Fakulti:	**FAKULTI KEJURUTERAAN ELEKTRIK**	
Nama Matapelajaran: MAKMAL TAHUN 3 Kod Matapelajaran : SKEL 3742	Semakan Tarikh Keluaran Pindaan Terakhir No. Prosedur	: 1 : 2015 : 2015 : **PK-UTM-FKE-(0)-10**

SKEL 3742

FAKULTI KEJURUTERAAN ELEKTRIK
UNIVERSITI TEKNOLOGI MALAYSIA
SKUDAI, JOHOR

MICROPROCESSOR LAB

Multi-core Programming on Intel Atom Platform : Merge Sort using pthreads

Disediakan oleh	:	Disahkan oleh	: Ketua Jabatan
Nama	: Dr. Usman Ullah Sheikh	Nama	:
Tandatangan	:	Tandatangan	:
Cop	:	Cop	:
Tarikh	: 21 February 2017	Tarikh	:

Project Introduction:

A multi-core processor is a processor with multiple CPU cores capable of executing multiple software threads at the same time. However, in order to take advantage of this parallel processing

capability, the software must be multi-threaded. **pthreads** allows programmers to write parallel programs easily. **pthreads** enables the creation of shared-memory parallel programs.

The objectives of this project are:

1. To examine a serial problem and convert it to parallel processing.
2. To measure the performance gain by applying multi-threading.
3. To understand basic pthread commands, and perform compilation of pthread enabled programs.
4. To understand the benefits as well as the limits of parallel programming.

Project Tasks:

In this project-based laboratory, you are required to implement a sorting program using C programming. You need to implement the basic serial C implementation of mergesort. Based on the serial program, measure the execution time. From there on, devise an approach to improve the performance of this program using parallel programming using pthread.

Week 1

1. Determine the specification of the Norco Atom platform.
    a. How many physical processors, CPU cores, logical processors are available?
    b. What is the processor model used in the Norco Atom platform?
    c. Does the processor has HyperThreading capability? What is HyperThreading?
    d. Determine the Linux distribution used, the kernel version and the gcc compiler version.
2. Describe phtread.
3. Write and show the C code to time a section of code (profiling).
4. Show how to compile C code using gcc compiler.
5. Describe in pseudocode, how to perform **merge sort.**

Week 2

1. Implement the pseudocode to perform merge sort using C code.

2. Show that the C code produces the correct output. Use a small array of numbers to prove the correctness of the output. Generate the numbers randomly in code.
3. Increase the size of array and measure the time to sort. The measurement should not include any I/O process (e.g. read file, print to screen). When declaring very big array of data, you may have to allocate memory dynamically. Show how you perform dynamic memory allocation.
4. Plot the graph of array size vs. computation time you have obtained from step (3).
5. Describe how you are going to parallelize the serial merge sort code you have implemented. Also mention the expected speed improvement you would achieve.
6. What is the complexity of a merge sort algorithm?

Week 3

1. Using pthread, implement a parallel merge sort C code. Show the compiler option required to compile the code.
2. Vary the number of threads, size of array (as in Week 2) and measure the execution time.
3. Plot the graph of array size vs. computation time for different number of threads.
4. Does the speedup scale with number of processors?
5. Conclude your experiment results. Besides parallelism, suggest another way to improve sorting.

A report supported with the experiment results is expected to be produced at the end of this project. The collected data, analysis and plots of results should be well presented and discussed in detail in the report. The report must cover both the theoretical concepts and the experimental results. C code must be attached.