

NAME

rt.os/mlock/mlockall - lock the address space of a process testset

DESCRIPTION

Tests the **mlockall()** function for conformance to *System Interfaces and Headers, Issue 5* Page: mlockall().

TESTS**Compliance**

- 1 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A successful call to **mlockall()** shall return 0.

Deviance

- 2 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when *flags* is 0 shall return -1 and set *errno* to EINVAL.
- 3 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when *flags* contains unimplemented flags shall return -1 and set *errno* to EINVAL.
- 4 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5* and the implementation requires a process to have appropriate privilege to call **mlockall()**:
A call to **mlockall()** when the calling process does not have the appropriate privilege to perform the requested operation shall return -1 and set *errno* to EPERM.
- 5 If `_POSIX_MEMLOCK` is not defined and the implementation codes not support the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** shall return -1 and set *errno* to ENOSYS.

Untestable Aspects

- 1 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when *flags* contains MCL_CURRENT shall cause all of the pages currently mapped by the address space of a process to be memory resident.
Reason for omission: There is no portable test method for this requirement.
- 2 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when *flags* contains MCL_FUTURE shall cause all of the pages mapped by the address space of a process in the future, when those mappings are established, to be memory resident.
Reason for omission: There is no portable test method for this requirement.
- 3 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when some or all of the memory identified by the operation could not be locked when the call was made shall return -1 and set *errno* to EAGAIN.
Reason for omission: There is no portable test method for this requirement.
- 4 If `_POSIX_MEMLOCK` is defined or the implementation supports the **mlockall()** function as defined in *System Interfaces and Headers, Issue 5*:
A call to **mlockall()** when locking the pages mapped by the specified range would exceed an implementation-dependent limit on the amount of memory that the process may lock may return -1 and set *errno* to ENOMEM.
Reason for omission: The specification is not adequate for testing as it does not define

under what conditions this error will be reported.

RELEASE

Release VSRT 5.0.0a1

Copyright (c) 1997 X/Open Company Ltd., A member of The Open Group.

All rights reserved