresses or not with the isValidIP function. This function returns a bool. Returns "true" if IP is valid and "false" otherwise. You can use the strtok string function of C for this part. For an address to be valid following conditions must hold.

- Every Ip address consists of four numbers separated by three dots. In the provided file, all addresses will have this format so that you do not need to check this condition further.
- Every number in these four sections must be in the range of 0 to 255. For instance, 192.255.0.67 is a valid IP address. However, 192.256.0.67 is not a valid IP address.
- IP addresses must not contain any characters other than '.' and digits.
- IP addresses must not contain unnecessary zeros before them. For instance, 192.01.168.2 is not a valid IP address since 01 can be represented as 1.

4.2 toHex:

This function converts a given IP address that is in base 10 to hexadecimal. It needs to convert numbers in each of the four sections separately. These hexadecimal numbers need to be separated by '.' signs again. For example, "192.117.63.253" is converted as "c0.75.3f.fd". Once again, there should not be unnecessary zero characters. For example, "10.117.63.253" is converted as "a.75.3f.fd". Here the first section must not contain "0a". We will give only valid IP addresses to this function; therefore, you do not need to check valid IP conditions here.

4.3 checkQuery:

This function checks if the given query string is in the given IP address or not and returns a bool accordingly. This function operates on Ip addresses that are converted to the hexadecimal base. This function tries to match queries only from the start of the IP address. For example if our IP address is "c0.a8.1.1" and query string is "c0.a8" it returns true. However if query is "c0.a8" and our IP address is "a0.c0.a8.1" it returns false. First, three sections of the IP address indicate network ID, and the last section indicates host ID. This implementation allows us to check if the current IP belongs to a given network.

4.4 Keeping the stats of the network:

You are provided with a queries.txt file containing some query strings that we will look for in our valid IP addresses. These query strings are given in hexadecimal base. They can vary in length. We will be searching for different string lengths and section numbers. The code that is provided already reads this file for you. These query strings are stored in "input_lines_query". You need to keep the statistic of each query string. You need to keep the number of occurrences for each query. You can use your checkQuery function implementation here. Store the occurrence counts in the "query_stats" array with the initial order of queries.

4.5 modifyIP:

This function will allow us to modify a given IP address. For the sake of simplicity, these modifications have certain constraints. This function takes `search_num`, `IP`, and `new_num` inputs. For the given `IP`, we search for the `search_num` string, and when we find the first occurrence of this string in `IP`, we replace it with `new_num`. We do this operation only if the length of the `search_num` is equal to `new_num`. Otherwise, we return the same input. For instance for the `IP "c0.75.3f.fd"`, `search_num "75.3f"` and `new_num "60.35"` our new `IP` is `"c0.60.35.fd"`. However, if our `new_num` is `"60.5"` we do not perform any modifications.

5. Submission

As with Assignment 1, 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.

1. Commit all the changes you make:

```
1 git commit -a -m "commit message"
```

Note: please use meaningful commit messages because

2. Push your work to GitHub servers:

```
1 git push origin main
```

You are expected to submit a `main.c` file. Please accurately read and follow the instruction mentioned in the `main.c` file as a comment. Enter your codes only between the specific spaces prepared at the `main.c` file. And please do not modify commented lines in the source code file.

6. Evaluation:

Your score will be computed out of a maximum of 100 points based on the following distribution:

- `isValidIP`: 20%
- `toHex`: 20%
- `checkQuery`: 20%
- Keeping the stats of the network: 20%
- `modifyIP`: 20%

Effective use of version control: 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 tasks coded and tested can be one commit. For each task, there should be at least one commit (with proper commit message) that includes just modifications on that function.

Important Note: We use automated plagiarism detection to compare your assignment submission with others and also the code repositories on Github and similar sites.

7. Oral Assessment

Important Note: We plan to ask randomly selected 10% of students to explain their approach verbally after the assignments are graded. And one may lose full credit if he or she fails from this oral part.

8. 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):

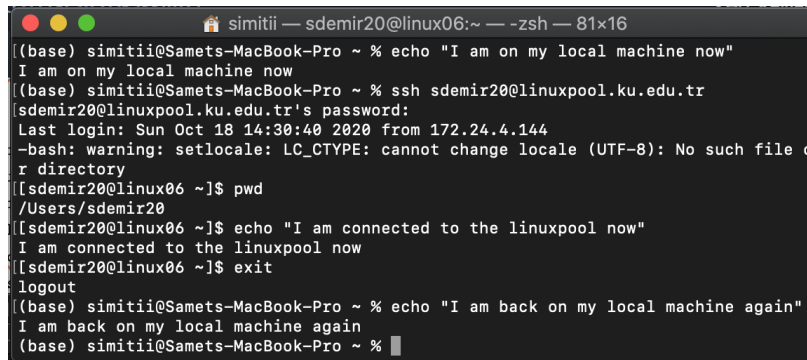
```
1 ssh USER@linuxpool.ku.edu.tr
```

(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 advise 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@Samets-MacBook-Pro ~ % echo "I am on my local machine now"
I am on my local machine now
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
simitii@Samets-MacBook-Pro ~ % echo "I am back on my local machine again"
I am back on my local machine again
simitii@Samets-MacBook-Pro ~ %
```

Figure 1: How to connect and disconnect using SSH

9. Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss the given assignments with your classmates, 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 web material as everything on the web has been written by someone else. See Koç University - Student Code of Conduct.

10. 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.**