NAME

mctl – memory management control

SYNOPSIS

/usr/ucb/cc [*flag* ...] *file* ...

#include <sys/types.h>
#include <sys/mman.h>

int mctl(*addr, len, function, arg* **)**
caddr_t *addr***;**
size_t *len***;**
int *function***;**
int *arg***;**

DESCRIPTION

mctl( ) applies a variety of control functions over pages identified by the mappings established for the address range [*addr, addr* + *len*). The function to be performed is identified by the argument *function*. Valid functions are defined in **mman.h** as follows:

MC_LOCK
Lock the pages in the range in memory. This function is used to support **mlock( )**. See **mlock**(3C) for semantics and usage. *arg* is ignored.

MC_LOCKAS
Lock the pages in the address space in memory. This function is used to support **mlockall( )**. See **mlockall**(3C) for semantics and usage. *addr* and *len* are ignored. *arg* is an integer built from the flags:

MCL_CURRENT	Lock current mappings
MCL_FUTURE	Lock future mappings

MC_SYNC
Synchronize the pages in the range with their backing storage. Optionally invalidate cache copies. This function is used to support **msync( )**. See **msync**(3C) for semantics and usage. *arg* is used to represent the *flags* argument to **msync( )**. It is constructed from an OR of the following values:

MS_SYNC	Synchronized write
MS_ASYNC	Return immediately
MS_INVALIDATE	Invalidate mappings

MS_ASYNC returns after all I/O operations are scheduled. **MS_SYNC** does not return until all I/O operations are complete. Specify exactly one of **MS_ASYNC** or **MS_SYNC**. **MS_INVALIDATE** invalidates all cached copies of data from memory, requiring them to be re-obtained from the object's permanent storage location upon the next reference.

MC_UNLOCK
Unlock the pages in the range. This function is used to support **munlock( )**. *arg* is ignored.

MC_UNLOCKAS
Remove address space memory lock, and locks on all current mappings. This function is used to support **munlockall( )**. *addr* and *len* must have the value 0. *arg* is ignored.

RETURN VALUES

mctl( ) returns 0 on success, –1 on failure.

ERRORS

mctl( ) fails if:

EAGAIN	Some or all of the memory identified by the operation could not be locked due to insufficient system resources.
EBUSY	**MS_INVALIDATE** was specified and one or more of the pages is locked in memory.
EINVAL	*addr* is not a multiple of the page size as returned by **getpagesize( )**.
EINVAL	*addr* and/or *len* do not have the value 0 when **MC_LOCKAS** or **MC_UNLOCKAS** are specified.
EINVAL	*arg* is not valid for the function specified.
EIO	An I/O error occurred while reading from or writing to the file system.
ENOMEM	Addresses in the range [*addr*, *addr* + *len*) are invalid for the address space of a process, or specify one or more pages which are not mapped.
EPERM	The process's effective user ID is not super-user and one of **MC_LOCK** **MC_LOCKAS**, **MC_UNLOCK**, or **MC_UNLOCKAS** was specified.

SEE ALSO

mmap(2), **memcntl**(2), **getpagesize**(3C), **mlock**(3C), **mlockall**(3C), **msync**(3C)

NOTES

Use of these interfaces should be restricted to only applications written on BSD platforms. Use of these interfaces with any of the system libraries or in multi-thread applications is unsupported.