3.7.5 RN_DELETE

NAME

rn_delete -- "Delete a Region"

SYNOPSIS

```
#include <memory.h>
uint rn_delete ( rnid )


            uint rnid;     /* region id as returned by rn_create or rn_ident */
```

DESCRIPTION

This directive deletes the specified region, provided that none of its segments is still allocated. After this directive has successfully executed, the executive will reject any *rn_getseg* and *rn_retseg* directives for the region.

RETURN VALUE

If *rn_delete* successfully deleted the region, then 0 is returned.

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

ERROR CONDITIONS

Invalid *rnid*.

Cannot delete -- outstanding segments.

NOTES

Not callable from ISR.

Will not cause a preempt.

3.7.6 RN_GETSEG

NAME

rn_getseg -- "Get a Segment"

SYNOPSIS

```
#include <memory.h>
uint rn_getseg ( rnid, size, flags, timeout, &segaddr )

        uint rnid;        /* region id as returned by rn_create or rn_ident */
        uint size;        /* segment size in bytes */
        uint flags;       /* directive options */
        uint timeout;     /* number of ticks to wait for memory */
                          /* 0 indicates forever */
        char *segaddr;    /* segment address - returned by this call */
```

The flags field values are defined as follows:

NOWAIT	set	if the task is to return immediately
	clear	if the task is to wait for memory

DESCRIPTION

This directive allocates a variable size segment from the region specified by the *rnid*. The address of the segment is returned to the caller in *segaddr*.

The actual segment length is a multiple of the region pagesize. Thus, the segment allocated may be larger than the requested size.

RETURN VALUE

If *rn_getseg* successfully allocated the segment, the address of the segment is returned in *segaddr* and 0 is returned.

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

ERROR CONDITIONS

Invalid *rnid*.

No memory available (no-wait only).

Timeout occurred before memory was available (wait with timeout).

Region has been deleted.

NOTES

Not callable from ISR.

Requester will be blocked when the wait option is selected and the memory is not available.

3.7.7 RN_RETSEG

NAME

rn_retseg -- "Return a Segment"

SYNOPSIS

```
#include <memory.h>
uint rn_retseg ( rnid, segaddr )


          uint rnid;        /* region id as returned by rn_create or rn_ident */
          char *segaddr;    /* segment address as returned by rn_getseg */
```

DESCRIPTION

This directive returns a segment to its region. If possible, the segment is merged with neighboring segments. The resulting segment then becomes available for subsequent allocation, or allocation to tasks already waiting.

RETURN VALUE

If *rn_retseg* successfully returned the segment, then 0 is returned.

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

ERROR CONDITIONS

Invalid *rnid*.

Segment not from specified region.

NOTES

Not callable from ISR.

May cause a preempt if a task waiting for memory becomes ready as a result of this call and has a higher priority than the running task, and the preempt mode is in effect.

3.7.8 PT_CREATE

NAME

pt_create -- "Create a Partition"

SYNOPSIS

```
#include <memory.h>
uint pt_create ( name, paddr, length, bsise, flags, &ptid, &bnum)


        uint name;      /* user defined 4-byte partition name */
        char *paddr;    /* physical start address of partition */
        uint length;    /* physical length in bytes */
        uint bsise;     /* sise of buffers in bytes */
        uint flags;     /* partition attributes */
        uint ptid;      /* partition id - returned by this call */
        uint bnum;      /* number of buffers in partition - returned by this call */
```

Flags field values:

GLOBAL	set	to indicate the partition is a multiprocessor global resource.
	clear	to indicate the partition is local

DESCRIPTION

This directive allows the user to create a partition of fixed size buffers from a contiguous memory area. The partition id will be returned in *ptid* by the executive to use for *pt_getbuf* and *pt_retbuf* directives for the partition. The number of buffers created by the executive will be returned in *bnum*.

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

The executive may use memory within the partition for partition and buffer data structures. Therefore, the product of the buffer count and buffer size will be slightly less than the length of the partition.

By setting the GLOBAL value in the flags field, the *ptid* will be sent to all processors in the system, to be entered into a global resource table. The system is defined as the collection of interconnected processors.

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

RETURN VALUE

If *pt_create* successfully created the partition, the *ptid* and *bnum* are filled in and 0 is returned.