NAME

ddi_mem_alloc, ddi_mem_free – allocate and free sequentially accessed memory

SYNOPSIS

#include <sys/ddi.h>
#include <sys/sunddi.h>

int ddi_mem_alloc(dev_info_t **dip*, **ddi_dma_lim_t** **limits*, **u_int** *length*, **u_int** *flags*, **caddr_t** **kaddrp*, **u_int** **real_length*);

void ddi_mem_free(caddr_t *kaddr*);

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

ddi_mem_alloc( )

dip	A pointer to the device's **dev_info** structure.
limits	A pointer to a DMA limits structure for this device (see **ddi_dma_lim_sparc**(9S) or **ddi_dma_lim_x86**(9S)). If this pointer is **NULL,** a default set of DMA limits is assumed.
length	The length in bytes of the desired allocation.
flags	The possible flags **1** and **0** are taken to mean, respectively, wait until memory is available, or do not wait.
kaddrp	On a successful return, **kaddrp* points to the allocated memory.
real_length	The length in bytes that was allocated. Alignment and padding requirements may cause **ddi_mem_alloc( )** to allocate more memory than requested in *length*.

ddi_mem_free( )

kaddr	The memory returned from a successful call to **ddi_mem_alloc( )**.

DESCRIPTION

ddi_mem_alloc( ) allocates memory for DMA transfers and should be used if the device is performing sequential, unidirectional, block-sized and block-aligned transfers to or from memory. This type of access is commonly known as *steaming* access. The allocation will obey the alignment and padding constraints as specified by the *limits* argument and other limits imposed by the system.

Note that you must still use DMA resource allocation functions (see **ddi_dma_setup**(9F)) to establish DMA resources for the memory allocated using **ddi_mem_alloc( )**. **ddi_mem_alloc( )** returns the actual size of the allocated memory object. Because of padding and alignment requirements, the actual size might be larger than the requested size. **ddi_dma_setup**(9F) requires the actual length.

In order to make the view of a memory object shared between a CPU and a DMA device consistent, explicit synchronization steps using **ddi_dma_sync**(9F) or **ddi_dma_free**(9F) are required.

ddi_mem_free( ) frees up memory allocated by **ddi_mem_alloc( )**.

RETURN VALUES

ddi_mem_alloc( ) returns:

DDI_SUCCESS Memory successfully allocated.

DDI_FAILURE Allocation failed.

CONTEXT

ddi_mem_alloc( ) can be called from user or interrupt context, except when *flags* is set to **1**, in which case it can be called from user context only.

SEE ALSO

ddi_dma_free(9F), **ddi_dma_setup**(9F), **ddi_dma_sync**(9F), **ddi_iopb_alloc**(9F), **ddi_dma_lim_sparc**(9S), **ddi_dma_lim_x86**(9S), **ddi_dma_req**(9S)

Writing Device Drivers