COSC 3360 - Fundamentals of Operating Systems

Programming Assignment 3

Due date: Tuesday, 13 September 2022, 10:59 PM
Requested files: main.cpp (Download)
Type of work: Individual work

Similarity Threshold: 90%

Objective:

This assignment will introduce you to interprocess synchronization mechanisms in UNIX using named POSIX semaphores, pthread mutex semaphores, and pthread condition variables.

Problem:

For this assignment, you will modify your solution for programming assignment 1 to comply with the restrictions explained below.

Given the ASCII alphabet (256 characters), you need to implement a string matching solution to search for multiple patterns based on the following steps:

- Read the input from STDIN (the Moodle server will implement input redirection to send the information from a file to STDIN). The format of the input file is as follows:
  - A string representing the text
  - An integer representing the number of patterns.
  - n lines (where n is the number of patterns) with a string representing a pattern.

  Given the previous format, the following file represents a valid input file:

```
Carlos Alberto Rincon Castro
4
Ca
o
Rincon
w
```

- Create n POSIX threads (where n is the number of patterns to find in the provided text). Each child thread executes the following tasks:
  - Receives the text and a pattern from the main thread.
  - Implements the Rabin–Karp algorithm to determine the positions where the pattern appears in the text (positions' values start from zero).
  - Print the result for the string-search process.

Given the previous input, the expected output is:

```
SEARCH RESULTS:

Pattern "Ca" in the input text at position 0
Pattern "Ca" in the input text at position 22

Pattern "o" in the input text at position 4
Pattern "o" in the input text at position 13
Pattern "o" in the input text at position 19
Pattern "o" in the input text at position 27

Pattern "Rincon" in the input text at position 15

Pattern "w" not found
```

NOTES:

- **You can safely assume that the input files will always be in the proper format.**
- **You cannot use global variables. A 100% penalty will be applied to submissions using global variables.**
- **You must define the critical sections following the guidelines that we discussed in class.**
- **You must use POSIX threads. A penalty of 100% will be applied to submissions using a thread library other than the pthread library.**
- **You can only use named POSIX semaphores, pthreads mutex semaphores, or pthreads condition variables to achieve synchronization. Using pthread_join or sleep to synchronize your threads is not allowed (you must use pthread_join to guarantee that the parent thread waits for all its child threads to end before ending its execution). A penalty of 100% will be applied to submissions using the previous system calls to synchronize the child threads.**
- **You cannot use different memory addresses to pass the information from the parent thread to the child threads.**
- **You must use the output statement format based on the example above.**

Requested files

main.cpp

```
1  // Write your program here
```

VPL

◄ Programming Assignment 2

Jump to...