May 23, 2016

EE 356/CS 376 **September 20, 2017**
Project 3 – Threading Analysis **Due: October 2, 2017**

For this project you will develop a benchmark program to evaluate the effectiveness of threading on different machines. The benchmark program should consist main code which creates six double precision arrays of size N. Fill the first two arrays with random doubles on the interval 0 to 1. Write four methods which will operate on the first two arrays and produce a third result array. Since there are four methods you will have the original two random arrays plus four more result arrays. Your program should run the four methods and produce a single double precision answer which is the sum of all of the terms in the four result arrays.

The four methods are:
Method 1: Multiply – This method should consist of a single loop that runs N times and multiplies to double precisions arrays and places the result in a third double precision array. $r[i] = a[i] * b[i]$

Method 2: Pythagoras – This method should consist of a single loop that runs N times and produces a result array in which each term is the square root of the sum of the squares of the corresponding terms in the random arrays. $r[i] = \sqrt{a[i] + b[i]}$

Method 3: Trigonometry – This method should consist of a single loop that runs N times and produces a result array in which each term is the sum of the sin of one random array plus the cosine of the second random array. $r[i] = \sin(a[i]) + \cos(b[i])$

Method 4: Exponents – This method should consist of a single loop that runs N times and produces a result array in which each term is the sum the corresponding term in the first random array to the 0.8 power and the second random array term to the 0.25 power.
$r[i] = (a[i])^{0.8} + (b[i])^{0.25}$

Minimum Requirements:

1. Your project must use C# WPF.
2. Provide a user interface consisting of a start button, a text box for the sum, and a graph created using the charting tool which shows the machines time on each of four tests.
   A) Test 1 will run all four methods without threading.
   B) Test 2 will run all four methods using a simple user created threads (not background workers.)
   C) Test 3 will run all four methods using background workers with *no frills*.
   D) Test 4 will run all four methods using background workers which report progress on progress bars and support user cancellation. (You will need four progress bars and four cancellation buttons since all four background workers run at the same time.)
3. Run your program on at least three machines: one should have a single core, one should have a dual core, and one should have a quad core (or higher).

Extras:

1. Include an *About* box that gives the author, date, etc.
2. Print CPU information on the result screen.
3. Create a fifth test in which you implement progress bars in Test 2.
4. A help box that explains the details of the tests performed.

Turn in the following electronically to your instructor

1. A Word document titled `EE356P2XXX.docx` (where xxx is your three initials) which contains:
   - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
   - A short description of your project. Include an overview and any special features that you added that are not in the specifications. Be sure to include a list of the machines that you used for your testing that has the number of cores and the crystal speed.
   - Screen shots showing your results from each machine.

2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title `EE356P3XXX.zip` where `XXX` are your three initials and email them to your instructor.