

A Note on Using Real-Time OS M3T-MR32R --On Releasing a Task from the Acceptance or Call of a Rendezvous--

Please take note of the following problem in using the M3T-MR32R real-time OS:

- On releasing a task from the acceptance or call of a rendezvous
-

1. Versions Concerned

M3T-MR32R V.3.00 Release 1 through V.3.50 Release 2

2. Descriptions

- (1) When a task is in a WAIT state for the acceptance or call of a rendezvous (the case where Condition (4)-(a) satisfied), no rendezvous is acknowledged even if the task meets the rendezvous condition, or another task in the READY state with the same priority as the above task is interpreted as meeting the rendezvous condition and enters a rendezvous. As a result of the above operations, such a symptom as writing a message at an incorrect address arises, and the application program may not run properly.
- (2) When a task is in a WAIT-SUSPEND state for the acceptances or calls of rendezvous (the case where Condition (4)-(b) satisfied), the processing of system calls `tcal_por`, `pcal_por`, `tacp_por`, `pacp_por`, and `fwd_por` cannot terminate.

3. Conditions

This problem occurs if the following conditions are all satisfied:

- (1) Any of the following system calls is issued:
`acp_por`, `tacp_por`, `pacp_por`, `cal_por`, `tcal_por`, `pcal_por`, and `fwd_por`

- (2) On the port that is the object of the system call in (1) exists either of the following tasks:
 - (a) A task waiting for the call of a rendezvous if `acp_por`, `tacp_por`, or `pacp_por` issued
 - (b) A task waiting for the acceptance of a rendezvous if `cal_por`, `tcal_por`, `pcal_por`, or `fwd_por` issued
- (3) Either of the tasks in (2) is released from the WAIT or WAIT-SUSPEND state by issuing an `irel_wai` system call inside the interrupt handler invoked during the execution of the system call issued in (1).
- (4) The task that is the object of the `irel_wai` issued in (3) is in either of the following states:
 - (a) A Wait state for the acceptance or call of a rendezvous
 - (b) A WAIT-SUSPEND state for the acceptances or calls of rendezvous

4. Workaround

This problem can be circumvented either of the following ways:

- (1) Before and after issuing the system call concerned, disable and enable an interrupt respectively.

Example

```
void task(INT stacd)
{
    .....

    /* Disable an interrupt */
    asm(" mvfc    R0,PSW¥n"
        " and3    R0,R0,#0xFFBF¥n"
        " mvtc    R0,PSW¥n");

    ercd = acp_por(&rno,(VP)msg,&size,ID_por2,0x80);

    /* Enable the interrupt */
    asm(" mvfc    R0,PSW¥n"
        " or3     R0,R0,#0x0040¥n"
        " mvtc    R0,PSW¥n");
```

```
.....  
}  
-----
```

- (2) Perform the following steps to release a task from a WAIT state:
1. Issue an ista_tsk system call to invoke the task to release from a WAIT state.
 2. Issue a rel_wai system call from the invoked task mentioned above.

Example

```
-----  
void handler(void)  
{  
  
.....  
  
/* Comment out irel_wai, issue ista_tsk to invoke task  
to release from WAIT state, and pass ID of  
objective task to ista_tsk as startup code */  
  
/* irel_wai( ID_task1 ); */  
ista_tsk( ID_DoRelease,(INT)ID_task1 );  
  
.....  
}  
  
void DoRelease(INT tskid)  
{  
rel_wai(tskid);  
}  
-----
```

5. Schedule of Fixing the Problem

We plan to fix this problem in our next release of the product.

[Disclaimer]

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.

