CSCI 3232 Systems Software Assignment 6

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Mar 21, Saturday, 2020.**

**Note: Make sure you have practiced all relevant sample codes in slides and Folio’s example codes before you start this assignment. About makefile submission issue in Folio, see last slide in 3\_Pointers\_Functions.pptx.**

1. (50 pts) Write a C or C++ program **A6p1.c**(**pp**) that accepts one command line argument which is an integer *n* between 2 and 6 inclusive. Generate a string of 60 random upper case English characters and store them somewhere (e.g. in a char array). Use pthread to create *n* threads to convert the string into a complementary lower case version of the string (‘A’→’z’, ‘B’→’y’, ‘C’→’x’,…, ‘Z’→’a’). You should divide this conversion task among the *n* threads as evenly as possible. Print out the string both before and after conversion on two separate lines. Hint: it is not necessary to use two arrays and it is dangerous to have printing code in your thread function(s). You may refer to testThread.c, testThread2.c, pthread\_ex1.c, pthread\_ex2.c for examples. Note: if you do not use pthread to divide the conversion task among the threads, you may get zero points.
2. (50 pts) Write a C or C++ program **A6p2.c**(**pp**) that accepts one command line argument which is an integer *n* between 2 and 4 inclusive. Generate 60 random integers between 1 and 49 inclusive and store them in a 5 by 12 two dimensional integer array (e.g.,int a[5][12];). Use pthread to create *n* threads to square all 60 array elements. You should divide this update task among the *n* threads as evenly as possible. Print the array both before and after the update separately as 5 by 12 matrices. Hint: it is not necessary to use two arrays and it is dangerous to have printing code in your thread function(s). Note: if you do not use pthread to divide the update task among the threads, you may get zero points. A sample run of the programs in 1 and 2 is shown below. You do NOT need to submit screen shots. Instead submit your source file.

[kwang@computer][~/work/]$ ./A6p1 3

using 3 threads.

original upper case string:

CLEMZEFLUXAUWOXVZYOXEDTODEIZYPFBBJPCQVOKUQGTHFOIECFKHAAMHJLF

complementary lower case string:

xovnavuofczfdlceablcvwglwvrabkuyyqkxjelpfjtgsulrvxupszznsqou

[kwang@computer][~/work/]$ ./A6p2 4

using 4 threads.

original array:

3 24 34 2 30 14 36 11 41 25 32 3

16 49 23 34 20 18 48 14 17 2 10 22

1 26 48 6 43 2 24 45 25 9 3 10

22 43 25 18 24 7 20 39 12 47 24 31

20 27 49 36 28 14 13 33 44 12 44 38

updated array:

9 576 1156 4 900 196 1296 121 1681 625 1024 9

256 2401 529 1156 400 324 2304 196 289 4 100 484

1 676 2304 36 1849 4 576 2025 625 81 9 100

484 1849 625 324 576 49 400 1521 144 2209 576 961

400 729 2401 1296 784 196 169 1089 1936 144 1936 1444

Include a single makefile to compile your programs in both 1 and 2. Up to 4 points will be deducted if you don’t provide a working makefile. When grading problems 1 & 2, we will type **make** (after changing the name of your submitted makefile if necessary) and then type **./A6p1** **<n>** and **./A6p2 <n>**. That means, the two output program names should be A6p1 and A6p2 respectively.

Checklist of files to be submitted: A6p1.c(pp), A6p2.c(pp), makefile