

kill(1)

НАЗВАНИЕ

kill - завершить процессы или послать им сигнал

СИНТАКСИС

```
/usr/bin/kill -s сигнал pid...
/usr/bin/kill -l [ статус_выхода ]
/usr/bin/kill [ -сигнал ] pid ...
```

ОПИСАНИЕ

Утилита **kill** посылает сигнал процессу или процессам, заданным операндами **pid**.

Для каждого операнда **pid** утилита **kill** будет выполнять действия, аналогичные выполняемым функцией **kill(2)**, вызванной со следующими аргументами:

1. Значение операнда **pid** будет использоваться в качестве аргумента **pid**.
2. В качестве аргумента **sig** используется значение, заданное в опции **-s**, или **SIGTERM**, если эта опция не указана.

Процесс, которому посыпается сигнал, должен принадлежать текущему пользователю, но суперпользователь (**root**) может посыпать сигналы любым процессам.

Описание версий **kill**, встроенных в командные интерпретаторы, см. в разделе "[ПРИМЕЧАНИЯ](#)".

ОПЦИИ

Поддерживаются следующие опции:

-l

(Английская буква "эл".) Выдать все значения сигналов, поддерживаемые в данной реализации, если операнды не указаны. Если указан операнд **статус_выхода**, соответствующий значению специального параметра ? командного интерпретатора и вызову **wait** для процесса, работа которого прекращена сигналом, выдать имя сигнала, которым была прекращена работа процесса. Если указан операнд **статус_выхода**, представляющий собой целое число без знака - номер сигнала, - выдать имя соответствующего сигнала. В противном случае, результат не определен.

-s **сигнал**

Задает сигнал, который надо послать, используя одно из символьных имен, заданных в заголовочном файле `<signal.h>`. Значение сигнала распознается независимо от регистра символов. При этом префикс `SIG` указывать не надо. Кроме того, распознается символьное имя `0`, представляющее сигнал со значением ноль. Указанный сигнал будет посыпаться вместо стандартного, `SIGTERM`.

ОПЕРАНДЫ

Поддерживаются следующие операнды:

`pid`

Одно из следующих значений:

1. Десятичное целое, задающее процесс или группу процессов, которым надо послать сигнал. Какие процессы выбираются при указании положительного, отрицательного числа или нуля в качестве значения `pid` см. в описании функции `kill`. Если указан процесс номер 0, посыпается сигнал всем процессам в соответствующей группе процессов. Если первый операнд `pid` - отрицательный, перед ним надо указать два дефиса (--), чтобы он не интерпретировался как опция.
2. Идентификатор задания системы управления заданиями, определяющий фоновую группу процессов, которой надо послать сигнал. Идентификаторы заданий можно указывать только для вызовов `kill` в среде выполнения текущего командного интерпретатора.

Учтите, что идентификаторы заданий можно указывать только в системах, поддерживающих возможности управления заданиями.

`статус_выхода`

Десятичное целое, задающее номер сигнала или статус выхода процесса, прекращенного сигналом.

ИСПОЛЬЗОВАНИЕ

Номера (идентификаторы) процессов можно найти с помощью команды `ps(1)`.

Когда `kill` выполняется в отдельной среде, идентификаторы заданий не обязательно работают так, как ожидается. В следующих примерах:

```
nohup kill %1 &
system( "kill %1");
```

`kill` работает в отдельной среде, в которой не поддерживаются те же идентификаторы заданий, что и в исходном командном интерпретаторе.

РЕЗУЛЬТАТЫ

Когда опция `-l` не указана, стандартный выходной поток не используется.

Когда опция `-l` указана, символьное имя каждого сигнала будет выдаваться в следующем формате:

```
"%s%c", <сигнал>, <разделитель>
```

где <сигнал> выдается в верхнем регистре без префикса **SIG**, а <разделитель> - либо символ новой строки, либо пробел. Для последнего выдаваемого сигнала <разделителем> будет символ новой строки.

Когда указана опция **-l** и operand **статус_выхода**, символьное имя соответствующего сигнала выдается в следующем формате:

```
"%s\n", <сигнал>
```

ПРИМЕРЫ

Пример 1: Примеры использования команды **kill**.

Команды:

```
kill -9 100 -165  
kill -s kill 100 -165  
kill -s KILL 100 -165
```

посылают сигнал **SIGKILL** процессу с идентификатором 100 и всем процессам, идентификатор группы процессов которых равен 165, если, конечно, посылающий процесс имеет право посылки сигнала этим процессам, и процессы с соответствующими идентификаторами существуют.

Во избежание двусмысленности, если первый аргумент - отрицательное число, задающее номер сигнала или группы процессов, всегда предполагается, что это номер сигнала. Поэтому, чтобы послать стандартный сигнал группе процессов (например, 123), надо использовать команды следующего вида:

```
kill -TERM -123  
kill -- -123
```

ПЕРЕМЕННЫЕ СРЕДЫ

Описание следующих переменных среды, влияющих на работу **cat**, см. на странице справочного руководства **environ(5)**: **LC_CTYPE**, **LC_MESSAGES** и **NLSPATH**.

СТАТУС ВЫХОДА

Программа завершается со следующими значениями статуса выхода:

- | | |
|----|--|
| 0 | Для каждого операнда pid был найден хотя бы один соответствующий процесс и указанный сигнал был успешно обработан хотя бы одним из соответствующих процессов. |
| >0 | Произошла ошибка. |

АТРИБУТЫ

Описание следующих атрибутов см. на странице справочного руководства **attributes(5)**:

ТИП АТРИБУТА	ЗНАЧЕНИЕ АТРИБУТА
Доступен в пакете	SUNWcsu
CSI	включено

ССЫЛКИ

[csh\(1\)](#), [jobs\(1\)](#), [ksh\(1\)](#), [ps\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [wait\(1\)](#), [kill\(2\)](#), [signal\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [signal\(3HEAD\)](#)

ПРИМЕЧАНИЯ

sh

Командный интерпретатор **sh** имеет встроенную версию **kill**, обеспечивающую функциональные возможности команды **kill** при идентификации процессов по номеру задания. В **sh** используется следующий синтаксис:

```
kill [-sig] [pid] [%job]...
kill -l
```

csh

Командный интерпретатор **csh** тоже имеет встроенную команду **kill** со следующим синтаксисом:

```
kill [-sig] [pid] [%job]...
kill -l
```

Встроенная в **csh** команда **kill** посылает стандартный сигнал **TERM** или указанный сигнал процессам с указанными идентификаторами процесса, номерами задания, или текущему заданию. Сигналы задаются по номеру или по имени. Стандартного сигнала нет. Команда **kill** без аргументов не посылает сигнал текущему заданию. Если посыпается сигнал **TERM** или **HUP**, то заданию и процессу посыпается также сигнал **CONT** (продолжить).

- l Выдает имена сигналов, которые можно посыпать.

ksh

Синтаксис встроенной команды **kill** в командном интерпретаторе **ksh**:

```
kill [-sig] [pid] [%job]...
kill -l
```

Версия **kill** в **ksh** посылает либо сигнал **TERM**, либо указанный сигнал указанным заданиям или процессам. Сигналы задаются по номерам или по именам (как описано на странице справочного руководства [signal\(3HEAD\)](#), но без префикса "SIG"). Если посыпается сигнал **TERM** или **HUP**, заданию или процессу посыпается также сигнал **CONT**, если процесс остановлен. В качестве аргумента **job** можно указать идентификатор процесса, не входящего ни в одно из активных заданий. Вторая форма вызова, **kill -l**, выдает номера и соответствующие имена поддерживаемых сигналов.

Последнее изменение: 18 сентября 1995 года

