

Интерфейс командной строки UNIX

В. Е. Сорокин

1 Приёмы работы с командной строкой

1.1 Нотация

В документации к программному обеспечению для UNIX-подобных операционных систем для обозначения комбинаций клавиш принято пользоваться своеобразной нотацией, отличающейся от привычной пользователям DOS и Windows.

$\wedge m$	Ctrl+m
Meta-m или M-m	Alt+m или Esc, m

Специальной клавиши **Meta** на клавиатуре нет, ибо такие комбинации реализуются двумя способами. Комбинации с клавишей **Alt** безотказно срабатывают на серверных платформах, там где нет графического интерфейса пользователя. Но в условиях графического окружения, комбинации с клавишей **Alt** зачастую приводят к неожиданным результатам, поскольку эта клавиша, как и в окружении Windows, задействована для работы со всевозможными меню. Именно для таких случаев, когда графический пользовательский интерфейс мешает использовать клавишу **Alt**, и придумали вторую комбинацию, с клавишей **Esc**. Удерживать клавишу **Esc**, подобно клавишам **Alt** или **Ctrl**, не нужно. Клавиши нажимаются последовательно: сначала нажатие клавиши **Esc**, затем, отпустив её, — нажатие второй клавиши, например, **m**.

1.2 Навигация по командной строке

$\wedge a$	Перемещение курсора в начало строки, аналог Home
$\wedge e$	Перемещение курсора в конец строки, аналог End
$\wedge b$	Перемещение курсора на символ назад, аналог клавиши ←
Meta-b	Перемещение курсора на слово назад
$\wedge f$	Перемещение курсора на символ вперёд, аналог клавиши →
Meta-f	Перемещение курсора на слово вперёд

1.3 Редактирование командной строки

Tab	Автодополнение
$\wedge d$	Аналог клавиши Del; в буфер обмена не кладёт
$\wedge h$	Аналог клавиши Backspace; в буфер обмена не кладёт
$\wedge w$	Удалить слово слева от курсора; можно использовать другую комбинацию — Meta-Backspace
Meta-d	Удалить слово справа от курсора
$\wedge l$	Очистить экран, набираемая команда не изменяется
$\wedge k$	Очистить строку от текущего положения курсора до конца строки, и положить удалённый стринг в буфер

<code>^u</code>	Очистить строку от начала строки до текущего положения курсора, и положить удалённый стринг в буфер; в BSD — очистка всей строки
<code>^y</code>	Вклеить содержимое буфера, начиная с текущего положения курсора
<code>^_</code>	UNDO, отменить изменение; это комбинация клавиш <code>Ctrl+Shift+-</code>
<code>^t</code>	Поменять два символа местами. Встать на правый символ, далее — <code>^t</code> , в результате символы поменяются местами, а курсор переместится на следующую позицию
<code>Meta-l</code>	Перевод слова в нижний регистр
<code>Meta-u</code>	Перевод слова в верхний регистр
<code>Meta-c</code>	«Капитализация» слова: сделать первую букву заглавной
<code>^p</code>	«вверх» по истории
<code>^n</code>	«вниз» по истории

2 Подсказка для работы с редактором vi

Основная трудность при работе с `vi` — это многорежимность и неочевидность текущего режима. Ничего страшного в этом нет. Следует только помнить, что `vi` — самый первый экранный текстовый редактор, и, по сути, представляет собою строковый редактор с обложкой «вокруг» него. С точки зрения современного пользователя работа в `vi` выглядит несколько необычно.

- Правая часть клавиатуры в общем случае может частично или полностью не работать.
- Сразу после запуска — командный режим. Нажатие любой клавиши — команда. Поэтому такой режим часто называют «командным».
- Для набора текста следует переместить курсор в нужную позицию, и переключить редактор в режим «редактирования». Далее — обычный набор текста.
- По завершении работы с блоком текста, следует вернуться в командный режим, и переместить курсор в новую позицию, и т.д.

Именно в связи с тем, что правая часть клавиатуры в традиционных UNIX-системах не работает, важно запомнить команды переключения между режимами, иначе работать с `vi` затруднительно.

2.1 Переключение между режимами

<code>Esc</code> или <code>^c</code>	Переключение в командный режим.
<code>Ins</code> или <code>i</code>	Редактирование в режиме вставки с текущей позиции.
<code>a</code>	Переход в режим редактирования со следующей позиции.
<code>A</code>	Переход в режим редактирования с конца строки.
<code>o</code>	Вставка новой строки <i>под</i> текущей, и переход в режим редактирования.
<code>O</code>	Вставка новой строки <i>над</i> текущей, и переход в режим редактирования.
<code>r</code>	Замена единственного символа в текущей позиции, без перехода в режим редактирования.

Дополнительные возможности редактора vim

- В редакторе `vim` можно набирать текст в режиме «вставки» и «замены».

- Повторные нажатия **Ins** будут переключать режимы редактирования между вставкой и заменой.
- Клавиша **R** переводит из командного режима в режим редактирования в режиме «замены» с текущей позиции.
- Текущий режим редактирования отображается в левом нижнем углу.
- Текущее положение курсора в тексте отображается в правом нижнем углу.

2.2 Командный режим. Перемещения по тексту

Там, где не работает правая часть клавиатуры, перемещения по тексту осуществляются в командном режиме. Тогда четыре клавиши, расположенные по соседству друг с другом, служат для перемещения по тексту.

h	крайняя левая из четырёх — перемещение на одну позицию влево, аналог клавиши \leftarrow (крайняя слева);
j	похожа на стрелку \downarrow — перемещение на одну строку вниз, аналог клавиши \downarrow ;
k	от слова «кайф» — перемещение на одну строку вверх, аналог клавиши \uparrow ;
l	крайняя правая из четырёх — перемещение на одну позицию вправо, аналог клавиши \rightarrow (крайняя справа);
\$	Курсор — в конец строки, аналог клавиши End ;
0	Курсор — в начало строки, аналог клавиши Home ;
H	Курсор — на первую строку экрана;
L	Курсор — на последнюю строку экрана;
G	Курсор — на последнюю строку файла;
1G	Курсор — на первую строку файла;
nG	Курсор — на n-ую строку файла;

Листание

^u	Прокрутка текста на пол-экрана вверх, от Up .
^d	Прокрутка текста на пол-экрана вниз, от Down .
^f	Прокрутка текста на экран вперёд, к концу файла, от Forward .
^b	Прокрутка текста на экран назад, к началу файла, от Back .

2.3 Повторитель

Перед любой командой в командном режиме может быть указано число повторений команды. Например, если в командном режиме набрать **25j**, то курсор переместится на 25 строк вниз. Команда **12dd** удалит 12 строк, начиная с текущей. Про повторитель важно не забывать, это делает из, странного на первый взгляд редактора, один из удобнейших текстовых процессоров.

2.4 Командный режим. Редактирование

d1 или x	Эквивалентно нажатию клавиши Del .
d0	Удаление от текущей позиции до начала строки.
d\$ или D	Удаление от текущей позиции до конца строки.
dd	Удаление строки. 24dd — удалить 24 строки.

Примечание. Все удаления попадают в буфер обмена.

~	Изменение регистра символа.
J	Объединение текущей строки со следующей, ниже по тексту.
u	Откат. vi отменяет только последнее изменения текста, vim — последовательно отменяет все изменения, происходившие с момента последнего сохранения текста.
NyD	Взять N символов, начиная с текущего, в буфер обмена. D задаёт направление отсчёта символов. Например: 6y1 — взять в буфер обмена 6 символов, вправо от текущего, а 11yh — взять 11 символов влево от текущего.
yy	Взять строку в буфер обмена. Нуу — взять в буфер обмена N строк, начиная с текущей.
p	Вставка содержимого буфера <i>после</i> текущей позиции курсора. Строчное p
P	Вставка содержимого буфера <i>перед</i> текущей позицией курсора. Прописное P

Дополнительные возможности редактора vim

v	Начало визуального выделения посимвольно.
^v	Начало выделения в режиме редактирования. Блок текста выделяется просто перемещением курсора, затем — команда взятия в буфер y, с последующим переходом в командный режим.
V	Визуальное выделение построчно.
Y	Строка — в буфер обмена. Повторитель задаёт число строк.

2.5 Командная строка редактора

:w [filename]	Сохранить файл. При необходимости, указать имя файла. Можно использовать >> для дописывания в конец файла.
:e filename	Открыть файл filename для редактирования.
:e!	Что-то пошло не так? Отбросить все изменения, и заново открыть файл с диска.
:q	Закрыть файл и выйти.
:wq	Сохранить файл, закрыть и выйти.
:q!	Закрыть файл и выйти не сохраняя.
:help	Справка.
/pattern	Поиск по шаблону pattern вперёд.
?pattern	Поиск по шаблону pattern назад.
:s/шаблон/замена/	Замена первого совпадения с шаблоном в текущей строке.
:s/шаблон/замена/g	Замена <i>всех</i> совпадений с шаблоном в текущей строке.
:%s/шаблон/замена/g	Замена <i>всех</i> совпадений с шаблоном по всему файлу.

Дополнительные возможности редактора vim

:hardcopy	Твёрдая копия, печать.
-----------	------------------------

Если на редактирование открыто сразу несколько файлов, чтобы перемещаться между ними:

:next	Перейти к редактированию следующего файла.
:previous	Перейти к редактированию предыдущего файла.

Редактор `vim` также позволяет работать с несколькими окнами на одном экране.

<code>^w+n</code>	Открыть новое окно.
<code>^w+w</code>	Перейти в следующее окно.

Примечание 1. Вместе с редактором `vim` в систему, как правило, устанавливается программа `vimtutor`. Представляет собою короткий интерактивный «курс молодого бойца». Занимает 20-30 минут времени. Настоятельно рекомендуется для начального обучения.

Примечание 2. Редактор `vim` великолепно документирован, и любую справку можно получить не выходя из редактора, пользуясь встроенной справкой. Возможности этого редактора огромны. Это одна из самых мощных программ-редакторов. Содержание справки тщательно продумано, прекрасно структурировано, и написано на отличном `basic english`, понятном не только носителям языка, но и пользователям всего мира. Освоение даже малой толики возможностей этого редактора превращает работу с ним — в удовольствие.