

_exit - Man Page

завершение вызывающего процесса

Краткий обзор

```
#include <unistd.h>
```

```
noreturn void _exit(int status);
```

```
#включить <stdlib.h>
```

```
noreturn void _Exit(int status);
```

Требования к макросу Feature Test для glibc (см. `feature_test_macros(7)`):

```
_Exit():
```

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

Описание

`_exit()` немедленно завершает вызывающий процесс. Все открытые файловые дескрипторы, принадлежащие процессу, закрываются. Любые дочерние элементы процесса наследуются `init(1)` (или ближайшим процессом "subreaper", как определено с помощью операции `prctl(2)` `PR_SET_CHILD_SUBREAPER`). Родительскому процессу отправляется сигнал `SIGCHLD`.

Значение `status & 0xFF` возвращается родительскому процессу как статус выхода процесса и может быть собрано родителем с помощью одного из вызовов семейства `wait(2)`.

Функция `_Exit()` эквивалентна `_exit()` .

Возвращаемое значение

Эти функции не возвращаются.

Соответствие

POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD. Функция `_Exit()` была введена C99 .

Примечания

Для обсуждения эффектов выхода, передачи статуса выхода, зомби-процессов, отправленных сигналов и т. Д. См. `exit(3)`.

Функция `_exit()` похожа на `exit(3)`, но не вызывает никаких функций, зарегистрированных в `atexit(3)` или `on_exit(3)`. Открытые потоки `stdio (3)`

_exit - Man Page

завершения ожидающего вывода. Если задержка нежелательна, может быть полезно вызвать такие функции, как `tcflush(3)`, перед вызовом `_exit()`. Отменяется ли какой-либо ожидающий ввод-вывод и какой ожидающий ввод-вывод может быть отменен после `_exit()`, зависит от реализации.

Библиотека C /различия в ядре

В glibc до версии 2.3 функция-оболочка `_exit()` вызывала системный вызов ядра с тем же именем. Начиная с glibc 2.3, функция-оболочка вызывает `exit_group(2)`, чтобы завершить все потоки в процессе.

Системный вызов `raw _exit()` завершает только вызывающий поток, и такие действия, как повторное создание дочерних процессов или отправка `SIGCHLD` родительскому процессу, выполняются только в том случае, если это последний поток в группе потоков.

См. Также

`execve (2)`, `exit_group(2)`, `fork (2)`, `убить (2)`, `ожидание (2)`, `wait4(2)`, `waitpid (2)`, `atexit (3)`, `выход (3)`, `on_exit(3)`, `termios (3)`

Colophon

Эта страница является частью версии 5.13 проекта Linux *man-pages*. Описание проекта, информацию об ошибках и последнюю версию этой страницы можно найти по адресу [https://www.kernel.org/doc/man-pages /](https://www.kernel.org/doc/man-pages/).

Ссылка

`atexit (3)`, `клон (2)`, `демон (3)`, `выход (3)`, `exit_group(2)`, `ожидание (1)`, `fork (2)`, `gprof(1)`, `убить (2)`, `on_exit(3)`, `pagesend.8c(8)`, `persistent-keyring (7)`, `pmGetConfig(3)`, `pmNoMem (3)`, `prctl(2)`, `ptrace (2)`, `seccomp (2)`, `setsid(2)`, `shmop (2)`, `сигнал-безопасность (7)`, `сокет (7)`, `stress-ng (1)`, `системные вызовы (2)`, `система (3)`, `пользовательский брелок (7)`, `user-session-keyring (7)`, `vfork (2)`, `ожидание (2)`.

Man-страницы `exit(2)` и `_Exit(2)` являются псевдонимами `_exit(2)`.

2021-03-22 Руководство программиста Linux

`_exit` - Man Page

[Главная](#) [Блог](#) [0 нас](#)