

Homework - 1 CSE437
REAL TIME SYSTEM ARCHITECTURES

Hamza YOĞURTCUOĞLU - 171044086

a) HOW TO USE - Requirements:

1.Create your SubTimer object which implements ITimer

2.Construct registerTimers which can be different timers . These timers has different version :

+run the callback once at time point tp.

registerTimer(timepoint, callback)

+run the callback periodically forever. The first call will be executed as soon as this callback is registered.

registerTimer(period, callback)

+Run the callback periodically until time point tp. The first call will be executed as soon as this callback is registered.

registerTimer(timepoint,period,callback)

+Run the callback periodically. Before calling the callback every time, call the predicate first to check if the termination criterion is satisfied. If the predicate returns false, stop calling the callback.

registerTimer(predicate,period, callback)

3.You can see timer result on the console. When timer is ready.

4.Please check main.cpp to see calling orders and milliseconds.

b) Design :

When you create subtimer object. We create a thread in order to handle timers. Then main thread is calling registerTimers. Then it notify the thread of subtimer. We are looking minimum remaining time. Then we put to sleep the class timer. When minimum remaining time comes. Then thread of class wakes up. But main thread must notify thread of class. **We have just 2 threads which are class thread and main thread.**

c) Test scenarios :

- 1.Timer: You can create lot of timer at the same time.
- 2.Second and third type timers directly call the callback function.
- 3.Resource Acquisition is initialization. Main thread must the wait(join) in destructor thread of class and notify it.
- 4.If the callback is called outside 5 milliseconds' offset from the registered time point, the timer will report "Deadline error", but the callback will still be called.
- 5.If you didn't register any timer. All operations are finished. Then you will get report "You forget to register timer" .

Compiler and run :

firstly : → cmake . #cmake command if you are same direction '.' or you must give absolute path
secondly : → make #make command is create object code that means compile with g++.

if you get error message when source code is compiling . Please , try again "cmake ." then "make" without delete files.