Name: - Krishna Kumar Nayak

Place: - Kolkata, India

Applying for: - Software engineer C++

Subject: - Software Design Overview Document for Interview process.

Basic instruction for understanding of application.

- It should Debug in **Visual Studio 2017** latest version.
- In **x86 and Release mode** is best for run this code.
- All path already set just check the Window SDK version should be **10.0.17763.0** &
  Platform toolset should be **Visual Studio 2017 (v141).**
- In deployment folder you can get 2 .txt file **Array.txt** (which carry the array you want to give for test code)
  Input.txt (which carry the other input data like
  time, threshold, row…. etc)
- In Array.txt carry the array which will input for code and you can edit it.
- In Input.txt **Row length** (means how many data in one row in array)
  Row no. (means how many rows in this array)
  Threshold Value
  Process time

Coding Guideline

- All the variables starts from some initials like i_value, str_value, v_value….etc
  Means **i_** tends to integer value
  Str_ tends to structure
  v_ tends to vector
  deq_ tends to deque and so on….you can understand what was written and which type it is.
- I have divided all 4 threads in to 4 class & put all functionality of individual class in respective class.
  CDataGen
  CFilterThres
  CLabelling
  CTracingComp

- **CUtils** class for carrying all conversion like string to int, int to double…etc
  QueueAndLock class for carrying all sharable LOCK and QUEUE which can be access in all thread.
- **InputFile.h** for carry some global value which was use in some thread.
  Structures.h for carrying the all structure format which is used in all thread.

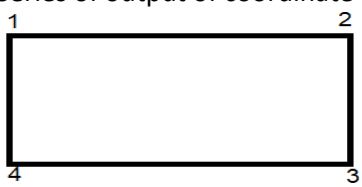
- All block will run parallelly and depend on the previous blocks result. It will start work from reading data from shared QUEUE between 2 blocks when 2nd block get signal from 1st block.
- 1st block will notify 2nd block when it will push the data in shared QUEUE.
- I have implemented the **Union-Find Algorithm** for implement the labelling the block, which link you had provided in document.
- **Locking mechanism**
  - One mutex lock is used for both notify the next thread and received signal from previous thread
  - In first thread the mutex is free then this thread notifies the second thread for processing its task if It is in wait state.
  - Second thread take the lock (if it is free or wait to free) and check the queue empty check. If empty then go to wait state with condition variable, after that the lock will be free for next use.

- **CDataGen:**
  - This thread read the data from file (Array.txt) which carry the input array.
  - And put the data from Queue_12.

- **CFilterThres:**
  - In this class it will get data from Queue_12 and calculate the threshold value.
  - And put it in Queue_23.
  - **<u>For first 4 data and last 4 data in every row, I put all zero (0) in Queue_23. Becoz for this we did not get enough value for threshold calculation. (that part you did not clarify in document)</u>**

- **CLabelling:**
  - In this thread we will get the data from Queue_23 and process for labelling.
  - Check all one (1) and labelled according to its neighbour element and calculate the size in every label.
  - When merge occurs in between 2 label the size also merged.
  - **<u>In document you mentioned some condition that could not be understood by me. That's why I did not implement it in my code.</u>**

    ""To reference and index all replaced labels that got connected with another label, maximum of 'two 1D arrays' of maximum size `m/2` each, is allowed *(need for this will be understood while implementing the code).* ""

  - Put all collected like co-ordinates, size, neighbours …etc in a structure and push it in next queue for thread 4.
- **CTracingComp:**
  - After getting signal it will start the work Tracing with getting data from Queue_23
  - It will calculate sizes of all label, maintain co-ordinates for smallest encompassing rectangle and handle print the output when a label has no future connection.
- **<u>For queue:</u>**
  - In queue operation in a single time only 1 thread can do any one operation (Push/Pull/EmptyCheck) At a time.
- 1 for push the data in queue
  2 for pull the data from queue
  3 for check the queue is empty or not.
- **<u>For output:</u>**
  It will show the output like this
  Label:
  Size:
  Co ordinates
  - In coordinates output I have taken the row/column count from 1 (not 0).
  - (x,y) x tend to x-axis and y tends to y-axis
  - Series of output of coordinate is: left top, right top, right down, left down
  - 

  - ***note** if an image is in one row, the coordinate will print in upper condition
    Ex: (1,5) (5,5) (5,5) (1,5)

Conclusion

- **I think block 3 & 4 has more dependable data sharing in between then, so I checked that if we put them together in one thread then the answer is coming more accurate. Kindly think about it.**
- **If you think the code need more enhancement, then inform me.**
- **I have checked my code with taking small data set(array), might be it will not more than this.**
- Lastly, I want to say, this code might be not the optimal solution. Something more enhancement may be added in future. But in short time period with my busy office hour I have coded to it in my knowledge.
- Might be some boundary condition I have missed.
- If some time you want clarification in my code, then feel free to contact me.

Best regards
Krishna Kumar Nayak
Ph- 8686416488
Email- krishna.kumar.nayak14@gmail.com