



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'mprotect.3p' command

\$ man mprotect.3p

MPROTECT(3P) POSIX Programmer's Manual MPROTECT(3P)

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

mprotect ? set protection of memory mapping

SYNOPSIS

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

int mprotect(void *addr, size_t len, int prot);
```

DESCRIPTION

The mprotect() function shall change the access protections to be that specified by prot for those whole pages containing any part of the address space of the process starting at address addr and continuing for len bytes. The parameter prot determines whether read, write, execute, or some combination of accesses are permitted to the data being mapped. The prot argument should be either PROT_NONE or the bitwise-inclusive OR of one or more of PROT_READ, PROT_WRITE, and PROT_EXEC. If an implementation cannot support the combination of access types specified by prot, the call to mprotect() shall fail.

An implementation may permit accesses other than those specified by prot; however, no implementation shall permit a write to succeed where

PROT_WRITE has not been set or shall permit any access where PROT_NONE alone has been set. Implementations shall support at least the following values of prot: PROT_NONE, PROT_READ, PROT_WRITE, and the bitwise-inclusive OR of PROT_READ and PROT_WRITE. If PROT_WRITE is specified, the application shall ensure that it has opened the mapped objects in the specified address range with write permission, unless MAP_PRIVATE was specified in the original mapping, regardless of whether the file descriptors used to map the objects have since been closed.

The implementation may require that addr be a multiple of the page size as returned by sysconf().

The behavior of this function is unspecified if the mapping was not established by a call to mmap().

When mprotect() fails for reasons other than [EINVAL], the protections on some of the pages in the range [addr,addr+len) may have been changed.

RETURN VALUE

Upon successful completion, mprotect() shall return 0; otherwise, it shall return -1 and set errno to indicate the error.

ERRORS

The mprotect() function shall fail if:

EACCESS The prot argument specifies a protection that violates the access permission the process has to the underlying memory object.

EAGAIN The prot argument specifies PROT_WRITE over a MAP_PRIVATE mapping and there are insufficient memory resources to reserve for locking the private page.

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.

ENOMEM The prot argument specifies PROT_WRITE on a MAP_PRIVATE mapping, and it would require more space than the system is able to supply for locking the private pages, if required.

ENOTSUP

The implementation does not support the combination of accesses

requested in the prot argument.

The mprotect() function may fail if:

EINVAL The addr argument is not a multiple of the page size as returned by sysconf().

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Most implementations require that addr is a multiple of the page size as returned by sysconf().

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

mmap(), sysconf()

The Base Definitions volume of POSIX.1-2017, <sys_mman.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. 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.opengroup.org/unix/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.