

Emacs для начинающих HOWTO (Emacs Beginner's HOWTO)

Jeremy D. Zawodny, jzawodn@wcnet.org, перевод Alex Ott ott@phtd.tpu.edu.ru v1.7, 14 Октября 1999

Этот документ описывает редактор Emacs для пользователей Linux. Он предполагает некоторое знакомство с редактором vi или подобным. Последние версии этого документа обычно доступны с <http://www.wcnet.org/jzawodn/emacs/>

Содержание

1 Введение	1
2 Запуск Emacs	3
3 Режимы Emacs	7
4 Настройка Emacs	11
5 Популярные пакеты	14
6 Другие ресурсы	16
7 Благодарности	18

1 Введение

1.1 Авторские права

Авторские права принадлежат Jeremy D. Zawodny © 1998 - 1999. Этот документ может распространяться и изменяться в соответствии с требованиями GNU General Public License. Копия GPL доступна по адресу <http://www.gnu.org/copyleft/gpl.html>

1.2 Назначение документа и его аудитория

Этот документ предназначен для пользователей Linux, которые заинтересованы в том, чтобы немного узнать о Emacs и научиться работать с ним. Этот документ был начат как план короткого учебника, о котором я объявил на митинге Группы пользователей Linux в Toledo: <http://www.talug.org/>. С тех пор он немного вырос, из-за получения полезных советов, которые я получил от общественности. Подробности смотрите в разделе Благодарности.

Честно говоря, в этом документе нет ничего специфичного в отношении Linux. Он может быть применен почти к любому виду Unix и даже к Emacs, работающему на Microsoft Windows. Но поскольку этот документ является частью Linux Documentation Project, я указал, что он разработан для пользователей Linux — потому, что это так и есть.

И в конце концов, тем из вас, кто предпочитает название GNU/Linux простому названию "Linux" (прочитайте <http://www.gnu.org/gnu/linux-and-gnu.html> для того, чтобы понять почему) предлагается подставлять GNU/Linux там где в этом документе встречается просто Linux. Поскольку я не согласен с причиной и смыслом этой идеи, я не хотел заставлять себя писать GNU/Linux.

1.3 Что такое Emacs?

Emacs выглядит по-разному для разных людей. В зависимости от того, кого вы спросите, вы получите любой из следующих ответов:

- Текстовый редактор
- Почтовый клиент
- Программа чтения новостей
- Текстовый процессор
- Религия
- Интегрированная среда разработки (IDE)
- Все, во что вы хотите ее превратить!

Но для наших целей мы притворимся, что он является текстовым редактором — изумительно гибким текстовым редактором. Мы углубимся в этот вопрос позже. Emacs был написан Richard Stallman основателем Фонда Свободного программного обеспечения (Free Software Foundation): <http://www.fsf.org/> и проекта GNU <http://www.gnu.org/>, и он до сих пор сопровождает эту программу.

Emacs является одним из наиболее популярных и мощных текстовых редакторов, используемых в Linux (и Unix). Он находится на втором месте по популярности после **vi**. Он известен своим огромным количеством возможностей? возможностью легкой настройки и отсутствием ошибок. Огромное количество возможностей и возможность настройки является следствием того, как Emacs был спроектирован и реализован. Без углубления в детали я просто укажу, что Emacs не является "только редактором". Это редактор, в большинстве своем написанный на языке программирования **Lisp**. Основой Emacs является полнофункциональный интерпретатор языка Lisp, который написан на языке C. Только самые базовые и низкоуровневые части Emacs написаны на языке C. Большая часть редактора в действительности написана на языке Lisp. Так что, Emacs имеет встроенный язык программирования, который может использоваться для настройки, расширения и изменения поведения редактора.

Emacs также является одним из старейших редакторов. Он использовался тысячами программистов на протяжении последних 20 (?) лет, что означает, что существует много дополнительных пакетов к нему. Эти дополнения позволяют Emacs делать такие вещи, которые Stallman вероятно даже не считал возможными в то время, когда он начал работать над Emacs. Более подробно об этом в следующих разделах.

Существует много других серверов и документов, которые дадут вам более лучшие обзоры Emacs, его истории и сопутствующие материалы. Вместо того, чтобы делать попытки воспроизвести их здесь, я предполагаю, что вы посмотрите некоторые из мест, перечисленных в разделе **6** (Другие ресурсы) этого документа.

1.3.1 Версии и переносы на другую системы

Необходимо сказать, что в действительности существует два разных редактора Emacs: GNU Emacs и XEmacs. Оба произошли имеют общее наследие и делят большинство одинаковых свойств. Этот документ ориентирован на GNU Emacs (версии 20.3), но большинство того, что вы читаете в нем, также применимо к XEmacs и более ранним версиям GNU Emacs. В течении всего документа я буду ссылаться на "Emacs". Когда я буду делать это, то помните про это различие.

1.3.2 Получение Emacs

Получить Emacs очень легко. Если вы используете популярный дистрибутив Linux, такой как Debian, RedHat, Slackware, или какой-то другой, то Emacs вероятно является дополнительным пакетом, который вы можете установить с носителя на котором расположен дистрибутив. Если это не так, то вы можете взять исходные тексты Emacs и скомпилировать их сами. Посетите сервер проекта GNU для того, чтобы узнать точное расположение исходных текстов: <http://www.gnu.org/software/emacs/emacs.html>

2 Запуск Emacs

2.1 Запуск Emacs и выход из него

Как новый пользователь, вы вероятно захотите запустить Emacs просто для пробы. Войдя в Emacs и захотев выйти, вы однако можете не определить как это сделать. Так что, если вы никогда ранее не использовали Emacs, сделайте это сейчас. В командной строке наберите `emacs` и нажмите клавишу ввода (`enter`). Emacs должен запуститься. Если он не запускается, то либо он не установлен, либо не находится в каталогах, в которых происходит поиск программ.

После того как вы увидели Emacs, вам необходимо знать как выйти из него. Комбинацией клавиш для выхода из Emacs является `C-x C-c`. Обозначение `C-x` означает, что удерживается клавиша `Ctrl` и нажимается клавиша `x`. В этом случае вам необходимо продолжать удерживать `Ctrl` и нажать `c` для окончания сеанса работы.

Сочетания клавиш, используемые в Emacs покажутся вам странными, чужими и могут даже быть неудобными для вас первое время — особенно если вы пользователь `vi`. В отличии от `vi`, Emacs не имеет разных режимов для редактирования текста и команд.

Для повторения: `emacs` запустит Emacs. А `C-x C-c` произведет выход из Emacs.

2.1.1 Что вы увидите

После старта Emacs займет все окно X window (или экран, если вы запускаете его на консоли вместо X Window System). Вы увидите меню вдоль верха окна, некоторый текст расположенный в центральной части окна и несколько строк внизу окна.

Это будет выглядеть примерно как этот набросок в ASCII:

```
+-----+
|Buffers Files Tools Edit Search Mule Help      |
|                                                |
|Welcome to GNU Emacs, one component of a Linux-based GNU system. |
|                                                |
|                                                |
| ...                                           |
|---1:---F1  *scratch*          (Lisp Interaction)--L1--All-----|
|For information about the GNU Project and its goals, type C-h C-p. |
+-----+
```

Замечание: Emacs обычно заполняет весь экран/окно. Я сократил предыдущий пример для того, чтобы сохранить место. Вы также увидите приветственное сообщение при первом запуске Emacs. Я опустил его и вместо него вставил "...". Приветственное сообщение просто определяет точную версию используемого вами Emacs, и сообщает вам о том как использовать помощь и другие вещи.

Строка меню Самой верхней строкой пользовательского интерфейса Emacs является меню. Если вы работаете в X, то вы увидите его как традиционное выпадающее меню, которое вы можете использовать с помощью мыши. Иначе вам придется использовать Клавиатурные сокращения (не описаны в этом документе) для доступа к меню.

Строка статуса и мини-буфер Из последних двух строк интерфейса Emacs, самой верхней является строка статуса. Она содержит информацию о буфере в котором вы работаете, о режиме в котором находится Emacs, а также другую информацию. Сейчас просто представьте что там находится и все.

Самая нижняя строка называется **мини-буфер**. Он отделен от основного буфера строкой статуса, который мы только что обсуждали. Вы можете думать о нем как о "командной строке". Это место где вы наберете команды для Emacs и где появляются сообщения о том, что вы делаете.

Вы обнаружите, что то, что мы называем строкой статуса обычно в документации, относящейся к Emacs называется строкой режима. Это место где Emacs отображает информацию о текущем режиме (режимах), а также другие вещи, такие как дата и время, номер строки, размер файла и почти все что вы захотите там увидеть.

2.2 Немного терминологии

В этом разделе описывается общая терминология, используемая в Emacs, с которой вы столкнетесь при использовании и чтении о Emacs.

2.2.1 Буфера и файлы

В отличие от других редакторов при открытии файла Emacs не оставляет его открытым все время, пока вы работаете с ним. Вместо этого Emacs считывает его в **буфер**, который находится в оперативной памяти. В течении редактирования буфера и другой работы с ним, с данными на диске ничего не происходит. Только когда вы сохраняете данные, буфер обновляет данные на диске. В этом подходе есть свои достоинства и недостатки, но для вас важно только то, что вы понимаете как это работает.

Как следствие этого, вы увидите термин "буфер" используемым в документации по Emacs, режимах, пакетах и прочих вещах. Просто осознавайте, что буфер означает "копию файла, которая сейчас находится в памяти". Необходимо также отметить, что не всегда буфер ссылается на файл на диске. Очень часто Emacs создает буфера как результат выполнения ваших команд. Такие буфера могут содержать результат выполнения команды, список выбора, и тому подобные вещи.

2.2.2 Точка (Point) и Область (Region)

В сленге Emacs, вы часто услышите или увидите ссылки на термин **точка (point)**. Честно говоря точка это позиция курсора. Настоящее различие между курсором и точкой скорее всего не актуально при начале работы с Emacs. Но если вы любопытны, думайте о нем этим способом. Курсор – это наглядное представление точки. Курсор всегда находится в определенной позиции символа текущего буфера. С другой стороны, точка живет *между символами* в этом буфере. Так что вы можете сказать, что если курсор находится на букве 'h' в слове "the", то точка находится между 't' и 'h'.

Подобно многим современным редакторам, Emacs позволяет выполнять операции (отступа, проверки слов, переформатирования, вырезания, копирования, вставки,...) над куском текущего буфера. Вы можете подсветить (или "пометить") блок текста, используя клавиатуру или мышь и затем выполнить операцию только над выделенным блоком текста. В Emacs такой блок текста называется **область (region)**.

2.2.3 Окна

Ок, это немного смутит тех, кто уже использовал графический интерфейс. Просто помните, что Emacs разрабатывался задолго до появления графического интерфейса пользователя и менеджеры окон были популярны.

Окно в Emacs — это область экрана, в которой отображается буфер. Когда Emacs запускается, у вас на экране отображается одно окно. Некоторые функции Emacs (такие как оперативная справка и документация) часто [временно] открывают дополнительные окна на экране Emacs.

Окна Emacs не имеют ничего общего с окнами X в плане графического интерфейса. Вы можете открыть дополнительные окна X для отображения буферов Emacs, например для сравнения двух файлов. На эти новые окна X ссылаются как на **фрэймы (frames)** в жаргоне Emacs. Читайте дальше.

2.2.4 Фрэймы (Frames)

В Emacs **фрэйм** это отдельное окно X window в котором отображается буфер Emacs. Но оба окна являются частью одного сеанса Emacs. Поведение буфера в чем-то (но не во всем) похоже на то, что происходит когда вы нажимаете Alt+N в Netscape Navigator.

2.3 Основы работы с клавиатурой

Этот раздел описывает основы использования клавиатуры Emacs. Почти как и любой мощный редактор, все что вы можете сделать в Emacs используя лишь несколько клавиатурных сокращений. Если вы являетесь пользователем vi, то понятие использования клавиш k, j, l, h для перемещения вверх и вниз на строку и вперед/назад на символ вероятно займет некоторое время для освоения. В действительности освоение этого может занять несколько часов или даже недель для того, чтобы освоить навигацию по файлу, используя различные комбинации клавиш доступных в vi.

Emacs не отличается от него. Существуют разные сочетания клавиш и команды, которые надо изучить. Подобно vi, вам надо знать хотя бы основы для того чтобы сделать разную работу. Затем постепенно вы вы расширите ваше знание и найдете более быстрые способы выполнения разных вещей.

2.3.1 Командные клавиши (Meta, Esc, Control, и Alt)

Как вы вскоре узнаете Emacs широко использует много-клавишные комбинации. Потому-что это не редактор, ориентированный на режимы, подобно vi, вы не должны думать о том, находитесь ли в "командном режиме" или в "режиме правки" до того как вы попытаете переместить курсор или выполнить команды. Вместо этого наберите правильную комбинацию ключей (клавиш) и Emacs сделает (обычно) то, что вы требуете.

Клавиши, которые использует Emacs в документации часто сокращаются как C (для клавиши Control или Ctrl) и M для (Meta). В то время как клавиатуры для современных персональных компьютеров имеют одну или несколько клавиш Ctrl, лишь некоторые из них имеют клавиши Meta. Вам необходимо мысленно подставлять либо Esc либо Alt для клавиши Meta. В большинстве стандартных конфигураций и Esc и Alt выполняют одну и ту же работу.

Так что когда вы увидите ссылку в любой относящейся к Emacs на сочетание C-x f, то оно означает "нажмите control-x а затем f." И если вы увидите ссылку на что-то подобное M-x shell, то это означает "нажмите alt-x и наберите слово shell".

Очень полезной командой для начинающих пользователей является команда M-x apropos или C-h a. apropos произведет поиск в справочной документации по всем функциям Emacs и постарается найти разделы соответствующие введенному регулярному выражению. Это хороший способ для того, чтобы найти все команды относящиеся например к фрэймам. Просто наберите C-h a и затем слово frame.

2.3.2 Перемещение по буферу

Теперь, когда вы знаете какие сокращения что обозначают, вот вам список наиболее общих клавиатурных сокращений для перемещения внутри буфера:

Сочетание клавиш	Действие
C-p	Вверх на одну строку
C-n	Вниз на строку
C-f	Вперед на один символ
C-b	Назад на один символ
C-a	Начало строки
C-e	Конец строки
C-v	Вниз на одну страницу
M-v	Вверх на одну страницу
M-f	Вперед на одно слово
M-b	Назад на одно слово
M-<	В начало буфера
M->	В конец буфера
C-g	Окончить текущую операцию

И как вы могли и ожидать, клавиши перемещения курсора работают как надо. Клавиша Backspace может не работать, но это уже отдельная история.

2.3.3 Важные команды

Ок, теперь вы узнали как перемещаться по буферу, а что насчет команд открытия и сохранения файлов? Вот несколько основных команд.

До того как сразу приступить к этим командам я должен вкратце рассказать как они работают.

Все "командные сокращения (keystrokes)" в Emacs (те, которые M-x что-нибудь или C-что-нибудь) в действительности являются указателями на функции, которые являются частью Emacs. Вы можете вызвать любую из этих функций набрав M-x имя-функции и нажав Enter. Вы также можете использовать клавиатурное сокращение для этой функции (если оно имеется).

Например функция Emacs, которая сохраняет буфер на диск называется `save-buffer`. По умолчанию она также присвоена комбинации клавиш C-x C-s. Так что вы можете использовать либо эту комбинацию клавиш, либо вы можете набрать M-x `save-buffer` и достигнете тех же самых результатов.

Все наиболее часто используемые функции имеют клавиатурные сокращения. Некоторые из них перечислены ниже.

Сочетание клавиш	Функция	Описание
C-x C-s	<code>save-buffer</code>	Сохранить буфер на диске
C-x u	<code>undo</code>	Отменить последнюю операцию
C-c C-f	<code>find-file</code>	открыть файл на диске
C-s	<code>isearch-forward</code>	Поиск строки вперед
C-r	<code>isearch-backward</code>	Поиск строки назад
	<code>replace-string</code>	Поиск и замена строки
	<code>replace-regexp</code>	Поиск и замена с использованием регулярного выражения
C-h t	<code>help-with-tutorial</code>	Использовать интерактивный учебник

C-h f	describe-function	Показать справку по функции
C-h v	describe-variable	Показать справку по переменной
C-h x	describe-key	Показать данные, о том, что делает последовательность клавиш
C-h a	apropos	Поиск в справке по строке/регулярному выражению
C-h F	view-emacs-FAQ	Показать Emacs FAQ
C-h i	info	Документация по Emacs (Info)
C-x r m	bookmark-set	Установить закладку. Полезно при поисках
C-x r b	bookmark-jump	Перейти к закладке

После того, как вы попробуете многие из этих функций, вы узнаете, что они требуют ввода ваших данных. Ввод всегда происходит через мини-буфер. Это похоже на использование команд : в vi или большинство команд, которые вы используете в вашей любимой оболочке Unix. В Emacs доступны сотни встроенных функций. Вышеприведенный список это только маленький пример, который показывает те функции, которые я часто использую. Смотрите оперативную помощь для более полного списка всех доступных функций и более полной документации на то, что я описал выше.

2.3.4 Дополнение по Tab

Также как и многие популярные оболочки Unix (bash, csh, tcsh, ...) Emacs предлагает дополнение команд по клавише Tab. В действительности дополнение команд в bash было сделано уже после Emacs.

В качестве примера, попробуйте набрать M-x search и затем нажмите Tab. Emacs добавит знак минус в качестве указания того, что существует несколько возможных дополнений, но все они имеют знак минус как следующий знак. Нажмите Tab еще раз и Emacs покажет вам список возможных дополнений, из которых вы можете выбрать. Заметьте, что он делает это в *новом окне*. Он временно разбивает экран на два окна: одно из них содержит ваш буфер, а в другом находится список возможных дополнений для "search-". Вы можете нажать комбинацию C-g для прекращения процесса выбора и закрытия нового окна.

2.4 Учебник, Помощь и Info

Emacs поставляется с интерактивным учебником, который проведет вас через объяснение основных свойств редактирования и функций, которые должны знать все. Он также объясняет как использовать другие справочные свойства в Emacs.

Я сильно рекомендую вам провести некоторое время за учебником, если вы планируете достичь серьезных результатов в обучении Emacs. Как показано в таблице выше, вы можете запустить учебник через C-h t. Учебник само-документирован и предназначен для людей, которые только что начали работу с Emacs.

Если вы запускаете Emacs в X, то вы увидите что, самый правый пункт в строке меню озаглавлен Help (Помощь). При исследовании меню Help заметьте, что некоторые пункты имеют клавиатурные сокращения, и они перечислены справа в меню.

В заключение, для просмотра всего объема документации, доступной в Emacs, вы должны использовать команды M-x info или C-h i, которые запустят Info, просмотрщик документации Emacs.

3 Режимы Emacs

Режимы Emacs — это различное поведение и свойства, которые вы можете включить или выключить (или настроить, конечно). Режимы делают редактор (Emacs) одинаково полезным для написания документации, программирования на многих языках (C, C++, Perl, Python, Java и другие), создания

домашней страницы, работы с E-Mail, чтения новостей Usenet, отслеживания ваших должностей и даже игры в игры.

Режимы Emacs—это просто библиотеки кода на Lisp, который расширяет, изменяет и повышает возможности Emacs.

3.1 Главные режимы против второстепенных

Существует два типа режимов: Основные и Второстепенные. Различие между ними не так просто охватывается до тех пор пока вы не поработаете с несколькими из них включая и выключая их, но даст вам видение.

Одновременно может быть активен только один основной режим. Однако одновременно может использоваться несколько второстепенных режимов. Основные режимы ориентированы на определенный язык или задачу, в то время как второстепенные режимы являются маленькими и менее специфическими утилитами, которые пересекаются с многими задачами.

Это звучит немного абстрактно, так что давайте займемся примером. Существует режим, который я использую часто при написании простых текстовых файлов. Он называется `text-mode`. Этот режим был спроектирован для написания файлов в свободной форме, таких как файлы README. Он понимает как идентифицировать слова и параграфы и в общем выполняет все что требуется при использовании команд навигации.

Когда я пишу файлы для людей, я обычно хочу, чтобы они выглядели хорошо. Они должны иметь правильный перенос слов на определенной границе, и т.п. Для разрешения переноса слов я просто включаю второстепенный режим `auto-fill`. Этот режим пытается сделать все правильно, когда я набираю текст и достигаю конца строки. В действительности второстепенный режим означает, что его можно использовать с разными основными режимами. Режим выполняет то, что нужно в зависимости от того, в каком основном режиме он работает, поведение в `text-mode` отличается от поведения в `java-mode`. Я не хочу, чтобы мой код на Java переносился точно, также как и английский текст. Но я хочу, чтобы текст переносился в блоках комментариев Java! Режим `auto-fill` достаточно умен, чтобы делать различия между режимами.

Авторы различных режимов Emacs проделали огромную работу по проверке, того что те вещи, которые должны работать как второстепенные режимы работали как второстепенные режимы.

Если вы глянете на снимок экрана Emacs, то вы увидите, что строка режима показывает в каком режиме находится Emacs. В нашем случае он находится в режиме, который называется "Lisp Interaction", который является режимом по умолчанию. В действительности он полезен только в том, случае если вы пишете код на Lisp. (Но поскольку большая часть Emacs написана на Lisp, то почему-бы и нет?)

3.2 Режимы для программирования

Первое и самое главное—Emacs был создан программистом для программистов. Для всех языков программирования (даже не очень популярных) существуют высококачественные режимы. Я только вкратце опишу несколько из них.

Большинство режимов программирования разделяют общие характеристики. Обычно они делают некоторые или все из перечисленных свойств:

- Обеспечивает подсветку синтаксиса для языка.
- Обеспечивает автоматический отступ и форматирование кода языка.
- Обеспечивает контекстную помощь (по языку).
- Предоставляет интерфейс к вашему отладчику.
- Добавляет специфические для данного языка пункты в меню.

В дополнение к этому существуют некоторые режимы, которые не являются специфическими для конкретных языков, они помогают выполнять задачи общие для программирования в разных языках. Это вещи, такие как взаимодействие с программным обеспечением контроля версий, автоматического добавления комментариев в ваш исходный код, создание Makefile, обновление Change Logs и т.п.

Когда вы соедините все эти режимы вместе и рассмотрите стабильность и зрелость кода Emacs, то он сравним с коммерческими Интегрированными средами разработки (IDE) для таких языков как C++ и Java. И конечно, он свободен.

3.2.1 C/C++/Java

Поскольку синтаксис C, C++ и Java достаточно похож, то существует один режим Emacs, который обрабатывает все три языка (также как и Objective-C и IDL). Это очень зрелый и полный пакет, включенный в дистрибутив Emacs. Этот режим называется либо `cc-mode`, либо `CC Mode`.

Для более полной информации или для получения более новой версии посетите <http://www.python.org/emacs/>.

3.2.2 Perl

Существует два режима для редактирования кода Perl в Emacs. Первый называется `perl-mode` (как вы и ожидали), а второй называется `sperl-mode`. Я не делал обзора их историй и почему существует два режима (документация этого не объясняет), но кажется, что `perl-mode` был оригинальным режимом для редактирования кода на Perl в Emacs. Кажется он имеет меньше возможностей, чем `sperl-mode` и не имеет возможности распознать некоторые конструкции любителей Perl.

От себя я рекомендую `sperl-mode`, который кажется более активно сопровождается и имеет все возможности в которых я нуждаюсь. Вы можете найти самую последнюю версию здесь: <ftp://ftp.math.ohio-state.edu/pub/users/ilya/emacs>.

Но не принимайте мои слова близко к сердцу. Попробуйте поработать с обоими и выберите лучший для ваших нужд.

3.2.3 Python

Python (другой очень популярный скриптовый язык) также имеет режим для Emacs. Как я уже мог упомянуть он не распространяется с GNU Emacs, но распространяется с XEmacs. Хотя он работает достаточно хорошо в обеих версиях.

Вы можете получить `python-mode` с официального сервера Python <http://www.python.org/emacs/python-mode/>.

3.2.4 Другие

Также для помощи программистам существует много других режимов. Такие режимы могут помочь с такими вещами как:

- Скрипты Shell (Bash, sh, ksh, csh, ...)
- Awk, Sed, Tcl, ...
- Makefiles
- Change Logs
- Документация
- Отладка

И многое другое. Смотрите последний раздел этого документа, о том как найти режимы и дополнения.

3.3 Создание текстов

Прекрасные режимы Emacs *не* ограничены только написанием кода. Люди пишущие документацию (любого типа) также могут извлечь пользу из широкого выбора режимов Emacs.

3.3.1 Проверка грамматики (режим `ispell`)

Авторы многих типов документов нуждаются в проверке грамматики. Если у вас установлен GNU `ispell`, то вы можете просто набрать `M-x ispell` и выполнить проверку текущего буфера. Если `ispell` находит слова, которых он не знает, то он выдаст список возможных замен и позволит вам выбрать одну (или ни одной) из них. Это эквивалентно по возможностям проверке грамматики во многих коммерческих пакетах.

3.3.2 HTML (режим `html-helper`)

Если вы хотите создать файлы HTML, то вы должны попробовать `html-helper-mode`. Он доступен с <http://www.santafe.edu/nelson/tools/> вместе с документацией и сопутствующими утилитами.

Как видно из его имени, `html-helper-mode` обеспечивает много вещей для того, чтобы помочь людям, которые до сих пор пишут HTML вручную — старомодным способом.

Примечание переводчика: Я рекомендую использовать пакет `psgml`, кроме `html` он поддерживает разные типы `sgml` и различает разные стандарты `html`. Пакет доступен с . Идет в поставке XEmacs.

3.3.3 TeX (`tex-mode`)

Если вы пишете документы в TeX, то очень полезно заставить Emacs добавить цветное выделение обратных слэшей, скобок и других символов. `tex-mode` заботится об этом.

Хотя я сам не много работал с TeX, когда я работал с этим режимом, то он был достаточно удобен, делая исходные тексты TeX более читаемыми.

Примечание переводчика: Также существует пакет AUC TeX, доступный по адресу . Имеет больше возможностей, чем вышеупомянутый `tex-mode`. Идет в поставке XEmacs.

3.3.4 SGML (`sgml-mode`)

The document you're now reading was written in SGML (and probably converted to the format you're reading it in). `sgml-mode` provides all the basics for SGML documents: validation, highlighting, forward-tag, backward-tag, and much more. It is a standard part of Emacs.

Примечание переводчика: Я рекомендую использовать пакет `psgml`, он поддерживает разные типы `sgml`. Пакет доступен с .

3.4 Другие режимы

Конечно существует множество других удобных режимов, которые делают жизнь более легкой. Вот просто образец популярных режимов:

3.4.1 Контроль версий (режим `vc`)

`vc` предоставляет интерфейс к большинству популярных систем контроля версий (RCS, SCCS, CVS), чтобы сделать очень легкой работу с файлами, работу с выпусками и т.п. Режим является стандартной частью Emacs и описан в документации Emacs.

3.4.2 Режим Shell

Зачем переключаться в другое окно X или виртуальный терминал для того, чтобы просто выполнить несколько команд shell? Сделайте это из Emacs и сохраните себя от беспокойства. :-)

`M-x shell` запустит shell внутри буфера Emacs. Вы сможете выполнить большинство вещей, которые вы можете делать в обычной строке shell (за исключением запуска полно-экранных программ, таких как `vi` или `pine`), потому-что Emacs общается с обычным shell в фоновом режиме.

Режим является стандартной частью Emacs, так что вы найдете документации по нему в стандартной документации Emacs.

3.4.3 Telnet и FTP

Зачем переключаться в другое окно X или виртуальный терминал для того, чтобы просто запустить telnet или FTP? Выполните это из Emacs (работайте по образцу?)

Точно также, как и запуск shell внутри Emacs, вы можете запустить telnet и ftp. Попробуйте выполнить `M-x telnet` или `M-x ftp` для приобретения опыта. Смотрите документацию для описания деталей работы.

3.4.4 Man

Зачем переключаться в другое окно X или виртуальный терминал для того, чтобы просто прочитать справочную страницу? Сделайте это из Emacs. (Я обещаю, что остановлюсь).

Точно также, как и запуск shell внутри Emacs, вы можете читать справочные страницы. Попробуйте выполнить команду `M-x man`.

3.4.5 Ange-FTP

Цитата из документации по `ange-ftp`:

Этот пакет пытается сделать доступ из GNU Emacs к файлам и каталогам на FTP простым и прозрачным насколько это возможно. Подмножество общих функций обработки файлов расширено для взаимодействия с FTP.

Это означает, что вы можете работать с файлами на удаленных машинах также как, как и с локальными. Так что если вам нужно изменить файл на удаленном компьютере, то просто скажите Emacs открыть его (используя немного отличный синтаксис путевого имени) и он возьмет на себя все детали о подключении и получении файла. Затем при сохранении файла через `C-x C-s`, `ange-ftp` перехватит сохранение и запишет измененный файл обратно на удаленную машину.

Немного измененный синтаксис путевого имени выглядит так... Файл именуемый "myfile", в каталоге пользователя "user", на машине с именем "my.host.org" может быть открыт (`C-x f`):

```
/user@my.host.org:user/myfile
```

Он также является стандартной частью дистрибутива Emacs, так что вы можете найти документацию среди документации по Emacs.

Спасибо Etienne Grossmann (etienne@anonimo.isr.ist.utl.pt) за вышеприведенный пример.

4 Настройка Emacs

Вся настройка Emacs выполняется через код на Lisp. Вы можете изменять переменные, которые влияют на то, каким образом работает Emacs или вы можете добавить новую функцию к Emacs (или переопределить существующие функции — заменяя их на свои собственные).

4.1 Временная настройка

Во время экспериментов с настройкой Emacs вы вероятно захотите сделать ее на некоторое время. Если что-то пойдет неправильно, то вы сможете просто нажать на C-x C-c для выхода из emacs и запустить его и запустить его снова. После того как вы определите, какие изменения вы хотите сделать постоянными, вы можете добавить их в ваш файл `.emacs`, так что они будут действовать каждый раз при старте Emacs. Это обсуждается в следующем разделе.

4.1.1 Присвоение переменных

Самым легким способом настройки является изменение значений переменных в Emacs. Для того, чтобы сделать это используется примерно такой код:

```
(setq variable-name new-value)
```

Где `variable-name` это имя переменной, а `new-value` это значение, которое вы хотите задать для этой переменной. (В Lisp, вы связываете переменную и значение). Функция `setq` в Lisp аналогична оператору присваивания (обычно `=`) в других языках программирования.

Замечание: Я рассказываю без многих деталей ради упрощения. Вы также можете указать мне или другим на использования функций Lisp `set` и даже `setq-default`. Если вы действительно любознательны, то вы можете посмотреть их описание в руководстве по Emacs Lisp.

Давайте глянем на строку из моего файла `.emacs`

```
(setq-default transient-mark-mode t)
```

Переменная `transient-mark-mode` контролирует должна ли подсвечиваться область при ее выделении. Во многих приложениях с графическим интерфейсом пользователя, если вы щелкнете кнопкой мыши и сделаете протягивание для выделения текста, то он будет выделен цветом или инверсией. Emacs будет работать так же, если переменная `transient-mark-mode` будет установлена (в не-`nil` значение).

Что такое значение?

Ок. Немного отвлечемся. Большинство языков программирования имеют некоторое понятие о значениях `true/false` (истина/ложь). В C/C++ значение считается истинным, если оно имеет ненулевое значение. В Perl, не-`null` или ненулевое значение считается истинным. В Lisp, применяются те же идеи, но имена и символы отличаются.

Истина обычно записывается как `t`, а ложь (или `null`) записывается как `nil`. Подобно другим языкам, не-`nil` значения считаются истиной.

Для получения полного описания того, что делает `transient-mark-mode`, вы можете использовать оперативную справку. Наберите C-h v или M-x `describe-variable`, затем `transient-mark-mode`. Если вы ленивы, как и я, то вы можете воспользоваться дополнением имени переменной с помощью кнопки `Tab`. Просто наберите часть имени переменной и нажмите клавишу `Tab`. Если вы набрали недостаточно для уникальной идентификации имени, то вы можете увидеть все возможные варианты дополнений.

Другой часто используемой переменной является `fill-column`. Она сообщает Emacs насколько широким должен быть экран для переноса слов (и `auto-fill-mode` использует это значение). Для установки этой переменной в какое-нибудь значение вы можете набрать:

```
(setq fill-column 20)
```

Но оно в действительности ничего не делает. Вам необходимо сказать Emacs чтобы он **вычислил (evaluate)** выражение, которое вы набрали. Чтобы сделать это поместите точку (курсор) в конец выражения и наберите сочетание C-x C-e, что вызовет функцию `eval-last-sexp`. Когда вы сделаете это заметьте, что число 20 (или что вы набрали) отобразится в мини-буфере внизу экрана. Это просто возвращенное при вычислении значение.

Просто для проверки его работы наберите пару предложений. Если у вас включен `auto-fill-mode` (он может быть и не включен), то вы заметите, что слова переносятся на отметке 20 колонки. Иначе, после того как вы набрали некоторый текст, нажмите `M-q`, что вызовет функцию `fill-paragraph`. Она выполнит перенос слов.

4.1.2 Связывание с файлами

Вы можете настроить Emacs на автоматическое выполнение чего-нибудь при открытии файлов с определенным типом (подобно как многие графические среды автоматически запускают приложения если вы щелкаете на значке определенного файла). Например, я хочу чтобы Emacs автоматически переключался в `text-mode` при открытии файлов с расширением `.txt`. Хорошо, это уже происходит. :-) Теперь я хочу заставить Emacs всегда переходить в `text-mode` при открытии файлов с именем "README".

```
(setq auto-mode-alist (cons '("README" . text-mode) auto-mode-alist))
```

Без погружения в дебри программирования на Lisp, которое знать не обязательно (но будет не вредно его выучить), просто указываем, что переменная `auto-mode-alist` содержит список пар. Каждая пара содержит регулярное выражение и название режима Emacs. Если открываемый файл соответствует указанному выражению (в нашем случае это строка README), то Emacs переходит в указанный режим.

Странный синтаксис в приведенном примере используется потому, что мы хотим добавить пару в список режимов. Вы не захотите просто присвоить данные переменной `auto-mode-alist` с потерей уже содержавшихся в ней данных.

И если я хочу, чтобы Emacs автоматически переключался в `html-helper-mode` каждый раз, когда я открываю файл, оканчивающийся на `.html` или `.htm`, то я должен добавить следующие строки в мой файл `.emacs`:

```
(setq auto-mode-alist (cons '("\\.html$" . html-helper-mode) auto-mode-alist))
(setq auto-mode-alist (cons '("\\.htm$" . html-helper-mode) auto-mode-alist))
```

Существует бесконечное количество возможных значений.

4.2 Использование файла `.emacs`

После того, как вы провели некоторое время в работе с Emacs и имеете идеи о том, что надо настроить, вы вероятно захотите настроить некоторые вещи навсегда (или по крайней мере до тех пор, пока вы не измените свою точку зрения). Если вы будете использовать Emacs в каждодневной работе, то вы также заметите, что ваш файл `.emacs` увеличивается со временем. Это *Хорошо*, потому-что вы определили как заставить Emacs работать так, как **вы** хотите. Для многих программ является позором, что они позволяют сделать это.

В случае, если вы еще не догадались, Каждый раз при запуске Emacs, он ищет файл с именем `.emacs` в вашем домашнем каталоге. Ваш файл `.emacs`—это место, куда вы должны помещать код на Lisp, если хотите, чтобы он автоматически выполнялся при запуске, а также файл включает в себя те настройки, которые вы хотите использовать.

Вот другой пример из моего файла `.emacs`:

```
(setq inhibit-startup-message t)
```

Переменная `inhibit-startup-message` задает будет ли показывать Emacs приветственное сообщение на старте, или нет. После того как я посмотрел на него (потому-что я уже знал как найти помощь) я нашел способ отключить его.

Как упражнение попробуйте создать собственный файл `.emacs` и добавить эту строку в него. Затем выйдите из Emacs и запустите его снова. Вы не должны увидеть стартового сообщения. Часто когда вы читаете о режиме Emacs (или о пакете), то документация будет вам предлагать добавить некоторый код в ваш файл `.emacs` для того, чтобы режим или пакет работали как надо. GNU Emacs FAQ (C-h F) содержит некоторые пункты, относящиеся к файлам `.emacs`, которые вы можете найти полезными.

4.3 Пакет Customize

После того как Emacs начал приобретать популярность и продолжал развиваться, некоторые люди сказали "должен быть лучший способ, позволяющий начинающим пользователям настроить Emacs". И родился пакет `customize`.

`Customize` обеспечивает более интуитивный метод настройки частей Emacs. Чтобы попробовать работать с ним, либо выберите подменю `Customize` в вашем меню `Help`, или наберите `M-x customize`.

`Customize` группирует настройки в логические группы "Editing", "Programming", "Files" и т.п. Некоторые группы содержат подгруппы.

Если вы сделаете изменения, используя интерфейс настройки, то Emacs сохранит изменения в ваш файл `.emacs`. Это намного легче, потому-что вы можете легко изучать изменения, которые вы он сделал для вас.

Я не использую пакет `Customize` так, что я не могу сказать о нем больше..

4.4 Дисплей X Windows

Подобно всем нормальным X-приложениям, Emacs работает с вашими X-ресурсами. Это означает, что вы можете контролировать начальные цвета, геометрию и другие специфические для X вещи, также как и в `xterm`, `nxterm` или чем-то другом.

Вот соответствующий кусок моего файла `/.Xdefaults`:

```
emacs*Background: DarkSlateGray
emacs*Foreground: Wheat
emacs*pointerColor: Orchid
emacs*cursorColor: Orchid
emacs*bitmapIcon: on
emacs*font: fixed
emacs.geometry: 80x25
```

Посмотрите справочную страницу X для более детального описания X-ресурсов. Chris Gray (cgray4@po-box.mcgill.ca) также заметил:

В Debian, файл `/.Xdefaults` кажется не используется. Однако работающие в Debian могут поместить то, что вы хотите в `/etc/X11/Xresources/emacs` и могут получить то, что они делали в RedHat.

5 Популярные пакеты

В добавление к многим различным режимам доступным для Emacs, также существует много дополнительных **пакетов**. Я называю их пакетами, потому-что они являются больше чем просто режимами. Они часто включают дополнительные утилиты, или слишком большие, чтобы называть их режимами. В других случаях они являются программным обеспечением, которое расширяет или интегрирует другие режимы и пакет Emacs. Различие не совсем ясное, но это нормально.

5.1 VM (Почта)

Цитата из VM FAQ:

VM (Просмотр почты (View Mail)) является подсистемой Emacs, которая позволяет читать и работать с почтой внутри Emacs. Предоставляются команды для выполнения всех обычных действий, которые ожидаются от почтового агента для пользователя, такие как создание ответов, сохранение сообщений в папках, удаление сообщений и т.п. Существуют расширенные команды для выполнения таких команд как разделение и создание дайджестов, пересылка сообщений и организация представления сообщений соответственно различным критериям.

Когда я только начал использовать Emacs, я попробовал работать с VM. Я нашел, что он является великолепной заменой для Pine, Elm или большинства других программ для почты. Но я не хотел использовать разные программы для чтения почты и новостей. VM активно разрабатывается и хорошо поддерживается в настоящее время.

Он доступен по следующему адресу: <http://www.wonderworks.com/vm/>.

5.2 Gnus (Почта и новости)

Цитата из руководства GNUS:

Gnus является лабораторией по чтению сообщений. Он позволит вам посмотреть почти все как будто это группа новостей. Вы можете читать новости с его помощью, вы можете перемещаться по каталогам, вы можете работать с ftp с его помощью — вы даже можете читать новости с его помощью!

Gnus пытается дать возможности людям, кто читает новости теми же способами, которыми Emacs дает возможности людям, редактирующим текст. Gnus не устанавливает пределов того, что может сделать пользователь. Пользователи поощряются на расширение Gnus, для того, чтобы он вел себя как вы этого хотите. Программа не должна контролировать человека; люди должны иметь возможности делать все что они хотят используя (или неправильно используя) программу.

Сейчас я использую GNUS для чтения новостей и почты (как было указано выше). GNUS также активно развивается и поддерживается.

Он доступен по адресу: <http://www.gnus.org/>.

5.3 BBDB (органайзер)

BBDB это коварная база данных Big Brother, подобная органайзеру программа для Emacs, которая работает с большинством популярных пакетов для почты в Emacs (включая VM и GNUS).

Пакет доступен с: <http://pweb.netcom.com/simmonmt/bbdb/index.html>.

5.4 AucTeX (другой режим TeX)

AucTeX является другим режимом для редактирования файлов TeX.

Цитата с сервера AucTeX:

AUC TeX это расширяемый пакет, который поддерживает написание и форматирование файлов TeX для большинства вариантов GNU Emacs. Поддерживается много различных макропакетов, включая AMS TeX, LaTeX и TeXinfo.

Пакет доступен с: <http://sunsite.auc.dk/auctex/>.

6 Другие ресурсы

Этот раздел описывает книги, сервера, группы новостей, списки рассылки и другие места в которых вы можете найти больше информации о Emacs.

6.1 Книги

Существует несколько действительно хороших книг для изучения Emacs. В добавление к ним, вы также обнаружите, что много книг о Linux и Unix также содержат главы о Emacs (и vi).

6.1.1 Learning GNU Emacs

Авторы: Debra Cameron, Bill Rosenblatt, Eric S. Raymond

Издатель: O'Reilly & Associates - <http://www.ora.com/>

Вы можете купить ее на Amazon.com: <http://www.amazon.com/exec/obidos/ASIN/1565921526/>

Комментарий: Вероятно, что это лучшая книга с которой можно начать работу. После того, как вы прочитали HOWTO и посмотрели в FAQ эта книга станет полным и доступным учебником.

6.1.2 Writing GNU Emacs Extensions

Автор: Bob Glickstein

Издатель: O'Reilly & Associates - <http://www.ora.com/>

Вы можете купить ее на Amazon.com: <http://www.amazon.com/exec/obidos/ASIN/1565922611/>

Комментарий: Если после того, как вы поработали с Emacs некоторое время и решили, что вы хотите написать ваш собственный режим или хотите сделать вашу собственную расширенную настройку, то это книга для вас. Поскольку она не делает попытки научить вас Lisp, она содержит только краткое введение в язык.

6.1.3 Programming in Emacs Lisp: An Introduction

Автор: Robert J. Chassell

Из файла README:

Это элементарное в программирование на Emacs Lisp для людей, которые не являются программистами, и кто не заинтересован в программировании, но кто хочет настроить или расширить свою среду.

Вы можете получить это руководство через анонимный FTP с сервера GNU: <ftp://prep.ai.mit.edu/gnu/emacs/>.

Вы можете купить хорошо отпечатанную версию на Amazon.com: <http://www.amazon.com/exec/obidos/ASIN/1882114418/jeremydzawodny/>.

Комментарий: Это хорошее введение в Emacs Lisp — даже если вы хороший программист.

6.1.4 The GNU Emacs Lisp Reference Manual

Автор: Richard Stallman

Издатель: The Free Software Foundation - <http://www.fsf.org/>

Вы можете получить это руководство через анонимный FTP с сервера GNU: <ftp://prep.ai.mit.edu/gnu/emacs/>.

Комментарий: Это полное руководство по языку программирования Emacs Lisp.

6.2 Сервера

6.2.1 EMACsulation

EMACsulation это колонка сопровождаемая Eric Marsden, которая появляется в интерактивном журнале Linux Gazette, расположенном по адресу <http://www.linuxgazette.com/>. Наиболее свежая колонка во время написания находилась по адресу <http://www.linuxgazette.com/issue39/marsden.html>. Посмотрите в конце статьи ссылки на предыдущие выпуски.

6.3 Группы новостей

Ищите в списке вашего сервера группы новостей, которые содержат слово "emacs" и вы найдете достаточно много. Вот какие группы есть на моем сервере:

- comp.emacs
- comp.emacs.sources
- gnu.emacs
- gnu.emacs.bug
- gnu.emacs.help
- gnu.emacs.sources

6.4 Списки рассылки

Единственным списком рассылки, посвященным Emacs, который я знаю, является список NT-Emacs. Это список для людей, которые используют версию Emacs для Micro\$oft Windows. Смотрите NT-Emacs FAQ <http://www.cs.washington.edu/homes/voelker/ntemacs.html> для подробной информации.

6.5 Архив Emacs Lisp

Из README архива Emacs Lisp:

Архивы Emacs Lisp на [ftp.cis.ohio-state.edu](ftp://ftp.cis.ohio-state.edu) содержат разные части и пакеты кода Emacs Lisp. Emacs Lisp является языком, используемым для расширения редактора GNU Emacs, который создан Free Software Foundation. Хотя много кода на Emacs Lisp включено в дистрибутив GNU Emacs, много людей написало пакеты для взаимодействия с другими системами, лучшей поддержки редактирования программ на используемых языках программирования, добавления новых возможностей или для изменения поведения Emacs. Большинство содержимого архива было написано отдельными людьми и распространяется свободно через Internet по спискам рассылки info-emacs или info-gnu-emacs, или по группам новостей comp.emacs, gnu.emacs или gnu.emacs.sources.

Архив доступен через анонимный FTP по адресу <ftp://ftp.cis.ohio-state.edu/pub/emacs-lisp/>.

Замечание: Я могу сказать, что архив Emacs Lisp потихоньку устаревает, я вижу очень немного новых (или обновленных) пакетов, появляющихся там, хотя я знаю, что они существуют. Они посылаются в группу новостей comp.emacs.sources. (Вы можете исправить меня, если я не прав).

7 Благодарности

Следующие люди определили успех этого документа.

- Robert Vollmert *rvollmer@gmx.net*
- Larry Brasfield *larrybr@seanet.com*
- Etienne Grossmann *etienne@anonimo.isr.ist.utl.pt*
- Thomas Weinell *kf6mli@amsat.org*
- Adam C. Finnefrock *adam@bigbro.biophys.cornell.edu*
- Chris Gray *cgray4@po-box.mcgill.ca*
- Robert J. Chassell *bob@rattlesnake.com*
- Isaac To *kkto@csis.hku.hk*
- Matteo Valsasna *valsasna@elet.polimi.it*
- Tijs van Bakel *smoke@casema.net*