Subject: Re: time measurement :: RTIMING, TimeStop, GetTickCount
Posted by Didier on Wed, 09 May 2012 20:23:11 GMT
View Forum Message <> Reply to Message

I use this to measure timings, it works in linux and windows.

Timing.h

```
#ifdef WIN32
 #include <Windows.h>
 class windowsTiming
 {
  public:
   windowsTiming(void);

  public:
   typedef struct
   {
     signed __int64 nTicksCnt;
   } timeType;

   inline double diff_ms(timeType& p_start, timeType& p_end)
   {
    return (double(p_end.nTicksCnt-p_start.nTicksCnt)/double(m_TickPerSecond)*1000.);
   };

   inline timeType getTime(void)
   {
    timeType res;
    QueryPerformanceCounter((LARGE_INTEGER*)&res.nTicksCnt);
    return res;
   };

  private:
   signed __int64 m_TickPerSecond;
 };

 typedef windowsTiming HWTiming;

#else
 #include <time.h>
 class LinuxTiming {
  public:
   LinuxTiming() {};

  public:
   typedef timespec timeType;
```

```
  static inline timeType getTime()
  {
   timeType t;
   clock_gettime(CLOCK_REALTIME, &t);
   return t;
  }

  static inline double diff_ms(timeType& t1, timeType& t2)
  {
   return ((t2.tv_sec-t1.tv_sec)*1000.0 + (t2.tv_nsec-t1.tv_nsec)/1000000.0);
  }
 };

 typedef LinuxTiming HWTiming;
#endif
```

Timing.c

```
#ifdef WIN32
  windowsTiming::windowsTiming(void)
  {
   QueryPerformanceFrequency((LARGE_INTEGER*)&m_TickPerSecond);
  };
#endif
```

To use it in a cross plateform way, do the following:

```
HWTiming timeMeasurer;

HWTiming::timeType  startTime = timeMeasurer.getTime();

.... do you're thing !-)

HWTiming::timeType  endTime = timeMeasurer.getTime();

double delta = timeMeasurer.diff_ms(endTime, startTime);
```