NAME

mincore - determine residency of memory pages

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/mman.h>

int
mincore(*const void *addr*, *size_t len*, *char *vec*);

DESCRIPTION

The **mincore**() system call determines whether each of the pages in the region beginning at *addr* and continuing for *len* bytes is resident or mapped, depending on the value of sysctl *vm.mincore_mapped*. The status is returned in the *vec* array, one character per page. Each character is either 0 if the page is not resident, or a combination of the following flags (defined in <*sys/mman.h*>):

MINCORE_INCORE	Page is in core (resident).
MINCORE_REFERENCED	Page has been referenced by us.
MINCORE_MODIFIED	Page has been modified by us.
MINCORE_REFERENCED_OTHER	Page has been referenced.
MINCORE_MODIFIED_OTHER	Page has been modified.
MINCORE_PSIND(i)	Page is part of a large ("super") page with size given by index i in the array returned by getpagesizes(3).
MINCORE_SUPER	A mask of the valid MINCORE_PSIND() values. If any bits in this mask are set, the page is part of a large ("super") page.

The information returned by **mincore**() may be out of date by the time the system call returns. The only way to ensure that a page is resident is to lock it into memory with the mlock(2) system call.

If the *vm.mincore_mapped* sysctl is set to a non-zero value (default), only the current process' mappings of the pages in the specified virtual address range are examined. This does not preclude the system from returning MINCORE_REFERENCED_OTHER and MINCORE_MODIFIED_OTHER statuses.

Otherwise, if the sysctl value is zero, all resident pages backing the specified address range are examined, regardless of the mapping state.

IMPLEMENTATION NOTES

Prior to the introduction of MINCORE_PSIND() in FreeBSD 13.0, MINCORE_SUPER consisted of a single bit equal to MINCORE_PSIND(1). In particular, applications compiled using the old value of MINCORE_SUPER will not identify large pages with size index 2 as being large pages.

RETURN VALUES

The **mincore**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **mincore**() system call will fail if:

[ENOMEM]	The virtual address range specified by the *addr* and *len* arguments is not fully mapped.
[EFAULT]	The *vec* argument points to an illegal address.

SEE ALSO

madvise(2), mlock(2), mprotect(2), msync(2), munmap(2), getpagesize(3), getpagesizes(3)

HISTORY

The **mincore**() system call first appeared in 4.4BSD.