

CIS 2500: Assignment 1 Specification

Due Sunday January 24, 11:55pm

Problem

Write a program that accepts two files as input: A template file, and a data file. The template file will contain variables, and their values will be defined inside the data file. Your program should output the template file, with the variables replaced with their associated values, to a third output file.

Musts

The following requirements must be met to achieve a grade >0%:

- Your program must compile on the Raspberry Pi (SoCS OS), with no errors or warnings using ***-std=c99 -Wall***.
- Your program must not make use of any global or static variables (with the exception of `storeval.c`)
- Your program must make use of the functions declared in `storeval.h` to store and retrieve values parsed from the data file
- You may not make any changes to the provided `storeval.c`, `storeval.h`, or `utlist.h` files
- Your program must make use of at least 4 different user defined functions
- Your program is submitted before the due date, and your submission is complete (see section: *Submission*)

Description

When started, your program should accept three command line arguments (using `argc/argv`). The first is a relative file path to the template file. The second is a relative file path to the data file. The third is a relative file path to the output file.

Your program should start by reading variable names and values from the data file, and store them using the functions from `storeval`.

After reaching the end of the data file, it should begin reading the template file line by line (hint: `fgets`) and replace any variable name with the value. It should make use of functions from `storeval` to retrieve the previously stored values based on the variable name. After modifying each line, your program should print the line to the output file and repeat the process until the entire template file is finished.

Before termination, your program should print the following to `stdout`:

- Whether execution was successful or not
- The number of variables found in the data file and template file

- The number of successful variable replacements
- The number of failed replacements (likely a result of the variable name not existing in the data file)

After execution, the output file should exist and contain the template with the variables replaced by their values.

Template File

The following is an example template file:

```
Dear {recipName},  
I understand that we haven't spoken since I ran away with {friend}...  
  
But I thought you should know that we have a son.  
His name is {sonName}. He has your {feature}.  
  
Sincerely,  
{sendName}
```

Data File

The following is an example data file:

```
recipName=Fork  
friend=Cup  
sonName=Spork  
feature=hair  
sendName=Spoon
```

Output File

Given the above examples, the following would be outputted:

```
Dear Fork,  
I understand that we haven't spoken since I ran away with Cup...  
  
But I thought you should know that we have a son.  
His name is Spork. He has your hair.  
  
Sincerely,  
Spoon
```

Additional Notes

- Variables in the template file will always be in the format:
 - o {variableName}
- { } will not appear in the template file, except to define variables for replacement.

- A variable used in the template file may not appear in the data file. In the event this occurs, use “MISSING_DATA” as the replacement, and count this as a failed replacement.
- A variable in the data file doesn’t necessarily need to be used in the template file.
- A line in the template file can have 0, 1, or multiple variables.
- A line in the template file can be blank.
- The data file will always be in the format:
 - o variableName=variable value<newline>
- The variable name won’t contain any spaces, but the variable value might. Neither will contain =, {, or }.
- A line in either file will always be less than 255 characters.
- A variable name and value will always be less than 25 characters.

Expectations

- A readme, which adheres to the course guidelines, is included in the submission.
- Your code adheres to the course coding style guidelines.
- Your submission is organized with src, include, assets (if necessary) and doc folders, as per course guidelines.
- Your program follows all specifications described in this document.
- Your program handles error cases appropriately.

Submission

The following materials constitute a complete submission:

- 1) **Source Code:** All source code you have written, storeval.c, storeval.h, uthash.h, and any other files necessary to compile and run your program.
 - 1a) storeval.c, storeval.h, and uthash.h will be provided to you through the course website and may not be modified for any reason
- 2) **Documentation:** A file called *README*, describing how to compile, run, and usage your program. The file should also describe any known limitations, and list all references you have used.

There should be no compiled binary in your submission.

Assignment submissions for each student will be pulled from their A1 repository available at <https://<username>@git.socs.uoguelph.ca/cis2500/<username>/A1>. Submissions will not be accepted through email. It’s recommended that you practice pushing to your repo well in advanced of the deadline to ensure you understand the process, and are able to sort out any technical issues before the assignment is due.

It's a good idea to clone your repository to a separate directory after submission, and make certain that it contains what you intended, compiles, and runs.

Academic Integrity

By submitting your assignment, you verify that you have exclusive control over your submission via your password.

You certify that:

- 1) You have read and understood the University policy on academic integrity;
- 2) You have completed the Computing with Integrity Tutorial on Moodle; and
- 3) You have achieved at least 80% in the Computing with Integrity Self Test.

You assert that your submission is your own.

You have appropriately acknowledged any and all material (data, images, ideas or words) that you have used, whether directly quoted or paraphrased. Furthermore, you certify that you prepared the submission specifically for this course.

If you have any concerns with these assertions, you must contact the instructor 72 hours prior to the assignment deadline.