NAME

rmallocmap, rmallocmap_wait, rmfreemap – allocate and free resource maps

SYNOPSIS

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

struct map ∗rmallocmap(unsigned long *mapsize***);**

struct map ∗rmallocmap_wait(unsigned long *mapsize***);**

void rmfreemap(struct map ∗*mp***);**

ARGUMENTS

mapsize Number of entries for the map.

mp A pointer to the map structure to be deallocated.

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

DESCRIPTION

rmallocmap( ) dynamically allocates a resource map structure. The argument *mapsize* defines the total number of entries in the map. In particular it is the total number allocations that can be outstanding at any one time.

rmallocmap( ) initializes the map but does not associate it with the actual resource. In order to associate the map with the actual resource a call to **rmfree**(9F) is used to make the entirety of the actual resource available for allocation starting from the first index into the resource. Typically the call to **rmallocmap( )** is followed by a call to **rmfree**(9F)**,** passing the address of the map returned from **rmallocmap( )**, the total size of the resource, and the first index into actual resource.

The resource map allocated by **rmallocmap( )** can be used to describe an arbitrary resource in whatever allocation units are appropriate such blocks, pages, or data structures. This resource can then be managed by the system by subsequent calls to **rmalloc**(9F)**, rmalloc_wait**(9F)**,** and **rmfree**(9F)**.**

rmallocmap_wait( ) is similar to **rmallocmap( )**, with the exception that it will wait for space to become available if necessary.

rmfreemap( ) deallocates a resource map structure previously allocated by **rmallocmap( )** or **rmallocmap_wait( )**. The argument *mp* is a pointer to the map structure to be deallocated.

RETURN VALUES

Upon successful completion, **rmallocmap( )** and **rmallocmap_wait( )** return a pointer to the newly allocated map structure. Upon failure, **rmallocmap( )** returns a NULL pointer.

CONTEXT

rmallocmap( ) and **rmfreemap( )** can be called from user, kernel or interrupt context.

rmallocmap_wait( ) can only be called from user or kernel context.

SEE ALSO

rmalloc(9F), **rmalloc_wait**(9F), **rmfree**(9F)

Writing Device Drivers