

ТЕА - ПОЛНОЕ НОГОВОДСТВО

Петр "Roxton" Семилетов

<peter.semiletov@gmail.com>

Эта книжечка - общественное достояние (public domain).

Содержание

1. [Необходимое предисловие](#)
2. [Интерфейс](#)
 - 2.1. [Главное окно](#)
 - 2.2. [Перетаскивание](#)
 - 2.3. [Диалоговые окна Открыть/Сохранить файл](#)
3. [Как ТЕА ладит с разными кодировками?](#)
4. [О редактировании и просмотре субтитров формата SRT](#)
5. [Трансцендентальное сияние меню Файл](#)
6. [Меню Правка](#)
7. [Поиск](#)
 - 7.1. [Обычные функции поиска и замены](#)
 - 7.2. [Цветовые пометки](#)
8. [Вёрстка](#)
 - 8.1. [Обо всем этом](#)
 - 8.2. [Утилиты HTML](#)
 - 8.3. [Общие функции](#)
 - 8.4. [Списки](#)
9. [Функции](#)
 - 9.1. [Общие](#)
 - 9.2. [Календарик](#)
 - 9.3. [IDE](#)
 - 9.4. [Проверка орфографии](#)
 - 9.5. [LaTeX](#)
 - 9.6. [УНИТАЗ](#)
 - 9.7. [Скрипты](#)
 - 9.7.1. [Скрипты с точки зрения пользователя](#)
 - 9.7.2. [Скрипты с точки зрения разработчика](#)
 - 9.8. [Таблицы замен](#)
 - 9.9. [Сниппеты](#)
 - 9.10. [Инструменты](#)
 - 9.11. [Формат](#)
 - 9.12. [Insert](#)
 - 9.13. [Регистр](#)
 - 9.14. [Числа](#)
 - 9.15. [Строка](#)
 - 9.16. [Сортировка](#)

- 9.17. [Фильтр](#)
 - 9.18. [Код Морзе](#)
 - 10. [Запуск](#)
 - 11. [Навигация](#)
 - 12. [Вид](#)
 - 12.1. [Функции меню Вид](#)
 - 12.2. [Подменю Ко](#)
 - 13. [Настройки](#)
 - 13.1. [Вступление](#)
 - 13.2. [Переключатели](#)
 - 13.3. [Функции](#)
 - 13.4. [Интерфейс](#)
 - 13.5. [Пределы](#)
 - 13.6. [Редактор](#)
 - 13.7. [Пути](#)
 - 14. [Квас](#)
 - 14.1. [Действия](#)
 - 14.2. [Вставка в редактор](#)
 - 15. [Опции командной строки](#)
 - 16. [Об авторе](#)
- A. [Ресурсы](#)

Глава 1. Необходимое предисловие

Tibi et igni

Осенью 2000 года я начал делать текстовый редактор под Windows, который вначале назывался Typewriter, а к сентябрю 2001 года был переименован в ТЕА. До 2004 был одним из самых популярных в Рунете текстовых редакторов. Аккурат под Новый год я приостановил разработку [ТЕА \(win32\)](#), чтобы сконцентрировать свои усилия на создании Linux-версии ТЕА. Однако, Win32-версия снова в процессе постоянной обработки, а к тому же еще и имеет статус public domain (общественное достояние), и код ее открыт. Но эта документация относится к Linux-версии.

Я не программист. Я [писатель](#) и журналист. Иногда делаю программы, которые мне нужны. Последние годы я писал только в ТЕА и перейдя в Linux, не мог отказаться от такой привычки. Надо сказать, что ТЕА for Linux - моя первая программа на Си и моя первая серьезная программа под Linux.

Летом 2005 года я сделал форк ТЕА и назвал его Chai. Chai использовал в качестве текстового движка GtkSourceView, а также зависел еще от нескольких библиотек: GnomeVFS и LibGconf. Немного позже обе ветки были сведены в одну общую, под традиционным названием ТЕА. По умолчанию скрипт *configure* настраивает исходник для сборки в режиме обычного ТЕА. Чтобы воспользоваться преимуществами GtkSourceView, надо сконфигурировать исходник командой: *./configure --enable-chai*, а затем дать привычные команды *make*, *make install*. Что до пакетов для различных дистрибутивов, то большинство из них собрано, насколько я

знаю, с GtkSourceView.

Пожалуйста, прочтите это ноговодство внимательно, потому что ТЕА местами отличается от общепринятых понятий.

Кстати, можете свободно задавать мне вопросы, вносить предложения и слать патчи на адрес - peter.semiletov@gmail.com

Now playing: Nirvana - I Hate Myself And I Want To Die

Глава 2. Интерфейс

2.1. Главное окно

Вот расположение элементов управления в ТЕА, сверху вниз:

- 0. Заголовок окна.
- 1. Панель инструментов (если включена в **Настройках**).
- 2. Меню.
- 3. Область вкладок (табов) с текстом.
- 4. Штука, называемая Логмемо, куда выводятся всякие информационные сообщения.
- 5. Поле ввода, которое далее будет фигурировать в тексте как **Знаменитое поле ввода**. Оно служит для поиска текста, ввода разных значений. Кнопка **Найти** справа от поля ввода играет ту же роль, что нажатие клавиши Enter в этом поле, либо нажатие Ctrl-F. Например, чтобы найти в тексте какое-то слово, достаточно набрать его в Знаменитом поле ввода и нажать Enter или Ctrl-F. А чтобы перейти на строку номер такую-то, введите ее номер в это поле, и нажмите Alt-G.

Рядом с кнопкой **Найти** находится кнопка **Открыть**. Чтобы быстро открыть файл, вы можете ввести его имя (полный путь) в Знаменитом поле ввода и нажать эту кнопку. При этом используется кодировка, выбранная в **Настройки > Кодировки > Кодировка по умолчанию для открытия файла**.

- 6. Стока состояния. Здесь выводятся всякие полезные данные о текущем файле. Можно смотреть и номера текущей строки и символа, если поставить галочку в **Настройки > Интерфейс > Показывать положение курсора в строке состояния**. По умолчанию эта штук отключена, дабы не нагружать зря процессор и не раздражать глаза.

2.2. Перетаскивание

Если вы перетаскиваете в окно ТЕА текстовые файлы, то они открываются для

редактирования. Если же файл - картинка, то в текущий документ будет вставлен соответствующий элемент разметки (например, тэг IMG). Я проверял это с Konqueror и GQView.

Текстовые файлы при перетаскивании открываются в кодировке, указанной в **Настройки > Кодировки > Кодировка по умолчанию для открытия файла**.

2.3. Диалоговые окна Открыть/Сохранить файл

В TEA эти окна большие. Их размер можно задать в **Настройки > Интерфейс**. Там же есть опция **Особый режим "Enter" для окон открытия/сохранения**. Когда она включена, нажатие Enter в этих окнах, где бы ни был фокус ввода, срабатывает как нажатие на кнопку ОК. Еще есть список "**Особые папки TEA**" - с его помощью вы можете быстро попасть в папки, где TEA хранит ваши снippets, шаблоны, таблицы замен, файлы сессий и скрипты. Пользуйтесь на здоровье! Кроме того, очень вкусен березовый сок.

Глава 3. Как TEA ладит с разными кодировками?

По умолчанию вам доступны следующие кодировки - это кодировка текущей локали (для жителей Украины и России, в старых дистрибутивах Linux обычно это были KOI8-R или KOI8-U, а нынче та же UTF-8) и UTF-8. Если обе кодировки совпадают, т.е. кодировка локали равна UTF-8, то двух UTF-8 в списке, понятное дело, не будет. Будет одна. При открытии файла вы можете также использовать автоматическое определение кодировки. Для этого надо выбрать в списке доступных кодировок пункт *autodetect*. Но! Кроме этого, следует указать, какие именно кодировки подлежат автоматическому определению. Их надо выбрать в **Настройки > Кодировки > Определять автоматически**, поставив галочки напротив нужных кодировок.

Иногда функция автоматического определения кодировки не срабатывает. Вы можете задать кодировку, которая будет использоваться автоматическим определением в случае, если TEA не был способен определить кодировку сам. Для этого поставьте галочку на "При неудачном автоопр., исп. эту кодировку:", а ниже в списке **Кодировка:** выберите кодировку по вкусу. По сути, значения, выставленные по умолчанию, можно не трогать.

Чтобы добавить другие кодировки в список доступных для открытия/сохранения файлов, надо пойти в **Файл > Настройки > Кодировки**, где на вкладке **Настроить кодировки** добавить нужные вам кодировки. Для этого надо нажать на кнопку **Добавить** - появится диалоговое окно с выбором кодировок. Чтобы удалить кодировку из списка **Выбранные**, надо выбрать в нем кодировку и нажать кнопку **Удалить**.

После этого все списки кодировок, использующиеся в ТЕА, будут обновлены - это касается окон открытия и сохранения файлов, меню **Вид > Кодировки**, внутреннего списка кодировок в движке поиска текста в файлах. Кстати о меню **Кодировки** (подменю **Вид** - оно служит не для изменения текущей кодировки, а просто перечитывает файл с диска в выбранной кодировке. Так что если вам нужно перевести файл из одной кодировки в другую, используйте **Сохранить как**, а не **Кодировки**.

При перетаскивании файлов в окно ТЕА, а также при открытии оных из Знаменитого поля ввода, используется кодировка, выбранная в **Настройки > Кодировки > Кодировка по умолчанию для открытия файла**.

Глава 4. О редактировании и просмотре субтитров формата SRT

ТЕА предоставляет некоторые функции для удобного редактирования файлов с субтитрами формата SRT. В меню **Вид** есть пункт **Просмотр в Mplayer**, который нужен для быстрого просмотра редактируемых SRT-субтитров в Mplayer'e. ТЕА берет время из текущей секции (где стоит курсор), запускает Mplayer и перематывает фильм к нужной позиции. А как выбрать фильм, который будет запускаться? Очень просто - через **Файл > Открыть иначе > Открыть фильм**. То есть, общий способ работы с субтитрами таков. Выбираете фильм, потом открываете для редактирования файл субтитров, редактируете его, и периодически просматривается результат с помощью **Просмотр в Mplayer**.

Замечу, что для предварительного просмотра такой-то секции субтитров, лучше выбрать предыдущую по отношению к текущей секцию. Потому, что когда ТЕА запускает Mplayer и перематывает фильм к нужному месту (чье время он извлекает из текущей функции), Mplayer показывает субтитры, начиная только со следующей секции.

Глава 5. Трансцендентальное сияние меню Файл

Новый - создает новый документ в кодировке UTF-8.

Добавить... > Добавить в Закладки - помещает в список Закладок запись на текущий файл. Эта запись включает в себя - имя файла, кодировку и позицию курсора (чтобы ее потом восстановить и прокрутить текст к тому месту, где вы завершили с ним работу).

Добавить... > Добавить в список автосохранения - помещает текущий файл в список автосохранений. Файлы из этого списка сохраняются автоматически при их закрытии. Чтобы убрать файл из этого списка, надо редактировать конфиг автосохранения в меню **Файл > Файлы настроек > Список автосохранения**.

Фигня - в оригинале *Crapbook*. Это текстовый файл, который автоматически

сохраняется при его закрытии. Удобен для разных заметок, цитат и прочей фигни. По Alt-M, если Фигня уже открыта, происходит переключение на ее документ.

Новый Квас - создает новое окно с файловым навигатором под названием Квас. Подробнее читайте [Что такое Квас?](#).

Шаблоны - в этом подменю доступны шаблоны, сохраненные в `$HOME/.config/tea/templates`. Дабы создать новый файл на основе шаблона, достаточно выбрать один из последних.

А чтобы сделать шаблон, надо сохранить любой документ с помощью пункта меню **Сохранить иначе > Сохранить как шаблон**. Если вы желаете открыть шаблон для редактирования, то в окне открытия файлов переместитесь в папку шаблонов, выбрав ее в списке "**Особые папки ТЕА**".

Открыть - открыть файл[ы]. Там еще есть список, из которого можно выбрать кодировку открываемого файла - смотрите внимательно (звучит эдак по-китайски). Изначально доступны лишь две кодировки - UTF-8 и кодировка текущей локали. Подробнее о работе ТЕА с кодировками читайте [Как ТЕА работает с разными кодировками?](#)

Если вы открываете несколько файлов одновременно, то следите, чтобы их кодировки были одинаковы - т.е. были в той кодировке, которая выбрана в списке. Иначе ТЕА будет ругаться. Для отечественных пользователей лучше будет включить автоопределение кодировок. Сам им пользуюсь, нареканий почти никаких. Есть одно - если файл слишком маленький - несколько десятков байт, то в нем просто может не хватить слов для анализа кодировки.

В окнах открытия и сохранения ТЕА использует следующие, определенные в движке GTK+ 2, сочетания клавиш:

- Подняться на каталог вверх - Alt-Up
- Спуститься на каталог вниз - Alt-Down
- Перейти в домашний каталог - Alt-Home

Now playing: Yab Yum - Dead Moon Ritual (Source Mutation Mix)

О том, как ТЕА открывает RTF-файлы. Да, ТЕА умеет читать RTF-файлы. По крайней мере, текст разобрать можно. А кодировку по умолчанию для RTF-файлов можно настроить в **Настройки > Кодировки > Кодировка по умолчанию для RTF**.

ТЕА может открывать в режиме *только чтения* архивированные (.gz, .zip, .bz2, .tar.gz, .tar.zip, .tar.bz2) файлы, но при условии, что в архиве только один текстовый файл, а его содержимое - в кодировке UTF-8 или содержит текст только латинскими буквами. Кроме того, ТЕА открывает в режиме чтения файлы форматов OpenDocument, OpenOffice.org Writer, KWord, Microsoft Word и AbiWord. Правда, надо, чтобы у вас были установлены *gunzip*, *bzip2* и *tar*. А для чтения Microsoft Word (DOC) надо установить программу **antiword**, которая найдется в любом дистрибутиве Linux.

Сохранить - ну да, самое оно. Разумеется, сохранение производится в той

кодировке, в которой вы открыли файл.

Сохранить иначе > Сохранить версию - сохраняет содержимое текущего файла, но под именем, основанным на текущих дате/времени и имени исходного файла. Для параноиков и для меня.

Вернуться к сохраненному - загружает с диска содержимое текущего файла и вставляет его вместо редактируемого текста.

Сохранить иначе > Сделать резервную копию - делает копию текущего файла.

Файлы настроек > Файл слов для автозамены - таковая автозамена включается в **Настройках**, на странице **Редактор**. Примечание: в текущей версии TEA автозамена совсем отключена. Ну а список автоматически заменяемых слов редактируется здесь. Файл - простой, как валенок - состоит из строк формата: **слово=другое слово**. Например:

```
Linux=GNU/Linux
sxe=straight edge
FSOL=Future Sound Of London
hl=highlighting
н-ск=Новосибирск
```

Слова зависят от регистра. Изменения вступают в силу сразу после сохранения этого файла. Как же работает автозамена? Когда вы набираете текст, то после каждого нажатия пробела или ввода символа пунктуации (точки, запятые и т.д.) TEA смотрит, не было ли введено слово из списка автозамены. Ежели да, то это слово заменяется на то, что стоит напротив него в списке после знака равенства.

Файлы настроек > Файл закладок - открывает в редакторе файл закладок, чтобы вы могли удалить ненужные элементы. Выглядит коряво, зато функционально и удобно. Список закладок будет обновлен после сохранения файла закладок.

Сохранить как - кстати, в этом окне есть список, где можно выбрать кодировку, в которой будет сохранен файл.

Сессии - из этого меню можно открыть сессию. Сессия в TEA - аналог сессии в Opera, то есть список файлов. Выбирая сессию, вы открываете сразу все файлы из ее списка.

Чтобы сохранить сессию, надо воспользоваться пунктом меню **Сохранить иначе > Сохранить сессию**. Вы сохраняете список с открытыми сейчас файлами, и потом можете их загрузить одним кликом. В списке для каждого элемента сохраняются: имя файла, его кодировка и текущая позиция в тексте.

Открыть иначе > Открыть файл сессии - открывает файл сессии как обычный текстовый файл, то есть не загружая список. Нужно, если вы хотите удалить файлы из этого списка. А чтобы удалить саму сессию, т.е. ее файл, надо пойти в **Настройки > Обслуживание > Сессии**, там есть кнопочка "Удалить" для выбранной в списке сессии.

Файлы настроек > Главный конфиг - открыть файл настроек TEA (`$HOME/.config/tea/tea_rc`). Вы можете редактировать главный файл настроек TEA вручную (изменения вступят в силу сразу после его сохранения), однако проще использовать

окно **Настроек**.

Файлы настроек > Конфиг клавиш - настройка горячих клавиш. Старый способ привязки сочетаний клавиш к пунктам меню. Проще использовать для этого интерфейс в **Настройки > Клавиатура**. Из каких-то соображений ручной способ тоже остается в силе. В любом случае, клавиатурные привязки сохраняются в файле `$HOME/.config/tea/tea_hotkeys`, который должен содержать текст только в UTF-8. Изменение настроек вступает в силу после сохранения этого файла. Можно навешивать клавиши даже на те пункты меню, которые создаются динамически, например на закладки.

Каждая строка в `tea_hotkeys` имеет простой формат: *название пункта меню=сочетание клавиш*. Например:

```
Реверс=Alt Shift R  
Настройки=F12
```

Несколько замечаний. *Название является регистра-зависимым*. Вы должны писать его точно так же, как в меню ТЕА. А вот сочетание клавиш - от регистра не зависит. Есть три модификатора, *Alt*, *Shift* и *Ctrl*. При указании надо разделять их пробелами. Пример: *Настройки=shift ctrl F5*

Печатать - да, ТЕА умеет печатать текст, выводить его на принтер. Покамест это всё - ни предварительного просмотра, ни выбора шрифта и колонтитулов. Потом может появится.

Глава 6. Меню Правка

Отступ/Отступ обратно - сдвигает текст вправо или влево, вставляя/удаляя символы табуляции либо пробелы (в зависимости от того, что у вас включено в **Настройки > Редактор**). Причем всё это работает как с одной строкой, так и многострочным выделением.

Поменять - меняет местами содержимое буфера обмена и выделенного текста.

Копировать имя текущего файла - копирует в буфер обмена полное имя (т.е. с путем) текущего файла.

Вставить в новый - вставляет текст из буфера обмена в новый файл.

Копировать/вырезать в новый - оно самое. Отличие от "вставки в новый" - эти две функции идут в обход стандартного буфера обмена, то есть в системный клипборд ничего не попадает.

Добавить в Фигню - добавляет выделенный текст в Фигню, независимо от того, открыта она или нет.

Удалить - удаляет весь или выделенный текст.

Глава 7. Поиск

7.1. Обычные функции поиска и замены

Кроме совмещенного окна **Поиска и замены**, в ТЕА реализован более удобный и быстрый способ вести поиски нужного вам... догадайтесь чего? Правильно - текста! А то зачем вам искать, например, овощ брюкву? Брюква - она сама вас найдет.

Смысл таков. Набираете текст, который хотите (а вы ведь хотите?) найти, в... да, в знаменитом поле ввода. Вот в нем набираете. И жмете Enter. Или Ctrl-F. При этом важно, чтобы курсор был в тексте - активируйте его клавой или мышью. Дело в том, что когда вы открываете файл, то курсор иногда появляется, только когда вы редактируете текст или клацаете мышью. Так что сначала поставьте в тексте курсор, а потом предпринимайте поиски. Поиск ведется начиная с позиции курсора.

Заменить ну все - снова на помощь нам приходит Знаменитое поле ввода. На сей раз оно используется в специальном формате, а именно: *старый текст~новый текст*, например: *геморрой~трудности*.

Найти в файлах - предоставляет вам окно, из которого можно искать текст в файлах, указав каталог, откуда начинать поиск, и шаблон имени файла - например, **.txt* - все текстовые файлы. Поиск осуществляется по выбранной из списка кодировке. Список найденных файлов можно сохранить как сессию, а также копировать все файлы из этого списка в указанный вам каталог. Двойной щелчок по файлу в списке открывает этот файл в редакторе. Кнопка "Очистить список" очищает список найденных файлов.

Найти локальные ссылки - сканирует текущий документ на предмет обнаружения ссылок на локальные файлы. Все найденные имена преобразует в полные пути к ним и наполняет ими меню **Нав > Ссылки**, откуда можно:

1. Загрузить файл в ТЕА (если текстовый). Загрузка осуществляется в кодировке текущего документа.
2. Просмотреть файл внутренней или внешней смотрелкой (подробнее см. о функции **Открыть в месте курсора**), если файл - картинка.
3. Переключиться на таб с документом, если он уже открыт.

В **Настройки > Переключатели** можно включить автоматическое сканирование ссылок при открытии документа.

Найти SRC - ищет в текущем документе все локальные SRC (картинки, флэшки и так далее) и помещает их в меню **Нав > Ссылки**, откуда вы, понятное дело, можете их открывать. Это гораздо удобнее, чем вручную искать картинки в тексте.

7.2. Цветовые пометки

Отметить. В ТЕА существует особый тип выделения текста. Функция **Отметить** служит именно для него. Используя ее, вы можете выделить несколько фрагментов

текста в одном документе. Цвет такого выделения настраивается в **Настройки > Цвета > Передний план/фон отметок**.

Чтобы **отметить** текст, выделите его и примените функцию **Отметить**. Чтобы снять такую отметку, установите курсор на отмеченный текст и снова примените функцию **Отметить**. Выделять при этом фрагмент вручную не нужно - просто ткните в него курсором. А чтобы удалить все отметки, используйте пункт меню **Снять отметки со всего**.

Что можно делать с отмеченным текстом? Копировать. Это пункт меню **Копировать отмеченное**.

Найти и отметить - ищет текст из Знаменитого поля ввода в текущем документе. Если находит, то раскрашивает его цветом, который можно задать в **Настройки > Цвета > Передний план/фон отметок**. Затем, вы можете копировать отмеченный таким образом текст, используя пункт меню **Правка > Копировать отмеченное**.

Отметить строки длиной > N символов. Введите в Знаменитое поле ввода количество символов, затем примените эту функцию, и редактор отметит другим цветом в текущем документе все строки, чья длина превышает заданное количество символов.

Глава 8. Вёрстка

8.1. Обо всем этом

Вначале я расскажу, как работает меню **Вёрстка**. В нем есть подменю **Режим верстки**, с помощью которого вы можете переключать режимы, в котором работает меню **Вёрстка**. Режимы таковы: HTML, XHTML, Wikipedia, TeX и Docbook. Под TeX подразумевается в основном LaTeX.

Не все пункты меню поддерживают все режимы. Например, функция **Сделать таблицу** не умеет делать таблицу в режиме LaTeX.

8.2. Утилиты HTML

Проверить битые ссылки - проверить ссылки. Проверяются ссылки на локальные файлы, а также HTTP-ссылки. Все внешние ссылки кроме HTTP трактуются TEA как "битые".

Вес документа - выводит в Логмэмо вес документа, включая локальные SRC - то есть картинки и прочее. Заодно выводится и подробная инфа, какой файл сколько весит.

Встроенный шаблон - вставляет в документ HTML-шаблон, который встроен в TEA.

Заключить выделенную ссылку в тэги - довольно бесполезная функция.

Например, если вы хотите заключить `www.foo.bar` или `ftp.foo.bar`, либо `mailto:foo@bar.com` в элемент `a href`, то попробуйте выделить ссылку и применить

эту функцию.

Убить все тэги! - удаляет все тэги из HTML или Docbook-документа.

Конвертировать тэги в entities - если у вас есть некий HTML или XML код, и вам надо отобразить его в документе, то выделите и обработайте этой функцией. В результате получится нечто вроде этого:

demo

Перевести текст в HTML - чисто, с помощью CSS конвертирует обычный текст в HTML, разбивая текст на параграфы и правильно делая для них отступы.

8.3. Общие функции

Некоторые из описанных ниже функций имеют двойственную природу, то бишь если текст был выделен, то обрамляют его некоторыми тэгами, иначе же просто вставляют эти тэги в текущую позицию курсора. Для остальных функций надо сначала выделить текст, а уж потом их применять.

Комментарий - закомментировать выделенный текст. Работает для HTML, CSS, Pascal и C/C++.

Цвет - выбор цвета! Когда выделен некий текст, то он "раскрашивается" элементом *span* с соответствующим цветовым атрибутом в CSS'e, в противном случае просто вставляет HTML-цвет туда, где курсор.

Картинка - являет пред очи диалог выбора файла. Выбирайте. В текущий файл вставится тэг изображения с заполненными параметрами - названием, размерами и т.п. Поддержка форматов зависит от GTK+: PNG, JPEG, WBMP (старый формат для мобилок), GIF... Полагаю, что для вставки такого тэга лучше использовать Квас (там есть функция **Вставить картинку**), Imageplane или перетащить в документ файл-картинку из какого-нибудь файлового менеджера вроде Konqueror или смотрелки GQview.

Таблицы > Сделать таблицу - если вы ожидали увидеть мастера таблиц или мастера починки телевизоров, то его следует искать в другом месте. Я сам не пользуюсь никакими визардами, но всегда хотел иметь простое средство создания таблиц. И я его сделал. Надо лишь ввести размеры таблицы в ЗНАМЕНИТОЕ поле ввода, и применить эту функцию. Традиционно, формат будет такой: *количество рядов~количество столбцов*. Например, вы хотите таблицу из 2 рядов по 4 столбца в каждом ряду. Тогда вам нужно ввести: 2~4

8.4. Списки

Поэлементный список - работает в режимах TeX и Docbook. Вот вам пример поэлементного списка:

- Scorn
- Nirvana

- Guano Apes

Нумерованный список - работает в TeX (как enumerated список) и Docbook (как orderedlist) режимах. Пример такого списка:

1. Scorn
2. Nirvana
3. Guano Apes

Глава 9. Функции

9.1. Общие

Чтобы использовать функции обработки текста, текст надо перед этим выделить. Если текст не выделен, TEA будет обрабатывать весь текст документа, а в редких случаях (например, при изменении регистра букв) - только текущее слово.

Статистика документа - как журналист, я более чем часто смотрю статистику по тексту, ибо количество символов означает для меня деньги :) Теперь представьте, как меня задолбало каждый раз нажимать кнопку OK или Закрыть во всяческих окнах статистики... Поэтому в TEA статистика выводится прямо в log_memo. Экономит время и нервы.

Читабельность документа - я сомневаюсь в пользе от этой функции для отечественных пользователей. Для вычисление некоего индекса читабельности используются две формулы (Automated Readability Index (ARI) и Coleman-Liau), которые применяются в образовательной системе США. Есть ли толк в этих индексах для английских текстов и русских, да и вообще любых - я не знаю, мне было просто любопытно реализовать эту штуку в TEA.

9.2. Календарик

Маленький календарик для разных нужд. Например, вам может понадобиться вычислить время между двумя датами. Как быть? Выбираем в календаре первую дату и нажимаем кнопку **дата1**. Затем выбираем вторую дату и нажимаем кнопку **дата2**. Щелкаем по кнопке **Подсчитать** и получаем результат в логмемо главного окна.

Кроме того, из Календарика вы можете вставлять в текущий документ дату, в разных форматах - для этого есть соответствующие кнопки внизу окна календаря.

9.3. IDE

В TEA присутствуют базовые средства среды разработки. Это ведение проектов, запуск проекта на сборку командой make, и запуск исполняемого файла, полученного в результате сборки. Кроме того, при выявлении компилятором ошибок

они отображаются в логемо цветом, который можно задать в **Настройки - Цвета - IDE**. И далее при двойном щелчке на сообщении об ошибке ТЕА откроет нужный файл (если он еще не открыт) и переставит курсор на строку, в которой найдена ошибка. Меню IDE предлагает такие пункты:

Новый проект - создать новый проект. Появится окно с полями, которые надо заполнить. Это:

(1) Имя проекта - название проекта.

(2) Директория с makefile - каталог, где лежит makefile проекта.

(3) Директория с исходным кодом - каталог, в котором находится исходник вашего проекта. Исполняемый файл - файл, который будет получен в результате работы make, и который будет запущен командой **Запустить**. Вам нужно указать только имя файла, без пути к нему.

Свойства проекта - позволяет редактировать свойства текущего проекта. Не забывайте потом сохранить проект.

И еще одна полезная функция, относящаяся к IDE - это помещенный в меню **Вид** пункт **Переключить исходник/заголовок**. Работает только для C/C++ файлов и в том случае, если и исходник, и заголовок находятся в одной директории.

Обновить тэги с помощью ctags - используя программу **ctags**, создает в каталоге текущего файла файл *tags*. Далее, когда этот файл создан, вы можете использовать функцию **Нав - Найти определение функции**, чтобы быстро перейти к месту определения C/C++ функции.

9.4. Проверка орфографии

ТЕА для проверки орфографии использует Aspell. В меню "**Языки для проверки орфографии**" вы видите установленные словари. Выбор пункта меню означает проверку *всего документа* соответствующим словарем. Выбранный язык запоминается, и потом вы просто можете использовать функцию "**Проверить орфографию**" вместо того, чтобы выбирать из меню нужный язык.

Замечу, что ТЕА при этом не предлагает слова на замену. Он только отмечает их другим цветом, который задается в **Настройки > Цвета > Цвет ошибок для проверки орфографии**.

Когда вы исправите помеченное как ошибочное слово, то ТЕА не изменит его окраску автоматически. Собственно говоря, проверка орфографии в ТЕА представляет собой частный случай подсветки синтаксиса. Поэтому, если вы хотите обновить раскраску, то выберите снова тот же пункт меню.

А дабы убрать раскраску вообще, воспользуйтесь **Вид > Скрыть подсветку**

Чтобы получить список предположительно верных написаний слова, поставьте курсор на слово, подчеркнутое как ошибочное, и примените пункт меню "**Предположить верное написание**".

9.5. LaTeX

Как работать с TeX/LaTeX в TEA? Прежде всего надо настроить параметры в **Настройки > Команды > LaTeX**, где следует прописать команды запуска программ, относящихся к LaTeX. Примечание - не добавляйте символ апмерсанда (&) в конце этих команд. И еще - макрос %s обозначает в команде имя текущего файла. Как вы можете заметить, пункты меню **LaTeX** служат для вызова заданных в настройках команд. Разумеется, у вас должны быть установлены программы, которые упоминаются в командах. По идее, прописанные по умолчанию команды вполне рабочие, главное удостоверьтесь, что поставили из своего дистрибутива все нужные пакеты с LaTeX-утилитами.

Чтобы сделать dvi-файл из текущего TeX/LaTeX документа, используйте пункт меню **Обработать в LaTeX**. Если все в порядке, теперь вы можете перевести dvi в **ps** или **PDF**. Для этого воспользуйтесь пунктами меню **dvi в PS** или **dvi в PDF**.

Получившийся в результате файл будет лежать в том же каталоге, что и dvi-файл. Об имени файла позаботится TEA.

Чтобы просмотреть dvi-файл, созданный из текущего LaTeX-документа, используйте **Просмотр dvi**. Точно так же, для **PS** и **PDF**, которые созданы в TEA из LaTeX-файлов, есть пункты меню **Просмотр PS** и **Просмотр PDF**.

Итак, алгоритм создания в TEA документа PDF из LaTeX таков: **Обработать в LaTeX**, затем **dvi в PDF**, и сразу посмотреть результат можно с помощью **Просмотр PDF**.

Если же у вас установлен пакет **pdflatex**, то все еще проще: **Обработать в pdflatex**, и просмотр как обычно.

9.6. УНИТАЗ

УНИТАЗ расшифровывается как Универсальный текстовый анализатор. Надо сказать, что в Windows-версии TEA УНИТАЗ работает намного быстрее - увы, из-за юникодной природы линуксового TEA я не могу применить старый алгоритм (вернее, могу, но он получается слишком тормозным) сортировки, так что...

Как бы ни было, вы можете вызывать УНИТАЗ с сортировкой и без. При этом должен быть открыт некий документ. УНИТАЗ выдаст статистику по количеству каждого слова из документа, а еще скажет, сколько слов всего, и сколько уникальных, то есть количество слов БЕЗ их повторов.

Ну а функция **Извлечь слова** действует примерно так же, как и классический УНИТАЗ, только не присовокупляет к результату числовую статистику. То есть вы просто получаете список слов из текущего документа. Это полезно для составления каких-нибудь словарей и т.д.

9.7. Скрипты

9.7.1. Скрипты с точки зрения пользователя

В TEA, вы можете использовать скрипты для обработки текста точно так же, как и встроенные функции. То есть, грубо говоря - выделяете текст, вызываете из меню скрипт. Если скрипт принимает какой-нибудь управляющий параметр, то его, этот параметр, следует вписать в Знаменитое поле ввода.

TEA "понимает" скрипты, написанные на Python, Perl and Bash (Sh). Чтобы установить скрипт, просто скопируйте его в **\$HOME/.config/tea/scripts**. Если скрипт находится в архиве, то распакуйте его. TEA "увидит" установленные скрипты после перезапуска.

9.7.2. Скрипты с точки зрения разработчика

Как написать скрипт для TEA? Да очень просто. Во-первых, вам нужно знать, как TEA передает текст в скрипт, и как получает обработанный текст обратно, чтобы заменить им выделение или все содержимое документа.

TEA запускает каждый скрипт с одним или двумя параметрами. Первый параметр всегда указывает на имя файла, в котором содержится переданный из TEA текст. Текст этот - либо выделенный фрагмент, либо весь текст целиком (текущего документа). То есть, чтобы получить из TEA текст, вам нужно в коде скрипта прочитать содержимое файла, чье имя передано в первом параметре к скрипту.

Второй параметр (если существует) содержит имя файла, в который TEA записывает текст из Знаменитого поля ввода. Итак, чтобы получить содержимое Знаменитого поля ввода, вам нужно внутри скрипта прочитать файл, на имя которого указывает второй параметр скрипта.

При обработке текста обратите внимание на кодировку - TEA передает текст в скрипты в кодировке UTF-8 и предполагает, что в этой же кодировке текст будет возвращен ему обратно. А как возвращать? Тоже просто. Надо записать обработанный текст в тот же файл, откуда текст был прочитан. То есть в файл, чье имя в первом параметре к скрипту.

Ниже я приведу пример UTF8-безопасного скрипта на Python. Этот скрипт "переворачивает" регистр символов полученного текста и возвращает обработанный текст обратно в TEA.

```
import sys
import string
import codecs
f = codecs.open(sys.argv[1], "r", "utf-8" )
u = f.read()
f.close
u = u.swapcase()
f = codecs.open(sys.argv[1], "w+", "utf-8" )
f.write(u)
f.close
```

Итак, `sys.argv[1]` (первый параметр скрипта) содержит имя файла, который нам нужно прочитать, чтобы получить текст из ТЕА. Когда мы обработали текст, мы записываем его все в тот же файл, имя которого содержится в `sys.argv[1]`. Обратите внимание на использование кодеков (в Python). Это необходимо для правильной работы скрипта с UTF-8.

И другой пример - эдакий калькулятор. Он получает выделенный текст, а в тексте - какое-то математическое выражение. Скрипт вычисляет его и возвращает результат. Поскольку мы имеем дело только с числами, мы можем не заботиться о кодировке:

```
import sys
f = file (sys.argv[1], 'r')
s = f.read()
f.close
t = eval (s)
f = file (sys.argv[1], 'w+')
f.write (str (t))
f.close
```

А что, ежели нам нужно получить некий параметр, с помощью которого пользователь, введя этот параметр в Знаменитое поле ввода, хочет повлиять на работу скрипта? В Python, для получения этого параметра, надо прочитать файл, чье имя находится в `sys.argv[2]`. А в Bash-скрипте, используйте `$1` для получения имени файла с текстом документа, и `$2` - для получения имени файла с содержимым Знаменитого поля ввода.

Если вы собираетесь пожертвовать какие-то скрипты для коллекции скриптов на сайте ТЕА, то было бы здорово, если бы вы дали свой скрипт со статусом "общественное достояние" (public domain). Во всяком случае, не под собственнической (proprietary) лицензией. Еще в комментариях внутри скрипта укажите авторство и описание скрипта.

9.8. Таблицы замен

Таблица замен - это чудесное средство для замены в одном документе сразу множества слов, а не по одному. Сначала вы составляете таблицу - в виде обычного текстового файла. Примерно так:

```
кошка=собака
мяукает=лает
```

Сохраняем этот файл через меню **Файл > Сохранить иначе > Сохранить как таблицу замен**. Опа! Имя сохраненного вами файла появилось в меню **Функции > Таблицы замен**. Теперь в другом документе пишем текст: "кошка мяукает", выделяем этот текст и выбираем пункт меню с нашей таблицей, то есть применяем таблицу к выделенному тексту. Что получилось? В тексте произошли множественные замены и теперь он гласит: "собака лает". Замечательно!

Если же вы взалчите отредактировать таблицу, то в окне открытия файла есть меню, или список - **Особые папки ТЕА**. Там есть пункт "**Таблица**". Выберите ее и в списке файлов будут отображены файлы таблиц. И открывайте нужный для

редактирования.

9.9. Сниппеты

Сниппет - это фрагмент текста, который можно вставить в текущий документ.

Каждый сниппет хранится в отдельном файле в директории

`$HOME/.config/tea/snippets`. Имя файла сниппета является пунктом меню в

Сниппетах. То есть меню **Сниппеты** представляет собой список файлов директории со сниппетами. Чтобы создать новый сниппет, вам надо:

1. Написать какой-то текст.
2. **Файл > Сохранить иначе > Сохранить как сниппет.** (Только в UTF-8!!!)
3. Всё!

Сниппетом вы можете оформить выделенный текст. Выделение представляется в сниппете макросом `%s`, который при применении сниппета будет заменен на выделенный текст. Пример сниппета, который оформит выделенный текст как href-ссылку:

```
<a href="%s">%s</a>
```

Надо сказать, что если текст не выделен, то сниппет просто вставится в текст.

9.10. Инструменты

Инструмент массовой перекодировки - служит для массового перекодирования файлов. Вы задаете образец поиска файлов к указанном вами же каталоге, выбираете исходную кодировку (список "Из:") и кодировку, в которую собираетесь перевести содержимое файлов (список "В:"), а затем нажимаете кнопку "**Найти файлы согласно шаблону и перекодировать их**". Попутно можно включить конвертирование окончания строки (LF или CR/LF).

Сделать снимок экрана/окна/области/из буфера обмена - все эти пункты меню предназначены для создания скриншотов. Изображения сохраняются в каталоге текущего документа, а значит, вам надо его предварительно открыть, а уж потом делать снимки. Имя файла, куда будет сохранена картинка, следует записать в Знаменитое поле ввода. Например: `test.jpg`. Полный путь вводить не надо.

По расширению файла ТЕА выбирает формат. Поддерживаются такие расширения: tif, tiff, jpg, jpeg, png (файлы при этом сохраняются в полноцветном PNG, а не в том, что с палитрой до 256 цветов), bmp. Для JPEG можно задать степень сжатия - делается это в окне Настроек, на странице **Функции** - там есть опция **Степень сжатия JPEG**. Кроме того, там же можно указать задержку (в секундах) между выбором функции снимка и его осуществления. В это время главное окно ТЕА временно спрячется и вы сможете что-то сделать с окнами других программ или с рабочим столом.

9.11. Формат

Убить форматирование - убивает форматирование в выделенном фрагменте.

Убить форматирование в каждой строке - убивает форматирование в каждой строке выделенного фрагмента, по возможности соблюдая разделение оного на строки.

Перенос строк в позиции - жесткий перенос строк/слов в позиции, которую вы должны задать в знаменитом поле ввода. Работает примерно как команда fold без параметра -s.

Перенос строк на словах в позиции - то же, однако как команда fold уже с параметром -s, то есть переносит только на пробелах, таким образом соблюдая деление на слова.

9.12. Insert

Строка *Сейчас играет* из Amarok - вставляет в документ название текущей песни из плейера Amarok. Формат вставляемой строки задается в **Настройки > Функции > Формат строки *сейчас играет***.

Дата/Время - вставляет текущие дату и время по заданному вами формату. Формат задается в **Настройки > Функции > Формат даты и времени** строкой особого формата. По умолчанию это `%d/%m/%Y %T`, что в итоге дает дату и время такого вида: *08/02/2004 12:50:02*. Полное описание синтаксиса форматасмотрите в английской версии документации, здесь же я приведу только немного оттуда:

- `%a` - день недели сокращенно, в зависимости от текущей локали.
- `%A` - день недели полностью, в зависимости от текущей локали.
- `%b` - месяц сокращенно, в зависимости от текущей локали.
- `%B` - месяц полностью, в зависимости от текущей локали.
- `%c` - предпочтительное отображение календарного времени для текущей локали.
- `%C` - век года.
- `%d` - день месяца, десятичный.
- `%D` - дата формата `%m/%d/%y`.
- `%F` - дата формата `%Y-%m-%d`.
- `%H` - час, от 00 до 23.
- `%I` - час, от 01 до 12.
- `%j` - день года, от 001 до 366.
- `%m` - номер месяца (от 01 до 12).
- `%M` - минута (от 00 до 59).
- `%n` - новая строка.
- `%p` - AM или PM.
- `%P` - аналогично, но в зависимости от локали.

- %r - полное календарное время в формате AM/PM для текущей локали.
- %R - часы и минуты в десятичном формате %H:%M.
- %s - сколько секунд прошло с 1970-01-01 00:00:00 UTC. //високосные секунды выпадают?
- %S - секунды, от 00 до 60.
- %t - таб.
- %T - время дня в формате %H:%M:%S.
- %u - День недели (от 1 до 7), понедельник = 1.
- %U - номер недели от 00 до 53), начиная с первого воскресенья первого дня первой недели (00).
- %V - аналогично, однако с 01 по 53, и с первого понедельника.
- %w - день недели, от 0 до 6, воскресенье = 0.
- %x - предпочтительный формат даты для текущей локали.
- %X - предпочтительное для текущей локали время дня.
- %y - год без тысячелетия, от 00 до 99. То есть не 2004, а 04.
- %Y - год по григорианскому календарю.

9.13. Регистр

БОЛЬШИЕ БУКВЫ/маленькие буковки (Ctrl-вверх/вниз) - изменяет регистр выделенного текста или текущего слова.

9.14. Числа

Вычислить - вычисляет математическое выражение, выделенное в тексте, и выводит результат в Логмемо. Внутри выражения допускаются скобки, а также операторы +, -, *, ^, /. Вот несколько примеров понимаемых ТЕА математических выражений:

(2/3)*123
5+5+5
1-200*89

Арабское в римское - допустим, выделяете в тексте 1977, применяете эту функцию, получаете MCMLXXVII.

Счетчик - генерирует список из целых чисел. Формат таков:
начальное_значение~конечное~[шаг]. (Кстати, по умолчанию параметр шага = 1).
Пример, где первое число, единица - начальное значение счетчика, второе (десять) - конечное, и параметр шаг равен пяти:

1~10~5

И вот результат (звучит, как в рекламе):

5
10
15
20
25
30
35
40
45
50

А можно и так: `1~10`, то есть опускаем параметр шага, и он приравнивается к единице.

9.15. Страна

Разбить на TeX-параграфы - разбивает текст на параграфы формата TeX, то есть вставляя между каждым параграфом пустую строку. Если вы переводите обычный текст в TeX, то эту функцию целесообразно применить прежде всего.

Конвертировать обычные кавычки в TeX - работает только для двойных кавычек. Конвертирует их к TeX'овые парные.

Пронумеровать строки - нумерует выделенные строки в соответствии с заданным форматом. Формат очень дурной: `printf`-подобный формат~шаг счетчика~начальное значение счетчика. Два последних параметра опциональны, и по умолчанию их значения равны единице.

В первом параметре для представления строки и счетчика используются макросы `%s` и `%d` соответственно. То есть при вызове функции нумерации, строка формата будет применена к каждой строке выделенного текста. И там, где в строке формата будет `%s`, этот макрос заменится на текст из обрабатываемой строке. А там, где `%d` - на числовое значение счетчика. Вот несколько примеров:

```
%d.)%s  
%d.)%s~10  
%d.)%s~10~4
```

Во второй строке мы задаем еще и шаг, он равен 10. В третьей строке мы задаем и шаг, и начальное значение счетчика (4).

Вы также можете использовать `%d` после `%s`, т.е. `%s (%d)`. Например, при обработке "чистого", без нумерации, списка групп форматом `%s (%d)` мы получим такую нумерацию:

```
Nirvana (1)  
Scorn (2)  
Napalm Death (3)  
Defecation (4)  
Neck (5)  
JR Ewing (6)
```

Fall (7)
Meathook Seed (8)
Guano Apes (9)
Led Zeppelin (10)

Конвертировать табы в пробелы - и так ясно. Но, сначала нужно ввести в знаменитое поле ввода число - количество пробелов на один таб. То бишь каждый таб будет переведен в указанное вами количество пробелов.

Конвертировать пробелы в табы - обратная функция. На сей раз в знаменитом поле ввода должно быть число, характеризующее количество пробелов, каждое нахождение которых функция будет переводить в символ табуляции.

Реверс - переворачивает текст. Было *roxton*, будет *notxor*.

Кодировать e-mail адрес супротив спама -- делает выделенный в тексте адрес e-mail невидимым для спаммерских сканнеров, конвертируя текст в числовые коды, которые, тем не менее, воспринимаются браузером как текст. Например, если вы посмотрите в исходник этого документа, то [эта ссылка](#) будет выглядеть как куча мусора. Я взял идею из одного номера [LinuxGazette](#).

Удалить пустые строки - не думаю, что это такая уж полезная функция, но пусть будет.

Удалить дубликаты - удаляет повторяющиеся строки.

Применить шаблон к каждой строке - и снова работаем со знаменитым полем ввода :) Например, я хочу добавить тэг br в конец каждой строки выделенного текста. Я пишу в знаменитом поле:

```
%s<br>
```

И применяю функцию (конечно же, я не забыл выделить текст). В итоге я получаю требуемый br в конце каждой строки. Другой пример - я хочу заключить каждую строку в тэг HREF, чтобы строка была как в SRC, так и в надписи. Ввожу в знаменитое поле ввода:

```
<a href="%s">%s</a>
```

Получаю искомое :) Как вы поняли, макрос %s при работе функции подменяется текстом каждой обрабатываемой строки.

9.16. Сортировка

Сортировать строки - в алфавитном порядке.

Обратный порядок строк - лучше поясню на примере. Было:

Line 1
Line 2
Line 3

Будет:

Line 3

Line 2

Line 1

9.17. Фильтр

Извлечь в каждой строке - оставляет в каждой строке подстроку между двумя символами-разделителями. Разделители эти надо прописать в Знаменитом поле ввода, в формете *первый разделитель~второй разделитель*, то бишь между разделителями должна стоять тильда.

Вот вам пример. У нас есть две строки, в которых мы хотим оставить только текст, заключенный в кавычки:

```
g_print ("hello\n");
g_print ("world\n");
```

Пишем в Знаменитом поле ввода шаблон обработки: "~", а затем выделяем текст и применяем функцию. В результате получается следующее:

```
hello\n
world\n
```

Извлечь перед разделителем в каждой строке, Извлечь после разделителя в каждой строке. Функции, сходные с предшествующей, однако в Знаменитое поле ввода надо прописать только один разделитель. Например, если вы хотите убрать в каждой строке комментарии, что в Знаменитое поле ввода надо ввести "//" (без кавычек) и применить функцию **Извлечь перед разделителем в каждой строке**.

Убить строки, содержащие фразу. Выделяете текст. В знаменитом поле ввода набираете некую фразу. Применяете эту функцию. Все строки, содержащие эту фразу, будут удалены (в выделенном вами фрагменте, разумеется).

Убить все строки без фразы. А вот тут наоборот - убиваются те строки, что НЕ содержат заданной фразы.

Убить строки, содержащие регулярное выражение и Убить строки без регулярного выражения. То же, что и две функции выше, только образец искомого текста задается в виде регулярного выражения. Что это даст? Например, чтобы удалить все строки, которые начинаются с "Иван" (т.е. это может быть и Иванова, и Иванов, и Иван-дурак), надо в Знаменитое поле ввода ввести "Иван*" и использовать фильтр **Убить строки, содержащие регулярное выражение**.

Символ "звездочка" означает, что после этого символа в строке могут быть какие угодно символы, лишь бы "Иван" было. Либо, бы хотите отфильтровать строки со словом, которое начинается на "ов" и заканчивается на "ка". То бишь это и овечка, и овчарка, и овсянка. Стока-фильтр будет такова: "ов*ка".

Убить строки длиной < N символов - убивает в выделенном тексте строки, длина которых меньше N символов. N - число, веденное в Знаменитое поле ввода.

Убить строки длиной > N символов - убивает в выделенном тексте строки, длина которых больше N символов. N - число, веденное в Знаменитое поле ввода.

9.18. Код Морзе

Кодировать в морзянку (англ.). Переводит в азбуку Морзе, на английском. Было:

Drum and bass

Будет:

-.. -.- ..- -.- .-. -.. -... - -

TEA помещает одиночный пробел между двумя соседними кодами Морзе, и два пробела ставит между словами.

Кодировать из морзянки (англ.). Переводит из морзянки на английский. При этом предполагается, что коды разделены одинарными пробелами, а слова - двойными.

Глава 10. Запуск

Это очень хорошее меню. Отсюда вы можете запустить текущий документ в какой-нибудь внешней программе, например браузере. Элементы меню настраиваются в файле конфигурации, который можно открыть пунктом меню **Файл > Файлы настроек > Список внешних программ для запуска**. Файл этот состоит из строк, каждая такого формата: *название элемента меню=команда*. Например:

opera=opera %s &

При выполнении команды, макрос %s будет заменен на имя текущего файла.

Амперсанд в конце строки команды нужен, чтобы после запуска внешней программы управление передавалось TEA, то есть чтобы TEA не ждала завершения внешней программы.

Глава 11. Нав[игация]

Сохранить позицию - сохранить позицию в тексте, чтобы потом на нее можно было перейти с помощью соседнего пункта меню - **Перейти на сохраненную позицию**. Вот так сидишь, лапоть нюхаешь, а в TEA между тем такие чудесные штуки есть.

Просмотр в Mplayer - эта функция нужна для быстрого просмотра редактируемых SRT-субтитров в Mplayer'e. TEA берет время из текущей секции (где стоит курсор), запускает Mplayer и перематывает фильм к нужной позиции. А как выбрать фильм, который будет запускаться? Очень просто - через **Файл > Открыть иначе > Открыть фильм**.

Перейти к предшествующему табу - переключиться к последнему табу перед тем табом, на который вы переключились.

Перейти к началу/концу блока - удобная функция для перемещения по коду. Для Си-подобных языков или PHP. Начало блока - {, конец - }. То бишь по Alt-влево/вправо вы можете прыгать к началу или концу текущего блока.

Открыть в месте курсора - допустим, есть у вас в HTML-документе некая ссылка на

локальный файл. Так вот, вы можете открыть его следующим образом. Поставьте курсора на имя файла в этой ссылке. Примените сию функцию. Ежели файл - картинка, то он откроется во встроенной смотрелке. А ежели вы хотите использовать внешнюю, то в **Настройки > Переключатели** включите опцию **Использовать внешнюю смотрелку**, а затем пропишите команду ее запуска в **Настройки > Команды > Внешняя смотрелка изображений**. По умолчанию там уже прописан запуск отличной утилиты из состава пакета ImageMagick, который точно найдется у вас в дистрибутиве.

Если же файл - не изображение, то он откроется в ТЕА как обычный текстовый файл, причем в той кодировке, какова у текущего файла (в коем сделана ссылка). *А если ссылка не файл, а ссылка на локальную для документа метку, то редактор прыгнет в место, где эта метка определена.*

И еще один режим работы функции - функция может принять на обработку выделенное в тексте имя файла. Это удобно, если имя файла не обрамлено кавычками.

Править в месте курсора - то же, что и см. выше, но исключительно для картинок, и работает со внешним редактором. Редактор задается тоже в **Настройках** в разделе **Команды**, и больше ничего включать не надо.

Найти определение функции - установите курсор на имени функции и используйте этот пункт меню, чтобы ТЕА перенес вас в файл и место в нем, где эта функция определена. Для поддержки этой возможности надо, чтобы у вас была установлена программа **ctags**. Эта программа вызывается автоматически при **Найти определение функции**, чтобы создать файл *tags*, который ТЕА использует, чтобы найти определение функции. Файл *tags* создается в каталоге текущего файла. Если файл *tags* уже создан, заново он не создается. Его можно обновить из **Функции - IDE - Обновить тэги с помощью ctags**.

Глава 12. Вид

12.1. Функции меню Вид

Профили - тут доступен список сохраненных вами профилей. Чтобы сохранить профиль под таким-то именем, воспользуйтесь пунктом меню **Сохранить профиль**. В профиле сохраняются размеры и положение главного окна. При загрузке профиля эти размеры и положение восстанавливаются.

Режим подсветки. В этом замечательном меню вы можете выбрать режим подсветки синтаксиса конкретно для текущего документа. Этот пункт меню доступен в тех версиях ТЕА, которые собраны без использования GtkSourceView.

Показывать картинки в тексте - показывать картинки прямо в тексте рядом с их IMG/SRC-тэгами.

Обновить подсветку. Это чтобы вручную обновить подсветку синтаксиса. Нужно в ТЕА, собранном без использования GtkSourceView.

Скрыть подсветку - убирает раскраску текущего документа. Полезно после проверки орфографии.

Переключить видимость картинок - режим, в котором картинки показываются прямо в тексте рядом с их IMG-тэгами. Может помочь, если вы запутались, где какая картинка, и надо всё быстро вспомнить.

Imageplane - вызывает к жизни вспомогательное окно Imageplane. Оно отображает все картинки, которые находятся в каталоге текущего текстового файла. В окне есть кнопочка **Вставлять тэги**. Если она включена, то двойной щелчок по картинке вставит в текущий документ тэг изображения для этой картинки. А если галочка на опции не стоит, то картинка будет просто показана во встроенной смотрелке картинок. Всяк да зрит!

Переключить исходник/заголовок - эта функция работает для файлов на С и С++.

12.2. Подменю Ко

Здесь вы можете переключить кодировку текущего документа. Кодировки доступны те же, что и при открытии/сохранении файла (см. где-то в начале этой доки, как добавлять новые кодировки). При переключении кодировки из этого меню **содержимое текущего файла перечитывается с диска заново**, только в другой кодировке. Поэтому использовать меню Ко следует только в том случае, если вы видите вместо текста какую-то чушь. А если вы просто СОХРАНИТЬ документ в другой кодировке, то для этого надо выбрать эту иную кодировку в диалоге **Сохранить как**.

Глава 13. Настройки

13.1. Вступление

Начиная с версии 3.0 ТЕА наконец заимел окно Настроек, поэтому возня с основным конфигом ТЕА отменяется. Ура, товарищи. Постепенно остальные, вспомогательные конфиги тоже будут переведены под графическую оболочку.

В окне **Настроек** есть две очень важные кнопки, большие такие. Кнопка **Закрыть** просто закрывает это окно. А кнопка **Сохранить и применить** делает то, что на ней написано, а именно сохраняет все опции в главном конфиге ТЕА, а затем перечитывает его, тем самым применяя.

Я бегло пройдусь по секциям **Настроек**. Опции разбросаны по секциям в порядке, который оставляет желать лучшего, так что если вы в новой версии ТЕА не находите часть опций на своем месте, то они явно были перенесены в какой-то другой раздел этого окна.

13.2. Переключатели

В этом разделе помещены всякие переключатели, которые обычно рассеяны по разным вкладкам. TEA же группирует их, что мне представляется более удобным.

Использовать снippetsы - включает или выключает механизм снippetов. Зачем его выключать? В зависимости от быстроты вашего компьютера и количества снippetов их обновление может тормозить запуск TEA. Поэтому, если вам потребуется [временно] отключить снippetsы, это можно сделать.

Подсветка синтаксиса при открытии файла - включать или нет подсветку при открытии файла.

Сканировать ссылки при открытии файла - если включено, ссылки будут просканированы и помещены в **Нав > Ссылки**.

13.3. Функции

Формат функции Цвет - тут можно переопределить поведение функции **Вёрстка > Цвет**. Макро `@color` служит, чтобы задать, где будет размещен сгенерированный цвет, а макро `@text` - куда напишется выделенный текст в строке-результате (который заменит выделенный текст).

13.4. Интерфейс

Показывать полный путь в заголовке окна - можете выключить, если делаете скриншоты и вам неохота, дабы зрители этих скриншотов видели, где лежат ваши файлы.

13.5. Пределы

В этом разделе задаются всякие предельные значения - наибольшее количество элементов в истории последних файлов и т.д.

13.6. Редактор

Режим верстки по умолчанию - режим по умолчанию, в котором работают пункты меню **Верстка**.

Размер таба в пробелах - задает, сколько визуально пробелов будет в 1 символе табуляции.

Вставлять пробелы по клавише TAB - если включено, то TEA вставляет при нажатии на TAB столько пробелом, сколь указано в опции выше.

Включить автосохранение - именно то, что здесь написано. При этом можно и нужно задать интервал сохранения. Автоматически сохраняются все документы, открытые в момент срабатывания функции.

Автозавершение тэгов - включите эту опцию, если хотите, чтобы тэги, которые вы набираете, закрывались автоматически при нажатии на пробел после открывающего тэга. То есть, вводите открывающий тэг, жмете пробел, и редактор сам добавляет закрывающий тэг.

Автозамена - включает обработку редактором списка автозамены (смотрите Файл > Файлы конфигурации > Файл слов для автозамены). А там - формат простой. Каждая строка имеет вид *слово которое было=слово на которое надо заменить старое слово*.

13.7. Пути

Тут вы можете переопределять разные пути по умолчанию. Только не забывайте еще включать их использование, если нужно. И отключать, если не нужно. "Умолчательные" пути для сохранения и открытия файлов - это значит, в какой директории открываются окна выбора, куда сохранять или откуда открывать.

Глава 14. Квас

Для Кваса у меня зреют большие планы, но покамест он ничего более, чем замена стандартному окну открытия файлов. Клавиша **Backspace** в Квасе - переход на директорию выше. В адресной строке можно ввести путь, нажать **Enter** и Квас перейдет в указанный каталог. Квас может открывать текстовые файлы и показывать во встроенной смотрелке графические. Текстовые файлы открываются в кодировке, выбранной в списке кодировок (слева, под меню).

Внизу рядом с адресной строкой есть две кнопки. Если ввести в строку адреса некий путь и нажать "Перейти", то Квас отправит вас в указанный каталог. Вторая кнопка, **Imageplane**, открывает окно Imageplane с картинками из каталога, который вы просматриваете в Квасе.

У Кваса есть своё меню, к описанию которого я сейчас приступлю.

14.1. Действия

Обновить Квас - перечитать список файлов в текущем Квасе. Квас не умеет делать этого автоматически.

Справка о файле - выдает в лог-мемо информацию о текущем (выделенном) файле. При этом незримо используется стандартная утилита **file**.

Запустить с... - запустить текущий (в смысле выделенный) файл некой внешней программой, которую надо указать в адресной строке. При этом надлежит указать просто имя программы, то есть, например, *opera* или *kwrite*.

14.2. Вставка в редактор

Вставить картинку - вставляет в текущий редактируемый документ IMG-тэг на

файл изображения, который является текущим в Квасе.

Глава 15. Опции командной строки

--crapbook - запускает TEA с открытой Фигней.

--charset=charset_name - задать кодировку для файла, чье имя следует за этим параметром. Например, мы хотим открыть file1 в кодировке CP1251, а file2 в кодировке UTF-8:

```
tea --charset=cp1251 file1 --charset=utf-8 file2
```

Замечу, что действие этого параметра распространяется именно на файл, указанный следом за параметром. Затем значение внутренней переменной сбрасывается в "умолчальное" значение кодировки для открытия файла (которая задается в **Настройки > Кодировки**).

Глава 16. Об авторе

Петр Владимирович Семилетов, родился в Киеве в 1977 году. ИТ-журналист, писатель и композитор. Вегетаринец, атеист. Барабанщик и иногда вокалист в группе **Какая-то**. Гитарист и один из вокалистов группы **Костный мозг**. Подробнее смотрите мой [сайт](http://www.roxton.kiev.ua/) (<http://www.roxton.kiev.ua/>).

Приложение А. Ресурсы

[сайт TEA](http://tea-editor.sourceforge.net/) (<http://tea-editor.sourceforge.net/>)

[Сообщество TEA в Live Journal.](http://community.livejournal.com/tea4linux/) (<http://community.livejournal.com/tea4linux/>)

Рассылка новостей TEA - чтобы подписаться на нее, просто пошлите пустое письмо по адресу tea-announce-subscribe@linux.kiev.ua

[Домашняя страница Петра Семилетова, разработчика TEA.](http://www.roxton.kiev.ua/)
(<http://www.roxton.kiev.ua/>)