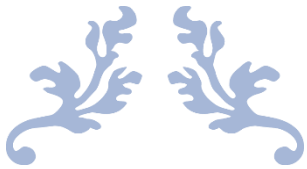

REAL TIME SYSTEMS ARCHITECTURES

Homework 1 Report

20 NOVEMBER 2019

AHMET YUŞA TELLİ
151044092

In this homework, we create a “ITimer” abstract class. From this class, we derived a “YTimer” class. In these classes, have 4 different functions with the same name “registerTimer()”. These functions:

1) void registerTimer(const Timepoint &tp, const TTimerCallback &cb);

   In this function, we call cb() function given time tp. Tp is time_point in chrono library. When we doing this, we take this time with now() (in chrono). Check the now() time to time point tp in a loop.

2) void registerTimer(const Millisecs & period, const TTimerCallback &cb);

   In this function, we do same thing previous function but we call cb() function with given “period” parameter periodically. This function runs forever, just call cb() periodically. In ınfinite loop, firstly we call cb() function then check the now() time to tp time.

3) void registerTimer(const Timepoint &tp, const Millisecs & period, const TTimerCallback &cb);

   The third function, we call cb() function periodically but until the time point tp. Firstly we check the now() time and tp time in a loop. Then we call cb() function. After that, we wait until the period time.

4) void registerTimer(const TPredicate &pred, const Millisecs & period, const TTimerCallback &cb);

   The last function, firstle we call pred() function, if it returns false the function is exit. If pred is true, we call cb() function, then we wait for a period time. We check every time pred() function.

All these functions we take cb() execution time, if the time is not 5 milliseconds we report “Deadline error” to user.

Thread Part:

In homework we should use only one thread for run cb() function. But I could not using one thread. In main function, I use 4 different threads.

If we want to use one thread, we should do this way:

In YTimer class's constructor, we create a thread and some flags or mutexes which one is better way. The thread has a thread_function. The class has a queue. The thread function use this queue.

All registerTimer() functions, when they called, they push their own parameters or specific flags. Or mutex lock when they push the queue.

In thread function, check the queue and take the first function parameters then call the function for do the job.

In class's deconstructor we should thread join in the end.

In shortly, we have a thread when the class's object created with empty queue. After registerTimer() functions call, the queue starts to fill up. The thread using this queue, pop first function and do its job.