


# 1 Базовые команды







## 1.1 Перемещения

1.  $(n)h$  — влево на 1 символ
2.  $(n)l$  — вправо на 1 символ
3.  $(n)j$  — вниз на 1 символ
4.  $(n)k$  — вверх на 1 символ

## 1.2 Перемещения по слову и строке

1.  $(n)w ((n)W)$  — в начало следующего слова
2.  $(n)e ((n)E)$  — в конец текущего слова
3.  $(n)b ((n)B)$  — в начало текущего слова
4.  $\$$  — в конец строки
5.  $0$  — в начало строки
6.  $\wedge$  — на 1-й непустой символ строки
7.  — на 1-й непустой символ след. строки

## 1.3 Перемещения по экранам


1.  + F — вперед на 1 экран
2.  + B — назад на 1 экран
3.  + D — вперед на 1/2 экрана
4.  + U — назад на 1/2 экрана
5.  + E — на 1 строку вверх (без перемещения курсора)
6.  + Y — на 1 строку вниз (без перемещения курсора)
7.  $z + \text{Enter}$  — переместить текущ. строку в верх экрана
8.  $z .$  — переместить текущ. строку в центр экрана
9.  $z -$  — переместить текущ. строку в низ экрана

Сочетаются с (c), (d), (y):

1. H — переход в начало экрана
2.  $(n)H$  — на  $n$  строк ниже верхней
3. L — переход в конец экрана
4.  $(n)L$  — на  $n$  строк выше нижней
5. M — переход в середину экрана

## 1.4 Перемещения по тексту

Сочетаются с (c), (d), (y), (e), (E), кроме gg, G:

1.  $(n)(, )$  — в начало текущего/след. предложения
2.  $(n)\{, \}$  — в начало текущего/след. абзаца
3.  $(n)[[, ]]$  — в начало текущего/след. раздела
4. gg — в начало файла
5. G — в конец файла
6.  $(n)G (n + \text{Ctrl} + G)$  — на строку  $n$
7.  + G — узнать номер строки
8.  $(n)+$  — на 1-й символ след. строки
9.  $(n)-$  — на 1-й символ предыдущей строки

## 1.5 Поиск

1.  $(n)f(\text{символ})$  —  $n$ -е вхождение символа в строке
2.  $(d, c, y)/(\text{текст})$  — поиск фразы в документе
3.  $(d, c, y)?(\text{текст})$  — поиск фразы назад
4. n — повторить поиск (в том же направлении)
5. N — повторить поиск (в обратном направлении)
6.  $/ + \text{Enter}$  — повторить вперед от текущего направления
7.  $? + \text{Enter}$  — повторить назад от текущего направления
8. % — поиск парного символа (скобки)
9.  $(n)^*$  — поиск следующего слова под курсором
10.  $(n)\#$  — поиск слова под курсором в обратн. напр.

## 1.6 Поиск в текущей строке

Сочетаются с (c), (d), (y), кроме ;, ':

1.  $(n)f(x)$  — следующее ( $n$ -е) вхождение  $x$
2.  $(n)F(x)$  — предыдущее ( $n$ -е) вхождение  $x$
3.  $(n)t(x)$  — символ перед след. вхождением  $x$
4.  $(n)T(x)$  — символ перед пред. вхождением  $x$
5. ; — повторить поиск в пределах строки
6. ' — повторить поиск в обратном направлении

## 1.7 Вставка и добавление текста

1. **i** — начать вставлять текст на место курсора
2. **I** — вставлять текст в начале строки
3. **(n)i** — вставить текст *n* раз
4. **p (P)** — вставить строку из буфера
5. **a** — добавить текст после курсора
6. **A** — добавить текст в конце строки
7. **(n)o** — создать пустую строку под курсором
8. **(n)O** — создать пустую строку над курсором
9. **"(bufname)p** — вставка из определенного буфера

## 1.8 Замена текста

1. **r** — заменить имеющийся символ
2. **R** — заменить более одного символа
3. **(n)cw** — замена слова после курсора
4. **(n)cb** — замена слова до курсора
5. **c\$ (C)** — замена до конца строки
6. **c0** — замена от начала строки
7. **(n)cc** — замена всей строки
8. **c%** — от курсора до конца блока скобок
9. **(n)** — смена регистра букв
10. **(n)s** — заменить неск. символов
11. **(n)S** — удалить строку и вставить текст
12. **(n)J** — объединить строку со следующей

## 1.9 Удаление текста

Считается вырезанием в буфер обмена:

1. **(n)x** — удалить символ на курсоре
2. **(n)X** — удалить символ перед курсором
3. **(n)dw** — удалить слово после курсора
4. **d\$ (D)** — строку после курсора
5. **d0** — удалить строку до курсора
6. **(n)de** — от курсора до начала другого слова
7. **(n)db** — удалить слово до курсора
8. **(n)dd** — удалить всю строку
9. **di** — удалить внутри скобок
10. **d%** — удалить блок со скобками
11. **(n)d(h, l, j, k)** — с перемещением посимвольно
12. **"(bufname)(n)d(command)** — удаление в именованные буферы

## 1.10 Копирование текста

1. **(n)yw ((n)yW)** — слово после курсора
2. **(n)yb ((n)yB)** — скопировать слово до курсора
3. **(n)yy ((n)Y)** — скопировать всю строку
4. **y\$** — скопировать строку после курсора
5. **y0** — скопировать строку до курсора
6. **(n)ye** — скопировать до начала другого слова
7. **yi** — скопировать внутри скобок
8. **y%** — скопировать блок со скобками
9. **(n)y1 (yh)** — скопировать символ
10. **"(bufname)(n)y(command)** — копирование в определенный буфер (от *a* до *z*)

## 1.11 Повторение и отмена команд

1. **(n).** — повторение команды
2. **(n)u** — отмена команды
3. **U** — отмена изменений в строке
4. **[Ctrl] + R** — откат отката
5. **"(bufnum)p** — восстановить из буферов
6. **"1ru.u.u... — перебор буферов один за другим (+ u.)**

## 1.12 Отметка места

1. **mx** — поставить метку *x* сюда
2. **'x** — поставить курсор на 1-й символ строки с меткой
3. **`x** — поставить курсор на помеченный символ
4. **"** — поставить курсор на предыдущую метку (место)
5. **"** — поставить курсор на строку предыдущей метки (места)

## 1.13 Команды открытия файла

1. **\$ vi file** — открыть файл
2. **\$ vi +n file** — открыть файл на строке *n*
3. **\$ vi + file** — открыть файл на последней строке
4. **\$ vi +/text file** — открыть файл на вхождении *text* (если в шаблоне пробелы, то шаблон заключается в кавычки)
5. **\$ vi -R file** — открыть файл в режиме чтения

## 2 Команды редактора **ex**

### 2.1 Базовые команды

1. `:(n)p` — показать  $n$ -ю строку
2. `:(n,m)p` — показать диапазон строк
3. `:vi` — открыть редактор `vi`
4. `:delete (:d)` — удаление строк
5. `:move (:m)` — перемещение строк
6. `:copy (:co, :t)` — копирование строк
7. `:(n,m)d,m,t` — действия с диапазоном строк
8. `:=` — вывести полное число строк
9. `:.=` — вывести номер текущей строки
10. `:/ (text)/=` — вывести номер строки шаблона

### 2.2 Адресация строк

1. `.., $ d` — удалить с текущей строки до конца
2. `:%d` — удалить все строки в файле
3. `:(n), .m$` — переместить строки с  $n$  до текущей в конец файла
4. `:%t$` — скопировать все строки и поместить их в конец
5. `.., .±(n)d` — удалить строки с текущей по  $n$
6. `:(n), $m.-(m)` — переместить строки с  $m$  до конца на строку, выше текущей на  $n$
7. `.., +(n)#` — отобразить номера строк с текущей до  $n$ -й
8. `..-, .+t0` — копировать строки от строки выше курсора до строки ниже него в начало файла

### 2.3 Шаблоны поиска

1. `:/ (text)/d` — удалить строку с шаблоном
2. `:/ (text)/+d` — удалить строку ниже строки с шаблоном
3. `:/ (text1)/, /(text2)/d` — удалить от строки с 1-м шаблоном до строки со 2-м
4. `.., /(text)/m(n)` — переместить текст от текущей строки до строки  $n$

### 2.4 Глобальный поиск

1. `:g/(text)` — последнее вхождение шаблона в файле
2. `:g/(text)/p` — найти и вывести все строки с шаблоном
3. `:g!/(text)nu` — найти и вывести все строки с их номерами, не содержащие шаблон
4. `:(n,m)]g/(text)/p` — вывести все строки с шаблоном с номерами из диапазона

### 2.5 Копирование одного файла в другой

1. `:(n)read file` — считать содержимое другого файла в строку  $n$  (можно  $\$$  или  $0$ )
2. `:/ (text)/read file` — поместить файл в текущий после строки с шаблоном

### 2.6 Глобальная замена

1. `:s/old/new` — заменить первое вхождение текста
2. `:s/old/new/g` — заменить во всех вхождениях
3. `:(n,m)s/old/new/g` — заменить во всех вхождениях в диапазоне строк
4. `:1, $s/old/new/g` — замена во всем файле
5. `:(n,m)s/old/new/gc` — спрашивать при каждой замене
6. `:g/(text)/s/old/new/g` — поиск и замена по разным шаблонам

### 2.7 Поиск по регулярным выражениям: метасимволы

1. `.` — одиночный символ (и пробел)
2. `*` — повторение любого кол-ва раз символа перед или после `*`
3. `^` — если в начале РВ, то требуется, чтобы следующее РВ находилось в начале строки; если не в начале РВ, то обозначает себя же; если стоит в начале скобок `[ ]`, то меняет их значение на обратное (все, кроме этих)
4. `$` — если в конце РВ, то требуется, чтобы следующее РВ было в конце строки; не в конце РВ означает себя же
5. `\` — последующий спец. символ должен рассматриваться как обычный
6. `[ ]` — соответствует любому из символов в скобках; диапазон задается тире; можно размещать несколько диапазонов вместе В скобках необходимо экранизировать лишь `\`, `-` и `]`.

7. \ ( \) — сохраняет шаблон между ними во временном буфере (9 штук — \1... \9)
8. \< \> — символы в начале или в конце слова; можно исп. не в паре; использование сразу обоих (конструкция «только если шаблон является словом целиком») приведет к поиску текстов, окруженных всем, чем угодно
9. — PB, использовавшееся в последнем поиске; работает только при обычном поиске /

## 2.8 Метасимволы в строках замены

При глобальной замене метасимволы PB имеют особое значение только в строке поиска (первой части) команды. Поэтому следует определить регулируемую строку замены.

1. \n — заменяется текст, соотв. *n*-му буферу
2. \ — требуется, чтобы след. спец. символ рассматривался как обычный
3. & — находясь в строке замены, заменяется на весь текст, соотв. строке поиска
4. — найденная строка заменяется на текст, опред. в последней команде подстановки
5. \u, \l — требуется, чтобы след. символ в строке замены исправлялся на прописной или строчный
6. \U, \L, \e, \E — все след. символы преобр. в прописные или строчные, пока не встретится конец строки замены или \e, \E.

## 3 В программировании

### 3.1 Отступы

1. [Ctrl] + T — перевести курсор на след. уровень отступа
2. [Ctrl] + D — вернуть курсор на пред. уровень отступа
3. ^ + [Ctrl] + D — курсор назад на начало текущей строки; следующая начнется на автоматическом отступе; полезно на директивах препроцессора
4. 0 + [Ctrl] + D — переместить курсор на начало строки со сбросом уровней отступа
5. (n)» — сместить *n* строк на 8 пробелов вправо
6. (n)« — сместить *n* строк на 8 пробелов влево

## 4 Команды Vim

### 4.1 Варианты запуска Vim

1. vim -b *file* — редактирование в двоичном режиме
2. vim -c *command file* — выполнится команда редактора *ex*
3. vim -C *file* — запуск в режиме совместимости с vi
4. vim -d (vimdiff) *file* — режим поиска разный diff
5. vim -E *file* — улучшенный режим *ex* (расширенные PB и т.д.)
6. vim -g *file* — графический режим
7. vim -m *file* — отключение режима записи: нельзя изменять буферы
8. vim -o (vim -O) *file* — открывать все файлы в отдельных окнах (при -O окна разделены вертикально)
9. vim -z (rvim) *file* — ограниченный режим (нет доступа к системным функциям и внешним интерфейсам)



### 4.2 Команды перемещения

1. [Ctrl] + [End] (*count*) — перейти в конец файла (на последний символ строки *count*)
2. [Ctrl] + [Home] — перейти на 1-й непробельный символ 1-ой строки
3. *count*% — перейти на соотв. процент файла; курсор помещается на 1-й печатаемый символ
4. ::go *n* (*n* go) — перейти на *n*-й байт в буфере

### 4.3 Визуальный режим

1. v — включение режима
2. (n)aw ((n)aW) — выделить *n* слов и пробелов (если они есть); w ищет слова, разделенные знаками препинания, W — разделенные пробелами
3. (n)iw ((n)iW) — выбрать *n* слов; w ищет слова, ограниченные знаками препинания, W — пробелами
4. as, is — добавить предложение или внутреннее предложение (сам текст без окружающих его непечатаемых и т.п. символов)
5. ap, ip — добавить абзац или внутренний абзац

#### 4.4 Расширенные регулярные выражения

1. `\|` — указывает на варианты слов
2. `\+` — соотв. одному или более предшествующим РВ
3. `\=` — соотв. одному или ни одному из предшествующих РВ
4. `\{n,m\}` — соотв. макс. кол-ву предшествующих РВ в диапазоне  $n-m$  (от 0 до 32000)
5. `\{n\}` — соотв.  $n$  предшествующим РВ
6. `\{n,\}` — соотв. как можно большему кол-ву предшествующих РВ, но не меньше  $n$
7. `\{,m\}` — соотв. как можно большему кол-ву предшествующих РВ, в диапазоне  $0-m$
8. `\{-n,m\}` — соотв. минимальному кол-ву предшествующих РВ, в диапазоне  $n-m$
9. `\{-n\}` — соотв.  $n$  предшествующим РВ
10. `\{-n,\}` — соотв. наименьшему кол-ву предшествующих РВ, но не меньше  $n$
11. `\{-,m\}` — соотв. наименьшему кол-ву предшествующих РВ, в диапазоне  $0-m$
12. `\i` — соотв. любому символу идентификатора согласно опции `isident`
13. `\I` — как `\i`, но исключая цифры
14. `\k` — соотв. любому ключевому слову согласно опции `iskeyword`
15. `\K` — как `\k`, но исключая цифры
16. `\f` — соотв. любому символу имени файла согласно опции `isfname`
17. `\F` — как `\f`, но исключая цифры
18. `\p` — соотв. любому печатаемому символу согласно опции `isprint`
19. `\P` — как `\p`, но исключая цифры
20. `\s` — соотв. любому пробельному символу
21. `\S` — соотв. всему, что не явл. пробелом или табуляцией
22. `\b` — символ Backspace
23. `\e` — 
24. `\r` — возврат каретки
25. `\t` —  (табуляция)
26. — соотв. последней использовавшейся строке замены
27. `\(...\)` — группировка для `*`, `\+` и `\=`, также делает доступным подтекст в команде замены (`\1` и т.д.)
28. `\1` — соотв. той же строке, кот. соотв. первому подвыражению в `\(...\)`