

1 Запуск Emacs

Для запуска GNU Emacs 29, просто наберите имя программы: `emacs`

2 Выход из Emacs

Приостановить работу Emacs (или свернуть в иконку при работе в X)	C-z
Выйти из Emacs	C-x C-c

3 Файлы

открыть файл в Emacs	C-x C-f
сохранить файл обратно на диск	C-x C-s
сохранить все файлы	C-x s
вставить содержимое другого файла в текущий буфер	C-x i
открыть нужный вам файл вместо текущего	C-x C-v
сохранить буфер в указанном файле	C-x C-w
Переключить состояние буфера “только для чтения”	C-x C-q

4 Получение помощи

Справочная система достаточно проста. Нажмите C-h (или F1) и следуйте инструкциям. Если вы в первый раз работаете с Emacs, то наберите C-u C-h t Russian RET для вызова **учебника**.

Скрыть окно справки	C-x 1
Прокрутить окно справки	C-M-v
argoros: показать команды, соответствующие строке	C-h a
показать имя функции, которая запускается клавишей	C-h k
получить информацию о функции	C-h f
получить информацию о режиме	C-h m

5 Исправление ошибок

прервать частично набранную или исполняемую команду	C-g
восстановить файл, потерянный при крахе системы	M-x recover-file
отменить нежелательные изменения	C-x u, C-_ или C-/
восстановить буфер в первоначальное состояние	M-x revert-buffer
перерисовать заполненный мусором экран	C-l

6 Нарастиваемый поиск

поиск вперед	C-s
поиск назад	C-r
поиск по регулярному выражению вперед	C-M-s
поиск по регулярному выражению назад	C-M-r
выбрать предыдущую строку поиска	M-p
выбрать следующую строку поиска	M-n
прекратить нарастиваемый поиск	RET
отменить эффект ввода последнего символа	DEL
прервать текущий поиск	C-g

Для повтора поиска в любом направлении используйте клавиши C-s или C-r. Если Emacs все еще производит поиск, C-g отменит только не выполненную часть.

7 Перемещение

единицы перемещения	назад	вперед
символ	C-b	C-f
слово	M-b	M-f
строка	C-p	C-n
начало (или конец) строки	C-a	C-e
предложение	M-a	M-e
абзац	M-{	M-}
страница	C-x [C-x]
выражение (sexp)	C-M-b	C-M-f
функция	C-M-a	C-M-e
переход в начало (или конец) буфера	M-<	M->
прокрутка экрана вниз	C-v	
прокрутка экрана вверх	M-v	
прокрутка влево	C-x <	
прокрутка вправо	C-x >	
установка текущей строки в центр экрана	C-u C-l	

8 Уничтожение и удаление

уничтожаемый объект	назад	вперед
символ (удаление, а не уничтожение)	DEL	C-d
слово	M-DEL	M-d
строка (до конца строки)	M-o C-k	C-k
предложение	C-x DEL	M-k
выражение	M- C-M-k	C-M-k
уничтожить область	C-w	
скопировать область в список уничтожений	M-w	
уничтожить до следующего вхождения символа <i>char</i>	M-z <i>char</i>	
вставить последний уничтоженный объект	C-y	
заменить только что восстановленный текст предшествующим уничтоженным текстом	M-y	

9 Пометка

установить пометки	C-@ или C-SPC
поменять местами курсор и пометку	C-x C-x
пометить <i>arg</i> слов	M-@
пометить параграф	M-h
пометить страницу	C-x C-p
пометить выражение	C-M-@
пометить функцию	C-M-h
пометить весь буфер	C-x h

10 Замена с запросом

интерактивная замена текстовой строки	M-%
с использованием регулярных выражений	M-x query-replace-regexp
В режиме интерактивной замены пользователю доступны следующие команды:	
заменить данное соответствие, перейти к следующему	SPC
заменить это соответствие, не перемещаться	,
пропустить соответствие без замены	DEL
заменить все оставшиеся соответствия	!
вернуться к предыдущему соответствию	^
выйти из режима “замена с запросом”	RET
войти в режим рекурсивного редактирования (для выхода используется	C-r
C-M-c)	

11 Использование нескольких окон

Команды, приведенные во втором столбце, применяются к другому фрейму (окну используемой оконной системы).

удалить все остальные окна	C-x 1	
разбить окно по горизонтали	C-x 2	C-x 5 2
удалить данное окно	C-x 0	C-x 5 0
разбить окно по вертикали	C-x 3	
выполнить прокрутку в другом окне	C-M-v	
переместить курсор в другое окно	C-x o	C-x 5 o
выбрать буфер в другом окне	C-x 4 b	C-x 5 b
показать буфер в другом окне	C-x 4 C-o	C-x 5 C-o
загрузить файл в другое окно	C-x 4 f	C-x 5 f
открыть файл в режиме чтения в другом окне	C-x 4 r	C-x 5 r
запустить Dired в другом окне	C-x 4 d	C-x 5 d
найти таг в другом окне	C-x 4 .	C-x 5 .
увеличить высоту окна	C-x ^	
сузить текущее окно	C-x {	
расширить текущее окно	C-x }	

12 Форматирование

сделать отступ для текущей строки (в зависимости от режима)	TAB
сделать отступ для области (в зависимости от режима)	C-M-\
сделать отступ для выражения (в зависимости от режима)	C-M-q
сделать отступ к заданной <i>arg</i> колонке	C-x TAB
вставить новую строку после курсора	C-o
сдвинуть остаток строки вертикально вниз	C-M-o
удалить пустые строки вокруг курсора	C-x C-o
объединить строку с предыдущей (со следующими <i>arg</i>)	M-^
удалить пустое пространство вокруг курсора	M-\
поместить только один пробел в позиции курсора	M-SPC
отформатировать текущий параграф	M-q
установить столбец заполнения	C-x f
установить префикс заполнения каждой строки	C-x .
установить начертание (<i>face</i>)	M-g

13 Изменение регистра

сделать буквы слова заглавными	M-u
сделать буквы слова строчными	M-l
сделать первую букву слова заглавной	M-c
сделать буквы области заглавными	C-x C-u
сделать буквы области строчными	C-x C-l

14 Минибуфер

Следующие клавиши (ключи) определены в минибуфере.

дополнить насколько возможно	TAB
дополнить до одного слова	SPC
дополнить и выполнить	RET
показать возможные дополнения	?
выбрать предыдущую строку из истории ввода	M-p
выбрать следующую строку из истории ввода	M-n
поиск в истории по регулярному выражению в обратном направлении	M-r
поиск в истории по регулярному выражению в прямом направлении	M-s
прервать команду	C-g

Наберите C-x ESC ESC для редактирования и повторения последней команды набранной в минибуфере. Наберите F10, чтобы работать с пунктами меню при помощи команд минибуфера.

15 Буфера

выбрать другой буфер	C-x b
выдать список всех буферов	C-x C-b
уничтожить буфер	C-x k

16 Перестановка данных

обменять местами символы	C-t
обменять местами слова	M-t
обменять местами строки	C-x C-t
обменять местами выражения	C-M-t

17 Проверка правописания

проверить текущее слово	M- $\$$
проверить все слова в области	M-x ispell-region
выполнить проверку всего буфера	M-x ispell-buffer

18 Теги

найти определение тега	M-.
использовать новый файл с тегами	M-x visit-tags-table
поиск по шаблону по всей таблице тегов	M-x tags-search
выполнить query-replace над всеми файлами	M-x tags-query-replace
продолжить поиск или поиск-замену тега	M-,

19 Командные процессоры

Выполнить команду командного процессора	M-!
выполнить команду командного процессора над областью	M-
пропустить содержимое области через команду командного процессора	C-u M-
запустить командный процессор в окне *shell*	M-x shell

20 Прямоугольные области

скопировать прямоугольную область в регистр	C-x r r
удалить прямоугольную область	C-x r k
вставить последнюю уничтоженную прямоугольную область	C-x r y
открыть прямоугольную область, сдвигая текст вправо	C-x r o
очистить прямоугольную область	C-x r c
предварить каждую строку строкой текста	C-x r t

21 Сокращения

добавить глобальное сокращение	C-x a g
добавить локальное для режима сокращение	C-x a l
добавить глобальное значение для данного сокращения	C-x a i g
добавить специфическое для режима значение данного сокращения	C-x a i l
явно вставить значение для сокращения	C-x a e
динамически дополнить предыдущее слово	M-/

22 Регулярные выражения

любой одиночный символ (за исключением символа новой строки)
 ноль или более повторений
 одно или более повторений
 ноль или одно повторение
 экранировать специальный символ *c*
 выбор между альтернативами (“или”)
 группировка
 текст в *n*-й группе
 граница слова
 не граница слова
объект

строка
 слово
 буфер

класс символов

явный набор
 символ-буква
 символ с синтаксисом *c*

. (dot)
 *
 +
 ?
 \c
 \|
 \(... \)
 \n
 \b
 \B

начало	соот-	конец	соот-
ветствия		ветствия	
^		\$	
\<		\>	
\‘		\’	
соответ-	соответ-	соответ-	соответ-
ствует им		ствует	дру-
		гим	
[...]		[^...]	
\w		\W	
\sc		\Sc	

23 Международные наборы символов

выбрать язык
 показать все возможные методы ввода
 разрешить или запретить метод ввода
 установить систему кодирования для следующей команды
 показать список всех систем кодирования
 выбрать предпочтительную систему кодирования

C-x RET l
 M-x list-input-methods
 C-\
 C-x RET c
 M-x list-coding-systems
 M-x prefer-coding-system

24 Информация (система Info)

запустить систему доступа к документации
 найти указанную функцию или переменную в документации
 Перемещение внутри узла (нодами):
 прокрутка вперед
 прокрутка назад
 перейти к началу узла
 Перемещение между узлами:
следующий узел
предыдущий узел
 перейти к **верхнему** узлу
 выбрать пункт меню по имени
 выбрать *n*-й пункт меню по номеру (1–9)
 перейти по ссылке (возврат с помощью 1)
 возвратится к последнему просмотренному узлу
 возвратится к узлу-каталогу
 Перейти к верхему узлу файла
 перейти к узлу, заданному именем
 Другие команды:
 запустить **учебник** по системе документации
 поиск темы в индексе
 поиск в узлах с помощью регулярных выражений
прекратить работу с документацией

C-h i
 C-h S

SPC
 DEL
 . (dot)

n
 p
 u
 m
 n
 f
 l
 d
 t
 g

h
 i
 s
 q

25 Регистры

сохранить область в регистре	C-x r s
вставить содержимое регистра в буфер	C-x r i
сохранить положение курсора в регистре	C-x r SPC
перейти к положению курсора, сохраненному в регистре	C-x r j

26 Клавиатурные макросы

начать определение клавиатурного макроса	C-x (
закончить определение клавиатурного макроса	C-x)
выполнить последний определенный макрос	C-x e
дополнить последний определенный макрос	C-u C-x (
задать имя для последнего определенного макроса	M-x name-last-kbd-macro
вставить в буфер определение макроса на языке Lisp	M-x insert-kbd-macro

27 Команды, связанные с Emacs Lisp

вычислить выражение перед курсором	C-x C-e
вычислить текущую функцию defun	C-M-x
вычислить область как выражение	M-x eval-region
считать и вычислить выражение из минибuffers	M-:
загрузить из стандартного системного каталога	M-x load-library

28 Простая настройка

настроить переменные и оформление (faces)	M-x customize
---	---------------

Создание глобальных привязок клавиш (ключей) на Emacs Lisp (примеры):

```
(global-set-key "\C-cg" 'goto-line)
(global-set-key "\M-#" 'query-replace-regexp)
```

29 Написание команд

```
(defun <command-name> (<args>)
  "<documentation>" (interactive "<template>")
  <body>)
```

Пример:

```
(defun this-line-to-top-of-window (line)
  "Reposition line point is on to top of window.
With ARG, put point on line ARG."
  (interactive "P")
  (recenter (if (null line)
                0
                (prefix-numeric-value line))))
```

Спецификация `interactive` сообщает о том, что аргумент надо считывать интерактивно. Для дополнительной информации наберите `C-h f interactive`.