12

Partition Manager

12.1 Introduction

The **partition manager** provides facilities to dynamically allocate memory in fixed-size units. The directives provided by the partition manager are:

Name	Directive Description
partition_create	Create a partition
partition_ident	Get ID of a partition
partition_delete	Delete a partition
partition_get_buffer	Get buffer from a partition
partition_return_buffer	Return buffer to a partition

12.2 Background

12.2.1 Definitions

A **partition** is a physically contiguous memory area divided into fixed-size buffers that can be dynamically allocated and deallocated.

Partitions are managed and maintained as a list of buffers. Buffers are obtained from the front of the partition's free buffer chain and returned to the rear of the same chain. When a buffer is on the free buffer chain, **RTEMS** uses eight bytes of each buffer as the free buffer chain. When a buffer is allocated, the entire buffer is available for application use. Therefore, modifying memory that is outside of an

allocated buffer could destroy the free buffer chain or the contents of an adjacent allocated buffer.

12.2.2 Building an Attribute Set

In general, an attribute set is built by a bitwise OR of the desired attribute components. The set of valid partition attributes is provided in the following table:

CONSTANT	DESCRIPTION	DEFAULT
LOCAL	local task	*
GLOBAL	global task	

Attribute values are specifically designed to be mutually exclusive, therefore bitwise OR and addition operations are equivalent as long as each attribute appears exactly once in the component list. An attribute listed as a default is not required to appear in the attribute list, although it is a good programming practice to specify default attributes. If all defaults are desired, the attribute **DEFAULT_ATTRIBUTES** should be specified on this call. The **attribute_set** parameter should be **GLOBAL** to indicate that the partition is to be known globally.

12.3 Operations

12.3.1 Creating a Partition

The **partition_create** directive creates a partition with a user-specified name. The partition's name, starting address, length and buffer size are all specified to the **partition_create** directive. **RTEMS** allocates a **Partition Control Block** (PTCB) from the PTCB free list. This data structure is used by **RTEMS** to manage the newly created partition. The number of buffers in the partition is calculated based upon the specified partition length and buffer size, and returned to the calling task along with a unique partition ID.

12.3.2 Obtaining Partition IDs

When a partition is created, **RTEMS** generates a unique partition ID and assigned it to the created partition until it is deleted. The partition ID may be obtained by either of two methods. First, as the result of an invocation of the **partition_create** directive, the partition ID is stored in a user provided location. Second, the partition ID may be obtained later using the **partition_ident** directive. The partition ID is used by other partition manager directives to access this partition.

12.3.3 Acquiring a Buffer

A buffer can be obtained by calling the **partition_get_buffer** directive. If a buffer is available, then it is returned immediately with a successful return code. Otherwise, an unsuccessful return code is returned immediately to the caller. Tasks cannot block to wait for a buffer to become available.

12.3.4 Releasing a Buffer

Buffers are returned to a partition's free buffer chain with the **partition_return_buffer** directive. This directive returns an error status code if the returned buffer was not previously allocated from this partition.

12.3.5 Deleting a Partition

The **partition_delete** directive allows a partition to be removed and returned to **RTEMS**. When a partition is deleted, the PTCB for that partition is returned to the PTCB free list. A partition with buffers still allocated cannot be deleted. Any task attempting to do so will be returned an error status code.

12.4 Directives

This section details the partition manager's directives. A subsection is dedicated to each of this manager's directives and describes the calling sequence, related constants, usage, and status codes.

12.4.1 PARTITION_CREATE - Create a partition

CALLING SEQUENCE:

```
procedure partition_create (
   name             : in     rtems.name;
   starting_address : in     rtems.address;
   length           : in     rtems.unsigned32;
   buffer_size      : in     rtems.unsigned32;
   attribute_set    : in     rtems.attribute;
   id               :    out rtems.id;
   result           :    out rtems.status_codes
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	partition created successfully
INVALID_NAME	invalid task name
TOO_MANY	too many partitions created
INVALID_ADDRESS	address not on four byte boundary
INVALID_SIZE	length or buffer size is 0
INVALID_SIZE	length is less than the buffer size
INVALID_SIZE	buffer size not a multiple of 4
MP_NOT_CONFIGURED	multiprocessing not configured
TOO_MANY	too many global objects

DESCRIPTION:

This directive creates a partition of fixed size buffers from a physically contiguous memory space which starts at **starting_address** and is **length** bytes in size. Each allocated buffer is to be of **buffer_length** in bytes. The assigned partition id is returned in **id**. This partition id is used to access the partition with other partition related directives. For control and maintenance of the partition, **RTEMS** allocates a PTCB from the local PTCB free pool and initializes it.

NOTES:

This directive will not cause the calling task to be preempted.

The **starting_address** and **buffer_size** parameters must be multiples of four.

Memory from the partition is not used by **RTEMS** to store the Partition Control Block.

The following partition attribute constants are defined by **RTEMS**:

CONSTANT	DESCRIPTION	DEFAULT
LOCAL	local task	*
GLOBAL	global task	

The PTCB for a global partition is allocated on the local node. The memory space used for the partition must reside in shared memory. Partitions should not be made global unless remote tasks must interact with the partition. This is to avoid the overhead incurred by the creation of a global partition. When a global partition is created, the partition's name and id must be transmitted to every node in the system for insertion in the local copy of the global object table.

The total number of global objects, including partitions, is limited by the **maximum_global_objects** field in the **Configuration Table**.

12.4.2 PARTITION_IDENT - Get ID of a partition

CALLING SEQUENCE:

```
procedure partition_ident (
   name   : in     rtems.name;
   node   : in     rtems.unsigned32;
   id     :    out rtems.id;
   result :    out rtems.status_codes
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	partition identified successfully
INVALID_NAME	partition name not found
INVALID_NODE	invalid node id

DESCRIPTION:

This directive obtains the partition **id** associated with the partition **name**. If the partition name is not unique, then the partition id will match one of the partitions with that name. However, this partition id is not guaranteed to correspond to the desired partition. The partition id is used with other partition related directives to access the partition.

NOTES:

This directive will not cause the running task to be preempted.

If **node** is **SEARCH_ALL_NODES**, all nodes are searched with the local node being searched first. All other nodes are searched with the lowest numbered node searched first.

If **node** is a valid node number which does not represent the local node, then only the partitions exported by the designated node are searched.

This directive does not generate activity on remote nodes. It accesses only the local copy of the global object table.

12.4.3 PARTITION_DELETE - Delete a partition

CALLING SEQUENCE:

```
procedure partition_delete (
   id     : in     rtems.id;
   result :    out rtems.status_codes
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	partition deleted successfully
INVALID_ID	invalid partition id
RESOURCE_IN_USE	buffers still in use
ILLEGAL_ON_REMOTE_OBJECT	cannot delete remote partition

DESCRIPTION:

This directive deletes the partition specified by **id**. The partition cannot be deleted if any of its buffers are still allocated. The PTCB for the deleted partition is reclaimed by **RTEMS**.

NOTES:

This directive will not cause the calling task to be preempted.

The calling task does not have to be the task that created the partition. Any local task that knows the partition id can delete the partition.

When a global partition is deleted, the partition id must be transmitted to every node in the system for deletion from the local copy of the global object table.

The partition must reside on the local node, even if the partition was created with the **GLOBAL** option.

12.4.4 PARTITION_GET_BUFFER - Get buffer from a partition

CALLING SEQUENCE:

```
procedure partition_get_buffer (
   id     : in     rtems.id;
   buffer :    out rtems.address;
   result :    out rtems.status_codes
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	buffer obtained successfully
INVALID_ID	invalid partition id
UNSATISFIED	all buffers are allocated

DESCRIPTION:

This directive allows a buffer to be obtained from the partition specified in **id**. The address of the allocated buffer is returned in **buffer**.

NOTES:

This directive will not cause the running task to be preempted.

All buffers begin on a four byte boundary.

A task cannot wait on a buffer to become available.

Getting a buffer from a global partition which does not reside on the local node will generate a request telling the remote node to allocate a buffer from the specified partition.

12.4.5 PARTITION_RETURN_BUFFER - Return buffer to a partition

CALLING SEQUENCE:

```
procedure partition_return_buffer (
   id     : in     rtems.id;
   buffer : in     rtems.address;
   result :    out rtems.status_codes
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	buffer returned successfully
INVALID_ID	invalid partition id
INVALID_ADDRESS	buffer address not in partition

DESCRIPTION:

This directive returns the buffer specified by **buffer** to the partition specified by **id**.

NOTES:

This directive will not cause the running task to be preempted.

Returning a buffer to a global partition which does not reside on the local node will generate a request telling the remote node to return the buffer to the specified partition.