Duplicate logical address.

Task not created on local node.

ISR cannot reference remote node.

NOTES

Can be called from within an ISR, except when the task was not created on the local node.

Will not cause a preempt.

3.8.7 MM_UNMAP

NAME

mm_unmap -- "Unmap Logical"

SYNOPSIS

```
#include <memory.h>
uint mm_unmap ( tid, laddr )


                uint tid;        /* task id as returned by t_create or t_ident */
                char *laddr;     /* logical start address */
```

DESCRIPTION

This directive removes the section starting at logical address *laddr* from the address space of the task identified by the *tid*.

RETURN VALUE

If *mm_unmap* was successful, then 0 is returned.

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

ERROR CONDITIONS

Invalid *tid*.

Unmapped logical address.

Task not created on local node.

ISR cannot reference remote node.

NOTES

Can be called from within an ISR, except when the task was not created on the local node.

Will not cause a preempt.

To return the segment to the region, the *rn_retseg* directive must be used.

3.8.8 MM_PREAD

NAME

mm_pread – "Physical read"

SYNOPSIS

```
#include <memory.h>
uint mm_pread ( paddr, laddr, length )

                    uint paddr;     /* physical start address */
                    char *laddr;    /* logical start address */
                    uint length;    /* length in bytes */
```

DESCRIPTION

The *mm_pread* directive reads from a physical address, and writes to the logical address in the calling task's address space. The length cannot span a section boundary.

RETURN VALUE

If *mm_pread* was successful, then 0 is returned.

If the call was not successful, no data is transferred and an error code is returned.

ERROR CONDITIONS

Unmapped logical address.

Length spans section boundary.

NOTES

Not callable from ISR.

Will not cause a preempt.

3.8.9 MM_PWRITE

NAME

mm_pwrite -- "Physical write"

SYNOPSIS

```
#include <memory.h>
uint mm_pwrite ( paddr, laddr, length )

        uint paddr;     /* physical start address */
        char *laddr;    /* logical start address */
        uint length;    /* length in bytes */
```

DESCRIPTION

The *mm_pwrite* directive reads from the logical address in the calling task's address space, and writes to a physical address. The length may not span a section boundary.

RETURN VALUE

If *mm_pwrite* was successful, then 0 is returned.

If the call was not successful, no data is transferred and an error code is returned.

ERROR CONDITIONS

Unmapped logical address.

Length spans section boundary.

NOTES

Not callable from ISR.

Will not cause a preempt.

3.8.10 MM_PTCREATE

NAME

mm_ptcreate -- "Create a Logical Partition"

SYNOPSIS

```
#include <memory.h>
uint mm_ptcreate ( name, paddr, length, bsize, laddr, 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 bsize;      /* size of buffers in bytes */
        char *laddr;     /* physical start address of partition */
        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 logical partition of fixed size buffers from a contiguous memory area. The partition is mapped into the caller's address space at the logical address specified in *laddr*. By creating logical partitions at the same logical addresses, partitions can be easily shared between processors.

The partition id will be returned in *ptid* by the executive to use for *pt_getbuf* and *pt_retbuf* directives for the partition.

The partition physical start address must be on the pagesize boundary.

The number of buffers created by the executive will be returned in *bnum*. 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