

September 9, 1988

Debug Extensions to RTEID

NOTES

Not callable from ISR.

Will not cause a preempt.

There is no requirement that the task identified by the *tid* be a controlled task.

Db_getmem will attempt to only read the requested data and will not access memory beyond the *laddr+length*. If *length* is 1, a byte wide read is performed. If *length* is 2, a word wide read is performed.

1.1.13 DB_SETMEM

NAME

`db_setmem` -- "Set a Task's Memory"

SYNOPSIS

```
uint db_setmem ( tid, laddr, bufaddr, length )
```

```
    uint tid;          /* task id as returned from t_create or t_ident */
    char *laddr;       /* logical start address */
    char *bufaddr;     /* buffer address */
    uint length;       /* length in bytes */
```

DESCRIPTION

The executive writes memory to the task identified in the *tid* field from the buffer identified in the *bufaddr* starting at the task's logical address *laddr* field for the length identified in *length*.

The task identified in the *tid* field may exist on the local processor, or any remote processor in the multiprocessing configuration if the task was created with the **GLOBAL** flag set (see *t_create*). This directive may be used to transfer data between any requesting task's buffer and a logical address belonging to the task identified by the *tid*.

RETURN VALUE

If *db_setmem* successfully writes the memory from the buffer, then 0 is returned.

If the memory was not successfully written from the buffer, an error code is returned.

ERROR CONDITIONS

Invalid *tid*.

Invalid *laddr*.

Bus Error occurred during the write.

NOTES

Not callable from ISR.

September 9, 1988

Debug Extensions to RTEID

Will not cause a preempt.

There is no requirement that the task identified by *tid* be a controlled task.

Db_setmem will only read the requested data and will not access memory beyond the *laddr+length*. If *length* is 1, a byte wide read is performed. If *length* is 2, a word wide read is performed.

1.1.14 DB_GETREG

NAME

db_getreg -- "Get a task's register"

SYNOPSIS

uint db_getreg (tid, regnum, ®ptr)

```

uint tid;           /* task id as returned from t_create or t_ident */
uint regnum;       /* register number */
union regval *regptr; /* pointer to register value - returned by this call */

union regval {
    uint i;
    float f;
}
    
```

The *regnum* field values are:

- | | |
|---------|------------------------------|
| S_STAT | Task's status byte values: |
| T_WTMEM | waiting for memory |
| T_WTMSG | waiting on message queue |
| T_WTEVT | waiting for event |
| T_WTSEM | waiting for semaphore |
| T_WTTIM | waiting for timeout |
| T_WTCTL | waiting on control |
| D_REG0 | Task's Processor Register D0 |
| D_REG1 | Task's Processor Register D1 |
| D_REG2 | Task's Processor Register D2 |
| D_REG3 | Task's Processor Register D3 |
| D_REG4 | Task's Processor Register D4 |
| D_REG5 | Task's Processor Register D5 |
| D_REG6 | Task's Processor Register D6 |
| D_REG7 | Task's Processor Register D7 |
| A_REG0 | Task's Processor Register A0 |
| A_REG1 | Task's Processor Register A1 |
| A_REG2 | Task's Processor Register A2 |
| A_REG3 | Task's Processor Register A3 |
| A_REG4 | Task's Processor Register A4 |
| A_REG5 | Task's Processor Register A5 |

A_REG6	Task's Processor Register A6
A_REG7	Task's Processor Register A7
H_SR	Status Register
H_PC	Program Counter
H_VOR	Vector Offset Register
H_USP	User Stack Pointer
H_ISP	Interrupt Stack Pointer
H_MSP	Master Stack Pointer
H_VBR	Vector Base Register
H_CACR	Cache Control Register
H_CAAR	Cache Address Register
H_VBR	Vector Base Register
H_CACR	Cache Control Register
H_CAAR	Cache Address Register
FP_REG0	Task's Processor Register FP0
FP_REG1	Task's Processor Register FP1
FP_REG2	Task's Processor Register FP2
FP_REG3	Task's Processor Register FP3
FP_REG4	Task's Processor Register FP4
FP_REG5	Task's Processor Register FP5
FP_REG6	Task's Processor Register FP6
FP_REG7	Task's Processor Register FP7
FPCR	Task's Coprocessor Control Register
FPSR	Task's Coprocessor Status Register
FPIAR	Task's Coprocessor Instruction Address Register

DESCRIPTION

The executive returns the register value in the *regptr* field for the register identified in the *regnum* field and the task identified by the *tid*.

The task identified in the *tid* field may exist on the local processor, or any remote processor in the multiprocessing configuration if the task was created with the GLOBAL flags value set (see *t_create*).

RETURN VALUE

If *db_getreg* is successful, *regptr* is filled in and 0 is returned.

If the call was not successful, an error code is returned.

ERROR CONDITIONS

Invalid *tid*.