NAME

mwritten - track which pages in memory are modified over time.

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/mman.h>

int
mwritten(*void *addr*, *size_t len*, *int flags*, *void *buf*, *size_t *naddr*, *size_t *gran*);

DESCRIPTION

The **mwritten**() system call returns a list of pages modified since write tracking was cleared, or if never cleared then since allocation, in the region starting at *addr* and continuing for *len* bytes. If the specified region does not start and end on page boundaries, **mwritten**() will query between the nearest page boundaries which contain the region. The addresses of writes to memory are placed in *buf* and upon successful return *naddr* is updated to indicate the number of addresses outputted. The value of *gran* will also be updated to indicate the granularity of write tracking. Hence, for each address outputted to *buf*, the current process wrote to somewhere between that address and the next *gran* bytes onward. Currently, *gran* is always equal to system page size.

If there are so many writes that *buf* fills up, then **mwritten**() returns as soon as it does so. Callers can detect this condition if the function call succeeds but the value of *naddr* is unchanged, and in this case the final address which was queried or cleared before returning can be found in the last outputted adddress, *buf[naddr - 1]*. To ensure all modifications in the region are found, and all written statuses cleared if requested, always check the value of *naddr* and call **mwritten**() again starting from the last outputted address if it returned early the first time.

The **mwritten**() system call accepts flags by *or*'ing the following values:

MWRITTEN_DEFAULT	Default behaviour. Returns addresses of modified regions in memory but does not alter their state.
MWRITTEN_CLEAR	Clears the written statuses of all memory in the specified region. Returns the addresses of modified regions in memory if *buf* and *naddr* are both specified and valid. If they are both NULL then **mwritten**() will return no output.
MWRITTEN_NOT_SHARED	Indicates that the memory region provided is never copy-on-written from. This may be faster, but beware that incorrect usage may lead to subtle bugs.

NOTES

This system call is designed primarily for usage within garbage collectors that require efficient write tracking. Whilst it does work on copy-on-written memory, no guarantees are made about its operation on shared memory. Furthermore, the effects of different threads calling **mwritten**() simultaneously are not isolated, and appropriate measures to prevent race conditions are required where necessary.

When clearing writes using the MWRITTEN_CLEAR flag, if the output is not required then setting *buf* and *naddr* to NULL is highly recommended, because this performs significantly faster.

RETURN VALUES

Upon successful completion, **mwritten**() returns 0. Otherwise, a non-zero value is returned and *errno* is set to indicate the error.

ERRORS

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

[EFAULT]	The provided output buffer starting at *buf* and which must be able to hold *naddr* pointers was not legal.
[EFAULT]	The address passed to *gran* was not a valid, allocated virtual address.
[EINVAL]	The start address given in the *addr* argument was not a valid, allocated virtual address.
[EINVAL]	The end address, that is *len* bytes after *addr*, was not a valid, allocated virtual address.
[EINVAL]	The argument *buf* was NULL when the flag MWRITTEN_CLEAR was not specified and the *naddr* argument was not also NULL.
[EINVAL]	The argument *naddr* was NULL when the flag MWRITTEN_CLEAR was not specified and the *buf* argument was not also NULL.
[EINVAL]	Some pages in the region of memory starting at *addr* and extending for *len* bytes onward were fictious or unmanaged.

SEE ALSO

minherit(2), mlock(2), mmap(2), mprotect(2), munmap(2), getpagesize(3), getpagesizes(3)