Document Title: Laboratory Sheet 3

Subject: CPP

Synchronization

1. Given two character strings s1 and s2. Using C and pthread to write a parallel program to find out the number of substrings, in string s1, that are exactly the same as string s2. The strings are ended with ‘\0’. For example, suppose number_substring(s1, s2) implements the function, then number_substring (“abcdab”, “ab”) = 2, number_substring(“aaa”, “a”)= 3, number_substring(“abac”, “bc”) = 0. Suppose the size of s1 and s2 are n1 and n2, respectively, and p threads are used, we assume that n1 mod p = 0, and n2 < n1/p. Strings s1 and s2 are stored in a file named “strings.txt”. String s1 is evenly partitioned among p threads to concurrently search for matching with string s2. After a thread finishes its work and obtains the number of local matching, this local number is added into a global variable showing the total number of matched substrings in string s1. Finally this total number is printed out. The format of the strings.txt is like this(the first string iss1 and the second one is s2):
   s1: Hello we are doing pthread testing with string.
   s2:in

   HINT: divide the s1 into two half and create two threads to search substring in this two half.

2. Demonstration of Race Condition in producer and consumer problem using thread implementation.

3. Write a program to implement producer consumer problem (Using MUTEX)
   Description: The producer-consumer problem (Also called the bounded-buffer problem.) illustrates the need for synchronization in systems where many processes share a resource. In the problem, two processes share a fixed-size buffer. One process (producer) produces information and puts it in the buffer, while the other process (consumer) consumes information from the buffer. These processes do not take turns accessing the buffer, they both work concurrently. Herein lies the problem. What happens if the producer tries to put an item into a full buffer? What happens if the consumer tries to take an item from an empty buffer? In order to synchronize these processes, we will block the producer when the buffer is full, and we will block the consumer when the buffer is empty. So the two processes, Producer and Consumer, should work as follows:
   Procedure for doing the experiment:
   1. Declare variable for producer & consumer as pthread-t-tid produce tid consume.
   2 . Declare a structure to add items, semaphore variable set as struct.
   3 .Read number of items to be produced and consumed.
   4. Declare and define semaphore function for creation and destroy.
   5 . Define producer function.
   6 . Define consumer function.
   7 . Call producer and consumer.
   8. Stop the execution.

4. Modify the producer-consumer program from Task 3 to use mutexes and condition variables for synchronization. Compare this implementation to the previous one without synchronization mechanisms.

5. Describe a scenario where the use of a barrier would be beneficial in a multithreaded program. Write a code example in C that demonstrates the implementation of a barrier using pthreads. Discuss the role of a barrier in achieving synchronization among threads in your example.