

サブスクリプション型がnewの場合に処理が停止することがある問題

2018/02/28 10:34 - n-miyamoto

ステータス:	終了	開始日:	2018/02/28
優先度:	通常	期日:	
担当者:	n-miyamoto	進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:	RELEASE_1_2_0		
説明			
サブスクリプション型をnewにした場合に、データ転送中にコネクタを切断すると処理が停止することがあるため原因の調査、修正を行う。			

履歴

#1 - 2018/02/28 13:26 - n-miyamoto

- ステータスを担当から解決に変更

- 進捗率を0から100に変更

RingBuffer.pyの以下の箇所で停止していることを確認しました。

```
def write(self, value, sec = -1, nsec = 0):
    try:
        self._full_cond.acquire()
        if self.full():
            #省略
            elif not overwrite and timedwrite: # "block" mode
                #省略
                ret = self._full_cond.wait(wait_time)
```

PublisherNewはバッファフルの時にwrite関数を呼ぶとブロックするようになっています。

このブロックはread関数内で解除できます。

```
def read(self, value, sec = -1, nsec = 0):
    #省略
    self._full_cond.acquire()
    full_ = self.full()

    if full_:
        self.advanceRptr()
        self._full_cond.notify()
    else:
        self.advanceRptr()

    self._full_cond.release()
```

ただ、PublisherNewのpushNew関数ではデータ読み込み時にread関数を使用していないため、バッファフルのブロックが解除されません。

```
def pushNew(self):
    self._rtcout.RTC_TRACE("pushNew()")
    try:
        self._buffer.advanceRptr(self._buffer.readable() - 1)

        cdr = self._buffer.get()
        self.onBufferRead(cdr)

        self.onSend(cdr)
        ret = self._consumer.put(cdr)

        if ret != self.PORT_OK:
            self._rtcout.RTC_DEBUG("%s = consumer.put", OpenRTM_aist.DataPortStatus.toString(ret))
            return self.invokeListener(ret, cdr)

        self.onReceived(cdr)
        self._buffer.advanceRptr()
```

```
return self.PORT_OK
```

```
except:  
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())  
    return self.CONNECTION_LOST
```

このため、バッファフル、バッファエンプティのブロック解除の処理をadvanceRptr関数、advanceWptrに移動しました。

```
def advanceRptr(self, n = 1, unlock_enable=True):  
    if unlock_enable and n > 0:  
        self._full_cond.acquire()  
        full_ = self.full()  
        guard = OpenRTM_aist.ScopedLock(self._pos_mutex)  
        if (n > 0 and n > self._fillcount) or /  
            (n < 0 and n < (self._fillcount - self._length)):  
            return OpenRTM_aist.BufferStatus.PRECONDITION_NOT_MET
```

```
self._rpos = (self._rpos + n + self._length) % self._length  
self._fillcount -= n
```

```
if unlock_enable and n > 0:  
    if full_  
        self._full_cond.notify()  
        self._full_cond.release()
```

```
return OpenRTM_aist.BufferStatus.BUFFER_OK
```

```
def advanceWptr(self, n = 1, unlock_enable=True):  
    if unlock_enable and n > 0:  
        self._empty_cond.acquire()  
        empty = self.empty()  
        guard = OpenRTM_aist.ScopedLock(self._pos_mutex)  
        if (n > 0 and n > (self._length - self._fillcount)) or /  
            (n < 0 and n < (-self._fillcount)):  
            return OpenRTM_aist.BufferStatus.PRECONDITION_NOT_MET
```

```
self._wpos = (self._wpos + n + self._length) % self._length  
self._fillcount += n
```

```
if unlock_enable and n > 0:  
    if empty:  
        self._empty_cond.notify()  
        self._empty_cond.release()
```

```
return OpenRTM_aist.BufferStatus.BUFFER_OK
```

#2 - 2018/09/13 09:17 - n-miyamoto

- ステータスを解決から終了に変更