PROGRAMMING ASSIGNMENT 1
Dynamic Memory Management in C

Subject: - Malloc, calloc, realloc, free, file IO and string
TA: Cemil Zalluhoğlu

Due Date: 27.03.2019 - 23:59:59

1 Introduction

In this assignment, you are expected to implement a simple file and word processing program using pointers and dynamic memory allocation functions. Please read this document in its entirety. It contains essential information about the assignment and how you will be assessed and provides hints for you to succeed at this assignment.

2 File and Word Processing Program

The file and word processing program that you will implement consist of specific commands. With this program, you can create and delete files in the virtual environment, append text to the data, remove the desired parts from the text, replace a word with another word, and print the data and files on the screen in various formats. These commands will explain in detail in the following sections of the document.

2.1 Commands

2.1.1 Create a file

- "-n" parameter defines the filename which you want to create(your filename must **include the extension**)
- "-t" parameter defines the text of the file. if no text is entered, "**Empty File**" will be written to the file by default.
- You must use dynamic memory allocation functions when you **create a file** and **add a new file to your file repository**.
- the order of parameters is **not important**.

To create a file, first you must use "create" command. It takes two parameters.

Usage of the create command:

Sample 1: `create -n temp.txt -t C programming is awesome`

Sample 2: `create -t I love C programming.  -n temp.doc`

Sample 3: `create -n temp.txt -t`
`print -n temp.txt -t`
`Output: Text: Empty File`

Sample 4: `create -t -n temp.doc`
`print -n temp.doc -t`
`Output: Text: Empty File`

2.1.2 Delete a file

To delete a file, first you must use "delete" command. It takes one parameter.

- "-n" parameter defines the filename which you want to delete.
- You must use dynamic memory allocation functions when you **delete a file**.

Usage of the delete command:

Sample : `delete -n temp.doc`

2.1.3 Remove text from the file

This command uses for removing the part of text from the file. It takes three parameters.

- "-n" parameter defines the file to be processed on.
- "-s" parameter represents the starting indices of the part to be removed from the text.
- "-l" parameter represents the length of the part to be removed from the text.

Usage of the remove command:

The content of the "temp.txt" before run the command:

`temp.txt : C programming is awesome`

Sample command : `remove -n temp.txt -s 5 -l 3`

The content of the "temp.txt" after run the command:

`temp.txt : C promming is awesome`

2.1.4 Append text to the file

This command is used to append the given text to the specified file. It takes two parameters.

- "-n" parameter defines the file to be processed on.
- "-t" parameter defines the text which you want to append the given file.
- the order of parameters is **not important**.

Usage of the append command:

Sample 1 :

The content of the "temp.doc" before run the command:

```
temp.doc :  I love C programming
```

Sample command : `append -n temp.doc -t what about you?`

The content of the "temp.doc" after run the command:

```
temp.doc :  I love C programmingwhat about you?
```

Sample 2 :

The content of the "temp.doc" before run the command:

```
temp.doc :  I love C programming
```

Sample command : `append -t what about you?  -n temp.doc`

The content of the "temp.doc" after run the command:

```
temp.doc :  I love C programmingwhat about you?
```

2.1.5 Replace a word with another word

This command is used to replace one word with another word in the given file. It takes three parameters.

- "-n" parameter defines the file to be processed on.
- "ow" parameter defines the word which you want to replace with another word.
- "nw" parameter defines the word which you want to put in the file.
- You must use dynamic memory allocation functions when you **replace a word**.
- the order of parameters is **not important**.

Usage of the replace command:

The content of the "temp.txt" before run the command:

```
temp.txt :  C programming is awesome.  I love C program.
```

Sample command : `replace -n temp.txt -ow program -nw lesson`

The content of the "temp.txt" after run the command:

```
temp.txt :  C lessonming is awesome.  I love C lesson.
```

2.1.6 Print file(s) and data(s)

The "print" command is used in six different ways.

1. "-a" : print all files. You must display **each file name in a row**.
   Sample command : `print -a`
   Output:
   ```
   Filename 1:  temp.txt
   Filename 2:  temp.doc
   ```
2. "-e" : print the content of the files which have the given extension.
   Sample command : `print -e doc`
   Output:
   ```
   Filename 2:  temp
   Text:  I love C programming
   ```
3. "-c" print the content of the all files.
   Sample command : `print -c`
   Output:
   ```
   Num:  1
   Name:  temp
   Text:  C programming is awesome.
   Num:  2
   Name:  temp
   Text:  I love C programming
   ```
4. "-n" and "-t" : print the text of the given file.
   Sample command : `print -n temp.txt -t`
   Output:
   ```
   Text:  C programming is awesome.
   ```
5. "cw" : print the number of occurrence of the given the word in the given file.
   Sample command : `print -n temp.txt -cw program`
   Output:
   ```
   Text:  C programming is awesome.  I love C program.
   Number Of Occurrence of "program" :  2
   ```
6. "cs" : print the number of sentences in the given file. You only need to pay attention to fullstop(dot), question mark and exclamation mark.
   Sample command : `print -n temp.txt -cs`
   Output:
   ```
   Text:  C programming is awesome.  I love C program!  what about you?
   Number Of Sentences :  3
   ```

3 Report

Reporting is mandatory. Your report should cover :

- problem definition (limited to max. 3 sentences)
- solution approach (limited to max. 3 sentences)

- an explanation about where you use dynamic memory allocation functions (limited to a single sentence for each function.)

4 Grading Policy

Task	Point
Report	10
`create_file`	10
`delete_file`	10
`remove_text`	15
`append_text`	15
`replace_word`	15
`print_files`	15
`simplicity of your code`	10

5 Submission Notes

- The output of your program will be graded **automatically**. Therefore, any difference of the output (even a smallest difference) from the sample output will cause an error and you will get 0 from execution. **Keep in mind that a program that does not work 100% right is a program that works wrong.**
- **Test your program on "dev.cs.hacettepe.edu.tr" before submission**. Your submission will be compiled and executed in this machine and this machine only.
- Do not submit any file via e-mail. You will use online submission system to submit your experiments. Other type of submissions especially by e-mail WILL NOT BE ACCEPTED.
  `https://submit.cs.hacettepe.edu.tr/`
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. All the duplicate or Internet works (even if a citation is provided) are both going to be considered as cheating.
- Don't use your instructors as a google. Use google for general purposes then ask us for specific ones
- You can ask your questions through Piazza and you are supposed to be aware of everything discussed there.
- The submission format is given below:: This file hierarchy must be zipped before submission (Not .rar, only .zip files are supported by the system)

→ <student id.zip>
  → report.pdf
  → studentID_exp1.c