Take-Home Exam 3

- Your code is automatically graded using a script, and therefore, if your file/folder names are wrong you will receive a grade of zero. Please read and follow the instructions carefully.
- Server spec: CPU: 64-bit quad-core core i5 with 8GB of RAM and 6MB of shared L3 cache.
- Go to the folder **~/the3/** in your home directory on the server and put the following files in this directory and remove any compiled binary and test cases.
- Make sure your code compiles and runs without any error **on the server**. Your grade will be zero if any compile or runtime error occurs on the server. Any!
- You will receive the speed grade only if your code runs correctly.
- Just leave your final programs on the server. Don't email anything!
- Talking to your friends and classmates about this take-home exam and sharing ideas are OK. Searching the internet, books and other sources for any code is also OK. However, copying another code is not OK and will be automatically detected using a similarity check software. In such a case your entire take-home exam grade will be zero, not just the copied part.

pth_msort.c – Parallel merge sort algorithm in pthread (grade: 20% correctness, 80% speed)

Develop a parallel merge sort program according to the algorithm shown in the following figure, for any array size N=2^M (22 <= M <= 28). Write your program in C (not C++) using Linux pthread library. Use the provided pth_msort.c to start your work.

- 4 sort modules: Use any sort algorithm you wish for the 4 sort modules. You should launch one thread for each of the 4 sort modules, i.e., total of 4 threads for all sort modules.
- 2 middle merge modules: Use serial merge algorithm for the 2 middle merge modules. You should launch one thread for each of the 2 middle merge modules, i.e., total of 2 threads for all middle merge modules.
- The last merge module: You must parallelize the last merge module using 4 threads.

Compile:
gcc -O2 -pthread -lm pth_msort.c
Execute:
./a.out M
Parameter M is input command line argument to the main() function, e.g., **./a.out 22**