12.6. TIMER_CANCEL

Cancel a running event timer.

Synopsis

```
   timer_cancel( tmid )
```

Input Parameters

tmid	: timer_id	kernel defined timer identifier

Output Parameters

<none>

Completion Status

OK	timer_cancel successful
ILLEGAL_USE	timer_cancel not callable from ISR
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_ID	timer does not exist
OBJECT_DELETED	originally existing timer expired or has been canceled before operation

Description

This operation cancels an event timer previously started using the timer_event_after, timer_event_when or timer_event_every operations.

13. INTERRUPTS

ORKID defines two operations which bracket interrupt service code. It is up to each implementor to decide what functionality to put in these operations.

Observation:

The kernel may use int_enter and int_return to distinguish if Interrupt Service Routine code or task code is being executed. Typically int_return will be useed to decide if a scheduling action must take place in kernels with preemptive scheduling.

13.1. INT_ENTER

Announce Interrupt Service Routine entry.

Synopsis

```
int_enter( )
```

Input Parameters

<none>

Output Parameters

<none>

Completion Status

OK	int_enter successful

Description

This operation announces the start of an Interrupt Service Routine to the kernel. Its functionality is implementation dependent. The operation takes no parameters and always returns a successful completion status. It is up to a user task to set up vectors to the handler which makes this call.

13.2. INT_RETURN

Exit from an Interrupt Service Routine

Synopsis

```
   int_return ( )
```

Input Parameters

<none>

Output Parameters

<none>

Completion Status

<not applicable>

Description

This operation announces the return from an ISR to the kernel. Its exact functionality is implementation dependent, but will involve returning to interrupted code or scheduling another task. The operation takes no parameters and does not return to the calling code.

The behavior of int_return when not called from an ISR is undefined.

14. MISCELLANEOUS

This chapter contains the descriptions of miscellaneous operations.

In the current revision of **ORKID** these are restricted to address translation operations. These operations translate addresses of multi-ported memory from local processor addresses to the corresponding addresses on other ports and vice-versa.