Assignment 6

Adv C Programming Comprehensive Assignment

Title: Mastering Advanced C Programing Concepts - Basic pthread functions

Objective: The objective of this assignment is to reinforce your understanding of advanced C programming concepts.

Assignment Task:

Implementing a word detection program using pthread functions.

Basic thread creation and management was introduced in the lecture. Based on the charcount example in the lecture slides, implement a word detection program using pthread functions in C. In this assignment, you will exercise how to create and execute a thread using pthread library functions. Also figure out difference between pthread_join and pthread_detach in use.

Download assignment6.c and implement TODO lines by your own code. You should capture a screenshot of the execution result from the C code.

```
/*
* File: assignment6.c
```

```
* Owner: Yoonseok Yang
* Date: 4.18.2024
* Description: Implement a word detection program using pthread functions
in C
*/

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
void* word_detect_join(void *vargp);
void* word_detect_detach(void *vargp);
clock_t join_start_time;
clock_t join_end_time;
clock_t detach_start_time;
clock_t detach_end_time;

int main() {
    pthread_t tid = pthread_self();
    printf("main: %u\n", (unsigned int)tid);

    while(1) {
        char str[100];
        pthread_t tid;
        scanf("%99s", str);
        if(strcmp(str, "quit") == 0)
            break;
        // TODO: Create a peer thread for word_detect_join, detecting a word in
the str buffer using pthread_join
        // ....
        // TODO: Create a peer thread for word_detect_detach, detecting a word
in the str buffer using pthread_detach
        // ...
    }
    // TODO: Free the str buffer if you dynamically allocate the string memory
    return 0;
}

void* word_detect_join(void *vargp) {
    // TODO: Implement a peer thread using pthread_join to detect a word in
the argument passed
    // TODO: Print the detected word and frequency if there are multiple
detections
    // TODO: Calculate the elapsed execution time in the thread execution
using join_start_time and join_end_time
```

```
    // i.e., join_start_time = clock(); <... your thread function>; join_end_time = clock(); join_execution_time = join_end_time - join_start_time;
    return NULL;
}

void* word_detect_detach(void *vargp) {
    // TODO: Implement a peer thread using pthread_detach to detect a word in the argument passed
    // TODO: Print the detected word and frequency if there are multiple detections
    // TODO: Calculate the elapsed execution time in the thread execution using detach_start_time and detach_end_time
    return NULL;
}
```

Instructions for the implementation

1. Error handling:
   - Properly handle any error cases and print error messages in the error cases.
2. Comment:
   - Add comments on any of your lines. Describe why you added them in details.
3. Compile:
   - Make your own Makefile to compile.

Documentation

- Document your code thoroughly with comments explaining each section.
- Prepare a report PDF document summarizing everything such as each step or functions that you implemented, challenges faced, and what you've learned during this assignment.

Submission Guidelines:

- Submit the consolidated source code (a zip file) for the entire assignment.
- Include a README file with instructions for compiling and running your program.
- Ensure your code is well-documented with comments.

Assessment Criteria:

Your assignment will be assessed based on the following criteria:

- Submit your own work (80% in points) even though it doesn't execute as expected.
- Proper use of C data types, pointers, arrays, structures, and dynamic memory allocation.
- Efficiency and readability of the code.
- Effective use of memory, functions, and function pointers.
- Proper error handling and validation.
- Utilization of macros for code optimization.
- Utilization of C standard library functions.
- Quality and completeness of documentation.

Important Note: **Plagiarism will not be tolerated, and students are expected to produce their work independently.** Please perform this assignment by yourself and don't copy any solutions from any Generative AI applications like ChatGPT or Bard, your friends or online. If I find any things that imply plagiarism, you will lose the whole points and be reported.