Operating Systems (coe628)

Lab 5

Week of February 26, 2024

Objectives

- Learn how to use threads in C (using the POSIX pthread package)

Preparation

- Familiarize yourself with the POSIX thread library:
  http://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html#BASICS.
  You will use the following calls:
  - pthread_create
  - pthread_join
  - pthread _mutex _lock
  - pthread _mutex _unlock
  - pthread_self

- You will also use the function getpid ()

Requirements

1- When two or more threads access a shared variable at the same time a *race* condition may occur; this can affect the program correctness.

2- When two or more threads try to access a shared variable, the scheduler may switch from the active thread to another while the thread has not finished working on that variable. This may result in incorrect flow of execution.

3- In this exercise, you will create a situation where race condition occurs. You will implement a program where counter threads increment a shared variable. Since the counter threads do not have exclusive access to the counter variable when accessing it, the race condition happens and the variable is not incremented correctly.

1 Download the template Netbeans project from D2L.

2 In the code, each of the placeholder comments should be replaced with one or more instructions in order to complete the program. The required libraries have been added to the file but you may need to include more libraries if you follow a different approach. In the source file the function count increments the shared variable *globalNumber* by one for 10 times. The program creates five threads that run the counter function and therefore at the end, the value of the *globalNumber* variable should be 50.

3 The code you download should compile and run.