

# vi(1)

## НАЗВАНИЕ

**vi** - экранно-ориентированный (визуальный) дисплейный редактор, основанный на **ex**

## СИНТАКСИС

```
vi [-t тэг] [-r файл] [-l] [-L] [-wn] [-R] [-x] [-C]
    [-c команда] файл ...
view [-t тэг] [-r файл] [-l] [-L] [-wn] [-R] [-x] [-C]
    [-c команда] файл ...
vedit [-t тэг] [-r файл] [-l] [-L] [-wn] [-R] [-x] [-C]
    [-c команда] файл ...
```

## ОПИСАНИЕ

**vi (visual)** - экранно-ориентированный текстовый редактор, основанный на базовом редакторе **ex**. Можно использовать командный режим **ex** из **vi** и наоборот. Команды экранного режима описаны на этой странице справочного руководства; установка опций (типа автоматической нумерации строк или автоматического перехода на следующую строку при нажатии Enter), а также все команды строкового редактора **ex** описаны на странице справочного руководства **ex(1)**.

При работе в **vi**, изменения, которые вы делаете в файле, отображаются на экране терминала. Курсор на экране показывает текущую позицию в файле.

Эти утилиты обрабатывают и показывают символы дополнительного набора в соответствии с локалью, заданной в переменной среды **LC\_CTYPE** (см. **LANG** в **environ(5)**), за исключением того, что символ **x**, задаваемый в командах **r**, **f**, **F**, **m**, **t** и **T** (см. раздел "**СВОДКА КОМАНД**" ниже) должен быть однобайтовым. Вся обработка, включая поиск шаблонов с использованием регулярных выражений, выполняется над символами, а не над столбцами или байтами. В командном режиме утилиты распознают аргументы, указывающие количество символов. Многостолбцовые символы при использовании всей ширины экрана. Для показа ширины отображаемых многостолбцовых символов, выдается такое же количество ASCII-символов >.

### Опции командной строки

Редактор **vi** воспринимает следующие опции командной строки (ранее описывавшиеся в документации опции рассмотрены в разделе "**ПРИМЕЧАНИЯ**" на этой странице справочного руководства):

<b>-t тэг</b>	Редактирует файл, содержащий тэг, и устанавливает редактор в позицию, соответствующую определению тэга. Примечание: тэги в файле тэгов должны идти в порядке возрастания номеров строк.
<b>-r файл</b>	Редактирует файл после сбоя редактора или системы. (Восстанавливает версию файла, которая была в буфере в момент возникновения сбоя.)
<b>-l</b>	Устанавливает настройки для редактирования программ на языке LISP.
<b>-L</b>	Выдает имена всех файлов, сохраненных в результате сбоя редактора или системы.

<b>-wn</b>	Устанавливает стандартный размер окна в <b>n</b> строк. Эта опция полезна при использовании редактора по низкоскоростной линии.
<b>-R</b>	Режим только для чтения; устанавливается флаг только для чтения, предотвращающий случайную перезапись файла. Опция шифрования; при ее задании <b>vi</b> имитирует команду <b>X</b> редактора <b>ex</b> и предлагает пользователю ввести ключ. Этот ключ используется для шифрования и дешифрования текста с помощью алгоритма команды <b>crypt</b> .
<b>-x</b>	Команда <b>X</b> сама определяет, является ли прочитанный текст зашифрованным или нет. Временный файл буфера также шифруется с использованием преобразованной версии введенного пользователем ключа. См. <b>crypt(1)</b> . См. также раздел " <b>ПРИМЕЧАНИЯ</b> " в конце данной страницы справочного руководства.
<b>-C</b>	Опция шифрования; аналогичная опции <b>-x</b> , но <b>vi</b> имитирует команду <b>C</b> редактора <b>ex</b> . Команда <b>C</b> аналогична команде <b>X</b> , но любой прочитанный текст считается зашифрованным.
<b>-с команда</b>	Начинает редактирование с выполнения указанной <b>команды</b> редактора (обычно, команды поиска или позиционирования).

Аргумент **файл** задает один или несколько файлов для редактирования.

Вызов **view** совпадает с **vi**, но при этом устанавливается флаг только для чтения.

Вызов **vedit** предназначен для начинающих. Он совпадает с **vi**, но при этом флаг **report** устанавливается равным 1, устанавливаются флаги **showmode** и **novice**, а флаг **magic** сбрасывается. Такие стандартные установки упрощают изучение **vi**.

## Режимы vi

Командный режим	Обычный и первоначальный режим. После завершения других режимов происходит возврат в командный режим. Для отмены частично введенной команды используется клавиша ESC.
Режим ввода	Устанавливается при вводе одной из следующих команд: <b>a</b> , <b>A</b> , <b>i</b> , <b>I</b> , <b>o</b> , <b>O</b> , <b>c</b> , <b>C</b> , <b>s</b> , <b>S</b> , <b>R</b> . Затем можно вводить произвольный текст. Режим ввода можно прекратить нормально, нажав клавишу ESC, или аварийно, с помощью прерывания.
Последняя строка	Чтение ввода для <b>:</b> , <b>/</b> , <b>?</b> или <b>!</b> ; заканчивается нажатием клавиши Enter; символ прерывание отменяет выход из режима.

## СВОДКА КОМАНД

В последующих описаниях **CR** означает нажатие клавиши Enter, а **ESC** - нажатие клавиши Esc.

## Примеры команд

стрелки	клавиши со стрелками перемещают курсор
<b>h</b> , <b>j</b> , <b>k</b> , <b>l</b>	то же, что и клавиши со стрелками
<b>і</b> текст <b>ESC</b>	вставка текста
<b>cw</b> новое <b>ESC</b>	замена слова на <b>новое</b>
<b>easESC</b>	преобразование слово в множественное число (перейти в конец слова; добавить <b>s</b> ; выйти из режима ввода)
<b>x</b>	удаление символа
<b>dw</b>	удаление слова
<b>dd</b>	удаление строки

<b>3dd</b>	удаление трех строк
<b>u</b>	отмена предыдущего изменения
<b>ZZ</b>	выход из <b>vi</b> с сохранением изменений
<b>:q!CR</b>	выход, изменения игнорируются
<b>/textCR</b>	поиск текста
<b>^U, ^D</b>	прокрутка вверх или вниз
<b>:cmdCR</b>	любая команда <b>ex</b> или <b>ed</b>

## Счетчики перед командами **vi**

Перед некоторыми командами можно указывать числа. Они интерпретируются одним из следующих способов.

номер строки/столбца	<b>z, G,  </b>
сколько экранов прокрутить	<b>^D, ^U</b>
количество повторений	большинство других команд

## Прерывание, отмена

<b>ESC</b>	прекратить вставку или отменить неполную команду
<b>DEL</b>	прерывание, выход из режима

## Манипулирование файлами

<b>ZZ</b>	если файл изменен, записать и выйти; в противном случае, выйти
<b>:wCR</b>	записать изменения
<b>:w!CR</b>	форсированная запись, если права доступа изначально ее не допускают
<b>:qCR</b>	выход
<b>:q!CR</b>	выход, игнорируя изменения
<b>:e имяCR</b>	редактировать файл <b>имя</b>
<b>:e!CR</b>	повторить редактирование, игнорируя изменения
<b>:e + имяCR</b>	редактировать, начиная с конца
<b>:e +n имяCR</b>	редактировать, начиная со строки <b>n</b>
<b>:e #CR</b>	редактировать другой файл
<b>:e! #CR</b>	редактировать другой файл, игнорируя изменения
<b>:w имяCR</b>	записать файл <b>имя</b>
<b>:w! имяCR</b>	перезаписать файл <b>имя</b>
<b>:shCR</b>	запустить командный интерпретатор, а затем вернуться
<b>!:командаCR</b>	выполнить <b>команду</b> , а затем вернуться
<b>:nCR</b>	редактировать следующий файл в списке аргументов
<b>:n argsCR</b>	задать новый список аргументов
<b>^G</b>	показать текущий файл и строку
<b>:ta тэгCR</b>	поместить курсор на указанный <b>тэг</b>

Вообще, можно ввести любую команду **ex** или **ed** (такую как подстановка или глобальная команда), если перед ней указать двоеточие и завершить ввод нажатием Enter.

## Перемещение по файлу

<b>^F</b>	на экран вперед
<b>^B</b>	на экран назад
<b>^D</b>	прокрутить вниз пол-экрана
<b>^U</b>	прокрутить вверх пол-экрана
<b>nG</b>	перейти в начало указанной строки (по умолчанию - последней), где <b>n</b> - номер строки
<b>/шаблон</b>	следующая строка, соответствующая <b>шаблону</b>
<b>?шаблон</b>	предыдущая строка, соответствующая <b>шаблону</b>
<b>n</b>	повторить последнюю команду / или ? в прямом направлении
<b>N</b>	повторить последнюю команду / или ? в обратном направлении
<b>/шаблон/+n</b>	<b>n</b> -ная строка после <b>шаблона</b>
<b>?шаблон?-n</b>	<b>n</b> -ная строка перед <b>шаблоном</b>
<b>]]</b>	следующий раздел/функция
<b>[[</b>	предыдущий раздел/функция
<b>(</b>	в начало предложения
<b>)</b>	в конец предложения
<b>{</b>	в начало параграфа
<b>}</b>	в конец параграфа
<b>%</b>	на соответствующую скобку ( , ), { или }

## Перерисовка экрана

<b>^L</b>	очистить и перерисовать окно
<b>^R</b>	очистить и перерисовать окно, если <b>^L</b> соответствует клавише стрелка право
<b>zCR</b>	перерисовать экран так, чтобы текущая строка была вверху окна
<b>z-</b>	перерисовать экран так, чтобы текущая строка была внизу окна
<b>z.</b>	перерисовать экран так, чтобы текущая строка была посередине окна
<b>/шаблон/z-CR</b>	переместить строку, соответствующую <b>шаблону</b> , вниз окна
<b>zn.</b>	использовать окно размером <b>n</b> строк
<b>^E</b>	прокрутить окно вниз на 1 строку
<b>^Y</b>	прокрутить окно вверх на 1 строку

## Пометка и возврат

<b>``</b>	переместить курсор на предыдущий контекст
<b>"</b>	переместить курсор на первый непробельный символ в строке
<b>mx</b>	пометить текущую позицию однобайтовой буквой нижнего регистра <b>x</b>
<b>`x</b>	переместить курсор на метку <b>x</b>
<b>'x</b>	переместить курсор на первый непробельный символ в строке, помеченной буквой <b>x</b>

## Перемещение по строкам

<b>H</b>	на верхнюю строку экрана
<b>L</b>	на последнюю строку экрана
<b>M</b>	на среднюю строку экрана
<b>+</b>	на следующую строку, на первый непробельный символ
<b>-</b>	на предыдущую строку, на первый непробельный символ
<b>CR</b>	то же, что и <b>+</b>
<b>v</b> или <b>j</b>	на следующую строку, в тот же столбец
<b>^</b> или <b>k</b>	на предыдущую строку, в тот же столбец

## Посимвольное перемещение

<b>^</b>	на первый непробельный символ
<b>0</b>	в начало строки

<b>\$</b>	в конец строки
<b>I</b> или <b>-&gt;</b>	на символ вперед
<b>h</b> или <b>&lt;-</b>	на символ назад
<b>^H</b>	то же, что и стрелка влево (клавиша Backspace)
<b>пробел то же, что и стрелка вправо (пробел)</b>	
<b>fx</b>	найти следующий однобайтовый символ <b>x</b>
<b>Fx</b>	найти предыдущий однобайтовый символ <b>x</b>
<b>tx</b>	перейти на символ перед следующим однобайтовым символом <b>x</b>
<b>Tx</b>	перейти на символ после предыдущего однобайтового символа <b>x</b>
<b>;</b>	повторить последнюю команду <b>f</b> , <b>F</b> , <b>t</b> или <b>T</b>
<b>,</b>	выполнить команду, противоположную последней из команд <b>f</b> , <b>F</b> , <b>t</b> или <b>T</b>
<b>n </b>	перейти в столбец <b>n</b>
<b>%</b>	перейти на соответствующую скобку

### Слова, предложения, абзацы

<b>w</b>	на слово вперед
<b>b</b>	на слово назад
<b>e</b>	в конец слова
<b>)</b>	на следующее предложение
<b>}</b>	на следующий абзац
<b>(</b>	на предыдущее предложение
<b>{</b>	на предыдущий абзац
<b>W</b>	вперед на слово, отделенное пробелом
<b>B</b>	назад на слово, отделенное пробелом
<b>E</b>	в конец слова, отделенного пробелом

### Исправления в ходе вставки

<b>^H</b>	стереть последний символ (забой - клавиша Backspace)
<b>^W</b>	стереть последнее слово
<b>erase</b>	символ стирания терминала, то же, что и <b>^H</b> (забой)
<b>kill</b>	символ удаления строки терминала, стирает введенную строку
<b>\</b>	маскирует символы стирания и удаления строки
<b>ESC</b>	завершает ввод, возвращает в командный режим
<b>DEL</b>	прерывание, прекращает режим ввода
<b>^D</b>	обратно на один символ табуляции; переустановить левую границу автоматического выравнивания
<b>^^D</b>	<i>циркумфлекс (^)</i> после которого идет <b>Ctrl-d (^D)</b> ; обратно в начало строки; не переустанавливать левую границу автоматического выравнивания
<b>0^D</b>	обратно в начало строки; переустановить левую границу выравнивания
<b>^V</b>	замаскировать непечатный символ

*vi* использует ввод через **Ctrl** управляющих символов ASCII, а также восьмеричную запись для непечатных 8-битовых символов.

### Вставка и замена

<b>a</b>	добавление после курсора
<b>A</b>	добавление в конце строки
<b>i</b>	вставка перед курсором
<b>I</b>	вставка перед первым непробельным символом
<b>o</b>	новая строка под текущей
<b>O</b>	новая строка над текущей

<b>gx</b>	заменяет один символ на однобайтовый символ <b>x</b>
<b>RтекстESC</b>	заменяет символы на введенный <b>текст</b>
<b>~</b>	изменяет верхний регистр на нижний и наоборот

## Операторы

После операторов выполняется перемещение курсора, и они влияют на весь перемещаемый текст. Например, поскольку **w** перемещает курсор на слово вперед, **dw** удаляет слово, через которое перескакивает курсор. Удвоение оператора, например, **dd**, оказывает воздействие на целые строки.

<b>d</b>	удаление
<b>c</b>	изменение
<b>y</b>	копирование строк в буфер
<b>&lt;</b>	сдвиг влево
<b>&gt;</b>	сдвиг вправо
<b>!</b>	фильтрация строк через команду

## Другие операции

<b>C</b>	изменить оставшуюся часть строки ( <b>c\$</b> )
<b>D</b>	удалить оставшуюся часть строки ( <b>d\$</b> )
<b>s</b>	подставить символы ( <b>cl</b> )
<b>S</b>	подставить строки ( <b>cc</b> )
<b>J</b>	соединить строки в одну
<b>x</b>	удалить символы ( <b>dl</b> )
<b>X</b>	удалить символы перед курсором ( <b>dh</b> )
<b>Y</b>	скопировать строки в буфер ( <b>yy</b> )

## Вырезание и вставка

Вставка вставляет последний удаленный или скопированный текст; однако, если используется именованный буфер (с использованием символа нижнего регистра ASCII **a-z**), вставляется текст из соответствующего буфера.

<b>3yy</b>	скопировать 3 строки
<b>3yl</b>	скопировать 3 символа
<b>p</b>	вставить текст после курсора
<b>P</b>	вставить текст перед курсором
<b>"xp</b>	вставка из буфера <b>x</b>
<b>"xy</b>	копирование в буфер <b>x</b>
<b>"xd</b>	вырезание в буфер <b>x</b>

## Отмена, повторное выполнение, извлечение

<b>u</b>	отменить последнее изменение
<b>U</b>	восстановить текущую строку
<b>.</b>	повторить последнее изменение
<b>"dp</b>	восстановить <b>d</b> -тое последнее удаление

## АВТОРЫ

**vi** и **ex** базируются на программном обеспечении, разработанном в Калифорнийском университете в Бэркли, отделение компьютерных наук, Department of Electrical Engineering and Computer Science.

## ФАЙЛЫ

**/var/preserve**

стандартный каталог, в котором размещаются временные рабочие файлы; его можно изменить с помощью опции **directory** (см. описание команды set в [ex\(1\)](#))

**/usr/share/lib/terminfo/?/\***

скомпилированная база данных описаний терминалов

**/usr/share/lib/.COREterm/?/\***

подмножество скомпилированной базы данных описаний терминалов

**/usr/lib/locale/локаль/LC\_MESSAGES/uxed.abi**

файл сообщений для текущего языка (См. **LANG** в [environ\(5\)](#).)

## ССЫЛКИ

[ctags\(1\)](#), [ed\(1\)](#), [edit\(1\)](#), [ex\(1\)](#)

## ПРИМЕЧАНИЯ

Две опции, хотя они по-прежнему поддерживаются, были заменены в документации на опции, соответствующие *стандарту синтаксиса команд* (Command Syntax Standard - см. [intro\(1\)](#)). Опция **-r** без аргументов заменена опцией **-L**, а опция **+команда** - на **-с команда**.

Опции шифрования и соответствующие команды поставляются в составе пакета утилит шифрования (Encryption Utilities), который доступен только в Соединенных Штатах.

Изменение записей в файле **/usr/share/lib/terminfo/?/\*** или **/usr/share/lib/.COREterm/?/\*** (например, изменение или удаление некоторых записей) может повлиять на программы типа **vi**, предполагающие наличие корректной записи. В частности, удаление "тупого" ("**dumb**") терминала может вызвать неожиданные проблемы.

Программные табуляции, задаваемые с помощью **^T**, срабатывают только после автоматического выравнивания (autoindent).

Сдвиги влево и вправо на интеллектуальных терминалах не используют операций, задаваемых символами вставки или удаления терминала.

Copyright 1994 Novell, Inc.  
Copyright 2000 В. Кравчук, OpenXS Initiative, перевод на русский язык