

# Bash: Комбинации клавиш – Ctrl + U, Ctrl + W, Ctrl + L, Ctrl + C, Ctrl + D, Ctrl + Z.

Category: bash & sh, clear, GNU/Linux, комбинации клавиш  
2025-02-10

В терминале **GNU/Linux** комбинации клавиш Ctrl + U, Ctrl + W, Ctrl + L, Ctrl + C, Ctrl + D и Ctrl + Z имеют специфическое значение и используются для управления процессами.

## **Описание комбинаций.**

Разберём каждую из них подробнее.

### **Ctrl + U**

**Функция:** Удаление всей строки, которая находится слева от курсора.

**Описание:** Например, если вы вводите команду и хотите быстро очистить её, просто нажмите Ctrl + U, и всё, что вы ввели до курсора, будет удалено.

### **Ctrl + W**

**Функция:** Удаление слова или части слова, находящегося слева от курсора.

**Описание:** Если вы хотите удалить предыдущее слово в командной строке, просто нажмите Ctrl + W, и последнее введённое слово будет удалено.

### **Ctrl + L**

**Функция:** Активация утилиты clear. Утилита clear в **GNU/Linux** используется для очистки содержимого терминала, возвращая его к первоначальному состоянию. Эта команда полезна, когда вам нужно убрать все предыдущие команды и вывод, чтобы сосредоточиться на новых данных или для более удобного представления информации.

**Описание:** После выполнения этой команды экран терминала будет очищен, и вы увидите пустую строку в верхней части.

## **Ctrl + C**

**Функция:** Прерывание процесса.

**Описание:** Когда вы нажимаете Ctrl + C, это сигналы (SIGINT) отправляются текущему процессу, который работает в переднем плане. Это означает, что процесс получает уведомление о том, что его следует завершить. Обычно это используется для остановки программ, которые могут выполняться бесконечно или требуют длительного времени на выполнение. Например, если вы запустили скрипт или программу, и хотите его остановить, просто нажмите Ctrl + C.

## **Ctrl + D**

**Функция:** Завершение ввода (EOF – **end of file** – конец файла) .

**Описание:** Нажатие Ctrl + D отправляет сигнал «конец файла» (EOF) в терминал. Это полезно, когда вы вводите данные в интерактивной программе или в командной строке, где нужно прекратить ввод. Например, если вы работаете в оболочке, и хотите выйти из неё, вы можете нажать Ctrl + D, чтобы сигнализировать, что больше не будет ввода, и оболочка закроется. Также Ctrl + D часто используется для завершения сессии в интерактивных интерпретаторах, таких как Python.

## **Ctrl + Z**

**Функция:** Приостановка процесса.

**Описание:** Нажатие Ctrl + Z отправляет сигнал SIGTSTP, который приостанавливает текущий процесс и переводит его в состояние фонового ожидания (**background**). Таким образом, программа не завершает свою работу, а остаётся в памяти и может быть восстановлена позже. После нажатия Ctrl + Z, вы можете использовать команду fg для продолжения выполнения процесса в переднем плане или bg для его продолжения в фоновом режиме. Это удобно, если вам нужно временно приостановить работу программы, чтобы выполнить другие задачи в терминале.

## **Краткий итог .**

Ctrl + U – удаляет всю строку, которая находится слева от курсора.

Ctrl + W – удаляет слово или часть слова, находящегося слева от курсора.

Ctrl + L – очистка содержимого терминала.

Ctrl + C – прерывает выполнение программы.

Ctrl + D – завершает ввод (EOF) или закрывает сессию.

Ctrl + Z – приостанавливает процесс, позволяя продолжить его позже.

Эти команды являются важными инструментами управления процессами в терминале, позволяя пользователям эффективно взаимодействовать с выполняемыми программами.

## В чем разница между комбинациями?

Комбинации клавиш `Ctrl + C`, `Ctrl + D` и `Ctrl + Z` часто используются в терминале для выхода из программы, работающей на переднем плане (**foreground**), и передачи управления `bash`.

`Ctrl + C` выполняет прерывание процесса, то есть убивает его из оперативной памяти. Результат действия `Ctrl + D` такой же. Однако разница между этими двумя способами выхода есть, и заключается она во внутреннем механизме.

Нажатие `Ctrl + C` заставляет терминал послать сигнал **SIGINT** процессу, который на данный момент его контролирует. Когда **foreground**-программа получает сигнал **SIGINT**, она обязана прервать свою работу.

Нажатие `Ctrl + D` говорит терминалу, что надо зарегистрировать так называемый **EOF** (**end of file** – конец файла), то есть поток ввода окончен. `Bash` интерпретирует это как желание выйти из программы.

При работе в конкретной программе могут срабатывать оба способа, но может только один. Так команда `cat` без параметров запускается в интерактивном режиме, в котором то, что вы ввели, будет тут же выводиться на экран после нажатия **Enter**. Выйти из программы можно как с помощью `Ctrl + C`, так и `Ctrl + D`.

Жмем `Ctrl + C`:

```
$ cat
```

Ответ:



Жмем `Ctrl + D`:

```
$ cat
```

Ответ:



Выйти из интерпретатора **Python** с помощью `Ctrl + C` нельзя.

```
$ python
```

Пример:

```
Python 3.6.7 (default, Oct  
[GCC 8.2.0] on linux  
Type «help», «copyright»,  
>>> a = 15  
>>> a + 7  
22  
>>> (жмем Ctrl + C)  
KeyboardInterrupt  
>>> (жмем Ctrl + D)  
$
```

Ответ:



Комбинация клавиш **Ctrl + Z** посылает процессу сигнал, который приказывает ему остановиться. Это значит, что процесс остается в системе, но как бы замораживается. Само собой разумеется он уходит в **background** – в фоновый режим.

```
$ python3
```

Ответ:



С помощью команды **bg** его можно снова запустить, оставив при этом в фоновом режиме. Команда **fg** не только возобновляет ранее приостановленный процесс, но и выводит его из фона на передний план.

```
$ fg python
```

Ответ:



На изображении сначала запускается интерпретатор **Python**. Далее его работа останавливается с помощью **Ctrl + Z**. Командой **fg** **python3** мы возобновляем работу приложения и делаем его текущим. То, что мы вернулись в тот же самый процесс, видно по переменной **a**, значение которой остается доступным.

С помощью команды **ps** можно посмотреть процессы текущего сеанса **bash** (используя ключи можно просматривать и другие процессы), а команда **kill** позволяет удалять ненужные.

Здесь запускается **cat** и приостанавливается.

```
$ cat
```

Ответ:



Далее смотрим список процессов.

```
$ ps
```

Ответ:



Ключ -9 команды kill обозначает, что мы хотим полностью избавиться от процесса. В качестве аргумента передается идентификационный **номер процесса (PID)**.

```
$ kill -9 12302
```

Ответ:



```
$ ps
```

Ответ:



Отметим, что во многих консольных программах управляющие комбинации с клавишей Ctrl обычно обозначаются как ^C, ^Z и тому подобное, то есть вместо Ctrl используется знак ^.

Понятие «**процесса**» связано с понятием «**одновременного выполнения нескольких программ**». Процесс – это выполняющаяся программа, а точнее экземпляр выполняющейся программы, потому что в системе может существовать несколько процессов, запущенных на основе одной программы. Ядром операционной системы создается видимость одновременного выполнения множества процессов.

При необходимости системный администратор может запускать процессы, требующие длительного выполнения, в **background**, а на «**переднем плане**» выполнять другие команды.

Запустить процесс **в фоне** можно с помощью знака амперсанда &, который записывается в конце команды.

```
$ sleep 10s &
```

Ответ:



Если в текущем терминале было запущено и переведено в фон несколько одноименных процессов, то сделать один из них текущим с помощью команды `fg` имя\_программы не получится. PID также не используется как аргумент для `fg`.

```
$ ps
```

Ответ:



Здесь используется идентификатор процесса, который возвращает команда `jobs`. Причем для `fg` его надо использовать в формате `%ID`.

Нумерация идет снизу:

```
$ fg %2
```

```
$ fg %3
```

Ответ:



```
$ fg %1
```

Ответ:



## Оригиналы источников информации.

1. [younglinux.info](http://younglinux.info) «Ctrl + C, Ctrl + D, Ctrl + Z».