BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K. K. BIRLA GOA CAMPUS

II SEMESTER 2018-2019

Real-time Systems (BITS G553 & CS F444)

Assignment 3

Due date 24/02/2019 (11.59 P.M) **Marks 10**

Instructions:

(1) Please upload the assignment in photon
(The file name should be <your id number>.tar.gz). You are allowed to upload only once.

(2) This is an **individual** assignment. See section of handout for Malpractice regulations

(3) The programming assignments will be graded according to the following criteria

- Completeness; does your program implement the whole assignment?
- Correctness; does your program provide the right output?
- Efficiency; have you chosen appropriate algorithms and data structures for the problem?
- Programming style (including documentation and program organization); is the program well designed and easy to understand?
- Viva conducted by me.

DO NOT FORGET to include a README file (text only) in your tar.gz file with the following contents.

General README instructions

In the directory you turn in (please upload the assignment as a tar.gz file), you must have a text-only file called README, in which you will cover AT LEAST the following:

1. Your name. If you interacted significantly with others indicate this as well.
2. A list of all files in the directory and a short description of each.
3. HOW TO COMPILE your program.
4. HOW TO USE (execute) your program.
5. A description of the structure of your program.
6. In case you have not completed the assignment, you should mention in significant detail:
   - What you have and have not done
   - Why you did not manage to complete your assignment (greatest difficulties)
     This will allow us to give you partial credit for the things you have completed.
7. Document any bugs of your program that you know of. Run-time errors will cost you fewer points if you document them and you show that you know their cause. Also describe what you would have done to correct them, if you had more time to work on your project.

QUESTION 2: Experiencing Parallel Programming with Threads

Write a program to multiply N complex numbers using threads.

Input File Format:

<No. of complex numbers = N>
<Complex No. 0>
<Complex No. 1>
<Complex No. 2>
:
:
<Complex No. N-1>
The complex number will be of the form **"a+ib"** where a and b are integers.

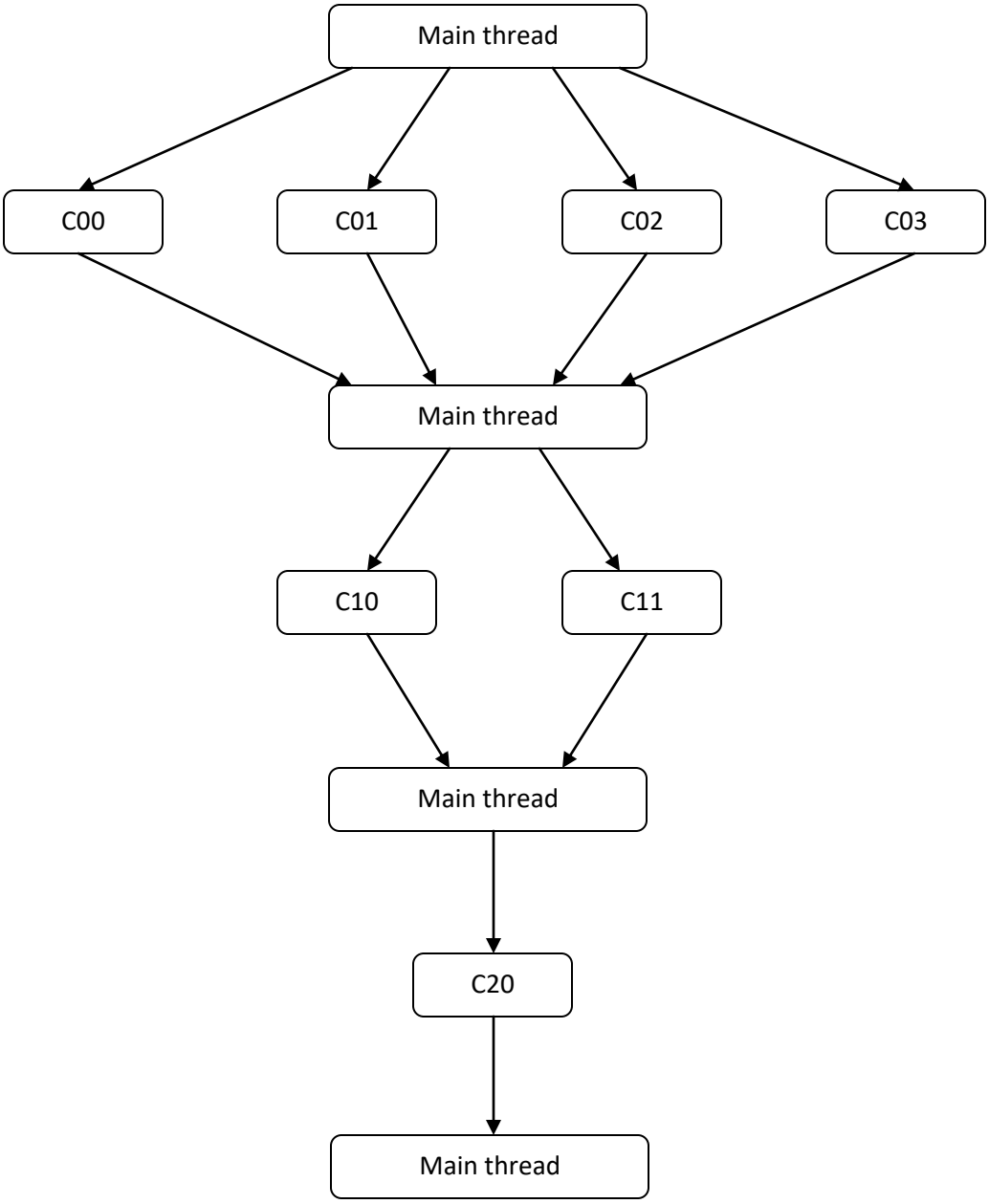
For e.g.

1 + i2 (a = 1, b = 2)
1 + i-1 (a = 1, b = -1)
-1 + i4 (a = -1, b = 4)
Procedure:

Each complex number is stored as a structure with a real and an imaginary integer member in it. The main thread should read N complex numbers from the file and store it as D[N] an array of structure in its local data space. The program should do the following

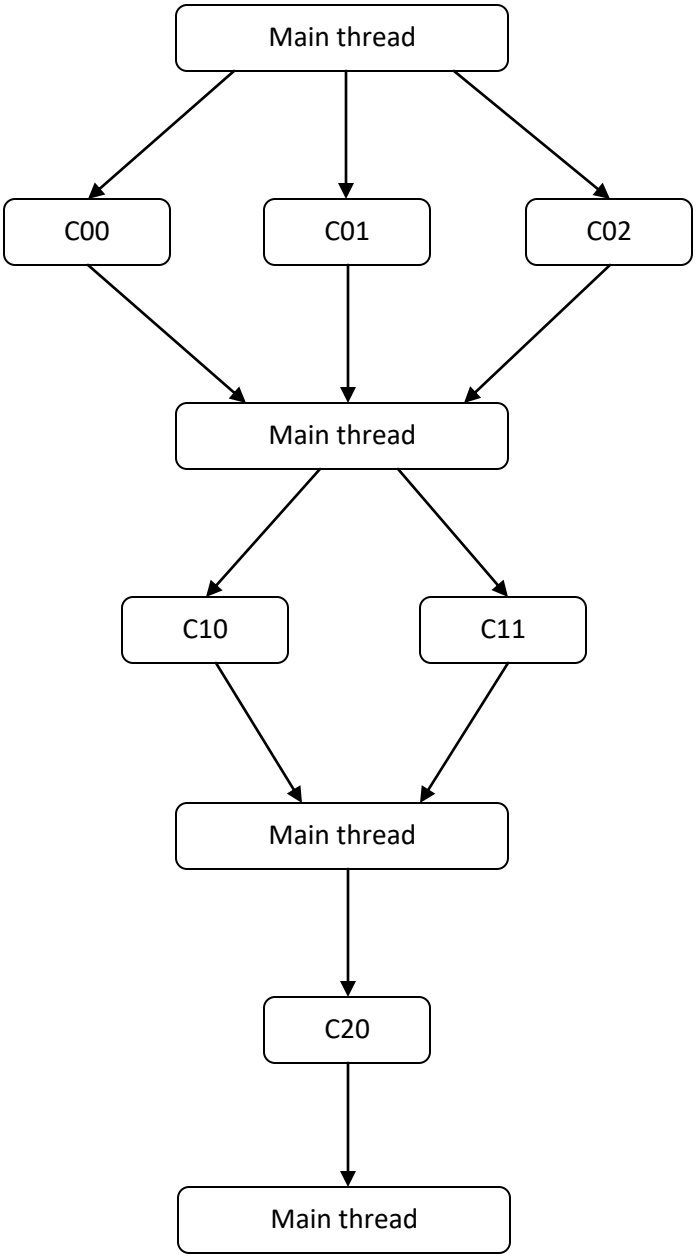
- Create [floor (N/2)] number of threads from the main thread
    - Each thread will compute product of $(2*i)^{th}$ input and $(2*i+1)^{th}$ input where i is the thread number [0 to [floor (N/2)]-1]. The product is stored in temporary output array in local space. [you need to pass the structures as argument and collect the result as return].
        - If N is an even number, the number of threads created for computation is equal to N/2 and the temporary output array elements will be equal to N/2.
        - If N is an odd number, the number of threads created for computation is equal to [floor (N/2)] and the temporary output array elements will be equal to [ceil (N/2)]. This is because the last input is transferred directly as output for the next processing.
    - After these [floor (N/2)] threads exit, the main process will print the intermediate results and will use this as the next input for the processing.
    - This process will continue till we get the final result, i.e., the number of elements in inter mediate output is exactly 1.
    - Print the final result from the main thread.

Thread hierarchy when N = 8



N = 8 will have 8 complex numbers (CN) as input
C00 computes the product of CN 0 and CN 1
C01 computes the product of CN 2 and CN 3
C02 computes the product of CN 4 and CN 5
C03 computes the product of CN 6 and CN 7
C10 computes the product of CN 0, CN 1, CN 2 and CN 3
C11 computes the product of CN 4, CN 5, CN 6 and CN 7
C20 computes the product of CN 0, CN 1, CN 2, CN 3, CN 4, CN 5, CN 6 and CN 7

Thread hierarchy when N = 7



N = 7 will have 7 complex numbers (CN) as input
C00 computes the product of CN 0 and CN 1
C01 computes the product of CN 2 and CN 3
C02 computes the product of CN 4 and CN 5
C10 computes the product of CN 0, CN 1, CN 2 and CN 3
C11 computes the product of CN 4, CN 5 and CN 6
C20 computes the product of CN 0, CN 1, CN 2, CN 3, CN 4, CN 5 and CN 6