NAME

unshare − disassociate parts of the process execution context

SYNOPSIS

#define _GNU_SOURCE
#include <sched.h>

int unshare(int *flags***);**

DESCRIPTION

unshare() allows a process to disassociate parts of its execution context that are currently being shared with other processes. Part of the execution context, such as the mount namespace, is shared implicitly when a new process is created using **fork**(2) or **vfork**(2), while other parts, such as virtual memory, may be shared by explicit request when creating a process using **clone**(2).

The main use of **unshare**() is to allow a process to control its shared execution context without creating a new process.

The *flags* argument is a bit mask that specifies which parts of the execution context should be unshared. This argument is specified by ORing together zero or more of the following constants:

CLONE_FILES
Reverse the effect of the **clone**(2) **CLONE_FILES** flag. Unshare the file descriptor table, so that the calling process no longer shares its file descriptors with any other process.

CLONE_FS
Reverse the effect of the **clone**(2) **CLONE_FS** flag. Unshare file system attributes, so that the calling process no longer shares its root directory, current directory, or umask attributes with any other process. **chroot**(2), **chdir**(2), or **umask**(2)

CLONE_NEWNS
This flag has the *same* effect as the **clone**(2) **CLONE_NEWNS** flag. Unshare the mount namespace, so that the calling process has a private copy of its namespace which is not shared with any other process. Specifying this flag automatically implies **CLONE_FS** as well.

If *flags* is specified as zero, then **unshare**() is a no-op; no changes are made to the calling process's execution context.

RETURN VALUE

On success, zero returned. On failure, −1 is returned and *errno* is set to indicate the error.

ERRORS

EINVAL
An invalid bit was specified in *flags*.

ENOMEM
Cannot allocate sufficient memory to copy parts of caller's context that need to be unshared.

EPERM
flags specified **CLONE_NEWNS** but the calling process was not privileged (did not have the **CAP_SYS_ADMIN** capability).

VERSIONS

The **unshare**() system call was added to Linux in kernel 2.6.16.

CONFORMING TO

The **unshare**() system call is Linux-specific.

NOTES

Not all of the process attributes that can be shared when a new process is created using **clone**(2) can be unshared using **unshare**(). In particular, as at kernel 2.6.16, **unshare**() does not implement flags that reverse the effects of **CLONE_SIGHAND**, **CLONE_SYSVSEM**, **CLONE_THREAD**, or **CLONE_VM**.

Such functionality may be added in the future, if required.

SEE ALSO

clone(2), **fork**(2), **vfork**(2), Documentation/unshare.txt

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.