

実行周期の設定値

2009/09/18 14:02 - 匿名ユーザー

ステータス:	終了	開始日:	2009/09/18
優先度:	低め	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 実行周期(exec_cxt.periodic.rate)に0.00045Hz以下(0Hzなど)の値を設定するとコンポーネントのCPU使用率が跳ね上がる。 以下の3パターンを試した結果、[A][B]に比べ[C][D]のCPU使用率がかなり大きくなりました。 [A] - onExecute : 使用 実行周期 : 1000Hz [B] - onExecute : 未使用 実行周期 : 1000Hz [C] - onExecute : 使用 実行周期 : 0Hz [D] - onExecute : 未使用 実行周期 : 0Hz * CPU使用率は"pidstat"コマンドにより測定しました。 環境は以下の通りです。 RTM-aist-0.4.2(c++) ubuntu-8.04			

履歴

#1 - 2010/01/08 17:22 - 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

C++版PeriodicExecutionContextの周期計算において、与えられた周波数からその逆数として計算される実行周期を秒以下であるとの仮定に基づいて計算していたため。
 実際の計算では、マイクロ秒単位の実行周期をint型で保持しており、周期の計算では、

周期: $\text{period} = 1000000/\text{hz}$

として計算を行っていた。指摘の条件 $\text{hz} = 0.00045$ の場合、

周期: $1000000/0.00045 = 2222222222.2222222222222222222222$

となり、int型の最大値2147483647を超えていた。
 この場合、最終的に周期は0となり正確な周期で実行できないだけでなくCPU時間を無駄に消費することとなる。

PeriodicExecutionContext

が保持する実行周期[s]をcoil::TimeValue型で保持し、coil::sleep()にもcoil::TimeValue型で直接与えることで、この問題はなくなる。
 ただし、TimeValueは内部的に、int型でsecとusecを保持しており、かつset_rate()関数では周波数をdouble型の引数で渡すことができるので、これらの変数をあふれさせるさらに小さすぎる周波数を与えることにより同様の現象がおこる可能セがある。

#2 - 2010/03/09 13:34 - 匿名ユーザー

[r1619](#)