

**NAME**

**mprotect** - control the protection of pages

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

*int*

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

**DESCRIPTION**

The **mprotect()** system call changes the specified pages to have protection *prot*.

The *prot* argument shall be PROT\_NONE (no permissions at all) or the bitwise *or* of one or more of the following values:

PROT\_READ The pages can be read.

PROT\_WRITE The pages can be written.

PROT\_EXEC The pages can be executed.

In addition to these standard protection flags, the FreeBSD implementation of **mprotect()** provides the ability to set the maximum protection of a region (which prevents **mprotect** from adding to the permissions later). This is accomplished by bitwise *or*'ing one or more PROT\_ values wrapped in the PROT\_MAX() macro into the *prot* argument.

**RETURN VALUES**

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

[EACCES] The calling process was not allowed to change the protection to the value specified by the *prot* argument.

[EINVAL] The virtual address range specified by the *addr* and *len* arguments is not valid.

[EINVAL] The *prot* argument contains unhandled bits.

[ENOTSUP] The *prot* argument contains permissions which are not a subset of the specified maximum permissions.

**SEE ALSO**

madvise(2), mincore(2), msync(2), munmap(2)

**HISTORY**

The **mprotect()** system call was first documented in 4.2BSD and first appeared in 4.4BSD.

The PROT\_MAX functionality was introduced in FreeBSD 13.