

3.7 Memory Management

The executive will support two different memory managers. A region manager provides allocation of variable sized memory segments. A partition manager provides allocation of fixed sized buffers.

3.7.1 Region Manager

A region is an area of physical contiguous memory from which the executive can dynamically allocate segments to an application. A segment is a variable length block of memory.

A region is created with the *rn_create* directive. Like all objects managed by the executive, a region has a 4 character name, and, once created, a 32-bit region id (*rnid*). Tasks other than the creator can use the *rn_ident* directive to obtain a region's *rnid*. The directives *rn_getseg* and *rn_retseg* allocate and return segments from the region.

Each region has an associated *pagesize*, specified when the region is created. The *pagesize* must be a power of 2. Segment lengths are always in multiples of this *pagesize*. For example, if a task requests a 700 byte segment from a region having a 512 byte *pagesize*, a 1024 byte segment is allocated.

When requesting a segment, if the request cannot immediately be satisfied, the requesting task may optionally wait (with or without timeout) for a segment to become available. If it elects to wait, the task is placed in a memory wait queue associated with the region. Tasks can be queued either by priority or FIFO. When a segment is returned, if possible it is merged with its neighbor segments and then the wait queue is searched. The first task, if any, whose request can be satisfied receives the segment.

In a multiprocessor system, regions may not be shared between processors. Segments may only be allocated or returned by tasks running on the processor from which the region was created. Hence, the GLOBAL flag used with the other create services is not supported by *rn_create*.

When a region is created, the executive must build data structures to manage the region. The memory containing these structures may itself be allocated from the region, in which case, the amount of allocatable memory within the region may be slightly less than the original size of the region.

The maximum number of regions that may exist at any one time is a configuration parameter.

The directives provided by the region manager are:

Directive	Function
<i>rn_create</i>	Create a region
<i>rn_ident</i>	Obtain id of a region
<i>rn_delete</i>	Delete a region
<i>rn_getseg</i>	Get a segment
<i>rn_retseg</i>	Return a segment

3.7.2 Partition Manager

A partition is a pool of equal sized buffers. *Pt_create* creates a partition in a physical contiguous memory area provided by the caller. Like all objects managed by the executive, partitions have a 4 character name, and, once created, a 32-bit partition id (*ptid*). Tasks other than the creator can use the *pt_ident* directive to obtain a partition's *ptid*. *Pt_getbuf* and *pt_retbuf* allocate and return buffers from the partition.

Each partition contains a specified number of fixed size buffers. The number and size of the buffers is specified when the partition is created.

In a shared memory multiprocessor configuration, partitions may be shared between processors. To do so, the caller must declare the partition GLOBAL when it is created. If a partition is GLOBAL, then the executive will arbitrate access to the partition.

Tasks may not wait for buffers. If no buffers are available an error number is returned.

When a partition is created, the executive must build data structures to manage the partition. The memory containing these structures may be allocated within the partition area provided by the caller, in which case, the partition may occupy slightly more memory than the simple product of the buffer count and buffer size.

The maximum number of partitions that may exist at any one time is a configuration parameter.

The directives provided by the partition manager are:

Directive	Function
<i>pt_create</i>	Create a partition
<i>pt_ident</i>	Obtain id of a partition
<i>pt_delete</i>	Delete a partition
<i>pt_getbuf</i>	Get a buffer
<i>pt_retbuf</i>	Return a buffer

3.7.3 RN_CREATE

NAME

`rn_create` - "Create a Region"

SYNOPSIS

```
#include <memory.h>
uint rn_create ( name, paddr, length, pagesize, flags, &rnid, &bytes )

    uint name;          /* user defined 4-byte region name */
    char *paddr;       /* physical start address of region */
    uint length;       /* physical length in bytes */
    uint pagesize;     /* region pagesize */
    uint flags;        /* region attributes */
    uint rnid;         /* region id - returned by this call */
    uint bytes;        /* available number of bytes - returned by this call */
```

The flags field values are:

PRIOR	set	to process wait list by priority
	clear	to process wait list by FIFO

DESCRIPTION

This directive allows the user to create a region from a physical contiguous memory area. The region id will be returned in `rnid` by the executive to use in `rn_getseg` and `rn_retseg` directives for the region.

The region physical start address specified in `paddr` will be long-word aligned by the executive. In systems with an MMU, the region physical start address must be on the `pagesize` boundary.

The available number of bytes within the region will be returned by the executive in the `bytes` field. Since the executive may use memory within the region for a region data structure, the number of bytes in `bytes` may be less than the number of bytes in `length`.

By setting the `PRIOR` value in the flags field, tasks which wait for segments from the region will be processed in task priority order. Otherwise, the tasks will wait in first in, first out (FIFO) order.

Regions may not be shared between processors in a shared memory multiprocessor configuration.

The maximum number of regions that can be in existence at one time is a configuration parameter.

RETURN VALUE

If `rn_create` successfully created the region, then `rnid` and `bytes` are filled in and 0 is returned.

If the region was not successfully created, an error code is returned.

ERROR CONDITIONS

Too many regions.

Paddr is not on a pagesize boundary (MMU only).

NOTES

Not callable from ISR.

3.7.4 RN_IDENT

NAME

`rn_ident` -- "Obtain id of a Region"

SYNOPSIS

```
#include <memory.h>
uint rn_ident ( name, &rnid )
```

```
uint name; /* user defined 4-byte region name */
uint rnid; /* region id - returned by this call */
```

DESCRIPTION

This directive allows a task to identify a previously created region by name, and obtain the `rnid` to use for `rn_getseg` and `rn_retseg` directives for the region.

The region must have been created by a task on the local processor. It may not be shared between processors in a shared memory multiprocessor configuration.

If the region name is not unique, the region id returned in `rnid` may not correspond to the region named by this call.

RETURN VALUE

If `rn_ident` directive succeeds, then the `rnid` is filled in and 0 is returned.

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

ERROR CONDITIONS

Named region does not exist.

NOTES

Can be called from within an ISR.

Will not cause a preempt.