# Joe - Текстовый редактор. Знакомство и возможности.

Опубликовано muff в Сб, 2009-08-15 22:24

Поскольку сам пользуюсь редактором joe, то после конечно первым делом займемся его установкой.

```
# cd /usr/ports/editors/joe
 # make install clean
```

В процессе установки могут подтянуться зависимости, если они еще не установлены:

- aspell;
- gettext;
- libiconv;
- perl.

По данному редактору инфы более чем предостаточно, соответственно не буду изобретать колесо, а наведу статью Алексея Федорчука от 2004.08.21

Эта статья была написана достаточно давно, по данным для joe версий 2.8.X-2.9.X. До недавнего времени ее нельзя было считать устаревшей, так как проект joe практически не развивался. Однако последнее время в нем наметилась определенная активность, ознаменовавшаяся выходом версий серии 3.X. Которые привнесли, кроме нескольких мелких усовершенствований, два кардинальных новшества - подсветку синтаксиса и возможность работы с кодировкой UTF-8. В результате чего исходно хороший редактор стал еще лучше...

- Обзор возможностей
- Система помощи
- Характерные особенности
- Макрокоманды
- Настройка joe
- Дополнительные настройки
- Заключительные соображения
- Приложение. Штатные команды joe

## Обзор возможностей

Текстовый редактор joe - типичный представитель консольных редакторов командного стиля, то есть ориентированных не на действие через меню, а на управление с помощью прямых директив. Название его можно перевести примерно как "редактор дядюшки Джо". Он создан Джозефом Алленом (Joseph H. Allen) при участии Ларри Форда (Larry Foard) и Гари Грея (Gary Gray). Это - открытая и бесплатная программа, доступная в исходниках на http://joe.sourceforge.net [1]. Она реализована для всех, насколько я знаю, Unix- и Unix-подобных систем. А некоторые конкретные врсии были собраны для Windows всякого рода и даже для DOS.

Запускается joe одноименной командой, можно - с именем файла, предназначенного для редактирования. В случае, если этого имени в природе не имеется, создается новый файл. Кроме этого, при запуске joe можно использовать ряд опций командной строки. Представление о них дает чтение страниц экранной документации (man joe).

Сразу после запуска joe выглядит весьма непрезентабельно: черный экран со светло-серым текстом - и все. Что делать дальше - остается пока неясным. Единственно, строка заголовка в верхней части экрана гласит, что с помощью комбинации клавиш **Control**+**K** - **H** можно вывести на экран систему помощи. И далее пролистывать ее с помощью **Escape** - **.** (точка) - вперед или **Escape** - **,** (запятая) - назад.

Так вот, внимательное знакомство с системой помощи дает представление о возможностях редактора. Каковые неожиданно оказываются весьма богатыми.

Думаю, понятно, что текстовый редактор позволяет вводить текст (в том числе и кириллический) и обеспечивает навигацию по нему. Последняя осуществляется двояко: либо с помощью собственных клавишных комбинаций (как правило, это **Control**+литера), о чем подробнее - в следующем разделе, либо - стандартных клавиш управления курсором (стрелок, **PageUp**, **PageDown**, **End**, **Home**). При этом клавиши эти введут себя обычным (с точки зрения пользователя DOS/Windows) образом, что отнюдь не само собой разумеющееся для консольных Unix-редакторов). Исключение - клавиша **End**, которая на некоторых типах терминалов и в большинстве эмуляторов терминала графического режима работать отказывается.

Вообще, следует заметить, что основным средством навигации по тексту в joe являются именно собственные команды, а не клавиши управления курсором. Во-первых, при наличии некоторого навыка, они обеспечивают большую скорость работы за счет того, что не требуется перемещения пальцев за пределы основной клавиатуры (поверьте, это действительно быстрее!). Во-вторых, команды joe, вызываемые клавишными комбинациями, функционируют абсолютно одинаково в любых консолях (по крайней мере за Linux-консоль, syscons FreeBSD, pcvt OpenBSD я ручаюсь - проверено лично) и эмуляторах терминалов. Тогда как для клавиш управления курсором возможно, как уже отмечено, аномальное поведение.

Мышь в joe поддерживается стандартным для Unix-консоли образом. То есть она выступает не как указательное устройство, а как средство выделения и копирования текстовых блоков. Это относится как к текстовой консоли, так и к эмуляторам терминалов графического режима.

Разумеется, в соответствие со своим званием joe позволяет и редактирование текстов, то есть: выделение блоков и отдельных знаков, их копирование, перемещение и удаление, форматирование абзацев (центрирование, лево- и правостороннее выравнивание и т.д.), вставку существующих файлов в текущий документ и запись выделенных фрагментов в виде отдельных файлов.

joe имеет функцию многоуровневой отмены и возврата отмененных операций. Встроенной проверки орфографии нет, однако можно подключить внешнюю программу (такую, как ispell), в том числе и для русскоязычных текстов. Имеются достаточно развитые средства поиска и замены, в том числе с использованием шаблонов и регулярных выражений. Есть возможность создания закладок (Bookmarcs) и перехода к ним, что необходимо при редактировании длинных структурированных документов.

В joe имитируется многооконный режим: поле текущего документа может быть разбито пополам, и далее каждое из них также может делиться сколь угодно дробно (правда, только по горизонтали). Обеспечена также одновременная работа со многими документами, каждый из которых может быть выведен в оконном или полноэкранном виде.

Количество одновременно открытых файлов, насколько мне удалось понять, не ограничено ничем. Более того, по сведениям моих корреспондентов, joe позволяет работать с документами, объем которых превышает объем ВСЕЙ доступной (то есть физической + виртуальной) памяти! Правда, проверить это мне не удалось...

В joe поддерживается собственный макроязык с достаточно прозрачным синтаксисом. Кроме того, имеется режим протоколирования макросов, что позволяет быстро наращивать его возможности.

И вообще, joe - очень настраиваемый редактор. Во-первых, имеется система интерактивной настройки ряда параметров, таких как перенос слов, абзацный отступ и т.д. Правда, установки эти действуют только в текущем сеансе. Для перманентных изменений необходимо редактирование конфигурационного файла. Однако здесь, учитывая возможность встраивания макрокоманд, возможности настроек становятся поистине безграничными.

И наконец, что немаловажно в наших условиях, joe корректно работает с кириллицей. При правильно русифицированной консоли не возникает никаких проблем ни с выводом, ни с вводом символов кириллицы. Более того, все клавишные комбинации работают и при латинской, и при русской раскладке клавиатуры. Правда, в последнем случае часто требуется дополнительное нажатие управляющей клавиши.

Иными словами, joe - вполне развитый и полнофункциональный текстовый редактор общего назначения. Он в равной мере пригоден для эпизодической работы по написанию скриптов, правке конфигурационных файлов и т.д., и для систематического использования при создании длинных структурированных текстов нарративного характера. А наличие языка макрокоманд допускает эффективно применять его и в специальных целях - для разметки html-страниц, верстки документов в TeX, не говоря уже о собственно программировании. Единственный недостаток joe в этом плане - отсутствие подсветки синтаксиса как в консоли, так и в эмуляторе терминала.

## Система помощи

Ознакомиться практически со всеми возможностями редактора joe можно посредством его системы помощи. Как уже говорилось, она выводится на экран нажатием комбинации клавиш **Control**+**K**-**H** и насчитывает семь секций, каждая из которых занимает собственный экран, перемещение между которыми осуществляется комбинациями клавиш **Meta**+**.** (вперед) и **Meta**+**,** (назад).

Первая секция, Basic, описывает действия наиболее общего плана: перемещения курсора (субсекция CURSOR), переходы по тексту (субсекция GO TO), операции с текстовыми блоками (субсекция BLOCK), команды удаления символов и текстовых фрагментов (субсекция DELETE), команды поиска, проверки орфографии, форматирования (субсекции SEARCH, SPELL, MISC), операции с файлами (субсекция FILE), а также выход из редактора.

Вторая секция посвящена описанию манипуляций с окнами - расщеплению (split) экрана, скрытию и показу открытых окон, переходу между окнами, изменению их размера.

В третьей секции собрано описание расширенных возможностей для редактирования текстов, как то:

- работа по протоколированию, записи и воспроизведению макросов (субсекция MACROS);
- команды для прокрутки текста (субсекция SCROLL);
- команды для взаимодействия с оболочкой (субсекция SHELL);
- средства установки закладок, наращиваемого поиска, ввода спецсимволов (секции

BOOKMARKS, I-SEARCH, QUOTE) и так далее.

Четвертая секция - расширенные возможности для программистов (команды перехода к регулярным выражениям, компилирования и отладки). В пятой секции дано описание сложных регулярных выражений. И, наконец, шестая секция - операции с командной строкой встроенной в редактор командной оболочки.

Таковы возможности справочной системы joe по умолчанию. Как будет показано ниже, пользователь может не только изменять имеющиеся секции, но и создавать собственные, в любом количестве и для любых целей.

Если мне удалось убедить читателя, что joe - вещь стоящая, имеет смысл подробнее ознакомиться с его характерными особенностями.

## Характерные особенности

Для эффективного использования joe следует четко уяснить, что это - командный редактор в чистом виде. То есть все действия по редактированию текста осуществляются соответствующими встроенными командами, к которым привязаны комбинации клавиш. В сущности, как будет показано ниже, это - макросы на собственном языке joe. Из чего следует, что, с одной стороны, система команд может быть сколь угодно наращена, с другой - что клавишные комбинации для них могут быть переопределены произвольным образом.

Впрочем, необходимости в последнем я не вижу. Структура предопределенных по умолчанию клавишных комбинаций проста и очень логична. За самыми простыми и частыми действиями для навигации и редактирования закреплены двухклавишные комбинации. Это, как правило, сочетание одновременно нажатых клавиш **Control** и буквенной. Последняя имеет, обычно, мнемонический смысл, хотя и не всегда прозрачный.

Полный список встроенных команд и привязанных к ним клавишных комбинаций дан в заключительном разделе этой статьи. Здесь же я приведу только основные примеры. Так, комбинация **Control**+**B** (от backward) перемещает курсор на один знак влево, **Control**+**F** (от forward) - на один знак вправо, **Control**+**Z** - переход к предыдущему слову, **Control**+**X** - к последующему слову, и так далее.

В некоторых случаях в качестве управляющей клавиши используется клавиша **Meta** (напомню, что ее эквивалент - нажатие и отпускание клавиши **Escape**): если в комбинации с ней набрать литеру **W** - курсор переместится на строку вверх, литеру **Z** - на строку вниз. Кроме того, **Meta** служит для вызова проверки правописания для слова (**Meta**+**N**) и всего файла (**Meta**+**L**). Нажатие клавиши **Meta** два раза подряд приводит к установке закладки (bookmark), которая маркируется произвольной цифрой, а **Meta**+**#** (где **#** - эта самая цифра) вызывает переход к установленной закладке. Правда, очевидно, что закладок не может быть больше 10; и к тому же по завершении сеанса они не сохраняются.

Все клавишные комбинации в joe не чувствительны к регистру, причем не только для буквенных, но и символьных клавиш. Так, для отмены последней операции (как уже говорилось, многоуровневой) зарезервирована комбинация **Control**+_ (знак подчеркивания), а для возврата отмененного действия - **Control**+^; однако в первом случае работает также комбинация **Control**+**-** (дефис или минус), во втором - **Control**+**6**.

Кроме того, все двухклавишные комбинации не чувствительны и к раскладке клавиатуры: сочетание клавиш, например, **Control**+**T** будет вызывать систему настройки joe и при

кириллической раскладке. Интересно, что для пролистывания страниц помощи вперед и назад при кириллической раскладке следует нажимать **Escape** и точку (или, соответственно, запятую) также в ее положении на русифицированной клавиатуре (то есть в нижнем правом углу для Windows-клавиатур и на верхнем регистре цифр 5 и 7, если мне не изменяет память, - для клавиатур с DOS-маркировкой).

Для более сложных или редких действий используются трехклавишные комбинации. Это почти исключительно одновременно нажатые **Control**+**K**, после чего нажимается литерная клавиша. Так, операции с блоками осуществляются следующим образом:

- **Control**+**K** - **B** отмечает начало выделяемого блока,
- **Control**+**K** - **K** - его конец,
- **Control**+**K** - **C** - копирует,
- **Control**+**K** - **M** - перемещает выделенный блок в позицию курсора,

и так далее. Трехклавишные комбинации также не чувствительны к регистру. И работают также и при кириллической раскладке клавиатуры. В этом случае только необходимо нажимать вторую литерную клавишу вместе с той же клавишей **Control**: то есть запись текущего файла при включении кириллической раскладки потребует комбинации **Control**+**K** - **Control**+**D**, вызов нового файла - **Control**+**K** - **Control**+**E**, и так далее.

В joe нет отдельной функции для переименования файла. Но при любой записи текущего документа следует запрос на подтверждение имени файла, изменить которое при этом никто не запрещает. Следует только помнить, что дальнейшая работа происходит с исходным, а не переименованным файлом.

Кроме того, в joe доступны еще некоторые действия с файлами. Так, комбинация **Control**+**K** - **R** вставляет текст из существующего файла в позицию курсора, **Control**+**K** - **W** - записывает выделенный блок в виде нового файла (разумеется, запросив предварительно его имя). С помощью комбинации **Control**+**K** - **E** можно открыть для редактирования другой существующий файл. При этом следует предложение ввести путь и имя, причем и для того, и для другого работает режим дополнения клавишей табуляции, как в командных средах bash или tcsh.

Между одновременно открытыми файлами возможен обмен данными: или с помощью команд joe (то есть выделением блока в первом файле и его копированием или перемещением во второй), или с помощью мыши - стандартным выделением и вставкой щелчком средней ее клавиши. Второй способ, естественно, может применяться и для обмена между разными копиями joe, запущенными на отдельных виртуальных консолях.

Одновременно открытые файлы могут быть представлены как в полноэкранном виде, так и каждый в своем окне. Для переключения между однооконным и многооконным режимами служит комбинация **Control**+**K** - **I**. Размер каждого из выведенных окон может быть увеличен или уменьшен (**Control**+**K** - **G** и **Control**+**K** - **T**, соответственно), правда, только с шагом в одну экранную строку. Переключение между открытыми документами, вне зависимости от режима, осуществляется комбинациями **Control**+**K** - **N** (вперед или вниз) и **Control**+**K** - **P** (назад или вверх).

К слову сказать, в joe возможен и независимый просмотр разных частей документа в отдельных окнах, для чего предназначена функция расщепления окна (**Control**+**K** - **O**). Ну и, конечно же, фрагменты из одной части файла могут быть легко перенесены в другую.

Универсальной комбинацией для окончания любой операции в joe является **Control**+**C**. С ее помощью закрывается окно с текущим документом; если он был единственным в данном сеансе, одновременно происходит и выход из редактора. В обоих случаях следует запрос на сохранение изменений, буде таковые имелись. Отказаться от выхода или закрытия файла можно повторным нажатием той же комбинации **Control**+**C**. Она же используется для прекращения любой длящейся во времени (спеллинг, поиск) или требующей подтверждения

операции.

Кроме этого, непосредственно из joe, без выхода, можно обращаться к командам оболочки (shell), причем - различными способами. Так, комбинация **Control**+**K** - **Z** обеспечивает временный выход в оболочку, где можно вводить любые ее команды. А по завершении операций - вернуться в среду joe можно комбинацией **fg**. То есть в данном случае мы имеем дело с обычной приостановкой текущей задачи.

Кроме этого, есть и более интересная возможность: открытие внутри joe, посредством комбинации **Control**+**K** - **'** (апостроф), самостоятельного окна с собственной командной оболочкой. Здесь можно выполнять любые команды с выводом их результатов на экран. После чего стандартной командой exit осуществляется выход из среды, а все результаты сохраняются обычным для joe образом в виде текстового файла: возможность неоценимая при создании и редактировании всякого рода скриптов.

Следует подчеркнуть, что в данном случае запускается именно встроенная в joe оболочка, не имеющая никакого отношения к пользовательской (login shell). Она не порождает нового процесса (в чем легко убедиться командой ps) и не наследует никаких переменных окружения от пользовательской оболочки. Интерактивные ее возможности достаточно слабы (не поддерживаются ни автодополнение, ни история команд). Однако она дает возможность интеграции возможностей joe по редактированию и команд оболочки для обработки текста.

Единичная команда оболочки может быть выполнена после нажатия комбинации клавиш **Meta**+**!**. В этом случае внизу экрана появляется приглашение командной строки в форме:

```
Program to run:
```

представляющее собой разновидность встроенной оболочки joe.

Наконец, в joe обеспечивается ввод специальных символов. Так, нажатие клавиши ` (обратный апостроф) приводит к предложению ввести первый знак десятичного (0-9), шестнадцатеричного (x) или восьмеричного (o) кода символа. Если же вместо кода нажать **Escape** - можно ввести любую esc-последовательность. То же самое можно сделать и после нажатия комбинации **Control**+\.

Специальные символы могут вводиться не только непосредственно в редактируемом тексте, но и в строке поиска. Это позволяет легко находить и глобально заменять лишние разрывы строк и тому подобные нарушения структуры.

Таковы основные возможности joe для редактирования текстов общего характера. Кроме этого, имеется ряд команд специального назначения, используемых при программировании (поиск ошибок, компилирование и проч.), однако их я за ненадобностью (мне) не изучал. Однако и сказанного достаточно для представления возможностей редактора joe.

## Макрокоманды

Если же штатных команд редактора joe оказывается недостаточно, можно прибегнуть к их самостоятельному конструированию. Для чего предназначен внутренний язык макрокоманд. К сожалению, он не описан в экранной документации. Однако получить представление о его синтаксисе и возможностях можно, изучив внимательно конфигурационный файл joerc (о чем

подробнее - в следующем разделе).

Кроме того, есть еще один простой и эффективный способ изучения макроязыка, совмещающий приятное с полезным, - режим протоколирования макрокоманд. Включается он комбинацией клавиш **Control**+**K** - **[** (открывающая квадратная скобка), вслед за чем следует ввести номер макроса (от 0 до 9), выступающий как в качестве его имени, так и в роли запускающей клавиши. Далее просто выполняются необходимые действия (например, вводится требуемый тэг html), после чего запись макроса останавливается комбинацией **Control**+**K** - **]** (закрывающая квадратная скобка). Для воспроизведения запротоколированного макроса используется комбинация **Control**+**K** - **#** (где **#** - указанный при записи номер макрокоманды).

С помощью протоколирования макросов можно автоматизировать ввод наиболее нужных для конкретной задачи символов и их наборов, не предусмотренных штатным образом. Например, основных тэгов html для разметки web-страниц, таких, как параграф, разрыв строки, заголовки нескольких уровней, таблицы и списки. А заодно - и изучить синтаксис языка. Поскольку в joe предусмотрена возможность помещения запротоколированных в данном сеансе макросов в тело существующего или нового документа (комбинацией клавиш **Meta**+**D** ).

Например, записанные макросы для ввода html-тэгов выглядят следующим образом:

```
General Structure
insf,"~/blank.html",rtn  ^K .k1    HTML Page
"",ltarw,ltarw,ltarw,ltarw,ltarw       .k1     H1
rtn,rtn,"",ltarw,ltarw,ltarw,ltarw,ltarw       .k2     H2
rtn,rtn,"\09","",ltarw,ltarw,ltarw,ltarw,ltarw        .k3     H3
rtn,rtn,"\09","\09","",ltarw,ltarw,ltarw,ltarw,ltarw .k4     H4

Body Text
rtn,rtn,"<p>","</p>",ltarw,ltarw,ltarw,ltarw    .k5     Paragraph
rtn,"<br>"      .k6  Break
rtn,rtn,"<pre>",rtn,rtn,"</pre>",uparw  .k7     Preformat
"<strong></strong>",ltarw,ltarw,ltarw,ltarw,ltarw,ltarw,ltarw,ltarw,ltarw .k8     Str
ong
"<em></em>",ltarw,ltarw,ltarw,ltarw,ltarw       .k9     Emphasis
rtn,rtn,"<div>",rtn,rtn,"</div>",uparw  ^K .k8  Division

Lists and Tables
rtn,rtn,"<ul>",rtn,rtn,"</ul>",uparw    ^K .k2  Unordered List
rtn,rtn,"<ol>",rtn,rtn,"</ol>",uparw    ^K .k3  Ordered List
"<li>,</li>",ltarw,ltarw,ltarw,ltarw,ltarw       ^K .k4  List Item
rtn,rtn,"",rtn,rtn,"<table></table>",uparw      ^K .k5  Table
```

Из чего вполне можно составить представление о командах языка: в первой колонке следуют разделяемые запятыми (без пробелов!) зарезервированные команды (rtn - **Enter**, ltarw - **Left**, uparw - **Up** и т.д.) и вводимые символьные значения в парных кавычках ("<p>"). Далее идут разделенные табулятором клавишные комбинации (^K 0 - **Control**+**K** - **0**), закрепленные за макросами, и имена макросов. Последние в оригинале представлены в виде Macro 0, Macro 1 и т.д. Но ничего не запрещает при редактировании придать им осмысленные имена. Да и сами макросы могут быть отредактированы должным образом в текстовом редакторе (том же joe, например).

Легко сообразить, что за один прием можно запротоколировать не более 10 макросов (маркированных цифрами от 0 до 9). Более того, они будут действовать только в течении данного сеанса: по выходе из редактора записанные макросы сами собой не сохраняются. Что, казалось бы, напрочь обесценивает данную возможность.

К счастью, это не так. И раз запротоколированные макрокоманды можно сохранить для дальнейшего использования. Да и количество их не обязано ограничиваться десятью. Как этого добиться?

Выясняется, что за воспроизведение макросов отвечает конфигурационный файл ~/.joerc, о котором подробнее будет сказано в следующем разделе. И потому достаточно поместить (с помощью **Meta**+**D** или из текстового файла) в соответствующую его секцию (какую - также скажу позднее) созданные команды, чтобы обеспечить их исполнение во всех последующих сеансах.

Более того, назначенные по умолчанию клавишные комбинации не являются обязательными. И их можно вручную заменить на любые другие, из числа не использованных ранее. После чего можно начать протоколирование команд сначала, нумеруя их от 0 до 9. Затем - повторить процедуру встраивания их ~/.joerc, и так далее, сколько потребуется (или до исчерпания комбинаторики клавиш). А поскольку количество незадействованных в joe клавишных комбинаций очень велико, то реально число созданных пользователем макрокоманд ограничивается только его фантазией и потребностями.

Таким образом можно легко автоматизировать процесс ввода тэгов HTML или XML, конструкций JavaScript, скриптов командной среды, разметки документов TeX, а также все, что потребуется впредь. Превратив joe в специализированный инструмент для решения почти любых задач.

## Настройка joe

Как я уже говорил, некоторые настройки joe можно выполнить интерактивно (вызвав их комбинацией **Control**+**T**). Однако они весьма ограничены и к тому же будут иметь силу только в текущем сеансе. Более интересные и богатые возможности открываются при редактировании конфигурационного файла joerc.

Пример такого файла обнаруживается в каталоге /usr/local/etc/joerc. Перво-наперво его надлежит скопировать с свой домашний каталог и переименовать в ~/.joerc - именно этот файл ищется в первую очередь при загрузке редактора. А затем он открывается в любом текстовом редакторе - лучше всего в самом же joe и, после соответствующего изучения, правится вручную.

В экранной документации синтаксис конфигурационного файла не описан, что мотивируется его простотой и понятностью. Это действительно почти так - смысл практически всех настроек можно понять из контекста (и комментариев).

Файл joerc разбит на четыре секции. Первая - это глобальные опции редактора, большая часть которых может быть задана также параметрами командной строки. Все они - односложные и имеют вид -имя_опции (установить данную опцию) или --имя_опции (отменить ее). Опция является установленной (или, напротив, отмененной), если ею начинается строка (первая колонка, в терминологии программы - это относится и ко всем остальным секциям .joerc). Если строка начинается с пробела или табуляции, дальнейшее ее содержание рассматривается как комментарий и не учитывается. Комментарием является и все, отделенное пробелом от имени опции.

Описание всех опций заняло бы слишком много места, поэтому обращу внимание только на некоторые ключевые моменты. Обязательно следует включить (то есть удалить пробел в

начале строки, если он имеет место быть) опцию

**-asis**

Это необходимо для правильного отображения символов кириллицы - иначе они будут показаны латинской транслитерацией.

Полезным представляется также установка опций:

**-lightoff**

обеспечивающей выключение подсветки выделенного блока после его перемещения или копирования, - иначе выделение это будет маячить перед глазами;

**-marking**

дающей подсветку текста между началом выделяемого блока и текущей позицией курсора. Можно отменить также создание страховых копий или, напротив, определить место для их помещения, отличное от исходных файлов. Это достигается включением опций

**-nobackups**

или

**-backpath path**

соответственно.

Интересная возможность - задание количества строк и колонок (знаков в строке) на экране, отличное от определенных для текущего терминала (виртуальной консоли) в целом, что задается опциями

**-lines #**
**-columns #**

где # - количество строк и знаков, соответственно.

В этой же секции настраивается вид статусной строки (вывод которой, впрочем, можно и отключить опцией -nosta). Он определяется двумя опциями -lmsg и -rmsg. Первая определяет компоненты, выводимые в левой части строки, вторая - в правой. В любой из них можно вывести индикацию режимов (забивки или вставки, переноса слов, автоотступов), имя файла и показатель его изменения, текущее положение курсора (в строках, колонках или знаках), и т.д. Первый знак левой части статусной строки - escape-последовательность, определяющая ее общий вид: инвертирование цветов, выделение подчеркнутым или полужирным начертанием, мерцание.

Например, строка вида

```
-lmsg \i%k%T%W%I%X %n %m%R %M
```

указывает, что в левой части статусной строки в инвертированном виде (черным по белому, \i) должны быть выведены

- префиксные ключи (%k), маркирующие включение режимов вставки/замены (%T), переноса слов (%W), автоотступа (%I), прямоугольного выделения (%X);
- имя редактируемого файла (%n);
- указание на модификацию файла (%m) и на режим "только для чтения" (%R);
- индикатор включения протоколирования макросов (%M).

Строка же вида

```
-rmsg  Row %r Col %c %o %O %u
```

выводит в правой части статусной линии номер текущей строки (Row %r) и колонки (Col %c) файла, смещение от начала в байтах (в десятичной, %o, и шестнадцатеричной, %O, формах, и системное время в 24-часовом формате(%u).

Возможен вывод и иной информации, как то:

- системного времени в 12-часовом формате (%t);
- индикации измененного файла символом *;
- ASCII-кода символа под курсором в десятичной (%a) или шестнадцатеричной (%A) форме;
- процента просмотра файла до позиции курсора (%p);
- общего количества строк в файле (%l);
- индикации запуска встроенной оболочки (%S).

Кроме того, внешний вид статусной строки можно изменить, придав ему, вместо или помимо инверсии, атрибуты подчеркивания (\u), полужирного начертания (\b), мерцания (\f).

Вторая секция конфигурационного файла - это локальные опции, которые можно определить отдельно для файлов различных типов. Для этого в ее составе создаются субсекции вида

```
* ??? ?????
*.html ????????? HTML
*.c ????????? ?? C
*rc ???????????????? ?????
```

и так далее. Напомню здесь, что знак маски (*) должен обязательно начинаться с начала строки, то есть занимать первую колонку.

Для каждой субсекции можно независимо задать такие параметры, как перенос слов (или его отключение), автоматический отступ, величину табуляции и т.д., а также, что интересно, назначить собственную раскладку клавиатуры, отличную от определенной для консоли в целом. Кроме того, с файлами любого типа можно связать макросы, выполняемые при их загрузке или записи.

Третья секция описывает вид экранов помощи, выводимых клавишной комбинацией **Control**+

**K** - **H**. Экраны эти могут быть изменены как с позиций внешнего вида (инверсия цветов, выделение или подчеркивание и т.д.), так и по существу. В частности, здесь можно задать специальные экраны помощи для собственных макрокоманд. Более того, редактированием этой секции можно выводить и подсказки на русском (или каком-либо еще) языке. В частности, для макросов ввода тэгов html, описанных в одном из предыдущих разделов, можно создать экран помощи вида:

```
{HTML
\i   Help Screen    turn off with ^KH    more help with ESC . (^[.)
\i \i\uGeneral\u     \uBody Text\u               \uList&Table\u
\u
\i \i^KF1 Blank  F5 P  F7 Pre ^KF8 Div  ^KF2 UL ^KF3 OL ^KF4 LI  F10 Href
^KF9
\i \iF1-F4 H1-H4 F6 Br F8 Strong F9 Em  ^KF5 Tab ^KF6 TR ^KF7 TD ^KF10 Name
}
```

Каждый такой экран заключается в фигурные скобки, предваряется собственным идентификатором (в примере - HTML), ему придаются атрибуты внешнего вида (в примере - инвертирование рамки, \i), задается строка подсказки, после чего следует просто перечисление клавишных комбинаций и их описание.

Четвертая секция - разного рода ключевые последовательности, или связки (key bindings), в том числе и макрокоманды. Они могут быть определены отдельно для всех окон (субсекция :windows, опять же начиная с первой колонки), окна редактируемого текста (субсекция :main), и так далее.

Внимательное рассмотрение секции показывает, что все описанные выше (и перечисленные в приложении) клавишные комбинации joe представляют собой макрокоманды, именно здесь и определенные. Из чего следует два вывод.

Первый - возможность переопределения клавиатурных комбинаций, назначенных для штатных команд joe по умолчанию. Например, если вам не нравится, что пролистывание экрана осуществляется комбинациями **Control**+**U** (назад) и **Control**+**V** (вперед), можно присвоить им иные значения (из числа свободных, разумеется).

Второй вывод - штатные команды joe могут быть (почти) неограниченно наращены собственными макрокомандами. Для чего достаточно их запротоколировать и поместить (напомню, посредством **Escape** - **D**) в соответствующее место четвертой секции (скорее всего, в раздел :windows или :main) файла ~/joerc. Более того, здесь же им можно присвоить и членораздельные имена (вместо данных по умолчанию Macro 1, Macro 2 и т.д.), во-первых, и изменить закрепленные клавишные комбинации (по умолчанию - **Control**+**K** - **0** и т.д.) - во-вторых. То есть полностью индивидуализировать редактор, не прибегая ни к каким сильнодействующим средствам (вроде перекомпиляции исходных текстов).

Пользовательские макросы следует помещать в одну из субсекций - :window или :main. В последней, кроме этого, целесообразно отредактировать имеющиеся там по умолчанию макросы проверки орфографии для работы с русским словарем. По умолчанию эти макросы описываются строками вида:

```
:def spellfile
:def spellword
```

для спеллинга всего текста и отдельных слов, соответственно. Они просто описывают параметры вызова внешней программы проверки орфографии (по умолчанию - ispell). И подключение русского словаря при этом, разумеется, не предусмотрено. Чтобы осуществить

это, следует просто добавить к обеим строкам после вызова исполнимой команды ispell опцию -d russian, в результате чего строки эти примут вид:

```
:def spellfile filt,"cat >ispell.tmp;ispell -d russian ispell.tmp \
</dev/tty >/dev/tty;cat ispell.tmp;/bin/rm ispell.tmp",rtn,retype
:def spellword psh,nextword,markk,prevword,markb,filt,"cat \
>ispell.tmp;ispell -d russian ispell.tmp </dev/tty >/dev/tty;tr -d
\<ispell.tmp '\\012';/bin/rm ispell.tmp",rtn,retype,nextword
```

Символы обратного слэша в конце строк означают только то, что каждая строка на самом деле не должна прерываться - вносить их в тело макроса не следует (нужно только позаботиться, чтобы опция переноса слов в редакторе была отключена).

## Дополнительные настройки

В начале своего повествования я упоминал, что ныне (начиная с версии 3.0) в joe появилась возможность подсветки синтаксиса для ряда языков программирования и разметки. Чтобы ей воспользоваться, нужно, во-первых включить соответствующую опцию в конфигурационном файле ~/.joerc - она расположена в секции Default local options и имеет вид

```
-highlight
```

Далее, требуются файлы описания цветов для синтаксических элементов различных языков. Примеры таких файлов расположены в /usr/local/etc/joe/synatx и охватывают языки Си (c.jsf), Assembler (asm.jsf), Fortran (fortran.jsf) и многие другие. Есть здесь и файлы описания языков командных оболочек (sh.jsf и csh.jsf), diff-файлов (diff.jsf), конфигов (conf.jsf), а также языков разметки (html.jsf и xml.jsf). Единственное, что остается с ними сделать - это отредактировать их в соовтетствие с предпочтительной цветовой гаммой (и, вероятно, сохранить в собственном домашнем каталоге под именами типа .html.jsf и так далее).

Наконец, последнее - это проверить соответствие файлов описаний в главном конфигурационном файле ~/.joerc. По умолчанию в какждой языковой субсекции они указываются там в виде:

```
-syntax html
```

и так далее, в предположении их стандартных имен и размещения (в указанном выше каталоге /usr/local/etc/joe/synatx.

## Заключительные соображения

И так, чем же может привлечь пользователя редактор joe? На мой взгляд, многим. Разумеется, если он, то есть пользователь, вообще испытывает потребность в консольном редакторе.

Во-первых, в его пользу - относительная (по сравнению, скажем, с vim или, тем более, emacs) простота освоения и использования. Немаловажно, что элементарные действия по вводу и редактированию текста в нем могут осуществляться (в том числе и) привычным для выходца из мира DOS/Windows способом.

Второй плюс - единообразность модели выполнения команд, особенно отчетливо проступающая в сравнении с emacs. Модель эта логична и легко запоминается, в том числе и благодаря мнемоническому характеру литерных клавиш, сочетающихся с управляющими.

Если же сравнивать joe с меню-ориентированными редакторами, такими, как mcedit или le, то его отличают а) очень высокая степень настраиваемости (практически не уступающая классическим редакторам командного стиля), и б) быстрота выполнения основных операций по вводу и редактированию. Впрочем, конечно, быстрота эта (как и во всех командных редакторах) достижима только при наработке определенного минимума практических навыков, желательно - доведенных до рефлекторного уровня.

Однако подчеркну, что такие навыки появляются достаточно быстро. И, кроме того, имеются альтернативные им, традиционные (для DOS/Windows) приемы навигации по тексту и прочее. Что приближает joe по простоте использования к меню-ориентированным редакторам и делает его пригодным и для эпизодического использования. Чего, в общем случае, нельзя сказать ни о vim, ни о emacs - эффективное их использование возможно только при постоянной практике.

Весьма удобными представляются средства одновременной работы с большим количеством документов и обмена данными между ними. Возможность независимого просмотра различных частей одного файла в отдельных окнах также следует отнести к числу достоинств (коими не обладает, например, le).

Достаточно просто реализована возможность вставки специальных символов, escape-последовательностей и тому подобных вещей. Что гармонично дополняется возможностью определения для joe клавиатурной раскладки, отличной от используемой в консоли по умолчанию.

Очень эффективно применение joe для составления и редактирования пользовательских сценариев командой оболочки - благодаря возможностям временного выхода в командную строку - раз, запуску единичной команды внутри редактора - два, и открытию почти полноценного сеанса командной оболочки изнутри него же (с возможностью записи в виде файла) - три. Судя по документации и конфигурационным файлам, есть средства и для более сложных программистских упражнений, но об их эффективности судить я не компетентен.

Наконец, главное достоинство joe - простота адаптации к специальным задачам, не требующая ни программирования на LISP, ни иных сильнодействующих средств. Она вполне достижима элементарным протоколированием макрокоманд и несложным их редактированием. Вероятно, такие возможности покажутся смешными записному программисту, но пользователю, профессионально связанному с подготовкой нарративных текстов, они в большинстве случаев более чем достаточны.

К принципиальным упущениям joe можно отнести, пожалуй, только отсутствие режима переноса символов без образования новой строки, подобного умолчальным для vim или emacs: режим word wrapping приводит к разрыву непрерывности абзаца (что в дальнейшем может создать сложности при автоматизированной обработке текста средствами sed или awk), а его отключение - к неудобству набора.

Иными словами, joe - достаточно мощный и удобный инструмент для работы с текстами. Он может быть использован и при периодических работах (вроде правки конфигурационных файлов или составления небольших документов). Но наиболее эффективно применять joe при повседневной работе с большими и структурированными нарративными текстами, особенно - предназначенными для публикации в Сети. Среди всех консольных редакторов joe отличается близким к оптимальному соотношением простоты, мощности и настраиваемости. А посему

беру на себя смелость рекомендовать его всем любителям работы в текстовом режиме, буде до сего времени они не приобрели иных пристрастий.

## Приложение. Штатные команды joe

В заключение приведу более-менее подробную справку по штатным командам редактора joe, примерно в соответствие с системой его встроенной помощи.

### Перемещения курсора:

- **Control**+**B** на один знак влево (эквивалентно **Left**);
- **Control**+**F** на один знак вправо (эквивалентно **Right**);
- **Control**+**P** на один знак вверх (в той же колонке по вертикали, эквивалентно **Up**);
- **Control**+**N** на один знак вниз (в той же колонке по вертикали, эквивалентно **Down**);
- **Control**+**Z** к предыдущему слову;
- **Control**+**X** к следующему слову.

### Переходы по тексту и скроллинг:

- **Control**+**U** к предыдущей экранной странице (эквивалентно **PageUp**);
- **Control**+**V** к следующей экранной странице (эквивалентно **PageDown**);
- **Control**+**A** к началу строки (эквивалентно **Home**);
- **Control**+**E** к концу строки (эквивалентно **End**);
- **Control**+**K**-**U** к началу файла;
- **Control**+**K**-**V** к концу файла;
- **Control**+**K**-**L** к строке по номеру;
- **Meta**+**0**...**9** переход к нумерованной закладке;
- **Meta**+**Control**+**[** установка нумерованной закладки;
- **Meta**+**B** к началу отмеченного (через **Control**+**K**-**B**) блока;
- **Meta**+**K** к концу отмеченного (через **Control**+**K**-**K**) блока;
- **Meta**+**W** смещение на одну строку вверх без изменения позиции курсора;
- **Meta**+**Z** смещение на одну строку вниз без изменения позиции курсора;
- **Control**+**K**- **<** смещение на один экран влево;
- **Control**+**K**- **>** смещение на один экран вправо.

### Действия с блоками:

- **Control**+K-**B** начало выделения блока;
- **Control**+K-**K** конец выделения блока;
- **Control**+K-**M** перемещение выделенного блока в позицию курсора;
- **Control**+**K**-**C** копирование выделенного блока в позицию курсора;
- **Control**+K - **W** сохранение выделенного блока в виде файла;
- **Control**+K - **Y** удаление блока;
- **Control**+**K**-**/** выделение всего текста для применения команд-фильтров оболочки, например, сквозной сортировки (sort), поиска фрагмента (grep) и т.д.

### Удаление:

- **Control**+**D** знака в позиции курсора (эквивалентно **Delete**);
- **Control**+**H** знака перед курсором (эквивалентно **Backspace**);

- **Control**+**Y** строки целиком;
- **Control**+**W** следующего слова (или пробела);
- **Control**+**O** предыдущего слова (или пробела);
- **Control**+**J** части линии справа от курсора до конца;
- **Meta**+**Y** разрыва строки;
- **Meta**+**O** линии от позиции курсора до начала;
- **`-#** вставка кода символа, # - номер кода в десятичной (начиная с 0), восьмеричной (начиная с o) или шестнадцатеричной (начиная с x) форме;
- **Control**+**** вставка Escape-последовательности;
- **Control**+**_** отмена последней операции;
- **Control**+**^** возврат отмененной операции.

## Поиск и спеллинг:

- **Control**+**K**-**F** текстового фрагмента;
- **Control**+**L** следующего вхождения;
- **Meta**+**R** наращиваемый поиск назад;
- **Meta**+**S** наращиваемый поиск вперед;
- **Meta**+**N** проверка орфографии по словам;
- **Meta**+**L** проверка орфографии для всего файла.

## Выход из редактора и команды оболочки:

- **Control**+**K**-**X** с записью;
- **Control**+**C** без записи, с запросом на подтверждение утраты сделанных изменений, буде таковые имелись; также закрытие текущего окна и прерывание операции;
- **Control**+**K**-**Z** временный выход в командную оболочку с переводом работы редактора в фоновый режим;
- **Control**+**K**-**'** открытие окна встроенной командной оболочки;
- **Meta**+**!** запуск одиночной команды во встроенной оболочке.

## Файловые операции:

- **Control**+**K**-**E** вызов файла для редактирования (в новом окне);
- **Control**+**K**-**R** вставка существующего файла в текущий (в позицию курсора);
- **Control**+**K**-**D** запись текущего файла с запросом имени.

## Управление окнами:

- **Control**+**K**-**O** разделение окна пополам (только по горизонтали);
- **Control**+**K**-**E** открытие файла в новом окне;
- **Control**+**K**-**G** увеличение размера текущего окна (с шагом в одну строку);
- **Control**+**K**-**T** уменьшение размера текущего окна (с шагом в одну строку);
- **Control**+**K**-**N** переход к нижнему (или предыдущему) окну;
- **Control**+**K**-**P** переход к верхнему (или последующему) окну;
- **Control**+**C** закрытие текущего окна;
- **Control**+**K**-**I** показать или скрыть все окна, кроме текущего (переключатель).

## Протоколирование макрокоманд:

- **Control**+**K**-**[ 0** ... **9** включение режима протоколирования макроса;
- **Control**+**K**-**]** выключение режима протоколирования макроса;
- **Control**+**K** - **0** ... **9** воспроизведение запротоколированного в текущем сеансе макроса по его номеру;
- **Meta**+**D** вставка записанных в данном сеансе макросов в текущий файл.

## Разное:

- **Control**+**K**-**J** форматирование абзаца;
- **Control**+**K**-**A** центрирование строки;
- **Control**+**K** -, уменьшение отступа;
- **Control**+**K-.** увеличение отступа;
- **Control**+**T** вызов настроек (имеющих силу только в текущем сеансе);
- **Control**+**R** обновление экрана;
- **Control**+**K**-**SPACE** вывод строки статуса в нижней части экрана (номер текущей строки и колонки, смещение от начала файла, ASCII-код символа под курсором);
- **Control**+**K**-**** повторение команды;
- **Meta**+**M** ввод регулярного выражения;
- **Meta**+**H** отправка сообщения (нечто вроде команды write, но на одном и том же терминале).

Материал взято **отсюда** [2]

**Источник (получено *2025-12-09 18:10*):**
http://muff.kiev.ua/content/joe-tekstovyi-redaktor-znakomstvo-i-vozmozhnosti

**Ссылки:**
[1] http://joe.sourceforge.net/
[2] http://unix.ginras.ru/apps/ed006.html