University of Massachusetts Dartmouth
CIS 370, Fall 2013
Lab 6
10/22/2013, 10/24/2013
POSIX Threads (Pthreads)

Objective

In this lab, you are going to understand and experiment with the Pthreads thread library to create and explore multithreaded programs.

Description

This lab assignment consists of two parts.

1. Fibonacci's Thread

The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, …. Formally it can be expressed as:

$fib_0 = 0$
$fib_1 = 1$
$fib_n = fib_{n-1} + fib_{n-2}$

Write a multithreaded C program to generate the Fibonacci sequence. The program should work as follows. The user will enter at the command line the number of Fibonacci numbers that the program is to generate. The program will then create a separate *CHILD* thread which will generate the Fibonacci sequence. When the child thread finishes execution, the *PARENT* thread will output the sequence. Each thread should print out its PID and TID before exiting.

Example:

Entered at the command line	Output
`$   ./lastnameFibonacci 2`	`0, 1`
`$   ./lastnameFibonacci 5`	`0, 1, 1, 2, 3`
`$  ./lastnameFibonacci 10`	`0, 1, 1, 2, 3, 5, 8, 13, 21, 34`

2. Threaded Prodigies

Write a C program which accepts 3 integer parameters. The program should create 3 parallel threads (4 including the initial "parent" thread) T1, T2, and T3.

- T1 should print its PID, its TID, then calculate the *sum* of the three integers.
- T2 should print its PID, its TID, then calculate the *product* of the three integers.
- T3 should print its PID, its TID, then calculate the *sum of the squares* of the three integers.

The "parent" thread should wait for all three of the above threads to exit before printing out its own PID, TID, all three values calculated by T1, T2, & T3, and then exit.

NOTE: All three threads should be created and execute in parallel (T1 should not exit before T2 is created, etc.). Sequential threading will receive no credit.

Hint: Use the function `atoi()` to convert an argument passed in through the command line into an integer.

To compile C code implementing threads, you will need to append "`-lpthread`" in the terminal.

Ex: `gcc -o lab6 lastnameProdigies.c -lpthread`

Deadline

Section 01 – Tuesday, 10/29/2013
Section 02 – Thursday, 10/31/2013

Resources

- **Chapter 4**, *Operating System Concepts*, Silberschatz, Galvin, Gagne (Course Text Book)
- http://www.advancedlinuxprogramming.com/alp-folder/alp-ch04-threads.pdf