Open-source Software Lab

Lab 3. Voogle: Bible Verse Search Program

30 Mar 2023

Shin Hong

Overview

Voogle

- Each team is asked to write a C program `voogle.c` that receives one or multiple searching terms, and returns each bible verse matched with all given terms
    - like Google for searching the NIV bible verse
    - The full text of NIV is given as text file
    - The resource is found in the `voogle` branch of `github.com/hongshin/learningC`
- You must use a given list of string functions properly in constructing your bible verse search program
    - use each function at least once
- Each team must submit the resulting source code and a video demo to HDLMS by 4 PM, 6 April

Voogle Workflow

1. receives a text line given in standard input
2. identifies up to 8 searching terms from the given text line
3. checks if the given searching terms are valid; terminate the program if they are invalid (i.e., print nothing if the input is invalid).
4. reads each line of `NIV.txt`, and prints the line to standard output, if it satisfies all conditions posed by all given searching terms

Example

```
$ ./voogle
abraham* issac -esau “ and “^D
Gen 25:11 After Abraham’s death, God blessed his son Issac, who then lived
neare Beer Lahai Roi.
1Ch 1:28 The sons of Abraham: Isaac and Ishmael. Descendants of Hagar
```

⋮

Searching Terms (1/2)

- A searching term specifies a condition of a bible verse text.
- If an input text contains more than one searching terms, these are separated by one or more whitespaces.

Searching Terms (2/2)

- A searching term is in one of the following forms:
  - *token* (i.e., an alphanumeric string without whitespace): a verse satisfies this condition iff a token in the verse matches with the given token in case-insensitive way
  - *token*`*` : a verse satisfies this condition iff there is a token in the verse whose prefix matches with *token* in case-insensitive way
  - `-`*token* : a verse satisfies this condition iff the verse has no token that matches with *token* in case-insensitive way
  - "*string*" (*string* can be any string that may contains whitespace): a verse satisfies this condition iff a substring is identical to *string*.
  - `book:`*code* : a verse satisfies this condition iff it is in the book of *code* (see appendix)
  - `chapter:`*num* : a verse satisfies this condition iff its chapter number is *num*. *num* must be a positive integer.

String Library Functions

- You are asked to study and use the following string related functions
  - `strchr`
  - `strcmp`
  - `strstr`
  - `strtok`
  - `tolower`
  - `atoi`
- Your program may use other string library functions as well

Video Demo

- In the video, you are asked to demonstrate that your bible search program is successfully built, and then works correctly in different use cases
- Also, you need to show the source code and explain how each string library function is used
- The demo video must take no more than 7 minutes
- Upload your demo video to YouTube, and write down the URL on the submission message

Evaluation

- Your result will be evaluated according to the following criteria:
  - all functionalities are correctly implemented
  - the source code is clean and comprehensible
  - each appointed string library function is properly used at least once in the source code
  - the demo video clearly describe that the program works correctly for various use cases, and the string libraries are properly used
- You will get extra points if you add new interesting features
- Good demo videos will be recognized

Other Instructions

- Open chat for Q&A - https://open.kakao.com/o/gII0Vjbf
- One submission for each team. No need to make duplicated submissions.
- No late submission will be accepted.
- Your demo video may be shared in the class, especially if it is to be recognized.
- You must use English in recording a video demo