

## OpenRTM-aist (C++) - バグ #1426

### 1.0.0のRingBuffer関連の問題

2010/04/22 17:39 - n-ando

<b>ステータス:</b> 終了	<b>開始日:</b> 2010/04/22
<b>優先度:</b> 通常	<b>期日:</b>
<b>担当者:</b>	<b>進捗率:</b> 100%
<b>カテゴリ:</b>	<b>予定工数:</b> 0.00時間
<b>対象バージョン:</b>	
<b>説明</b>	
<p>KEK仲吉さん安さんから指摘。</p> <p>とりあえずの対処として read()の前に InPortに対し isEmpty()を行っていました。この件を安さんが調査し次のような問題を見つけられましたので報告致します（以下、安さんからのメールの抜粋）。</p> <pre>  RingBuffer.hのwriteメソッドで、   従来は     if (empty_)   {       Guard eguard(m_empty.mutex);       m_empty.cond.signal();   }   advanceWptr(1);     となっていました。   これだと、readでバッファがなく待っていた状態で、writeが入り、   cond.signalでシグナルを出しますが、 advanceWptr(1)をする前に、   シグナルを出しているため、writeのバッファポインタがインクリ   メントされる前に、readがwake upする可能性が出てきてしまいます。   そこで、下記のように変更しました。     if (empty_)   {       Guard eguard(m_empty.mutex);       advanceWptr(1);       m_empty.cond.signal();   } else       advanceWptr(1);     これで、昨日の昼から走らせていますが、数秒で止まってしまう場合   があったり数十分で止まってしまったりする場合があるのに比べて、   いまだ走っていますので、つじつまがあう点を考えても、問題は解決   したものと思います。</pre>	
<b>関連するチケット:</b>	
関連している OpenRTM-aist (Java) - バグ #2327: RingBuffer関連の問題	終了 2012/01/16

#### 履歴

#1 - 2010/04/22 17:41 - n-ando

- ステータスを新規から終了に変更

- 進捗率を0から100に変更

修正しました。