

## PeriodicExecutionContextでのsleep時間の計算方法

2010/07/06 23:33 - n-ando

ステータス:	終了	開始日:	2010/07/06
優先度:	通常	期日:	
担当者:	n-ando	進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			

**説明**

PeriodicExecutionContextでのsleep時間は、RTCの実行時間が考慮されていないので修正する。  
実際のsleep時間は、(周期)-(全RTCの実行時間) にするべき。

## 履歴

#1 - 2011/04/29 20:56 - n-ando

- ステータス を 新規 から 終了 に変更
- 担当者 を n-ando にセット
- 進捗率 を 0 から 100 に変更

ECのsleep時間をRTCの実行時間を考慮して計算するように修正。

数百msオーダーでの正確さはあるが、それ以上 ( msオーダー ) ではあまり意味がない。

```
+++ trunk/OpenRTM-aist/src/lib/rtm/PeriodicExecutionContext.cpp 2011-04-29 11:54:13 UTC (rev 2075)
@@ -148,12 +148,24 @@
{
    m_worker.cond_.wait();
}
+ coil::TimeValue t0(coil::gettimeofday());
if (m_worker.running_)
{
    std::for_each(m_comps.begin(), m_comps.end(), invoke_worker());
}
m_worker.mutex_.unlock();
- if (!m_nowait) { coil::sleep(m_period); }
+ coil::TimeValue t1(coil::gettimeofday());
+ RTC_PARANOID("Period: %f [s]", (double)m_period);
+ RTC_PARANOID("Execution: %f [s]", (double)(t1 - t0));
+ RTC_PARANOID("Sleep: %f [s]", (double)(m_period - (t1 - t0)));
+ coil::TimeValue t2(coil::gettimeofday());
+ if (!m_nowait && m_period > (t1 - t0))
{
    RTC_PARANOID("sleeping... ");
    coil::sleep((coil::TimeValue)(m_period - (t1 - t0)));
}
+ coil::TimeValue t3(coil::gettimeofday());
+ RTC_PARANOID("Slept: %f [s]", (double)(t3 - t2));
} while (m_svc);

return 0;
```