ECE5720 Programming Assignment 1, due February 15 Spring 2019

1. Matrix-matrix multiplication

Recall the problem of matrix-matrix multiplication discussed in Lecture 1. There were two algorithms presented

- an elementary "row-by-column" approach, slides 36-37, and
- a blocked (or tiled) approach, slides 38-40.

You are asked to write a C code implementing these two sequential algorithms.

In class we introducd POSIX threads. You are asked to write a `pthread` C code for executing matrix-matrix multiplication.

3. General instructions

- there should be a single file for each code
- name the files as follows
  - `mynetid_mm_rbyc.c` for the row-by-column method
  - `mynetid_mm_tile.c` for the blocked method
  - `mynetid_mm_pt.c` for the `pthread` method
  - `mynetid` is your Cornell net ID.
- the first few lines in each file should say how the code is to be compiled and executed
- your codes must be written in standard C language and compiled by the `gcc` compiler (use the `-O3` optimization flag), please use only standard C directives (OS independent)
- if your codes generate compile errors on Linux machines they will be considere as non-compliant and will not earn any credit
- your code must be well documented so any person with limited knowledge of C could understand what the code is doing
- please clean the code so all spurious and debugging commands are removed
- your codes and findings (discussion of performance) must be described in a file `mynetid_hw1.pdf`, for clarity and to save space you can refer to relevant lines in the codes,
- all files need to be archived with the `tar` or `gzip` utilities, the archive file should be named `mynetid_hw1.suffix` where `suffix` is either `tar` or `zip`,
- submit your work on Blackboard.

3. Code specific instructions

- `mynetid_mm_rbyc.c` should take from the command line user supplied parameter `dim_n` which defines the dimension of two square matrices to be multiplied
- `mynetid_mm_tile.c` should take from the command line user supplied parameters
  - `dim_n` which is the dimension of two square matrices to be multiplied,
  - `tile_size` is the dimension of tiles
  - the user should be informed that `dim_n` has to be divisible by `tile_size`,
- `mynetid_mm_pt.c` should take from the command line user supplied parameters
  - `dim_n`, the dimension of two square matrices to be multiplied,
  - `nthreads`, the number of threads to be used
  - you can use any work assignment to threads you find applicable but try to make the code as efficient as possible
- the memory for the matrices should be allocated dynamically, for example by the following commands

```
/* using a single pointer */
     int *mat = (int *)malloc(r * c * sizeof(int));
     /* loop to fill-in the array */
     *(mat + i*c + j) = (read from file or set to f(i,j));

/* using an array of pointers */
     int *mat[r];
     for (i=0; i<r; i++)
       mat[i] = (int *)malloc(c * sizeof(int));
     /* loop to fill-in the array */
     mat[i][j] = (read from file or set to f(i,j));
```

- populate matrices by calls to the library function `drand48()`, set the seed by calling `srand48(1)`
- you have to time your codes, for a high resolution clock consult
  - `https://www.cs.rutgers.edu/~pxk/416/notes/c-tutorials/gettime.html`
  - `http://linux.die.net/man/3/clock_gettime`
- for pthreads directives consult

  `https://computing.llnl.gov/tutorials/pthreads/`

or any other convenient source, try to use only the simplest pthread constructs,

- if you relay on resources outside lecture notes but publically available, you need to cite sources in your write-up,
- follow the above to the letter

4. Benchmarking.

In this assignment you want to investigate the influence of various parameters on the performance (speed) of the codes. Please discuss the following points.

- How does the performance of the sequential row-by-column compare to the sequential blocked matrix-matrix multiplication?
- How does perfomance vary with different dimensions of tiles? What is the speed-up (or slow down) when tiling is used?
- In your `pthreads` code, how does the perfromance vary with varying number of `threads`?

Parameters:

(a) To assess the quality of your codes you need to banchmark your code for a range of dimensions and thread numbers.

(b) Please graph and tabulate your results and discuss your findings in the file `mynetid_hw1.pdf`.

Please present your tables and graphs in a way so they are easily readable. For example, if certain combinations of (number o threads, matrix dimension) do not bring any new information, you may omit them from your graphs. But then explain why you are omitting them.