



## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sys\_mman.h.0p' command***

**\$ man sys\_mman.h.0p**

sys\_mman.h(0P) POSIX Programmer's Manual sys\_mman.h(0P)

### 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

sys/mman.h ? memory management declarations

### SYNOPSIS

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

### DESCRIPTION

The <sys/mman.h> header shall define the following symbolic constants for use as protection options:

PROT\_EXEC Page can be executed.  
PROT\_NONE Page cannot be accessed.  
PROT\_READ Page can be read.  
PROT\_WRITE Page can be written.

The <sys/mman.h> header shall define the following symbolic constants for use as flag options:

MAP\_FIXED Interpret addr exactly.  
MAP\_PRIVATE Changes are private.  
MAP\_SHARED Share changes.

The <sys/mman.h> header shall define the following symbolic constants

for the `msync()` function:

`MS_ASYNC` Perform asynchronous writes.

`MS_INVALIDATE` Invalidate mappings.

`MS_SYNC` Perform synchronous writes.

The `<sys/mman.h>` header shall define the following symbolic constants

for the `mlockall()` function:

`MCL_CURRENT` Lock currently mapped pages.

`MCL_FUTURE` Lock pages that become mapped.

The `<sys/mman.h>` header shall define the symbolic constant `MAP_FAILED`

which shall have type `void *` and shall be used to indicate a failure from the `mmap()` function .

If the Advisory Information option is supported, the `<sys/mman.h>` header shall define symbolic constants for the `advice` argument to the `posix_madvise()` function as follows:

`POSIX_MADV_DONTNEED`

The application expects that it will not access the specified range in the near future.

`POSIX_MADV_NORMAL`

The application has no advice to give on its behavior with respect to the specified range. It is the default characteristic if no advice is given for a range of memory.

`POSIX_MADV_RANDOM`

The application expects to access the specified range in a random order.

`POSIX_MADV_SEQUENTIAL`

The application expects to access the specified range sequentially from lower addresses to higher addresses.

`POSIX_MADV_WILLNEED`

The application expects to access the specified range in the near future.

The `<sys/mman.h>` header shall define the following symbolic constants for use as flags for the `posix_typed_mem_open()` function:

`POSIX_TYPED_MEM_ALLOCATE`

Allocate on `mmap()`.

#### POSIX\_TYPED\_MEM\_ALLOCATE\_CONTIG

Allocate contiguously on `mmap()`.

#### POSIX\_TYPED\_MEM\_MAP\_ALLOCATABLE

Map on `mmap()`, without affecting allocatability.

The `<sys/mman.h>` header shall define the `mode_t`, `off_t`, and `size_t` types as described in `<sys/types.h>`.

The `<sys/mman.h>` header shall define the `posix_typed_mem_info` structure, which shall include at least the following member:

`size_t posix_tmi_length` Maximum length which may be allocated from a typed memory object.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int mlock(const void *, size_t);
```

```
int mlockall(int);
```

```
void *mmap(void *, size_t, int, int, int, off_t);
```

```
int mprotect(void *, size_t, int);
```

```
int msync(void *, size_t, int);
```

```
int munlock(const void *, size_t);
```

```
int munlockall(void);
```

```
int munmap(void *, size_t);
```

```
int posix_madvise(void *, size_t, int);
```

```
int posix_mem_offset(const void *restrict, size_t, off_t *restrict,  
size_t *restrict, int *restrict);
```

```
int posix_typed_mem_get_info(int, struct posix_typed_mem_info *);
```

```
int posix_typed_mem_open(const char *, int, int);
```

```
int shm_open(const char *, int, mode_t);
```

```
int shm_unlink(const char *);
```

The following sections are informative.

#### APPLICATION USAGE

None.

#### RATIONALE

None.

## FUTURE DIRECTIONS

None.

## SEE ALSO

<sys\_types.h>

The System Interfaces volume of POSIX.1-2017, `mlock()`, `mlockall()`, `mmap()`, `mprotect()`, `msync()`, `munmap()`, `posix_madvise()`, `posix_mem_off?`, `set()`, `posix_typed_mem_get_info()`, `posix_typed_mem_open()`, `shm_open()`, `shm_unlink()`

## 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](https://www.kernel.org/doc/man-pages/reporting_bugs.html).

IEEE/The Open Group

2017

sys\_mman.h(OP)