

unshare.2

ИМЯ

unshare - отделяет части процесса контекста выполнения

ОБЗОР

```
#define _GNU_SOURCE                /* Смотрите feature_test_macros(7) */
#include <sched.h>

int unshare(int flags);
```

ОПИСАНИЕ

Вызов `unshare()` позволяет процессу отделить части своего контекста выполнения, которые используются совместно с другими процессами. Часть контекста выполнения, например пространство имён монтирования, неявно делается общей при создании нового процесса с помощью `fork(2)` или `vfork(2)`, в то время как другие части, такие как виртуальная память, могут стать общими по явному запросу при создании процесса с помощью `clone(2)`.

Основное предназначение `unshare()` (ем позволить процессу контролировать свой общий контекст выполнения без создания нового процесса.

Значение аргумента `flags` представляет собой битовую маску, в которой указывается какие части контекста выполнения должны перестать быть общими. Значение составляется из нескольких следующих констант (через OR):

CLONE_FILES Обратный эффект флагу **CLONE_FILES** для `clone(2)`. Отделяет таблицу файловых дескрипторов таким образом, что вызывающий процесс больше не имеет общих файловых дескрипторов с другими процессами.

CLONE_FS Обратный эффект флагу **CLONE_FS** для `clone(2)`. Отделяет атрибуты файловой системы таким образом, что вызывающий процесс больше не имеет общих атрибутов корневого каталога, текущего каталога и `umask` с другими процессами (`chroot(2)`, `chdir(2)` или `umask(2)`).

CLONE_NEWIPC (начиная с Linux 2.6.19) Этот флаг имеет действие подобное флагу **CLONE_NEWIPC** для `clone(2)`. Отделяет пространство имён System V IPC таким образом, что вызывающий процесс будет иметь свою личную копию пространства имён System V IPC, неиспользуемую другими процессами. Задание данного флага автоматически устанавливает флаг **CLONE_SYSVSEM**. Для использования **CLONE_NEWIPC** требуется мандат **CAP_SYS_ADMIN**.

CLONE_NEWNET (начиная с Linux 2.6.24) Этот флаг имеет действие подобное флагу **CLONE_NEWNET** для `clone(2)`. Отделяет сетевое пространство имён таким образом, что вызывающий процесс будет иметь свою личную копию сетевого пространства имён, неиспользуемую другими процессами. Для использования **CLONE_NEWNET** требуется мандат **CAP_SYS_ADMIN**.

CLONE_NEWNS Этот флаг имеет действие подобное флагу **CLONE_NEWNS** для `clone(2)`.

Отделяет пространство имён монтирования таким образом, что вызывающий процесс будет иметь свою личную копию данного пространства имён, неиспользуемую другими процессами. Задание данного флага автоматически устанавливает флаг CLONE_FS. Для использования CLONE_NEWNS требуется мандат CAP_SYS_ADMIN.

CLONE_SYSVSEM (начиная с Linux 2.6.26) Обратный эффект флагу CLONE_SYSVSEM для clone(2). Отделяет семафоры System V таким образом, что вызывающий процесс будет иметь свою личную копию, неиспользуемую другими процессами. Для использования CLONE_SYSVSEM требуется мандат CAP_SYS_ADMIN.

CLONE_NEWUTS (начиная с Linux 2.6.19) Этот флаг имеет действие подобное флагу CLONE_NEWUTS для clone(2). Отделяет пространство имён UTS IPC таким образом, что вызывающий процесс будет иметь свою личную копию пространства имён UTS, неиспользуемую другими процессами. Для использования CLONE_NEWUTS требуется мандат CAP_SYS_ADMIN. Если значение flags равно нулю, то unshare() ничего не делает, то есть в контексте выполнения вызывающего процесса ничего не изменяется.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

При успешном выполнении возвращается 0. При ошибке возвращается -1, а errno присваивается значение ошибки.

ОШИБКИ

EINVAL В значении flags установлен недопустимый бит.

ENOMEM Не удалось выделить достаточно памяти для копирования части контекста вызывающего, которая должна быть отделена.

EPERM Вызывающий процесс не имеет требуемых привилегий для этой операции.

ВЕРСИИ

Системный вызов unshare() был добавлен в ядро Linux версии 2.6.16.

СООТВЕТСТВИЕ СТАНДАРТАМ

Системный вызов unshare() есть только в Linux.

ЗАМЕЧАНИЯ

Не все атрибуты процесса, которые могут использоваться совместно при создании нового процесса с помощью clone(2), могут быть отделены с помощью unshare(). В частности, начиная с ядра 2.6.16 в unshare() не реализована поддержка флагов, которые были имели обратное действие CLONE_SIGHAND, CLONE_SYSVSEM, CLONE_THREAD или CLONE_VM. Эти возможности могут быть добавлены позднее, если потребуется.

СМОТРИТЕ ТАКЖЕ

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