Bag of Tasks

Homework #4, CSCI 322, Winter 2016

Due: Monday, February 29, at midnight.

Palindromic words: Recall that a *palindrome* is a word that reads the same when reversed, such as "radar" and "noon." A *palindromic word* is a word that is also a word when reversed, such as "draw" and "live."

You will write two programs in **C** to find palindromic words, one sequential and one concurrent using `pthreads` to implement a bag of tasks.

Make each program a single, standalone file, so I can read them easily.

Sequential program: I have provided a copy of the `/usr/share/dict/words` file from my computer on the github repository with this file (so we can all start with exactly the same data). It is a file with one word per line, 99171 lines. The maximum length of a string in the file is 23. Your task is to find all the palindromic words in this file.

Name your program `palseq.c`. It should compile with just `gcc palseq.c` (no fancy libraries).

Input phase. Your sequential program starts by reading all the words in this file into an array of strings. Also keep an array of booleans of the same length as the word list, initialized to `false`.

Compute phase. Then take each word, one at a time, compute its reverse, and do a linear search through the entire array to see if the reverse of this word is in the file. If it is, mark it as `true` in the array, and increment a count of the total number of palindromic words found.

Output phase. After this loop is done, write out the palindromic words into a file named `resultseq.txt`, one word per line, and print the total number of palindromic words to standard output.

Concurrent program: After you have a working sequential program, modify it to use the bag-of-tasks paradigm.

Name this program `palconc.c`. It should compile with `gcc palconc.c -lpthread` (only the pthread library).

- Your parallel program should use `W` worker processes, where `W` is a command-line argument.
- Use the workers just for the compute phase. Reading the words and writing the palindromic words should be done sequentially, as before.
- There are 26 tasks, one for each lower-case letter of the alphabet. Each worker should count the number of palindromic words it finds that begin with that letter. An efficient representation for the bag of tasks is an array of 27 integers, each representing the starting position of that letter in the word array, and the last integer being one greater than the last "z" position. So task i would search the array of words from `words[task[i]]` to `words[task[i+1]]`. This array of tasks should be set up on the input phase.
- You may use `join` to synchronize the output phase with all the workers.
- The palindromic words should be written to `resultconc.txt` so that this file can be compared to `resultseq.txt`. They should be identical.