Short paper on Timer.h Implementation

Jose Antonio Gallardo Jaramago

September 18, 2018

First lines are include macros which allow us to use standard C functions, thread allows us to use posix threads, chrono is used to sleep for a concrete milliseconds amount of time, functional is used for binding callback function, future is used to retrieve thread variables result, and cstdio and iostream are used for input/output related tasks.

Timer.h have 4 functions: Connect, receives a parameter which type is callback and contains the function that is invoked on every timeout event if timeout is started. Start and stop functions manages private class variables using functions so variables are treated as atomic in order to control thread execution. SetPeriod change the timeout period. The two private variables are go and period, go controls if the timeout thread should run the code or not, and period is the period that timeout sleeps before doing the task, variables are atomic so multithreaded execution can be safe.