4.6. TASK_SUSPEND

Suspend a task.

Synopsis

```
task_suspend( tid )
```

Input Parameters

tid	: task_id	kernel defined task identifier

Output Parameters

<none>

Literal Values

tid	= SELF	the calling task suspends itself.

Completion Status

OK	task_suspend successful
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_ID	task does not exist
OBJECT_DELETED	originally existing task has been deleted before operation
OBJECT_PROTECTED	task in NOPREEMPT mode
TASK_ALREADY_SUSPENDED	task already suspended
NODE_NOT_REACHABLE	node on which task resides is not reachable

Description

This operation temporarily suspends the specified task until the suspension is lifted by a call to task_resume. While it is suspended, a task cannot be scheduled to run.

If the task's active mode has the parameter NOPREEMPT set the operation will fail and return the completion status OBJECT_PROTECTED, unless the task suspends itself. In which case the operation will always be successful.

4.7. TASK_RESUME

Resume a suspended task.

Synopsis

```
task_resume( tid )
```

Input Parameters

tid	: task_id	kernel defined task identifier

Output Parameters

<none>

Completion Status

OK	task_resume successful
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_ID	task does not exist
OBJECT_DELETED	originally existing task has been deleted before operation
TASK_NOT_SUSPENDED	task not suspended
NODE_NOT_REACHABLE	node on which task resides is not reachable

Description

The task_resume operation lifts the task's suspension immediately after the point at which it was suspended. The task must have been suspended with a call to the task_suspend operation.

4.8. TASK_SET_PRIORITY

Set priority of a task.

Synopsis

```
   task_set_priority( tid, new_prio, old_prio )
```

Input Parameters

tid	: task_id	kernel defined task id
new_prio	: integer	task's new priority

Output Parameters

old_prio	: integer	task's previous priority

Literal Values

tid	= SELF	the calling task sets its own priority.
new_prio	= CURRENT	there will be no change in priority.

Completion Status

OK	task_set_priority successful
ILLEGAL_USE	task_set_priority not callable from ISR
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_ID	task does not exist
OBJECT_DELETED	originally existing task has been deleted before operation
INVALID_PRIORITY	invalid priority value
NODE_NOT_REACHABLE	node on which task resides is not reachable

Description

This operation sets the priority of the specified task to new_prio. The new_prio parameter is specified as CURRENT if the calling task merely wishes to find out the current value of the specified task's priority (see also 4. Task Priority).

4.9. TASK_SET_MODE

Set mode of own task.

Synopsis

```
task_set_mode( new_mode, mask, old_mode )
```

Input Parameters

new_mode	: bit_field	new task mode settings
mask	: bit_field	significant bits in mode

Output Parameters

old_mode	: bit_field	task's previous mode

Literal Values

new_mode	+ NOXSR	XSRs cannot be activated
	+ NOTERMINATION	task cannot be restarted or deleted
	+ NOPREEMPT	task cannot be preempted
	+ NOINTERRUPT	task cannot be interrupted
	= ZERO	no mode parameter set
old_mode		same as new_mode
mask	+ NOXSR	change XSR mode bit
	+ NOTERMINATION	change NOTERMINATION mode bit
	+ NOPREEMPT	change NOPREEMPT mode bit
	+ NOINTERRUPT	change NOINTERRUPT mode bit
	= ALL	change all mode bits
	= ZERO	change no mode bits

Completion Status

OK	task_set_mode successful
ILLEGAL_USE	task_set_mode not callable from ISR
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_MODE	invalid mode or mask value

Description

This operation sets a new active mode for the task or its XSR. If called from a task's XSR then the XSR mode is changed, otherwise the main task's mode is changed.

The mode parameters which are to be changed are given in mask. If a parameter is to be set then it is also given in mode, otherwise it is left out. For both mask and mode, the logical OR (!) of the symbolic values for the mode parameters are passed to the operation.

For example, to clear NOINTERRUPT and set NOPREEMPT, mask = NOINTERRUPT ! NOPREEMPT, and mode = NOPREEMPT. To return the current mode without altering it, the mask should simply be set to ZERO.

4.10. TASK_READ_NOTE_PAD

Read one of a task's note-pad locations.

Synopsis

```
   task_read_note_pad( tid, loc_number, loc_value )
```

Input Parameters

tid	:	task_id	kernel defined task id
loc_number	:	integer	note-pad location number

Output Parameters

loc_value	:	word	note-pad location value

Literal Values

tid	=	SELF	the calling task reads its own note-pad

Completion Status

OK	task_read_note_pad successful
INVALID_PARAMETER	a parameter refers to an invalid address
INVALID_ID	task does not exist
OBJECT_DELETED	originally existing task has been deleted before operation
INVALID_LOCATION	note-pad number does not exist
NODE_NOT_REACHABLE	node on which task resides is not reachable

Description

This operation returns the value contained in the specified note-pad location of the task identified by tid (see also 4. Task Note-Pads). **ORKID** compliant kernels have a minimum of 16 note-pad locations, indexed via loc_number starting at one.