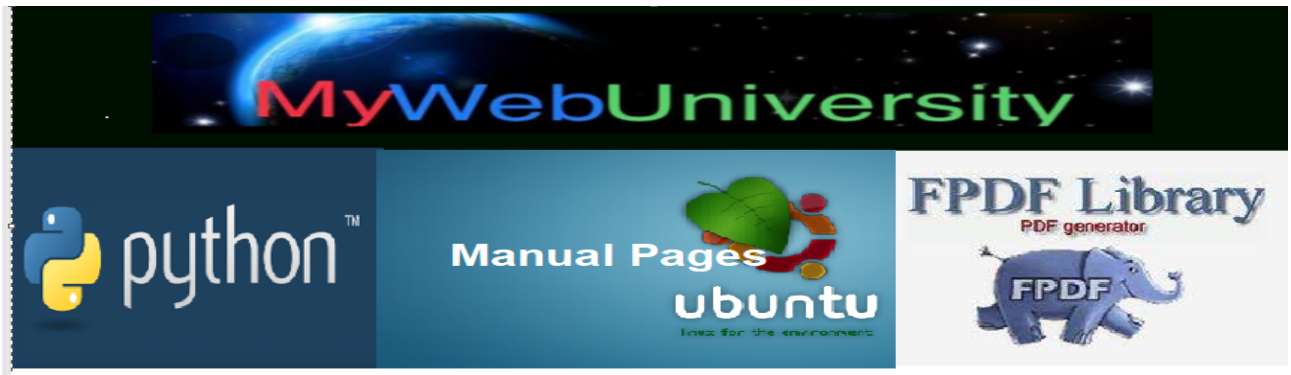

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'process_madvise.2'

$ man process_madvise.2

process_madvise(2) System Calls Manual process_madvise(2)

NAME

process_madvise - give advice about use of memory to a process

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

ssize_t process_madvise(size_t n;
                        int pidfd, const struct iovec iovec[n],
                        size_t n, int advice, unsigned int flags);
```

DESCRIPTION

The process_madvise() system call is used to give advice or directions to the kernel about the address ranges of another process or of the calling process. It provides the advice for the address ranges described by iovec and n. The goal of such advice is to improve system or application performance.

The pidfd argument is a PID file descriptor (see pidfd_open(2)) that specifies the process to which the advice is to be applied.

The pointer iovec points to an array of iovec structures, described in iovec(3type).

n specifies the number of elements in the array of iovec structures. This value must be less than or equal to IOV_MAX (defined in <limits.h> or accessible via the call sysconf(_SC_IOV_MAX)).

If manipulating another process, or before Linux 6.13, the advice argument is one of the following values:

MADV_COLD

See madvise(2).

MADV_COLLAPSE

See madvise(2).

MADV_PAGEOUT

See madvise(2).

MADV_WILLNEED

See madvise(2).

Since Linux 6.13, when manipulating the calling process, any advice flag is permitted.

The flags argument is reserved for future use; currently, this argument must be specified as 0.

The n and iovec arguments are checked before applying any advice. If n is too big, or iovec is invalid, then an error will be returned immediately and no advice will be applied.

The advice might be applied to only a part of iovec if one of its elements points to an in? valid memory region in the remote process. No further elements will be processed beyond that point. (See the discussion regarding partial advice in RETURN VALUE.)

Since Linux 5.12, permission to apply advice to another process is governed by ptrace access mode PTRACE_MODE_READ_FSCREDS check (see ptrace(2)); in addition, because of the performance implications of applying the advice, the caller must have the CAP_SYS_NICE capability (see capabilities(7)).

RETURN VALUE

On success, process_madvise() returns the number of bytes advised. This return value may be less than the total number of requested bytes, if an error occurred after some iovec ele? ments were already processed. The caller should check the return value to determine whether a partial advice occurred.

On error, -1 is returned and errno is set to indicate the error.

ERRORS

EBADF pidfd is not a valid PID file descriptor.

EFAULT The memory described by iovec is outside the accessible address space of the process referred to by pidfd.

EINVAL flags is not 0.

EINVAL The sum of the iov_len values of iovec overflows a ssize_t value.

EINVAL n is too large.

ENOMEM Could not allocate memory for internal copies of the iovec structures.

EPERM The caller does not have permission to access the address space of the process pidfd.

ESRCH The target process does not exist (i.e., it has terminated and been waited on).

See madvise(2) for advice-specific errors.

STANDARDS

Linux.

HISTORY

Linux 5.10. glibc 2.36.

Support for this system call is optional, depending on the setting of the CONFIG_AD? VISE_SYSCALLS configuration option.

When this system call first appeared in Linux 5.10, permission to apply advice to another process was entirely governed by ptrace access mode PTRACE_MODE_ATTACH_FSCREDS check (see ptrace(2)). This requirement was relaxed in Linux 5.12 so that the caller didn't require full control over the target process.

SEE ALSO

madvise(2), pidfd_open(2), process_vm_readv(2), process_vm_write(2)