

Горячие клавиши в терминале Linux

```
ASH(1)
NAME
    bash - GNU Bourne-Again Shell
SYNOPSIS
    bash [options] [command_string | file]
COPYRIGHT
    Bash is Copyright (C) 1989-2016 by the Free Software Foundation, Inc.
DESCRIPTION
    Bash is an sh-compatible command language interpreter that executes com-
    mands read from the standard input or from a file. Bash also incorpo-
    rates useful features from the Korn and C shells (ksh and csh).

    Bash is intended to be a conformant implementation of the Shell and Util-
    ities portion of the IEEE POSIX specification (IEEE Standard 1003.1).
Manual page bash(1) line 1 (press h for help or q to quit)
```

Давным-давно, такие слова как "hot keys" и "keyboard shortcuts" мне не всегда удавалось перевести на русский без потери лица. Как-то раз, я написал "клавиатурные сокращения", чем сразу же привлёк косые взгляды и вызвал смелые медицинские фантазии... Но вроде бы сейчас принято везде говорить и писать "горячие клавиши". О них и поговорим.

Данная заметка — шпаргалка по линуксовой оболочке Bash и смежным компонентам. Часть приводимых ниже команд относится к библиотеке [Readline](#), часть — к [сигналам](#) Linux, однако такие подробности нам здесь не важны. Если вам приходится часто иметь дело с терминалом в Linux (и вы не меняли Bash на другой шелл), то будет очень полезно использовать эти самые "сокращения" на благо себе и в мирных целях. Текст написан для начинающих пользователей, но кто знает — может быть и вы найдёте в нём что-то новое и полезное для себя.

Для удобства будем считать, что по умолчанию под терминалом мы понимаем стандартную в настольной редакции Ubuntu программу "Терминал Gnome".

Вкладки

В программе "Терминал Gnome" предусмотрены вкладки, и работают они аналогично вкладкам в веб-браузере или файловом менеджере. Иными словами, если вам нужно несколько терминалов, вовсе не обязательно открывать *несколько окон*. Достаточно одного окна с *несколькими вкладками*. Ниже приведены горячие клавиши, относящиеся ко вкладкам:

Ctrl+Shift+T — открыть новую вкладку;

Ctrl+Shift+W или Ctrl+D — закрыть текущую вкладку (или весь терминал, если вкладка одна);

Ctrl+Shift+N — открыть новое окно терминала из текущего.

Со временем вы можете оказаться в ситуации, когда вкладок станет действительно много, и тогда возникнет вопрос о навигации между ними. Вам пригодятся следующие сочетания клавиш:

Ctrl+PgDn — перейти на следующую (справа) вкладку;

Ctrl+PgUp — перейти на предыдущую (слева) вкладку;

Ctrl+Shift+PgDn — сдвинуть вкладку вправо;

Ctrl+Shift+PgUp — сдвинуть вкладку влево.

Alt+1 — перейти на первую по счёту вкладку. Подставьте другую цифру для нужной вам вкладки. Данный способ позволяет «дотянуться» максимум до *десятой* по счёту вкладки.

Навигация

Три очень часто используемые комбинации для копирования и вставки текста, а также отмены выполняющейся команды:

Ctrl+Shift+C — копирование в буфер обмена;

Ctrl+Shift+V — вставка из буфера обмена;

Ctrl+C — прерывание выполняющейся команды или очистка текущей строки.

Для того чтобы выделить нужный текст в терминале, вам потребуется воспользоваться мышью. Тем не менее, в программе “Терминал Gnome” имеется встроенное средство поиска текста, которое позволяет искать как по обычному фрагменту, так и по регулярному выражению:

Ctrl+Shift+F — вызов встроенного поиска по любому тексту в терминале.

Если команда в терминале слишком длинная, или вы сделали опечатку в начале и не сразу это заметили, вы можете вернуться в начало строки. А затем — снова в конец. Вот как это сделать:

Ctrl+A — переместиться в начало строки;

Ctrl+E — переместиться в конец строки.

В терминале Linux можно перемещаться внутри строки также по словам и по отдельным символам (в последнем случае, это аналогично использованию клавиш с боковыми стрелками):

Ctrl+F — переместиться на 1 символ вперед;

Ctrl+B — переместиться на 1 символ назад;

Alt+F — переместиться к следующему слову;

Alt+B — переместиться в начало предыдущего слова.

Управление командами и процессами

Предыдущие команды касались навигации по терминалу и строке ввода команды. Далее стоит рассмотреть управляющие команды Bash, с помощью которых можно запускать, останавливать, ставить на паузу и возобновлять команды и процессы. Вы уже знаете, что запущенный в терминале процесс можно прервать по Ctrl+C, но полезно также знать и некоторые нюансы.

В терминале Linux вы можете не только завершать программы полностью, но и ставить их на паузу. Затем выполнение программы можно возобновлять, причём, как с возвратом интерактивной командной строки, так и без него:

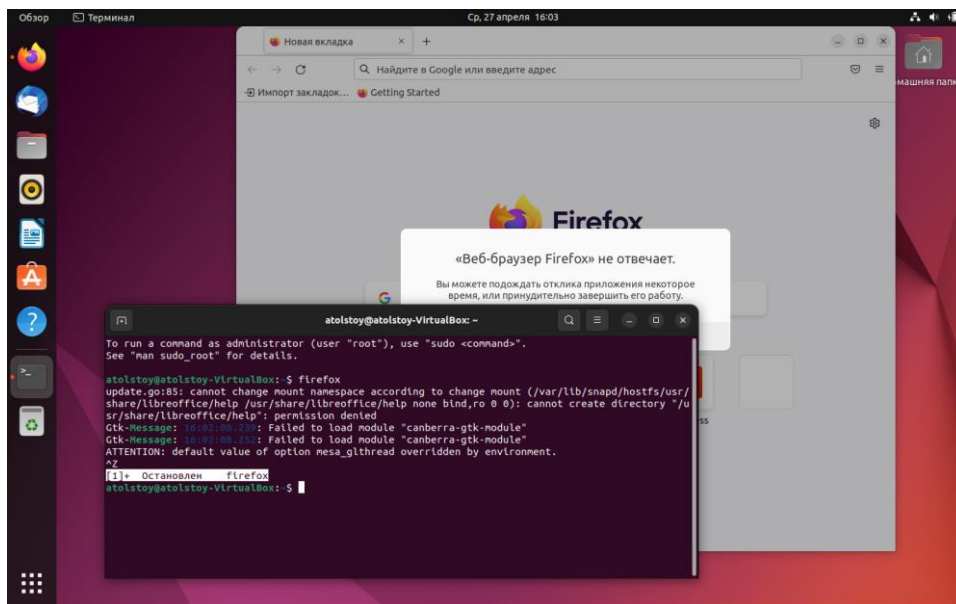
Ctrl+Z — приостановка процесса;

команда *bg* — возобновление процесса с возвратом командной строки (процесс продолжает выполнение в фоне);

команда *fg* — возобновление процесса, при котором он удерживает командную строку за собой (процесс выполняется на «переднем плане»).

Процессы также можно приостанавливать и возобновлять. Запустите какую-либо команду, например *htop*, и нажмите Ctrl+Z. Сначала будет казаться, будто команда завершилась, но она будет числиться в списке запущенных процессов (*ps -a*) и появится вновь после ввода команды *fg*.

Если повторить эксперимент с графическим приложением, например, введя команду *firefox*, то можно будет использовать для его «оживления» как *fg*, так и *bg*. При любом варианте приложение останется «закреплённым» за текущим терминалом: если вы закроете его, то оно тоже завершится.



После приостановки процесса `firefox` в терминале рабочая среда Gnome будет считать, что приложение «не отвечает».

Существует и другой тип «приостановки»: временное *прекращение вывода* выполняющейся команды. Как консольное, так и графическое приложение может быть запущено в терминале, в который будет выводиться текущая диагностическая информация. Иногда бывает очень удобно временно прекратить постоянный вывод сообщений без завершения самого приложения. Для этого пригодятся следующие сочетания клавиш:

`Ctrl+S` — прекратить обновление вывода команды;

`Ctrl+Q` — возобновить вывод команды.

История команд

Bash умеет запоминать все введенные вами команды. Пока терминал запущен, они хранятся в оперативной памяти компьютера, а при выходе из терминала записываются в долговременное хранилище в файле `~/.bash_history`.

Если вы точно знаете, что вводили нужную вам команду раньше, поищите её в истории:

`history` — вывод истории команд;

Если вы помните хотя бы часть команды, поиск можно уточнить:

`history | grep <часть команды>` — пример уточняющего поиска по истории команд.

У каждой команды в истории есть номер. Введите этот номер, поставив вначале восклицательный знак, и Bash выполнит соответствующую команду:

!151 — выполнить команду под номером 151 из истории;

!151: — показать команду номер 151, но не выполнять её;

!! — повторно выполнить последнюю введенную команду.

В Bash имеется интерактивный режим поиска по истории команд. Нажмите Ctrl+R и начните набирать часть команды. Bash сам предложит вам первый совпадающий вариант. Если он не подходит, нажимайте Ctrl+R дальше для перебора вариантов. Когда нужный вариант будет найден, нажмите Enter.

Интересно, что у этой клавиши ввода есть два аналога — вместо Enter можно нажать Ctrl+M или Ctrl+J.

Самый простой способ перемещаться по истории команд — стрелки «вверх» и «вниз» на клавиатуре. Они тоже имеют дубликаты:

Ctrl+P — вывести предыдущую команду;

Ctrl+N — вывести следующую команду.

Редактирование команд

Самое время рассмотреть средства редактирования команд — они в Bash весьма продвинутые. Удобное перемещение в начало и конец строки, выборочное удаление символов и слов — это лишь часть возможностей, которые могут пригодиться пользователю. За редактирование команд отвечают следующие сочетания клавиш:

Ctrl+U — удалить весь текст слева от курсора;

Ctrl+K — удалить весь текст справа от курсора;

Ctrl+W — удалить 1 слово или параметр слева от курсора;

Ctrl+D — удаление текущего символа (аналогично Del);

Ctrl+H — удаление предыдущего символа (аналогично Backspace);

Alt+D — удалить всё справа от курсора до ближайшего пробела;

Alt+Backspace — удалить всё слева от курсора до ближайшего пробела;

Alt+T — поменять местами текущее слово с предыдущим;

Esc+T — поменять местами два предыдущих слова;

Tab — автодополнение команды после ввода её первых символов.

Ещё одна любопытная деталь: у Bash имеется собственный буфер обмена, который работает независимо от стандартного буфера (как мы помним, копирование по Ctrl+Shift+C, вставка по Ctrl+Shift+V). Это важно, поскольку у первых трёх команд из предыдущего списка есть дополнительные функции:

они не просто удаляют часть текста, но и копируют его в тот самый отдельный буфер обмена Bash. Поэтому, будет справедливо уточнить:

Ctrl+U — вырезать и поместить в буфер обмена весь текст слева от курсора;

Ctrl+K — вырезать и поместить в буфер обмена весь текст справа от курсора;

Ctrl+W — вырезать и поместить в буфер обмена 1 слово или параметр слева от курсора;

Кстати, для вставки скопированного текста обратно сработает комбинация Ctrl+Y.

Напоследок

Конечно, выше я описал не все горячие клавиши: их гораздо больше, и полное описание содержало бы в себе кучу бородатой экзотики, унаследованной из древних университетских времён UNIX. В любом случае, не забывайте про `man bash` (например, там есть замечательный раздел *Commands for Moving*) и про `bind -P`.