

CSCI 340 – Operating Systems - Homework 4

Collaboration Policy: Individual Assignment

Total Points: 100

Let's make Merica great again objectives

Using the C template code and the pthread and curl libraries (attached to the Dropbox), please modify the code to do the following:

- Create and run a thread for any combination of web URLs provided in the `url.txt` file.
- All threads must be joined after completion.
- Create a file named **make_merica_great_gain.txt**. For each website (in the `url.txt`) every the value of every **href** attribute that contains the word “trump” is printed to this file (see comments in code that outline file format).
- Fully implement the parse function that locates and stores all the **href** attribute values that contain the word “trump” using the webpage field sorted in the `web_t` struct.
- This assignment does not use any locking mechanism. Even though this assignment defines an array that is a global variable, the threads are designed to be mutually exclusive (i.e. no two threads should operate on the same item in the global array).

Provided files

The three files are provided to you;

hw4.c: source code that includes a stubbed out version of the main function, and defines libraries, variables, and constants used in this assignment. Please note: you may not remove, modify existing code. If you wish to use a function or constant in one of the standard C language libraries, and the header file is not included, you may add it.

web.h: Header file that defines the command struct used in this assignment, constants, and the function prototypes to be completed by you. Please note: You may not add new function definitions to this header file.

web.c: The file containing the implementation of the functions listed in `web.h`. Having a different file for the implementation separates interface (the include file) from the implementation (the `.c` file).

The `hw4.c`, `web.h` and `web.c` files contain many comments that provide basic definitions and step-by-step instructions. Please read the comments carefully and follow their instructions.

Todo

In the `web.h` file you code a working solution within the corresponding `web.c` file for each function listed below.

- `void parse(web_t* web_struct)`

In the `hw4.c` file you code a working solution for each function listed below.

- `int main(int argc, char** argv)`
- `int write_url_file(char* file_path, web_t* web_array, int num_urls)`

Numerous comments are provided in the header or `.c` file to guide you in this assignment. Please read them carefully, the comments either provide basic step-by-step instructions or basic calculations. **Give yourself plenty of time.**

Collaboration and Plagiarism

This is an individual assignment, i.e. no collaboration is permitted. Plagiarism will not be tolerated. Submitted solutions that are very similar (determined by the instructor) will be given a grade of zero. Please do your own work, and everything will be OK.

Submission

Create a compressed tarball, i.e. `tar.gz`, that only contains the completed `hw4.c` and `web.c` files. The name of the compressed tarball must be your last name. For example, `ritchie.tar.gz` would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions). Submit the compressed tarball (via OAKS) to the Dropbox setup for this assignment. You may resubmit the compressed tarball as many times as you like, Dropbox will only keep the newest submission.

To be fair to everyone, late assignments will not be accepted. Exceptions will only be made for extenuating circumstances, i.e. death in the family, health related problems, etc. You will be given a week to complete this assignment. Poor time management is not excuse. Please do not email assignment after the due date, it will not be accepted. Please feel free to setup an appointment to discuss the assigned coding problem. I will be more than happy to listen to your approach and make suggestions. However, I cannot tell you how to code the solution. Additionally, code debugging is your job. You may use the debugger (gdb) or print statements to help understand why your solution is not working correctly, your choice.

Grading Rubric

Code compiles	15 points
Code runs without errors	5 points
URL test cases (8 cases 10 points each)	80 points
	100 points

In particular, the assignment will be graded as follow, if the submitted solution

- Does not compile: 0 of 100 points
- Compiles but does not run: 15 of 100 points
- Compiles and runs with no errors: 20 of 100 points
- Passes all 6 test cases developed by instructor: 100 of 100 points.