CppTimer

Generated by Doxygen 1.8.13

Contents

1 CppTimer

Generic C++ Timer for Linux

It's a wrapper around the Linux timers. There are two ways of using the timer: by overloading the `timer←Event()` method in the `CppTimer` class itself (fastest) or by registering a callback class called `Runnable` with an overloaded `run()` method.

## Installation

```
cmake .
make
sudo make install
```

Usage (overloading the timer event)

Include CppTimer.h in your program and link the static library `libcpptimer.a` to your project:

```
TARGET_LINK_LIBRARIES(your_project_title cpptimer rt)
```

### Create the Timer class

```
class MyTimer : public CppTimer {

        void timerEvent() {
                // your timer event code here
        }
};
```

where you override `timerEvent` with your function.

Run the Timer class

The timer is programmed in nanoseconds:

```
MyTimer myTimer;
// every 500000ns
myTimer.startns(500000);
```

or milliseconds:

```
// every 200ms
myTimer.startms(200);
```

As soon as start returns the timer fires instantly and then at the specified interval.

Demo program

To run `demo.cpp` just do `cmake .`, `make` and then `./demo`.

Callback version

Instead of overloading the `run()` method in the timer class you can overload the `run()` method in the `Runnable` class and then register this class with the timer class. Check out `demo_runnable` which demonstrates how to use this method.

Unit tests

Run:

```
ctest
```

That's it. Enjoy!

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

3 Class Index

3.1 Class List

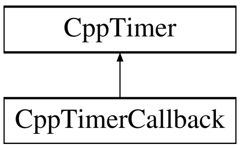
Here are the classes, structs, unions and interfaces with brief descriptions:

4 Class Documentation

4.1 CppTimer Class Reference

```
#include <CppTimer.h>
```

Inheritance diagram for CppTimer:



Public Member Functions

- CppTimer (const int signo=SIGRTMIN)
- virtual void startns (long nanosecs, cppTimerType_t type=PERIODIC)
- virtual void startms (long millisecs, cppTimerType_t type=PERIODIC)
- virtual void stop ()
- virtual ~CppTimer ()

Protected Member Functions

- virtual void timerEvent ()=0

4.1.1 Detailed Description

Timer class which repeatedly fires. It's wrapper around the POSIX per-process timer.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 CppTimer()

```
CppTimer::CppTimer (
             const int signo = SIGRTMIN )
```

Creates an instance of the timer and connects the signal handler to the timer. The default signal which is being used is SIGRTMIN but can be changed to other signals if other processes / threads use them.

Parameters

signo	The signal used by the timer.

4.1.2.2 ∼CppTimer()

```
virtual CppTimer::∼CppTimer ( ) [virtual]
```

Destructor disarms the timer, deletes it and disconnect the signal handler.

4.1.3 Member Function Documentation

4.1.3.1 startms()

```
virtual void CppTimer::startms (
             long millisecs,
             cppTimerType_t type = PERIODIC ) [virtual]
```

Starts the timer. The timer fires first after the specified time in milliseconds and then at that interval in PERIODIC mode. In ONESHOT mode the timer fires once after the specified time in milliseconds.

Parameters

millisecs	Time in milliseconds
type	Either PERIODIC or ONESHOT

4.1.3.2 startns()

```
virtual void CppTimer::startns (
              long nanosecs,
              cppTimerType_t type = PERIODIC )  [virtual]
```

Starts the timer. The timer fires first after the specified time in nanoseconds and then at that interval in PERIODIC mode. In ONESHOT mode the timer fires once after the specified time in nanoseconds.

Parameters

nanosecs	Time in nanoseconds
type	Either PERIODIC or ONESHOT

4.1.3.3 stop()

```
virtual void CppTimer::stop ( )  [virtual]
```

Stops the timer by disarming it. It can be re-started with start().

4.1.3.4 timerEvent()

```
virtual void CppTimer::timerEvent ( )  [protected], [pure virtual]
```

Abstract function which needs to be implemented by the children. This is called every time the timer fires.

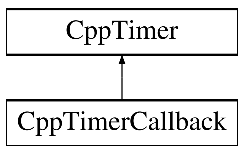
Implemented in CppTimerCallback.

The documentation for this class was generated from the following file:

- CppTimer.h

4.2 CppTimerCallback Class Reference

Inheritance diagram for CppTimerCallback:



Classes

- class Runnable

Public Member Functions

- void timerEvent ()

Additional Inherited Members

4.2.1 Member Function Documentation

4.2.1.1 timerEvent()

```
void CppTimerCallback::timerEvent ( )  [inline], [virtual]
```

Abstract function which needs to be implemented by the children. This is called every time the timer fires.

Implements CppTimer.

The documentation for this class was generated from the following file:

- CppTimerCallback.h

4.3 CppTimerCallback::Runnable Class Reference

The documentation for this class was generated from the following file:

- CppTimerCallback.h

Index