

**PROLOG**

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

**NAME**

`mlock`, `munlock` — lock or unlock a range of process address space (**REALTIME**)

**SYNOPSIS**

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

int mlock(const void *addr, size_t len);
int munlock(const void *addr, size_t len);
```

**DESCRIPTION**

The `mlock()` function shall cause those whole pages containing any part of the address space of the process starting at address `addr` and continuing for `len` bytes to be memory-resident until unlocked or until the process exits or *execs* another process image. The implementation may require that `addr` be a multiple of {PAGESIZE}.

The `munlock()` function shall unlock those whole pages containing any part of the address space of the process starting at address `addr` and continuing for `len` bytes, regardless of how many times `mlock()` has been called by the process for any of the pages in the specified range. The implementation may require that `addr` be a multiple of {PAGESIZE}.

If any of the pages in the range specified to a call to `munlock()` are also mapped into the address spaces of other processes, any locks established on those pages by another process are unaffected by the call of this process to `munlock()`. If any of the pages in the range specified by a call to `munlock()` are also mapped into other portions of the address space of the calling process outside the range specified, any locks established on those pages via the other mappings are also unaffected by this call.

Upon successful return from `mlock()`, pages in the specified range shall be locked and memory-resident. Upon successful return from `munlock()`, pages in the specified range shall be unlocked with respect to the address space of the process. Memory residency of unlocked pages is unspecified.

Appropriate privileges are required to lock process memory with `mlock()`.

**RETURN VALUE**

Upon successful completion, the `mlock()` and `munlock()` functions shall return a value of zero. Otherwise, no change is made to any locks in the address space of the process, and the function shall return a value of `-1` and set `errno` to indicate the error.

**ERRORS**

The `mlock()` and `munlock()` functions shall fail if:

**ENOMEM**

Some or all of the address range specified by the `addr` and `len` arguments does not correspond to valid mapped pages in the address space of the process.

The `mlock()` function shall fail if:

**EAGAIN**

Some or all of the memory identified by the operation could not be locked when the call was made.

The `mlock()` and `munlock()` functions may fail if:

**EINVAL**

The `addr` argument is not a multiple of {PAGESIZE}.

The `mlock()` function may fail if:

**ENOMEM**

Locking the pages mapped by the specified range would exceed an implementation-defined limit on the amount of memory that the process may lock.



**EPERM**

The calling process does not have appropriate privileges to perform the requested operation.

*The following sections are informative.*

**EXAMPLES**

None.

**APPLICATION USAGE**

None.

**RATIONALE**

None.

**FUTURE DIRECTIONS**

None.

**SEE ALSO**

*exec*, *exit()*, *fork()*, *mlockall()*, *munmap()*

The Base Definitions volume of POSIX.1-2008, <sys\_mman.h>

**COPYRIGHT**

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2013 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, Copyright (C) 2013 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. (This is POSIX.1-2008 with the 2013 Technical Corrigendum 1 applied.) In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.unix.org/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html).

