TCSS 422 — Computer Operating Systems
Spring 2015 — Homework Assignment 2

Due Date: Tuesday, Apr. 21

Guidelines

Homework should be electronically submitted to the instructor by the end of the day on the due date. A submission link is provided on the course Canvas page for this assignment.

Assignment Description

This assignment is intended to introduce you to the Pthreads library and give you further experience with C. In this assignment you will implement a small multithreaded program that identifies palindromic words. Starter code for this assignment is available on the course Canvas page for this assignment.

Implementation Specifications

There are four provided files for this assignment:

- `main.c` – The implementation file containing the program entry point, `main`, and some basic driver code.
- `palindrome.h`, `palindrome.c` – A header file and implementation file for the function you're to implement.
- `words.txt` – A data file containing a list of words that will be searched for palindromes.

Inside `palindrome.c` you'll find the skeleton for the function you're to implement, `multithreaded_findPalindromes`. You can add helper methods, new types, etc., but do not modify the interface to `multithreaded_findPalindromes`. The provided `main.c` file already loads the list of words from `words.txt`, invokes `multithreaded_findPalindromes`, and displays the results to help you test your implementation.

```
char ** multithreaded_findPalindromes(char ** words,
                                      int numOfWords,
                                      int * palindromesFound,
                                      int numThreads)
```

This function receives in parameter `words` an array of strings, where each string is a word (loaded from `words.txt`). The number of words in the array is specified in parameter `numOfWords`. This function should return an array of strings containing all of the words that are palindromes. The number of words in the returned array should be stored in the integer pointed to by parameter `palindromesFound` (the palindromes will therefore occupy indices 0 through `palindromesFound` – 1). The work of searching for palindromes in the array must be split equally across multiple threads, specifically the number of threads specified in parameter `numThreads`. Each thread will therefore examine approximately `numOfWords` / `numThreads` words.

Additional important points:

- The array returned by `multithreaded_findPalindromes` should be dynamically allocated.
- You may assume that the number of words in the array `words` is larger than the desired number of threads, but it may not be evenly divisible by the number of threads.
- The provided main function invokes `multithreaded_findPalindromes` with the number of threads equal to four, but your code should work with any number of specified threads.
- The returned array should use the same strings present in array `words`, i.e., you don't have to copy the strings.
- The words in the returned array may be in any order, they *don't* need to be in alphabetic order like the array `words`.

Deliverables

The following items should be submitted to the instructor. Failure to correctly submit assignment files will result in a 10% grade penalty.

1) The completed `palindrome.c` source file.
2) Any additional new or modified source files needed by `palindrome.c`.

Do not submit unmodified provided files, e.g., `words.txt`. Do not include any extraneous files, such as Eclipse IDE files, object files, or subversion files.