

Emacs и средства интерактивного общения

Alex Ott

14 сентября 2004 г.

1 Введение

Кроме поддержки "неинтерактивных" средств обмена сообщениями, таких как почта и новости Usenet, Emacs имеет поддержку практически всех видов интерактивного обмена информацией:

1. Instant Messaging. Сюда входит поддержка протоколов ICQ, Yahoo Messenger, AIM (AOL Instant Messaging), Jabber и MSN;
2. Internet Relay Chat (IRC).

Для каждого из перечисленных сервисов существует как минимум одна реализация, но часто существует две и более реализации, отличающиеся полнотой реализации и степенью поддержки. Некоторые из реализаций уже не развиваются, но часто просто работают. В данной статье я не буду описывать реализации которые уже не поддерживаются, и для которых существуют альтернативы.

2 Instant Messaging и Emacs

2.1 ICQ

ICQ — средство для быстрого обмена сообщениями, особенно популярное среди пользователей Windows, также может использоваться и из Emacs. В настоящее время существует одна реализация ICQ — `icq`¹, но она может использоваться только с XEmacs. Данная реализация поддерживает следующие возможности:

- Поддержку протокола ICQ версии 8 (версия из CVS);
- Регистрацию новых пользователей и смену пароля;
- Передавать сообщения и URL;
- Авторизовывать пользователей, обновлять мета-информацию и получать список контактов;

¹<http://icq.sf.net>

- Преобразовывать список контактов `misc` и `licq` в собственный формат;
- Изменять привязки клавиш, и писать функции на Emacs Lisp.

К сожалению данная реализация не поддерживает чаты, передачу файлов.

В связи с тем, что я не пользуюсь XEmacs, я не смог протестировать данную реализацию, но судя по документации, она достаточно просто устанавливается и настраивается.

2.2 Yahoo! Chat Messenger

Существует две реализации клиента Yahoo! Chat для Emacs. Первая из них называется ElGyach и доступна с <http://www.nongnu.org/elgyach>, а вторая называется yod.el и доступна с <http://users.starpower.net/marti1/yod.el.html>, однако ElGyach имеет больше возможностей по сравнению с yod.el, поэтому я остановлюсь на его описании более подробно.

2.2.1 Возможности ElGyach

ElGyach поддерживает следующие возможности:

- Сохранение буферов с сообщениями в файлах;
- Проверка грамматики;
- Подсветка сообщений;
- Поддержка шифрования с помощью AES;
- Механизм дополнений имен, команд и текста;
- Настройка цветов и атрибутов текста;
- Одновременное использование разных учетных записей;
- История сообщений, по которой можно перемещаться вперед и назад;
- Игнорирование сообщений от пользователей;
- Интеграция с другими модулями Emacs.

Более подробную информацию вы можете найти на страницах, относящихся к разработке ElGyach.

2.2.2 Установка ElGyach

Для установки ElGyach необходимо скачать его с <http://savannah.nongnu.org/projects/elgyach>, развернуть и скомпилировать (поскольку для общения с сервером Yahoo, ElGyach использует реализацию на языке C). Вот так будет выглядеть загрузка ElGyach в вашем файле инициализации:

```
(add-to-list 'load-path "path_to_elgyach")
(autoload "gyach" "gyach" "Autoload for ElGyach")
```

После загрузки вы можете задать необходимые параметры с помощью команд настройки Emacs — M-x customize-group gyach. Здесь вы можете задать имя и пароль пользователя Yahoo! Chat, а также имя программы **elgyach**, если она стоит в нестандартном месте.

2.2.3 Использование ElGyach

Для запуска ElGyach используется команда M-x gyach, которая создаст буфер процесса. Вы также можете использовать несколько процессов ElGyach одновременно. Это произойдет, если вы перед выполнением команды зададите префикс, вот так C-u M-x gyach. Однако вы не можете войти больше чем в один чат, используя одно и то же имя — это ограничение Yahoo! Chat.

2.2.4 Команды ElGyach

ElGyach поддерживает выполнение различных команд, задаваемых в форме /**имя**. В настоящее время доступны следующие команды:

- /**ignore User** и /**unignore User** — для игнорирования сообщений от определенных пользователей, и для отмены этого ограничения;
- /**highlight User** и /**unhighlight User** — для подсветки сообщений от определенного пользователя, и окончания использования данной настройки;
- /**version** — печатает номер версии ElGyach.

2.2.5 Настройка ElGyach

Основным параметром настройки являются имя и пароль пользователя, которые используются для подключения к Yahoo! Chat. Эти значения хранятся в переменных `gyach-yahoo-username` и `gyach-yahoo-password`.

Для обеспечения настройки параметров буфера ElGyach может использоваться код, который будет выполняться с помощью хука `gyach-mode-hook`.

ElGyach легко расширяется. При каждом вызове команды вида /**КОМАНДА**, будет вызываться функция `gyach-custom-КОМАНДА`. Таким образом, вы можете добавить свою функциональность с помощью небольшого объема кода.

2.3 AOL Instant Messaging (AIM)

Поддержка AIM в Emacs осуществляется пакетом TNT. Он позволяет обмениваться сообщениями, и участвовать в групповых чатах.

2.3.1 Установка пакета

Получить данный пакет можно с его домашней страницы, расположенной по адресу <http://tnt.sourceforge.net/>. Установка пакета достаточно проста — надо раскрыть архив, и настроить Emacs для его использования. Это выполняется с помощью следующих команд:

```
(add-to-list 'load-path "path_to_tnt_dir")
(require 'tnt)
```

Вот и все.

2.3.2 Работа с пакетом

Для работы с AIM вам нужно иметь уже зарегистрированное имя. Если у вас его нет, то вы можете зарегистрировать новое имя по адресу <http://www.aol.com/aim>.

Основная работа с пакетом осуществляется с помощью сочетаний клавиш, что позволяет удобно работать с пакетом. Все команды можно разделить на несколько частей, которые представлены в виде соответствующих разделов.

Подключение, отключение и статус Подключение к сервису AIM выполняется с помощью команды `tnt-open` (**C-x t o**). При подключении вам необходимо указать ваше имя и пароль, и если все пройдет нормально, то пакет сообщит о том, что вы подключились к системе. Чтобы каждый раз не вводить имя и пароль, вы можете задать их с помощью переменных `tnt-default-username` и `tnt-default-password`. Вы также можете использовать несколько разных имен для входа в систему. Более подробную информацию вы можете найти в разделе `XrefId[??]`.

После подключения, вы можете выполнять стандартные задачи — ведение переговоров, общение в чатах и т.п.

Изменение статуса с "доступен" в "занят" производится с помощью команды `tnt-away-toggle` (или **C-x t A**). Вы можете задать сообщение, которое будет отображаться для пользователей, которые будут пытаться соединиться с вами, когда вы находитесь в данном режиме.

Прекращение работы с AIM выполняется с помощью команды `tnt-kill` (**C-x t q**).

Обмен информацией TNT поддерживает два типа обмена информацией — прямые беседы между двумя пользователями, и общение к чатах.

Для прямого общения используется команда `tnt-im` (или сочетание клавиш **C-x t i**). При этом у вас запросят имя пользователя с которым вы хотите пообщаться, и после установления связи, будет открыт буфер, в котором вы сможете печатать свои сообщения и видеть ответы. Если пользователь есть в вашем списке пользователей, то вы можете использовать механизм дополнения для ввода имени.

Для начала общения в чатах используется команда `tnt-join-chat` (или сочетание **C-x t j**). Вы можете ввести имя существующего чата, или ввести новое, и тогда чат будет

создан. Как и в предыдущем случае, создается новый буфер, в котором и ведутся все переговоры. Для выхода из чата вы можете использовать `tnt-leave-chat` (или `C-x t l`).

В том случае, если кто-то другой присылает вам приглашение в чат, или сообщение, то вы можете принять его с помощью `tnt-accept` (`C-x t a`) или отклонить с помощью `tnt-reject` (`C-x t r`) — это может быть полезным, если вы заняты, или не хотите беседовать с данным пользователем. В том случае, если вы получаете несколько сообщений, или приглашений в чат, то они накапливаются в очереди, в которой их можно просматривать с помощью команд `tnt-next-event` (`C-x t n`) — для получения следующего сообщения, и `tnt-prev-event` (`C-x t p`) — для получения предыдущего сообщения.

Прочие команды Вы можете формировать списки друзей, для того, чтобы быстро набирать их имена. Для просмотра существующего списка используется команда `tnt-show-buddies` (`C-x t b`), а для редактирования — команда `tnt-edit-buddies` (`C-x t B`). Имена заданные в этом списке будут использованы для реализации механизма дополнений в строках ввода, там, где необходим ввод имен пользователей.

TNT может также работать как мост между AIM и электронной почтой. Это очень удобно, если вы хотите получать сообщения из AIM на ваш почтовый адрес. Это выполняется с помощью команды `tnt-toggle-email` (`C-x t M`). И для ее работы нужно задать две переменные `tnt-email-to-pipe-to` и `tnt-email-binary`.

В том случае, если у вас несколько зарегистрированных имен, то вы можете использовать команду `tnt-switch-user` (`C-x t s`) для переключения между разными именами.

Команда `tnt-mute` (`C-x t m`) может использоваться для включения и отключения звуковых сигналов, используемых TNT для оповещения пользователя.

2.3.3 Настройка

Настройку TNT можно осуществлять как с помощью стандартных механизмов настройки Emacs, так и с помощью кода на Emacs Lisp. Для вызова настройки можно использовать команду `M-x tnt-customize`, или `M-x customize-group tnt`.

Настройка входа в систему TNT позволяет пользователю использовать несколько учетных записей для использования. Для задания имени и пароля для входа, сразу после запуска TNT, используются переменные `tnt-default-username` и `tnt-default-password`. А для задания связей между именами и паролями в учетных записях, используется ассоциативный список `tnt-username-alist`. Но будьте осторожны с этими переменными — помните, что они хранятся в вашем файле настройки как обычный текст, и в случае установки неправильных прав на файл, могут быть прочитаны другими пользователями.

Настройка звука TNT позволяет настроить проигрывание разных сигналов в зависимости от разных событий. Настройка сигналов управляется с помощью переменных `tnt-beep-on-...`, например `tnt-beep-on-incoming-message`. Кроме этого, вы можете настроить то, с помощью какой программы будут проигрываться звуковые сигналы. Эти настройки определяются переменными `tnt-sound-exec` и `tnt-sound-exec-args`.

Прочие настройки Пользователь имеет возможность настроить множество других параметров, полное описание которых доступно в файле `README` из поставки TNT, и которые можно настроить используя буфер настройки Emacs.

2.4 Jabber

Поддержку сервиса Jabber в настоящее время осуществляют несколько пакетов пакета для Emacs — `ejab`, который уже не развивается (замороженная версия доступна с <http://ejab.sourceforge.net>), `Jabber.el`, который активно развивается и представляет достаточно полную и расширяемую поддержку протокола Jabber, и `smurno`, который развивается медленней, но обладает достаточным набором возможностей (он доступен с <http://www.nongnu.org/smyrno/>). Поскольку `Jabber.el` динамично развивается, то я сосредоточусь исключительно на данном пакете.

2.4.1 Получение и установка

Пакет требует для своей работы GNU Emacs 21 и Gnus 5.10.x (из него нужно только пара файлов — `sha1-el.el` и `hex-util.el`, так что вы можете их установить не устанавливая весь Gnus), или XEmacs с поддержкой Mule, а также установленными пакетами `gnus`, `net-utils` и `mule-ucs`. Кроме этого, если установлен пакет FLIM, то `Jabber.el` может использовать его для поддержки SASL.

Установка пакета достаточно проста — надо просто скачать его дистрибутив с <http://emacs-jabber.sourceforge.net>, развернуть его и настроить загрузку в стартовом скрипте Emacs'а, например следующими командами:

```
;; если вы установили jabber.el в нестандартное место
(add-to-list 'load-path "path_to_jabber.el_dir")
(require 'jabber)
```

После загрузки пакета, необходимо задать имя пользователя и сервера, что делается с помощью команды M-x `jabber-customize`. С помощью этой команды можно и зарегистрировать нового пользователя, если перед выполнением команды задать префикс.

2.4.2 Основные действия

После настройки, `Jabber.el` готов к работе, необходимо только подключиться к выбранному серверу Jabber.

Подключение, отключение и состояние Подключение к серверу выполняется одной командой — M-x `jabber-connect`, после подключения с помощью команды M-x `jabber-send-presence` можно отправить на сервер уведомление о присутствии. Эта команда дает возможность вам указать статус присутствия, если вы оставили статус пустым, то будет установлено значение по умолчанию равное `Online`.

Для отключения от сервера Jabber, просто выполните команду M-x `jabber-disconnect`, и ваше подключение будет закрыто.

`Jabber.el` позволяет вам изменять состояние вашей доступности. Это можно сделать как явно, с помощью команды M-x `jabber-send-presence`, так и программно, с помощью этой

же функции. Кроме этого, вы можете запрограммировать сервер Jabber на уведомление вас, об изменении статуса нужного вам человека. Это выполняется с помощью команды `M-x jabber-send-subscription-request`.

Переговоры (chatting) После подключения, пользователю становится доступен буфер `roster'a`, находясь в котором он может выполняться различные действия. Одно из таких действий — переговоры (или `chatting`). Для начала переговоров, нужно лишь выбрать персону с кем вы будете переговариваться и набрать `C-c C-c`, и будет создан буфер с именем `*-jabber-chat-: -ИМЯПЕРСОНЫ-*`, в котором вы и сможете вести переговоры.

Групповые чаты Для подключения к групповому чату пользователь может использовать команду `M-x jabber-groupchat-join`, или использовать сочетание клавиш `C-c C-m` находясь в буфере `roster'a`. После подключения к чату, будет создан буфер с именем `*-jabber-groupchat-: -ИМЯЧАТА-*`, что позволяет одновременно участвовать в нескольких обсуждениях. Работа с буфером осуществляется точно так же, как и при обычных переговорах. Для выхода из группового чата просто наберите `M-x jabber-groupchat-leave`.

Кроме этого, если вы являетесь владельцем данного чата, пакет позволяет произвести его настройку, используя команду `M-x jabber-groupchat-get-config`.

Работа с буфером roster Буфер `roster'a` называется `*-jabber-*`, и содержит в себе список контактов вашего `roster'a`. С использованием данного буфера проводятся все операции, которые требуют указания Jabber ID, такие как переговоры, групповые и многопользовательские чаты, получение информации `roster'a` и работа с сервисами.

Работа с сервисами jabber Jabber.el позволяет пользователю получить доступ к большинству возможностей предоставляемых Jabber. Сюда входят — просмотр списка сервисов, поиск в каталогах, регистрация для использования сервисов, а также поддержка набора специализированных команд (`ad-hoc`).

Для использования каких-либо сервисов, вам сначала нужно их найти. Это может выполняться двумя способами — поиском (`discovering`) и просмотром (`browsing`). Первый способ является предпочтительным, поскольку использует новые протоколы. Для поиска сервисов могут использоваться две команды — `M-x jabber-get-disco-items` и `M-x jabber-get-disco-info`, которые выдают информацию либо о конкретном сервисе, либо о сервисах, относящихся к выбранному сервису. Эти же команды доступны через меню, которое вызывается сочетанием клавиш `C-c C-i` из буфера `roster'a`.

Поиск по сервисам выполняется с помощью команды `M-x jabber-get-search`, или выбором пункта меню, отображаемого с помощью сочетания клавиш `C-c C-s`. После выполнения запроса, вам будет показана форма поиска, которую необходимо заполнить для его выполнения.

Регистрация производится аналогично проведению поиска, только используется команда `M-x jabber-get-register`. С помощью данного сервиса, вы можете например, сменить свой пароль, или зарегистрироваться для использования конкретного сервиса. Как и в случае с поиском, при регистрации, модуль отобразит форму регистрации, которую вам необходимо будет заполнить.

Для работы со специализированными командами, пользователь сначала должен получить их список, поддерживаемых сервером. Это выполняется с помощью команды M-x `jabber-ahc-get-list`. Для выполнения специализированной команды, используется команда M-x `jabber-ahc-execute-command`. Специализированные команды могут использоваться, например, для того, чтобы удаленно изменить состояние вашего клиента, или другие вещи.

2.4.3 Настройка и расширение возможностей

Настройка клиента Настройка клиента выполняется с помощью команды M-x `jabber-customize` или, что тоже самое, с помощью команды M-x `customize-group jabber`. Переходя по разделам группы, вы можете настроить как основные параметры, такие как имя, сервер, так и другие, например уведомления о событиях, параметры начертаний (`faces`) и различные хуки.

Этот процесс является интуитивно-понятным, и поэтому я не буду заострять на нем внимание. Список переменных, которые используются для настройки с помощью этих команд, приведен в руководстве для Jabber.el.

Расширение возможностей пакета Jabber.el позволяет пользователю расширять возможности пакета, предоставляя возможности по приему и отправке новых типов запросов и ответов, осуществлять доступ к данным соединения и многое другое.

XML, который используется для общения с сервером jabber, представляется в виде списков, которые создаются модулем `xml.el` — каждый тэг представляется в виде списка, первым элементом которого является символ представляющий тэг, за тем следует список из списков атрибутов, и затем идут вложенные тэги и содержимое.

Текущее состояние roster'a представляется списком с именем `*jabber-roster*`. В данном списке можно найти подробную информацию о группах, состоянии соединения и прочих параметрах.

Подробное описание процесса расширения возможностей пакета приведено в разделе "Hacking and extending" справочной страницы Jabber.el

2.5 MSN

Судя по информации в Internet, кроме вышеперечисленных модулей, также существует и рудиментарная поддержка служб мгновенного обмена сообщениями MSN. Реализация этого сервиса доступна с <http://www.geocities.co.jp/SiliconValley-SanJose/7474/EmacsLisp.html>, но страницы написаны на японском языке, и поэтому информации не так уж и много, хотя по скриншоту видно, что клиент работает.

3 IRC

IRC является достаточно популярным сервисом в среде пользователей Unix, которых много среди пользователей Emacs, и это привело к появлению нескольких пакетов, поддерживающих данный сервис. В настоящее время наиболее активно развивающимся является

пакет Erc, но существует и альтернативная реализация, Zenirc, которая также имеет много возможностей, но уже практически не поддерживается (если вы хотите посмотреть на него, то вы можете найти исходные тексты по адресу <ftp://ftp.splode.com/pub/zenirc>).

Дальнейший рассказ будет в основном про пакет Erc, хотя может быть я буду упоминать и его конкурента.

3.1 Основные возможности пакета Erc

Пакет Erc предоставляет для пользователя множество возможностей, вот основные:

- Поддержку работы с несколькими каналами и серверами. Каждый канал работает в своем буфере;
- Уведомления;
- Дополнения ников;
- Поддержку разных языков;
- Отслеживание активности на каналах;
- Расширение возможностей, за счет написания скриптов;
- Подсветку ключевых слов и различных действий;
- Взаимодействовать с другими средствами обмена сообщениями, используя BitlBee (более подробно о настройке Erc для использования вместе с BitlBee смотрите страницу <http://www.emacswiki.org/cgi-bin/wiki?BitlBee>;
- и многое, многое другое.

Пакет работает как с GNU Emacs, так и с XEmacs.

3.2 Установка пакета

Исходные тексты Erc доступны по адресу <http://sourceforge.net/projects/erc/>. Необходимо только раскрыть полученный архив, и поместить несколько команд в ваш файл инициализации, например вот так:

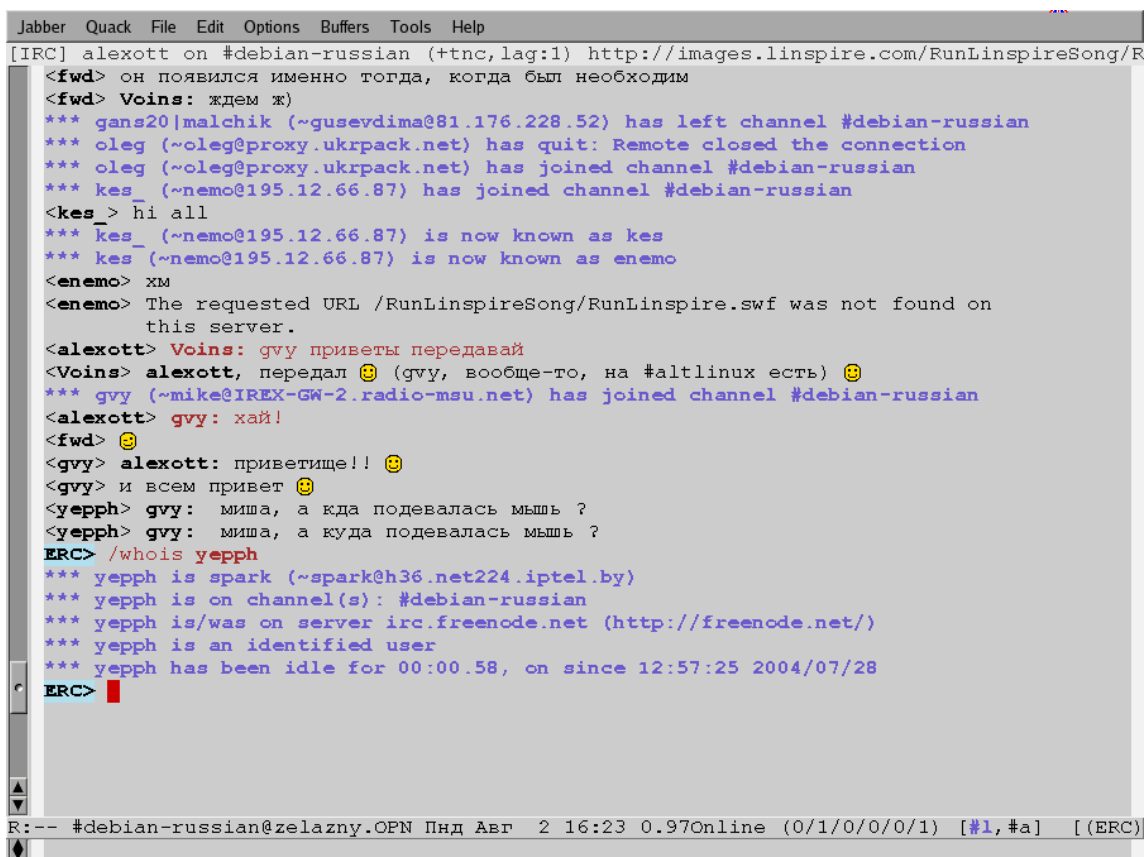
```
(add-to-list 'load-path "path_to_erc_dir")
(require 'erc)
```

Это минимальный набор команд, необходимых для загрузки данного пакета. Но Erc предоставляет намного больше возможностей, поэтому у человека, который часто пользуется данным пакетом, файл инициализации выглядит намного сложнее. Вы можете найти такие примеры на Emacs Wiki, или в разделе XrefId[?]. А более подробно про настройку я расскажу в разделе XrefId[?].

3.3 Основные операции

Начала работы вам необходимо набрать команды M-х `erc-select`, которая запросит у вас имя сервера, порт и другие параметры. Вы можете задать эти параметры один раз произведя настройку клиента (см. раздел [XrefId\[??\]](#)). В зависимости от настройки, клиент может сразу подключиться к нужным вам каналам — это выполняется пакетом `erc-autojoin`, который поставляется с `Егс`.

После того, как вы подключились к IRC-серверу, вы можете выполнять обычные команды IRC — подключение к новым каналам, получение информации и много других вещей. Команды вводятся в строке, которая начинается с `ERC>`. Окно канала может выглядеть как изображено на рисунке [XrefId\[??\]](#), кроме того, вы можете использовать стандартные команды Emacs для разделения фрейма на несколько окон и отображать в них разные каналы.



```
Jabber Quack File Edit Options Buffers Tools Help
[IRC] alexott on #debian-russian (+tnc,lag:1) http://images.linspire.com/RunLinspireSong/R
<fwd> он появился именно тогда, когда был необходим
<fwd> Voins: ждем ж)
*** gans20|malchik (~gusevdima@81.176.228.52) has left channel #debian-russian
*** oleg (~oleg@проxy.ukrpack.net) has quit: Remote closed the connection
*** oleg (~oleg@проxy.ukrpack.net) has joined channel #debian-russian
*** kes_ (~nemo@195.12.66.87) has joined channel #debian-russian
<kes_> hi all
*** kes_ (~nemo@195.12.66.87) is now known as kes
*** kes_ (~nemo@195.12.66.87) is now known as enemo
<enemo> хм
<enemo> The requested URL /RunLinspireSong/RunLinspire.swf was not found on
this server.
<alexott> Voins: gvy приветы передавай
<Voins> alexott, передал 😊 (gvy, вообще-то, на #altlinux есть) 😊
*** gvy (~mike@IREX-GW-2.radio-msu.net) has joined channel #debian-russian
<alexott> gvy: хай!
<fwd> 😊
<gvy> alexott: приветнице!! 😊
<gvy> и всем привет 😊
<yepph> gvy: миша, а кда подевалась мышь ?
<yepph> gvy: миша, а куда подевалась мышь ?
ERC> /whois yepph
*** yepph is spark (~spark@h36.net224.iptel.by)
*** yepph is on channel(s): #debian-russian
*** yepph is/was on server irc.freenode.net (http://freenode.net/)
*** yepph is an identified user
*** yepph has been idle for 00:00.58, on since 12:57:25 2004/07/28
ERC> █
R:-- #debian-russian@zelazny.OPN Пнд Авг 2 16:23 0.97Online (0/1/0/0/0/1) [#1,#a] [(ERC)]
```

Рис. 1: Окно сеанса `Егс`

Во время работы, `erc` изменяет строку состояния, в которой отображает имя канала и сервера, а также изменение статуса других каналов, в виде сокращения `#` и первой буквы имени канала, и разным цветом, синим — когда кто-то заходит на канал или покидает в него, и красным когда кто-то беседует на канале. В верхней строке окна отображается название канала, текущий лаг, и тема канала. Кроме этого, `erc` позволяет использовать стандартные механизмы дополнения имен пользователей и каналов, при их наборе.

Для того, чтобы закончить работу с IRC, наберите команды `erc-quit-server`.

3.4 Модули Erc

В составе Erc идет некоторое количество модулей, которые расширяют основную функциональность пакета. Для каждого из этих модулей пользователь может выполнить дополнительную настройку, управляющую его поведением. Часть модулей загружается автоматически, а некоторые требуют индивидуальной загрузки. Список автоматически загружаемых модулей хранится в переменной `erc-modules`.

Ниже приведен список наиболее часто используемых модулей, и того какие настройки для них могут выполняться.

3.4.1 `erc-autojoin`

Данный пакет позволяет запоминать на какие каналы заходил пользователь используя команды `/JOIN` и `/PART`, и при следующем сеансе использования `erc` восстановить сохраненную конфигурацию. Список автоматически подключаемых каналов сохраняется в переменной `erc-autojoin-channels-alist`. Данная переменная состоит из списков вида "имя сервера канал1 канал2"..., например:

```
(setq erc-autojoin-channels-alist
  '(("freenode.net" "#emacs" "#arch" "#squiddev")))
```

По умолчанию, данный режим включен, и используется.

3.4.2 `erc-button`

Данный модуль позволяет создать нажимаемые кнопки на основе текста буфера. По умолчанию данный режим включен, и используется для организации нажимаемых кнопок для самых разных объектов, таких как:

- адресов сайтов (URL's);
- запросов к Google;
- имен пользователей IRC;
- RFC (что позволяет сразу перейти к нужному документу);
- разделов документации в формате Info;
- интернет-времени.

Пользователи могут определять и свои типы, используя механизм сокращений (`abbrevs`). Кроме этого, пользователь имеет возможность перехода между кнопками, используя сочетание `C-c C-w C-b` для перехода на предыдущую кнопку, и `C-c C-w C-f` для перехода на следующую.

3.4.3 `erc-track`

Данный модуль позволяет отслеживать изменения в буферах `erc` — данный режим включен по умолчанию. При использовании данного модуля, в строке статуса отображается список каналов, в которых произошли изменения. По умолчанию, отслеживается вся активность — вход новых посетителей, выход с канала, и изменение имени, что не всегда нужно знать. С помощью переменной `erc-track-exclude-types` вы можете исключить некоторые команды из списка отслеживаемой активности. Например, вот так:

```
(setq erc-track-exclude-types '("JOIN" "NICK" "PART" "QUIT"))
```

Для перехода в буфер, в котором, в последнем была какая-либо активность, вы можете использовать сочетание клавиш `C-c C-space` или `C-c C-@`. В том случае, если нет больше активных каналов, данное сочетание переключает вас в тот буфер, в котором вы работали перед этим.

3.4.4 `emacs-completion`

Данный модуль позволяет использовать дополнение имен каналов и посетителей, а также команд `Erc`, используя клавишу `TAB`. Этот модуль разрешен по умолчанию, и для своей работы использует программируемый механизм дополнений, который реализуется библиотекой `rcomplete`, которая обычно поставляется с `Emacs`. В том случае, если у вас нет этой библиотеки, вы все равно можете воспользоваться возможностями дополнения, переключившись на использование старого механизма, правда это не рекомендуется делать — лучше установите библиотеку `rcomplete`.

3.4.5 `erc-fill`

По умолчанию `Erc` позволяет красиво заполнять и выравнивать отображаемый текст. По умолчанию используется режим заполнения, с переменным префиксом, зависящим от длины имени посетителя, который написал данное сообщение. Однако вы можете изменить эту настройку с помощью установки переменной `erc-fill-prefix`, в которой вы можете задать данные для заполнения — например, несколько пробелов подряд. Кроме этого, существуют и другие варианты заполнения, про которые вы можете прочитать на `Emacs Wiki`².

3.4.6 `erc-stamp`

Кроме всего прочего, `Erc` позволяет вам вставлять в текст буферов временные пометки, для того, чтобы вам было удобней просматривать журналы бесед, или точно находить нужную беседу. По умолчанию данный модуль не подгружается автоматически, так что вам необходимо добавить его к списку загружаемых модулей. Для настройки отображения можно задать несколько переменных, которые будут описывать вид отображаемых временных пометок, и другие параметры.

²<http://www.emacswiki.org/cgi-bin/wiki/ErcFilling>

3.4.7 `erc-netsplit`

Данный модуль обеспечивает определение отключения от сети целых IRC-серверов (так называемый `netsplit`), и подключение ранее отключенных серверов. Вы можете выполнять свой код в данных случаях, воспользовавшись хуками `erc-netsplit-hook` и `erc-netjoin-hook`.

3.4.8 `erc-autoaway`

Иногда вы отходите от клавиатуры на длительный срок, и можете забыть сообщить серверу, что вы отсутствуете. Чтобы это делалось автоматически, с Erc поставляется модуль `erc-autoaway`, который сам изменит режим после заданного отрезка времени. Данный модуль не загружается автоматически, так что вам нужно самим подключить его, с помощью команд, подобных этим:

```
(require 'erc-autoaway)
(setq erc-autoaway-idle-seconds 600)
(setq erc-auto-discard-away t)
```

первая команда подгружает сам модуль, вторая задает через сколько времени вы будете считаться отсутствующим, а последняя задает режим автоматического выхода из режима отсутствия, если вы что-то напишете в каком-либо из буферов.

3.4.9 `erc-log`

Erc позволяет вам сохранять журналы бесед на IRC. Для этого существует модуль `erc-log`. Однако, данная возможность не подключается автоматически, а требует ручной загрузки, например вот так:

```
(require 'erc-log)
(setq erc-log-channels-directory "~/irc/logs/")
```

первая команда загружает нужный модуль, а вторая задает расположение для журнальных файлов.

Сохранение журналов может производиться несколькими способами. Для обычных пользователей, наиболее приемлимым будет использование команды

```
(setq erc-save-buffer-on-part t)
```

что приведет к сохранению журналов при выходе из канала. Однако пользователи могут явно сохранить журналы с помощью сочетания клавиш `C-c C-l` или явно выполнив команду `M-x erc-save-buffer-in-logs`. Вы можете также поместить временные метки в сохраняемые журналы — это контролируется с помощью переменной `erc-hide-timestamps`.

Erc умеет восстанавливать часть содержимого предыдущих сессий используя журнальные файлы. Если вы хотите, чтобы в буфере отображались старые разговоры, то установите переменную `erc-log-insert-log-on-open` в истинное значение. По умолчанию, данная переменная равна `nil`.

Более подробную информацию о настройке журналирования, а также о возможных проблемах, и способах их решения читайте на Emacs Wiki³.

³<http://www.emacswiki.org/cgi-bin/wiki/ErcLogging>

3.4.10 `erc-nickserv`

В некоторых IRC-сетях вы можете зарегистрировать свое имя, и таким образом избежать возможных конфликтов между пользователями. Эта возможность реализуется с помощью сервиса под названием *NickServ*. По умолчанию, когда вы входите в такую сеть, в ответ на приглашение сервера, `erc` в мини-буфере запрашивает у вас пароль для регистрации. Чтобы избежать повторяющихся действий, был разработан модуль *erc-nickserv*, который идет в поставке `erc`, но не загружается автоматически. Настройку этого модуля можно выполнить с помощью следующих команд:

```
(require 'erc-nickserv)
(erc-nickserv-mode t)
(setq erc-prompt-for-nickserv-password nil)
(setq erc-nickserv-passwords
      '((freenode      (("nick" . "password")))))
```

Первые две команды загружают модуль и включают соответствующий режим. Третья команда отключает запрос пароля у пользователя при запросе от сервера. Ну а четвертая команда задает параметры ассоциативного списка, описывающего сервера, имена и пароли для них. Для каждого из серверов, вы можете задать одно или несколько имен, и соответствующие пароли для них. Заметьте, что в качестве ключа списка задается символ, соответствующий IRC-сети. Отображение между именами сетей и доменными именами серверов хранится в переменной `erc-networks`, которую вы при необходимости можете изменять.

3.4.11 `erc-truncate`

В том случае, если вы постоянно пользуетесь IRC, размеры буферов с разговорами, могут расти очень быстро, и поэтому может быть желательно держать их в рамках определенных размеров. Для этого используется модуль *erc-truncate*, которые следит за размерами буферов каналов, и при необходимости выбрасывает лишние данные. Данный модуль поставляется с `erc`, но не загружается автоматически.

Параметры модуль контролируются несколькими переменными. Переменная `erc-max-buffer-size` задает максимальный размер буфера, в байтах. А если установлена переменная `erc-truncate-buffer` то обрезание буферов будет происходить при сохранении журналов. Вы также можете явно выполнить обрезание буферов, используя команду M-x `erc-truncate-buffer`.

3.4.12 `erc-dcc` и `erc-xdcc`

Для `Erc` реализована поддержка прямой передачи данных между клиентами (DCC), которая реализуется с помощью модуля *erc-dcc*. Данный модуль не загружается автоматически, так что вы должны сделать это сами. После загрузки этого модуля, вам будут доступны команды `/dcc`, которые вы можете использовать для разговоров, и для передачи файлов между клиентами.

Кроме этого, в `Erc` реализована возможность организации файловых сервисов, на основе IRC. Эта возможность реализуется с помощью модуля *erc-xdcc*.

3.4.13 Прочие модули

Кроме описанных выше модулей существуют еще и другие:

- *erc-bbdb* — для интеграции ерс и BBDB;
- *erc-notify* — для организации оповещения, когда определенный пользователь входит или выходит из сети;
- *erc-smileys* — для преобразования smileys в изображения с соответствующими эмоциями;
- *erc-sound* — для проигрывания звуков при некоторых событиях;
- *erc-nicklist* — отображает список имен для данного канала в отдельном буфере;
- *erc-speedbar* — обеспечивает интеграцию ерс и speedbar;
- *erc-chess* — позволяет играть в шахматы с другими пользователями на IRC;
- *erc-lang* — реализует новую команду /lang для выбора языка на канале;
- *erc-list* — обеспечивает быстрый механизм получения списка каналов;
- *erc-menu* — добавляет меню к буферам Ерс, что может быть очень удобно, особенно для начинающих пользователей;
- *erc-speak* — позволяет проговаривать содержимое разговоров с помощью Emacspeak;
- *erc-page* — обеспечивает обработку /ctcp page команд;
- *erc-replace* — выполняет замены в тексте сообщений.

3.5 Настройка клиента

Настройку Ерс можно выполнить используя команду M-x `customize-group erc`. Появившееся окно настройки позволяет вам задать основные параметры клиента, а также перейти к настройке разных подсистем клиента. Но кроме визуальной настройки, вы можете и сами определять порядок инициализации, используя Emacs Lisp.

3.5.1 Настройка auto-query

Ерс позволяет настроить свое поведение в отношении личных сообщений. По умолчанию, они отображаются в буфере сервера, что не всегда удобно, когда ты разговариваешь с несколькими людьми. Однако, с помощью переменной `erc-auto-query` вы можете изменить поведение Ерс. Среди разрешенных значений:

- `'buffer` — показывать личные сообщения в отдельном буфере;
- `'window` — показывать новые сообщения в новом, выбранном окне;
- `'window-noselect` — то же, что и предыдущее, но без перехода в новое окно;

- `'bury` — создает новый буфер, но не переходит в него;
- `'frame` — открывает новый фрейм.

Вы можете создавать такие буфера не только для личных сообщений, но и для уведомлений сервера. Это делается добавлением соответствующего хука.

3.5.2 Игнорирование и скрывание лишней информации

С помощью Erc вы можете делать часть информации невидимой. Это относится как к посетителям серверов, так и к различным уведомлениям, поступающим от самих серверов.

Например, чтобы не отображать данные о входе и выходе с каналов, вы можете использовать переменную `erc-hide-list`, в качестве значения которой задается список игнорируемых команд. Например, вот так:

```
(setq erc-hide-list '("JOIN" "PART"))
```

Для временного игнорирования сообщений от какого-либо посетителя, вы можете использовать команду `/IGNORE`, но она будет действовать только до окончания сессии. Для установки постоянного игнорирования, вам необходимо в переменной `erc-ignore-list` задать список ников, или регулярных выражений, которые будут приводить к игнорированию сообщений от посетителей, чьи ники и сетевые адреса подпадают под заданное выражение.

Кроме этого, вы можете скрывать часть содержимого, основываясь на более сложных вещах, например, если кто-то произносит определенные фразы, или разговаривает с роботами, засоряя канал ненужным текстом. Как это сделать описано на странице Emacs Wiki⁴.

3.5.3 Настройка подсветки

Erc позволяет вам подсвечивать различные части сообщений на канале. Вы можете подсвечивать текст, имена пользователей, при этом по умолчанию есть разделение на "друзей" и "дураков":-) Вы можете задать список имен, и они будут отображаться с заданным оформлением. По умолчанию этот режим отключен, но вы можете его легко включить с помощью команды `M-x customize-group erc-match`. В этой группе вы также обнаружите опции для настройки списков друзей, подсветки вашего собственного имени, а также можете задать подсветку произвольного текста. Для настройки параметров отображения, воспользуйтесь группой настройки `erc-faces`.

3.5.4 Настройка кодировок каналов

Erc позволяет своим пользователям использовать разные кодовые страницы для разных каналов (конечно, это только для тех пользователей, которые используют Emacs с поддержкой `mule`). Используя переменную `erc-encoding-coding-alist` вы можете задать соответствие между именами каналов, и кодировками, которые используются на этих каналах. Например, я могу нормально общаться на немецко-говорящих каналах, хотя на остальных каналах будет использоваться кодировка `koï8-r`.

⁴<http://www.emacswiki.org/cgi-bin/wiki/ErcIgnoring>

3.5.5 Хуки

По реализации Erc, по большей части управляется событиями, которые обрабатываются соответствующими хуками. Существующие хуки можно разделить на несколько больших групп:

- Хуки для серверных сообщений;
- Хуки для /ctcp команд;
- Хуки для отображения и ввода;
- Прочие хуки.

Хуки для серверных сообщений Стандарт для IRC определяет, что сервер должен посылать команды клиентам в специальном формате. Сама команда может быть либо трехзначным числом, либо именем команды. При получении данных команд Erc запускает соответствующий обработчик, и при его выполнении также запускает соответствующий хук, который именуется `erc-server-команда-hook`.

Хуки для команд /ctcp Для команд /ctcp существует два вида хуков — когда вы получаете запрос на выполнение команды, и когда вы получаете ответ о выполнении команды. Соответственно запускаются хуки `erc-ctcp-query-команда-hook` и `erc-ctcp-reply-команда-hook`.

Хуки для отображения и ввода Эти хуки также делятся на две группы — для отображения, и для обработки ввода.

К хукам отображения относятся:

- `erc-insert-pre-hook` — вызывается перед вставкой сообщения в буфер;
- `erc-insert-modify-hook` — вызывается когда текст уже вставлен, но при этом, буфер в которые вставлен текст, сужается до самого текста;
- `erc-insert-post-hook` — вызывается после вставки текста, но может обрабатывать весь буфер.

К хукам ввода относятся:

- `erc-send-pre-hook` — вызывается перед отправкой введенного пользователем сообщения;
- `erc-send-modify-hook` — вызывается после вставки текста в буфер, но при этом, буфер в которые вставлен текст, сужается до самого текста;
- `erc-send-post-hook` — вызывается после вставки текста, но может обрабатывать весь буфер.

`erc-send-modify-hook` и `erc-send-post-hook` никаким образом не влияют на отправляемый на сервер текст, и в основном могут использоваться для изменения параметров отображения текста в буфере.

Прочие хуки

- `erc-mode-hook` — вызывается для каждого буфера `erc` — буферов каналов, буферов серверов и т.п.;
- `erc-timer-hook` — вызывается при получении любого события от сервера;
- `erc-before-connect` и `erc-after-connect` — вызываются перед подключением к конкретному серверу, а также после выдачи сервером сообщения дня (MOTD).

3.5.6 Пример файла инициализации `Erc`

Ниже приведен пример настройки `Erc`, который используется у меня. Я буду писать комментарии сразу в коде, чтобы сделать его понятным:

```
;; здесь мы просто загружаем erc
(add-to-list 'load-path "~/emacs/erc")
(require 'erc)

;; загружаем авто-подключение к каналам, и задаем список каналов для
;; подключения
(require 'erc-autojoin)
(erc-autojoin-mode t)
(setq erc-autojoin-channels-alist
      '(("relay2" "#dansguardian" "#emacs" "#arch" "#linux.de"
         "#c.de" "#squiddev")))

;;
(require 'erc-fill)
(erc-fill-mode t)

;; задаем персональные данные, хотя их можно задать и через
;; M-x customize-group erc
(setq erc-user-full-name "Alex Ott")
(setq erc-email-userid "alexott@gmail.com")

;; часть относящаяся к логированию переговоров на каналах
;; нужно ли вставлять старый лог в окно канала?
(setq erc-log-insert-log-on-open nil)
;; логировать переговоры на каналах?
(setq erc-log-channels t)
;; где будут храниться логи
(setq erc-log-channels-directory "~/.irclogs/")
;; сохранять ли логи при возникновении PART
(setq erc-save-buffer-on-part t)
;; убирать или нет временные отметки?
(setq erc-hide-timestamps nil)
;; максимальный размер буфера канала
(setq erc-max-buffer-size 500000)
```

Большую часть использованных переменных можно установить воспользовавшись командами настройки Emacs, я показал этот пример только для наглядности. Вы можете

найти большое количество примеров настройки на страницах Emacs Wiki⁵.

3.6 Дополнительная информация

Вы можете найти огромное количество информации о Erc на страницах Emacs Wiki по адресу <http://www.emacswiki.org/cgi-bin/emacs-en/EmacsIRCClient>. Там вы сможете найти множество советов по настройке клиента, а также по интеграции его с различными приложениями, например, клиентом для игры в шахматы.

⁵<http://www.emacswiki.org/cgi-bin/emacs-en/EmacsIRCClient>