

Управление системой посредством «Центра Управления»

Использование конфигураторов

В качестве конфигуратора выступает Alterator в Линукс Мастер.

Проблемы, ведущие к использованию конфигуратора

Проблема №1: Минимизация последовательностей действий администратора, в особенности по первоначальной настройке. Превращение последовательности действий в атомарную операцию.

Минимизация числа действий администратора.

Настройка сети вручную представляет собой весьма непростое дело, не говоря уже о том, что компьютер приходится настраивать каждый раз, когда он включается. На самом деле компьютер, конечно, настраивается автоматически, так как существуют конфигурационные файлы, в которые вписываются необходимые значения, читаемые и используемые при каждом запуске. Но есть ещё одна задача, которая все ещё не очень удобно решается, и не всякий сумеет её решить. Задача связана с начальной настройкой системы, в частности, при её установке, и перенастройкой компьютера, в том случае, если, например, меняются внешние параметры машины (месторасположение и т.п.).

Проблема состоит в том, что при первоначальной настройке компьютера администратору, пусть он даже достаточно квалифицированный специалист, необходимо проделать довольно большое количество различных действий. На примере ПСПО установку можно разбить на 14 шагов, и в каждом необходимо отвечать на какие-то вопросы, и соответственно, если мыслить в категориях редактирования файлов --- то менять довольно большое количество различных файлов конфигурации. А если надо настроить не одну машину, а несколько, причём по-разному, то одновременная автоматическая установка невозможна. Учитывая, что чем большее количество манипуляций надо произвести, тем выше вероятность ошибки, было бы неплохо минимизировать количество действий. Это и есть первая проблема.

Минимизация действий администратора может быть двух разных видов:

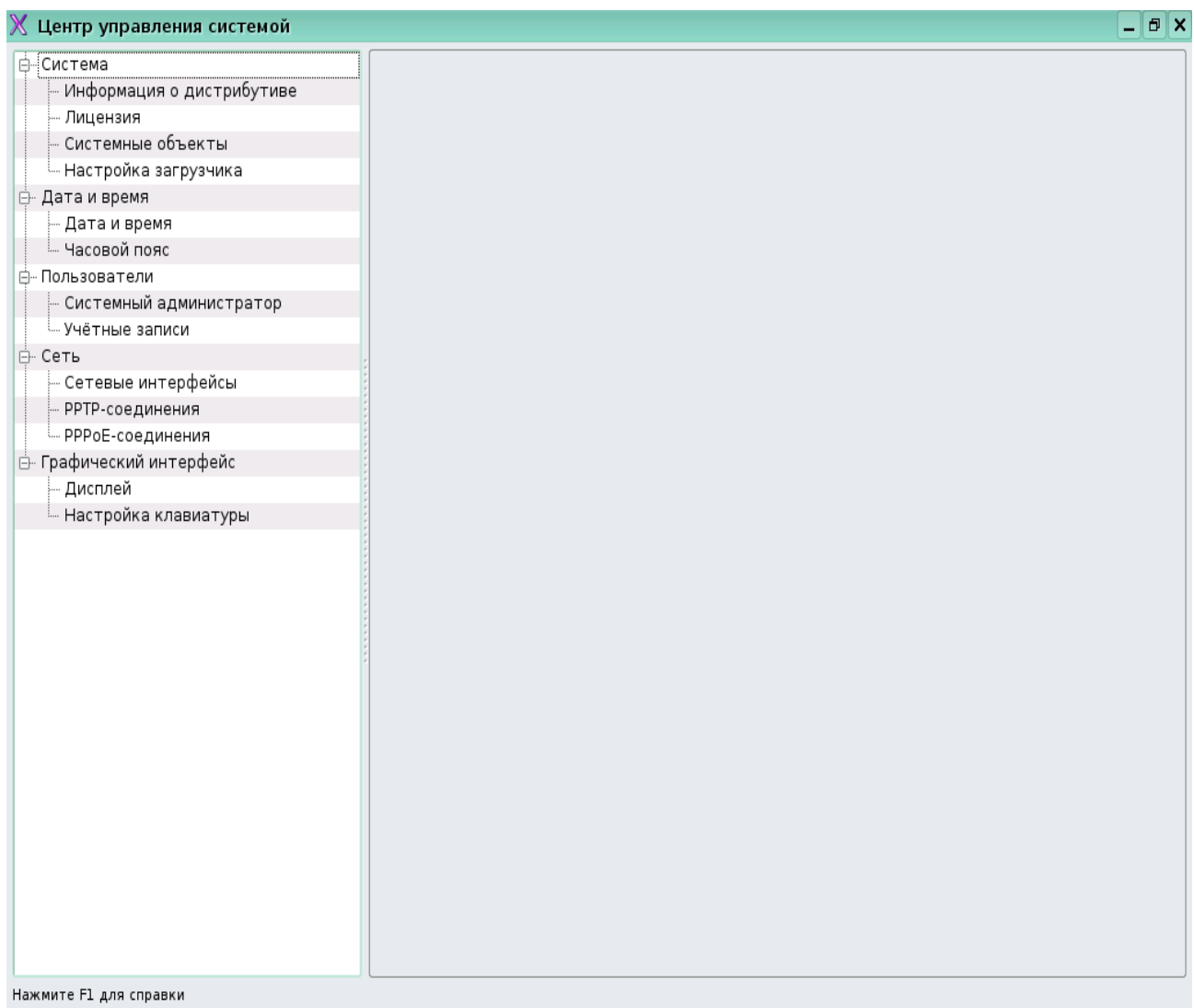
- первоначальная настройка;
- ситуация, когда однократное действие требует изменения многих многих файлов, т.е. необходимо полное воспроизведение всей последовательности действий.

Отсюда возникает требование превращения множества действий в атомарную операцию. Существует некий порядок работ, связанных, например, с разворачиванием почтового сервера. Как минимум, для этого требуется установить пакет и настроить его --- то есть отредактировать некоторое число файлов конфигурации. В любом случае системный администратор с большим опытом, после того, как он обнаружит, что ему три-четыре раза пришлось проделать одну и ту же последовательность действий, может написать сценарий на языке shell или языке Perl, который будет делать эту работу за него. Но для того, чтобы не разводить "зоопарк" мелких сценариев, каждый из которых производит какие-то мелкие действия, можно поступить по-другому --- добавить модуль к конфигуратору системы, который будет превращать последовательность фиксированных или параметризованных действий в, например, нажатие одной кнопки, что в случае опытного системного администратора равносильно запуску одного скрипта. В дальнейшем мы рассмотрим, как эта задача решается в ПСПО.

Работа оператора ЭВМ (неквалифицированного человека, отвечающего за машину).

Вторая проблема возникает, когда с действиями администратора должен справиться недостаточно квалифицированный человек. В советское время, когда компьютеры и микросхемы были большими, а объёмы памяти --- маленькими, такой человек назывался оператором. В наше время под оператором можно понимать две роли:

- первая --- это так называемый системный администратор, как его начали понимать с момента внедрения системы Windows, когда системное администрирование --- это систематическое выполнение последовательности действий "согласно предписаниям" и регулярная переустановка ОС в местах её "падения". Это можно назвать задачами повседневного администрирования (задачи, которые необходимо выполнять изо дня в день, и было бы очень неплохо, если они не были завязаны на повседневные навыки выполняющих). 30 лет назад задачами повседневного администрирования были: подать нужную ленту на нужное устройство, чтобы сделать резервное копирование, вкл/выкл, профилактический просмотр различных составляющих машины. Нынче же задачи таких операторов очень сильно расширились. Например, необходимо, чтобы оператор, не имея достаточных знаний о том, какие файлы редактировать, мог добавлять/удалять пользователей системы. Откроем Alterator и посмотрим, что он умеет.



Здесь представлена область деятельности оператора, которая, вобщем-то, совпадает с набором действий, которые должен выполнить любой человек, устанавливая ОС на свою машину. Кстати сказать, этот человек отчасти тоже оператор ЭВМ. Соответственно,

- вторая роль, которая сводится к роли оператора -- роль хозяина персональной ЭВМ. Если ваш компьютер находится у вас на работе, то вышеуказанными проблемами наверняка займётся специально обученный человек, а на домашней машине за вас никто этого не сделает, и вам придётся выработать некоторые навыки по управлению системой в режиме "недостатка информации". В идеале, разумеется, эта проблема должна быть исключена и любой пользователь должен быть технически грамотным. Linux хорошо для этого подходит, как весьма познаваемая система с большим количеством документации и развитым информационным пространством, но ведь недостаток информации может проистекать исключительно от недостатка времени --- например, вы мало используете машину (вам нужно и вы умеете выполнять лишь несколько разных действий), и, конечно, ради этого изучать все устройство системы вы вряд ли станете.

Конфигуратор

В чём проблема конфигуратора как такового? У нас есть некое множество точек воздействия на систему (условно говоря, системные конфигурационные файлы, хотя могут быть даже некие базы данных), которые для произведения пресловутых конфигурационных действий надо отредактировать. Причём надо понимать, что мы работаем в предположении, что пользователь конфигуратора не обладает всей полнотой информации, он не обязан знать всё, что происходит на самом деле при выполнении макрооперации.

- во-первых, согласно давней традиции, имеющейся в UNIX, а следом за ней и Linux, никакой унификации на структуру конфигурационных файлов и на то, где и как они располагаются, нет. Есть только соглашение, что они находятся в каталоге `/etc/`, хотя есть исключения; так, некоторые конфигурационные файлы располагаются в `/var/lib`.
- во-вторых, что более важно, --- ни в коем случае не рекомендуется (и в сообществе никто и не стремится) делать так, чтобы у всех конфигурационных файлов был единый синтаксис. Примеры различных конфигурационных файлов очевидны: есть файл `passwd`, который представляет собой строки, состоящие из полей, разделённых двоеточиями, есть конфигурационный файл сервера Apache, который использует синтаксис, схожий с SGML, есть некие службы, которые конфигурируются на языке XML с применением схем --- например, DE Gnome, и так далее. Если программа написана на shell, то и конфигурационный файл к ней обычно тоже написан на shell и представляет собой просто скрипт, который выполняется при запуске этой программы. Таким образом, в системе существует большое количество разнообразных синтаксисов, объединение которых считается нецелесообразным, поскольку в каждом случае решаются свои задачи и, соответственно, синтаксисы конфигурационных файлов заточены под конкретные решения. Это, как правило, не создает лишних трудностей системному администратору, для которого не проблема запомнить 6-7 разных синтаксисов или, что ещё более естественно, посмотреть документацию. Но это составляет существенную проблему, если мы хотим сделать обёртку, унифицированную по интерфейсу. К тому же, надо принимать во внимание не только содержание конфигурационных файлов, но и права доступа на эти файлы, состояние системы --- сервис, подвергающийся настройке, к примеру, может быть в данный момент запущен, а может быть остановлен.

Эти проблемы можно решить тремя способами:

1. решить в лоб, то есть, договориться о том, как должен выглядеть GUI, и потом написать упрощённый редактор конфигурационного файла, который производил бы свёртку множества действий в атомарную операцию для одного или нескольких файлов. Например, вместо редактирования ряда файлов в `/etc/net` (как было

описано ранее) мы бы запустили специальную программу, которая называлась бы "Сетевые профили" и считала бы все необходимые файлы, сформировала правильный вид и позволила бы нам менять с ее помощью нужные параметры.

- Недостатки этого подхода:
 - если взять все графические конфигурационные утилиты разных подсистем и собрать их вместе, то мы получаем решение лишь первой проблемы --- минимизации действий. Система остается не менее запутанной и непонятной обычному пользователю, так как вместо набора разнородных конфигурационных файлов мы получим набор не менее разнородных утилит.
 - Довольно тяжело обеспечить полное покрытие всех проблем. Это сделать можно, но это требует взаимодействия между всеми людьми, занимающимися этими утилитами. Более того, часто подобные утилиты разрабатываются не под задачу, а под решение.
 - Проблема синхронизации пространства имён --- многие данные запрашиваются во многих модулях повторно, и тем самым увеличивается возможность ошибки, когда те же данные неправильно введённые в другом модуле перечеркивают правильные введённые раньше.
- Достоинства:
 - Лёгкость сопровождения --- при изменении формата конфигурационного файла можно легко поменять и формат утилиты, особенно когда утилита настроена не на задачу, а на решение

Использование конфигураторов: продолжение

1. Решение в лоб (из предыдущего раздела)

Недостатки

- Вторая проблема заключается в обеспечении полного покрытия тех задач, которые нам нужно решать. Сделать это, разумеется, можно, но каждый раз приходится взаимодействовать с разными людьми, которые занимаются созданием соответствующих модулей. К примеру, нам нужно решить задачу "добавления пользователя". Это означает, что добавлять пользователя придется в `/etc/passwd`, в Samba, в хранилища типа LDAP и т. п. Заметим в скобках, что в линейке дистрибутивов ПСПО есть подсистемы, имеющие отдельный интерфейс конфигурирования, отличный от Alterator. Кроме того, конфигураторские интерфейсы часто разрабатываются не под задачу, а под инструмент. Типичный пример - графический конфигуратор `/etc/net`. Его использование, пожалуй, доставляет больше трудностей, чем непосредственное редактирование конфигурационных файлов. Чтобы эффективно с ним работать, надо знать, как именно функционирует `/etc/net`. Итак, здесь возникают значительные сложности в сопровождении.
- Третья проблема - это синхронизация пространства имен. С одной стороны, это проекция первых двух проблем, с другой - проблема более общего плана. К примеру, при настройке различных подсистем приходится вводить один и тот же IP-адрес компьютера. Такая процедура вызывает раздражение со стороны пользователя. А если допустить возможность ошибки? Если в разных конфигураторах ввести разные адреса, то гарантировать корректность функционирования всех конфигурируемых

таким образом подсистем будет невозможно. Таким образом, проблема унификации имен связана и с проблемой пересечения областей воздействия различных конфигураторов. Если для настройки некоторой службы нужно отредактировать, скажем, 10 файлов, а для настройки другой службы - 5, из которых 2 встречаются среди первых 10, то в случае разрозненных инструментов система, скорее всего, не будет функционировать вообще.

Достоинства

- Достоинство такого подхода состоит в легкости сопровождения каждого модуля в отдельности. Достаточно следить за обновлением служб, которые он конфигурирует, и адаптировать его в случае изменения синтаксиса тех или иных конфигурационных файлов. Здесь один из указанных ранее недостатков обращается в достоинство: когда инструмент конфигурации привязан не к задаче, а к инструменту, сопровождение его особенно легко. Конфигуратор для уже названного `/etc/net` пишется самими авторами, поэтому любое изменение в `/etc/net` сразу находит отражение в "редакторе настройки".

Заметим, что именно по этому принципу устроено большинство компонентов конфигулятора Debian GNU/Linux (не путать с Ubuntu). Среди требований, предъявляемых к пакетам, есть требование наличия специального конфигурационного скрипта, работающего по протоколу Debconf. При соблюдении этих требований инструмент и система его конфигурирования создаются одним и тем же человеком - мейнтейнером (либо сообществом мейнтейнеров). В каждый момент времени к любому модулю системы в любой части системы существует работающий конфигуратор. Другими словами, без конфигулятора сервис вообще не может существовать. С другой стороны, Debconf конфигурирует инструменты, а вовсе не решает задачи, что, очевидно, является существенным его недостатком. Считается, что эффективно пользоваться Debconf могут только специалисты, знающие, к каким конкретным изменениям в системе приведет, скажем, нажатие той или иной кнопки.

Другой пример - виртуализация в ALT Linux Server 4.0. Существует web-интерфейс, который позволяет эффективно управлять виртуализацией - если знать, что такое виртуализация, что такое `openvz`, Xen и пр.

2. Монолитная схема

Рассмотрим теперь монолитную схему построения конфигулятора: в одной большой (вероятно, модульной) программе перечислены решаемые задачи. Для каждой такой задачи существует отдельное решение, которое модифицирует конфигурационные файлы. Типичный пример более или менее монолитного конфигулятора - `webmin` (<http://www.webmin.com/>).

Достоинства

- Монолитный конфигулятор создается одной командой разработчиков. Для эффективной работы им достаточно адаптироваться к одному и тому же интерфейсу UI и API.
- Монолитный конфигулятор прозрачен для пользователя благодаря единому интерфейсу. Для работы с таким конфигуратором достаточно обучиться правильно заполнять все поля. Подобным образом конфигурируется, скажем, большинство подсистем в Windows. Как, к примеру, происходит подключение к провайдеру обычного пользователя? Ему выдается инструкция со скриншотами, в которой сказано: вписать такое-то значение в такое-то поле. Настройку могут производить те же самые люди, которые осуществляют собственно подключение домов. Заметим, что

данное достоинство может обернуться серьезным недостатком: с течением времени оно становится legacy (необходимостью поддерживать большое количество чего-либо старого, доставшегося по наследству). В какой-то момент придется изменять внешний вид, потому что изменились принципы функционирования. Сохранить старый интерфейс при новой "начинке" бывает необычайно тяжело. Отметим, однако, что разделение областей происходит по задачам, а не по инструментам. Задачи же меняются значительно реже: задача подключения к сети или добавления пользователя вряд ли изменится в ближайшие несколько лет, а вот инструмент для решения может и поменяться.

Недостатки

- Главный недостаток монолитного конфигуратора заключается в том, что его крайне сложно поддерживать: на это требуется очень много ресурсов. Если в Linux-системе немного изменился синтаксис одного из конфигурационных файлов, которыми конфигуратор умеет манипулировать, то нужно вовремя отследить это изменение и отреагировать на него, переписав соответствующую часть конфигуратора. При этом переписывание части может повлечь переписывание всего, начиная с интерфейса и заканчивая конкретным синтаксическим анализатором. Программист такой системы должен обладать талантами как в области системного администрирования (нужно хорошо представлять, как и что мы исправляем), так и в областях проектирования логики работы и программирования собственно интерфейса. По этой причине монолитный подход хорошо работает только в не очень быстро (лучше - последовательно) развивающихся системах с достаточно большим количеством разработчиков, занимающихся именно задачей построения конфигуратора. Выход новой версии, допустим, Red Hat Enterprise Linux сопровождается практически полным переписыванием того, что у них называется конфигуратором: это anaconda, их установщик, и несколько разрозненных модулей. Чтобы оценить объем прделываемой разработчиками RHEL работы, достаточно взглянуть на Changelog.

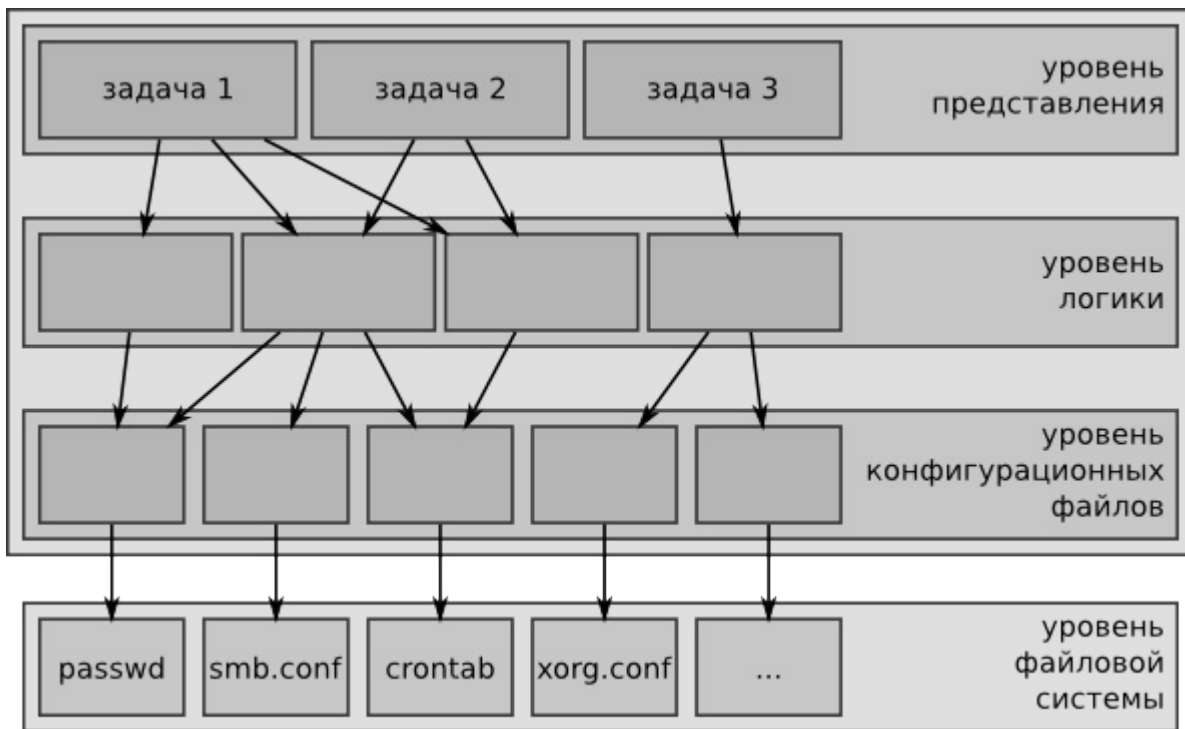
3. Трехслойная схема

Теория

Возможен и третий подход к построению конфигуратора - трехслойный. В нем разделяются действия трех типов:

- системного администратора, принимающего решения о том, как нужно управлять конфигурационными файлами;
- системного архитектора, определяющего логику работы и решающего, как нужно конфигурировать всю систему в целом;
- дизайнера, создающего пользовательский интерфейс.

Фактически, архитектура такого конфигуратора допускает деление как горизонтальное (по слоям), и вертикальное (по задачам):



Нижний уровень работает с конфигурационным файлом, средний - с логикой представления. Понятно, что уже на уровне представления одна задача может вылиться во взаимодействие сразу с несколькими конфигурационными файлами. Рассмотрим простой пример. Пусть на уровне представления есть следующая задача: добавить пользователя (в базу данных пользователей). Для оператора ЭВМ это означает, что где-то в интерфейсе есть список пользователей, обладающих тем или иным набором свойств. Он нажимает кнопку "добавить еще пользователя" и задает его свойства (пароль, домашний каталог, группу и пр.). На этом его роль заканчивается. На уровне же модификации конфигурационных файлов добавление пользователя - это достаточно непростая и зачастую неатомарная операция. Она не сводится к добавлению строки к `/etc/passwd`: есть еще Samba-пользователи и пр. Известно, что пользователи Samba хранятся в отдельной БД, причем список полей этой БД сильно отличается от списка полей в `/etc/passwd`. В Samba хранятся, к примеру, пароли пользователей, а в `/etc/passwd` хранятся лишь хэши паролей. Таким образом, на нижнем уровне операция добавления пользователя должна привести к модификации нескольких конфигурационных файлов. В зависимости от того, сколько есть "мест" в системе, ответственных за понятия пользователя, изменяется на уровне представления и содержимое того окна, куда оператор вводит данные. Например, там может быть такое поле: сделать домашний каталог создаваемого пользователя общедоступным по smb. Заметим, что человеку, который планирует эту систему, разбираться в том, какой синтаксис у `/etc/passwd`, а какой у `smbpasswd`, не следует. Не должен он и писать программу, которая редактирует (сама) файл `/etc/passwd`. Существуют специальные программы, занимающиеся модифицированием этого файла, например `adduser`, он же `useradd`. Подобные же утилиты существуют и для Samba. Человек, который реализует цепочку от представления и заполнения полей до нижнего уровня, не должен задумываться о том, с какими ключами вызывается та или иная программа.

Поэтому между уровнем логики и уровнем конфигурационных файлов существует, скажем так, прослойка, задача которой - унификация передаваемых данных. На уровне конфигурации специальный модуль распознает уже унифицированные низкоуровневые команды и превращает их в работу с `passwd` или что-либо иное (для другого модуля). Эти модули регистрируются в конфигураторе на уровне представления. Когда человек добавляет пользователя, он видит окно, а в нем два модуля: Samba и POSIX (Unix) users. Пользователь

уровню логики дает высокоуровневые команды, передавая данные из полей. Уровень логики транслирует это в низкоуровневые команды: модулю `posix` добавить пользователя с таким-то паролем, модулю `smb` добавить пользователя с таким-то паролем. После чего "нижние" модули делают свое дело и отвечают уровню логики на некотором унифицированном языке: "Я пользователя добавил, все в порядке", "Я пользователя пытался добавить, но там проблемы, ошибка вышла", - тогда уровень логики может сказать первому: "Отмени внесенные изменения" (иначе будет потеряна целостность системы). Управляет этим процессом именно уровень логики, а на уровень интерфейса выносятся сообщения об ошибке, оттранслированное, скажем, следующим образом: "При добавлении пользователя в Samba произошла ошибка, операция отменена". Преобразование низкоуровневых унифицированных команд в работу с различными файлами и программами осуществляется специальными сценариями на языке Shell (либо `awk`, `Scheme` или иных).

Точно так же происходит и простой просмотр данных. Уровень логики говорит: "У меня есть задача получить список пользователей, кто даст?" Модуль `Samba` отвечает: "Я", модуль `passwd` отвечает: "Я". Данные пойдут так, как описано, только уже снизу вверх.

Понятно, что разделение на три уровня в данной схеме принципиально. Если одна команда сверху транслируется в несколько действий снизу, то трудно будет при создании конфигуратора поддержать разделение труда среди администраторов, архитекторов и дизайнеров интерфейса. По трехслойной схеме устроен конфигуратор `YaST` в `SUSE`, хотя разделение на уровни там менее жесткое: возможно написать один большой `YaST`-модуль, который будет выполнять все части работы, за исключением собственно записи конфигурационного файла. Два верхних уровня в `YaST` вообще часто слиты в один. В конфигураторе же `Alterator`, используемом в дистрибутивах ПСПО, разделение соблюдается строже: выделены описание политик (уровень представления), трансляция команд и принятие решений (уровень логики) и собственно работа с конфигурационными файлами.

Проиллюстрируем описанную схему. Посмотрим список пакетов в нашем дистрибутиве, относящихся к `Alterator`:

```
[user@demo ~]$ apt-cache search alterator
alterator - ALT Linux configurator engine
alterator-apt - wrapper over apt-get
alterator-autoinstall - automatic non-interactive installer engine
alterator-backend-x11 - alterator backend for x11 setup and configuration
alterator-browser-qt - X11 Qt interface driver for alterator
alterator-chkconfig - alterator module for simple service setup
alterator-control - alterator module for control package
alterator-datetime - alterator module for date/time setup
alterator-design-server - stylesheets and images for server distribution
alterator-fbi - alterator on rails
alterator-firewall - alterator module for iptables firewall control
alterator-http - alterator http/cgi libraries
alterator-icons-lite - pictures for alterator
alterator-lilo - alterator module for lilo setup
alterator-lookout - dialog based interface for alterator
alterator-menu - alterator control center menu driver
alterator-net-common - helpers for etcnet administration
alterator-net-eth - alterator module for tcp/ip connections configuration
alterator-net-general - alterator module for general network settings
alterator-net-junior - alterator module for general network settings
alterator-net-pppoe - alterator module for pppoe connections configuration
alterator-net-pptp - alterator module for pptp connections configuration
alterator-net-wifi - alterator module for wi-fi connections administration
alterator-notes - alterator module for view license and release notes
alterator-pkg - additional package installation
alterator-root - alterator module for edit system administrator properties
alterator-sh-functions - helper functions for alterator shell based backends
```

alterator-squid - alterator module for Squid
alterator-standalone - System Management center
alterator-standalone-usermode - usermode bindings for alterator-standalone
alterator-sysconfig - alterator module for basic system settings
alterator-sysinfo - alterator module to view general system information
alterator-tzone - alterator module for timezone setup
alterator-ulogd - alterator module for network traffic statistics
alterator-users - alterator module for system users administration
alterator-wizardface - alterator's wizard like module aggregator
alterator-x11 - alterator module for Xorg setup and configuration
alterator-xkb - alterator module for XKB administration
design-alterator-browser-qt-junior - Junior design for alterator-browser-qt
docs-alterator_vm - Разбиение диска средствами программы установки
httpd-alterator - Apache HTTP Server (alterator edition)
alterator-vsftpd - alterator module for vsftpd configuration

На экране появился список основных компонентов и модулей Alterator'a. К примеру, alterator-lookout - это компонент, отвечающий за уровень представления, а alterator-xkb - модуль, управляющий настройками клавиатуры. Выведем список файлов в системе, относящихся к пакету alterator-xkb:

```
[user@demo ~]$ rpm -ql alterator-xkb
/usr/bin/xkbdatadump
/usr/bin/xkbmapconf
/usr/lib/alterator/backend3/template-xkb
/usr/lib/alterator/backend3/xkb
/usr/share/alterator/applications/xkb.desktop
/usr/share/alterator/ui/xkb
/usr/share/alterator/ui/xkb/avail_layout.scm
/usr/share/alterator/ui/xkb/html-messages.scm
/usr/share/alterator/ui/xkb/index.scm
/usr/share/locale/be/LC_MESSAGES/alterator-xkb.mo
/usr/share/locale/ru/LC_MESSAGES/alterator-xkb.mo
/usr/share/locale/uk/LC_MESSAGES/alterator-xkb.mo
/var/www/html/fbi
/var/www/html/fbi/xkb-layout.html
/var/www/html/fbi/xkb.html
```

Файлы из `/usr/lib/alterator/backend3` - это собственно back-end, исполняемые файлы из `/usr/bin` - вспомогательные утилиты. В `/usr/share/locale` лежат данные для локализации. Написанные на языке Scheme сценарии для интерфейса находятся в `/usr/share/alterator/ui` (в специальном подкаталоге), а в `/var/www/html/fbi` размещены файлы, относящиеся к Web-интерфейсу (обратим внимание на наличие двух разных способов представления). Заметим, что Alterator позволяет писать back-end'ы на чем угодно: подойдет и Shell, и awk, и Perl, и Scheme (который, как мы видим, в данном случае использовался для написания UI; заглянув в файл `avail_layout.scm`, мы увидим всего лишь 66 строк программного кода).

Недостатки

- Главный недостаток рассмотренной конструкции (да и составного конфигуратора вообще) - его "текучесть". Разобраться, какой именно протокол взаимодействия между уровнями "самый правильный", бывает весьма непросто: его описание постоянно претерпевает значительные изменения. То же и с пользовательским интерфейсом: он не связан с конкретным типом интерфейса, а предоставляет множество разнородных элементов управления. Конструирование библиотек для таких интерфейсов - задача весьма непростая.

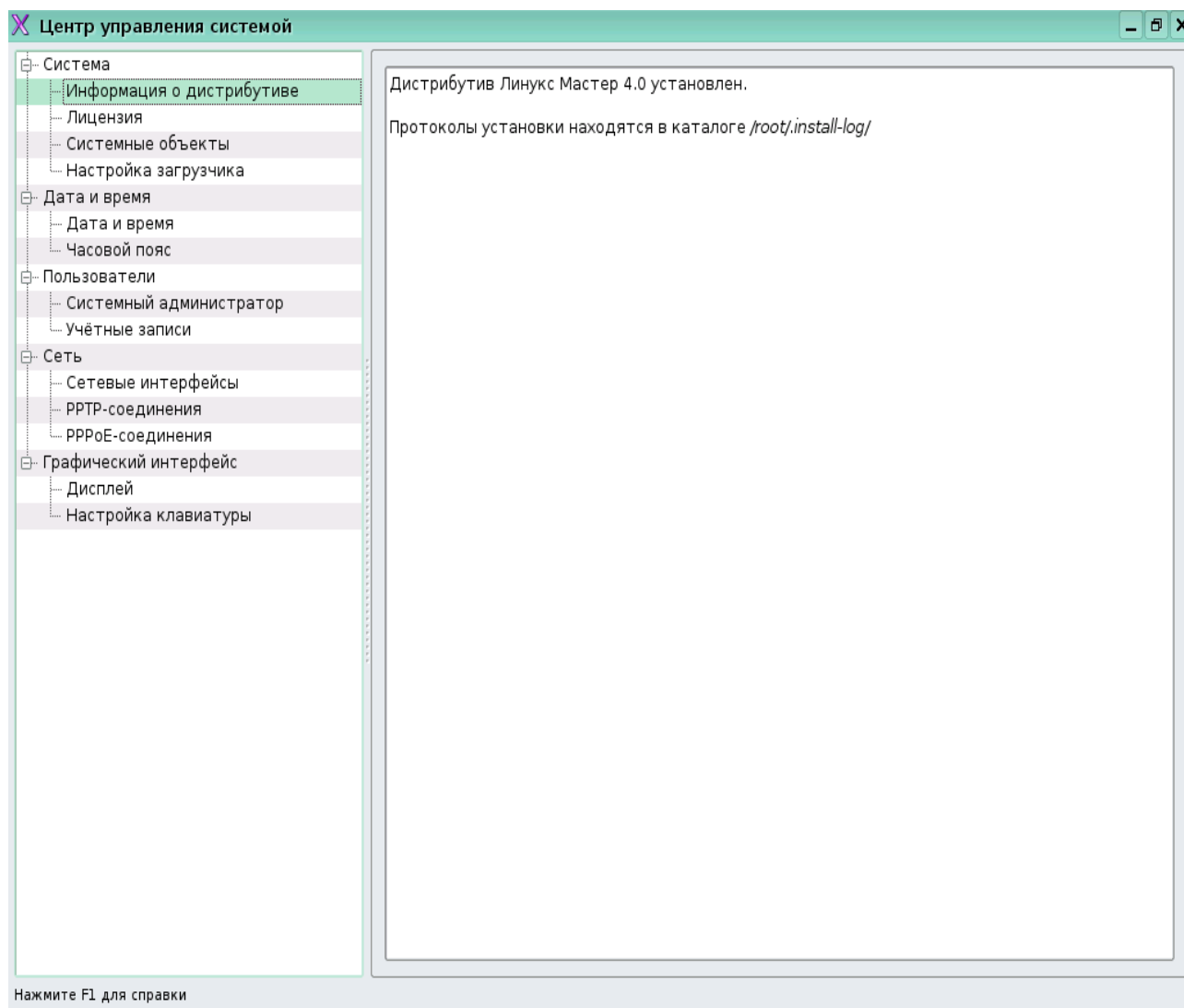
Достоинства

- Несмотря на то, что поддерживать "трехслойный" конфигурактор весьма непросто, это оказывается легче, чем при монолитном подходе. Главную часть Alterator'a традиционно поддерживают один-два человека, модули же пишутся гораздо большим количеством людей. Человеческих ресурсов при такой схеме расходуется существенно меньше, хотя модулей достаточно много.
- Главное достоинство "трехслойного" конфигуратора - независимость уровней. Для нижней пары уровней мы это уже видели, а для верхней пары это следует, к примеру, из существования разных пользовательских интерфейсов. В Alterator'e, как мы видели, есть возможность использования form-based interface (FBI) - это уровень представления для Web. При создании Alterator'a была возможность вставки еще одного промежуточного уровня, но от этой идеи отказались, так как в Qt и, допустим, HTML (или тексте) слишком разные способы представления данных и команд. Хорошим технологичным примером в этом смысле служит Debconf, но у него нет центрального слоя, управления логикой (он ориентирован на инструмент, а не на задачу). В RHEL же и Fedora используется целое семейство конфигураторов с различными типами интерфейсов.

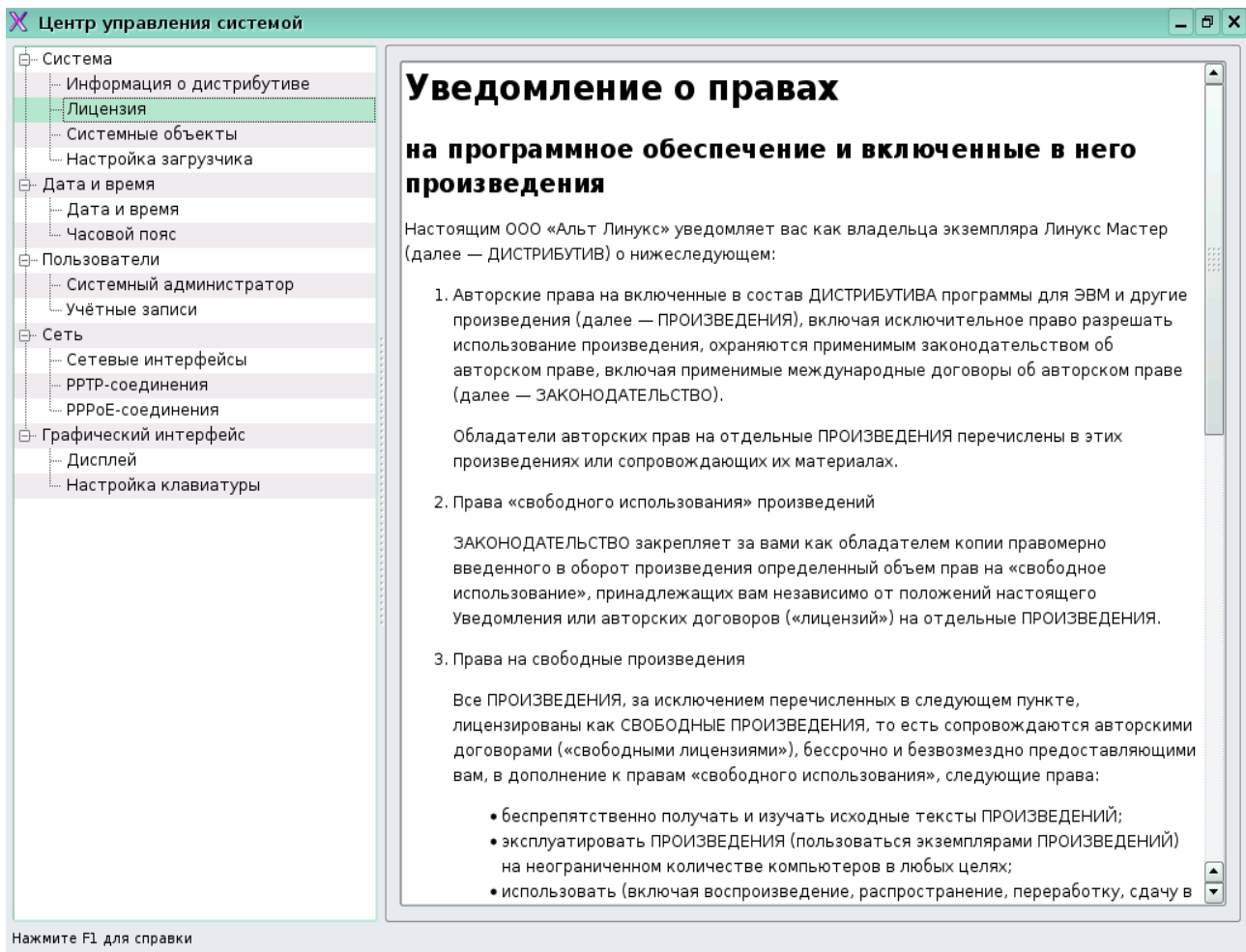
Настройка с использованием Центра управления

Как уже говорилось, в ПСПО используется модульный конфигуратор под названием Alterator, причем он используется как при установке системы, так и для настройки уже установленной системы. Qt-интерфейс этого конфигулятора можно вызвать из меню KDE (Настройка -> Центр управления системой). Для доступа к web-интерфейсу нужно подключиться по протоколу HTTPS к порту 8080 компьютера, набрав в адресной строке браузера <https://localhost:8080>, и пройдя авторизацию.

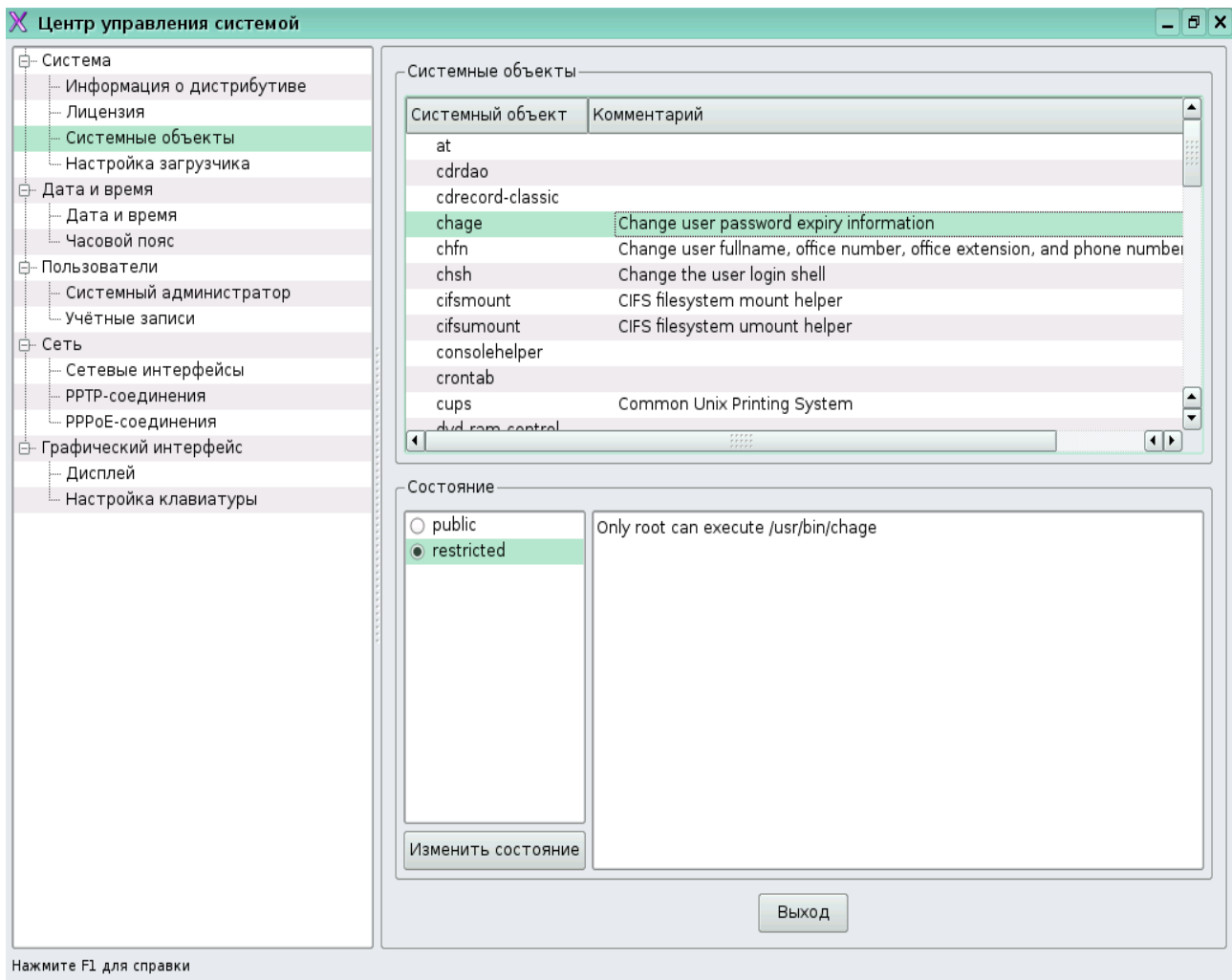
Первый пункт --- информация о дистрибутиве --- скорее информация для администратора. В случае успешной установки, выбрав этот пункт, можно увидеть следующее:



Следующий пункт --- специально написанная свободная лицензия. Это проверенная юристами лицензия, которая улаживает, среди прочего, проблемы со множественными лицензиями внутри дистрибутива (например, в дистрибутив входят несвободные драйверы для некоторых устройств). Эта лицензия замечательна с точки зрения указания прав, передаваемых пользователям, и снимает множество юридических вопросов, связанных с распространением и использованием ПСПО. С дисками дистрибутива распространяется бумажная копия этой лицензии.



Пункт "Системные объекты" - интерфейс к системе control, специфической для ПСПО и OWL.



В каждом пункте есть переключатель на несколько положений (которые описываются в соответствующем сценарии в каталоге `/etc/control.d/facilities`), например, приложением `cdwriter-classic` пользоваться могут либо члены группы `cdwriter`, либо только `root`, сервер печати `cups` может быть сетевым сервером или локальным и т.п. Если переключиться в консоль и дать команду `control`, то можно увидеть то же самое.

```
[root@demo ~]# control
at                public          (public restricted atdaemon)
cdrdao            public          (public restricted)
cdrecord-classic public          (public restricted)
chage             restricted      (public restricted)
chfn             restricted      (public restricted)
chsh             restricted      (public restricted)
cifsmount        wheelonly      (public wheelonly restricted)
cifsumount        wheelonly      (public wheelonly restricted)
consolehelper    public          (public wheelonly restricted)
crontab          public          (public restricted)
cups             server          (server local)
<...>
```

Рассмотрим эту систему подробнее. Возьмем случай службы CUPS. Есть два способа функционирования сервера CUPS. Для перевода из одного режима в другой, необходимо раскомментировать/закomentarить одну строку в конфигурационном файле. Это можно увидеть в файле `/etc/control.d/facilities/cups`, который является обычным сценарием на Shell:

```
. /etc/control.d/functions
```

```
CONFIG=/etc/cups/cupsd.conf
```

```
new_subst server \  
    '^Listen[[:space:]]+localhost:631[[:space:]]*$' \  
    's/^#\ (Listen[[:space:]]+\ localhost:631\)[[:space:]]*$/#\1/g'  
new_subst local \  
    '^#Listen[[:space:]]+localhost:631[[:space:]]*$' \  
    's/^#\ (Listen[[:space:]]+\ localhost:631\)[[:space:]]*$/#\1/g'
```

```
new_summary "Common Unix Printing System"  
new_help local "Only local utilities can work with cups"  
new_help server "External IPP interface are available for user"
```

```
control_subst "$CONFIG" "$*"
```

Система control, по сути --- API для сценариев на языке shell, которая реализует различные функции, например, поиск с заменой в конфигурационных файлах, изменение прав на файлы, а также позволяет определять для своих действий описания, используемые в конфигураторе. Так как используются сценарии shell, решается также вопрос свёртки множественных действий в атомарное. Благодаря этому, система control предоставляет достаточно удобные средства для управления работой сервисов и ограничением доступа пользователей к некоторым программам и/или устройствам.

Пункт "Настройка загрузчика":

Центр управления системой

Система

- Информация о дистрибутиве
- Лицензия
- Системные объекты
- Настройка загрузчика

Дата и время

- Дата и время
- Часовой пояс

Пользователи

- Системный администратор
- Учётные записи

Сеть

- Сетевые интерфейсы
- PPPoE-соединения
- PPPoE-соединения

Графический интерфейс

- Дисплей
- Настройка клавиатуры

Установить

Устройство	Размер	Описание
hda	6442 MB	Главная загрузочная запись

Показать разделы

Экспертные настройки

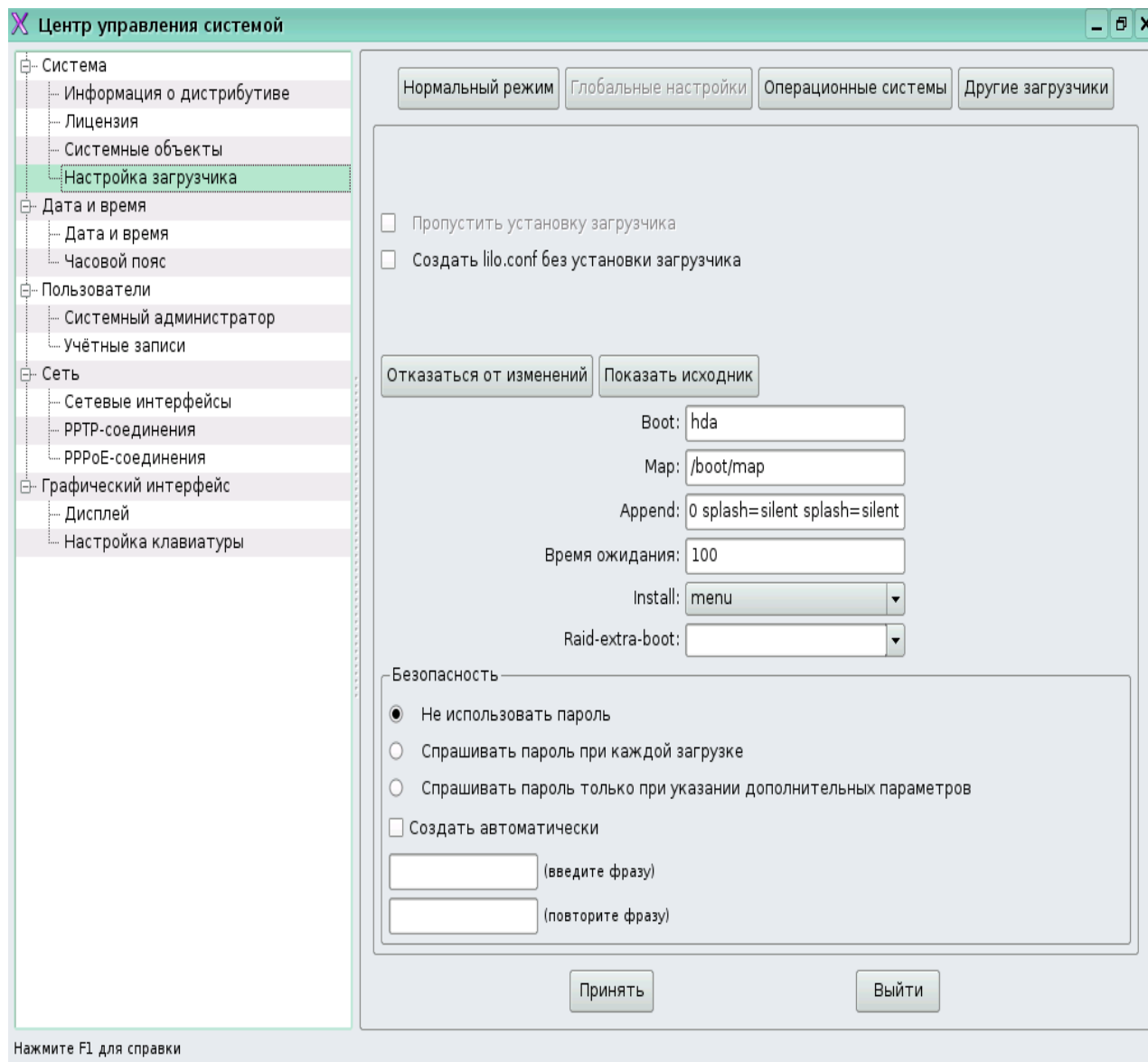
Принять

Выйти

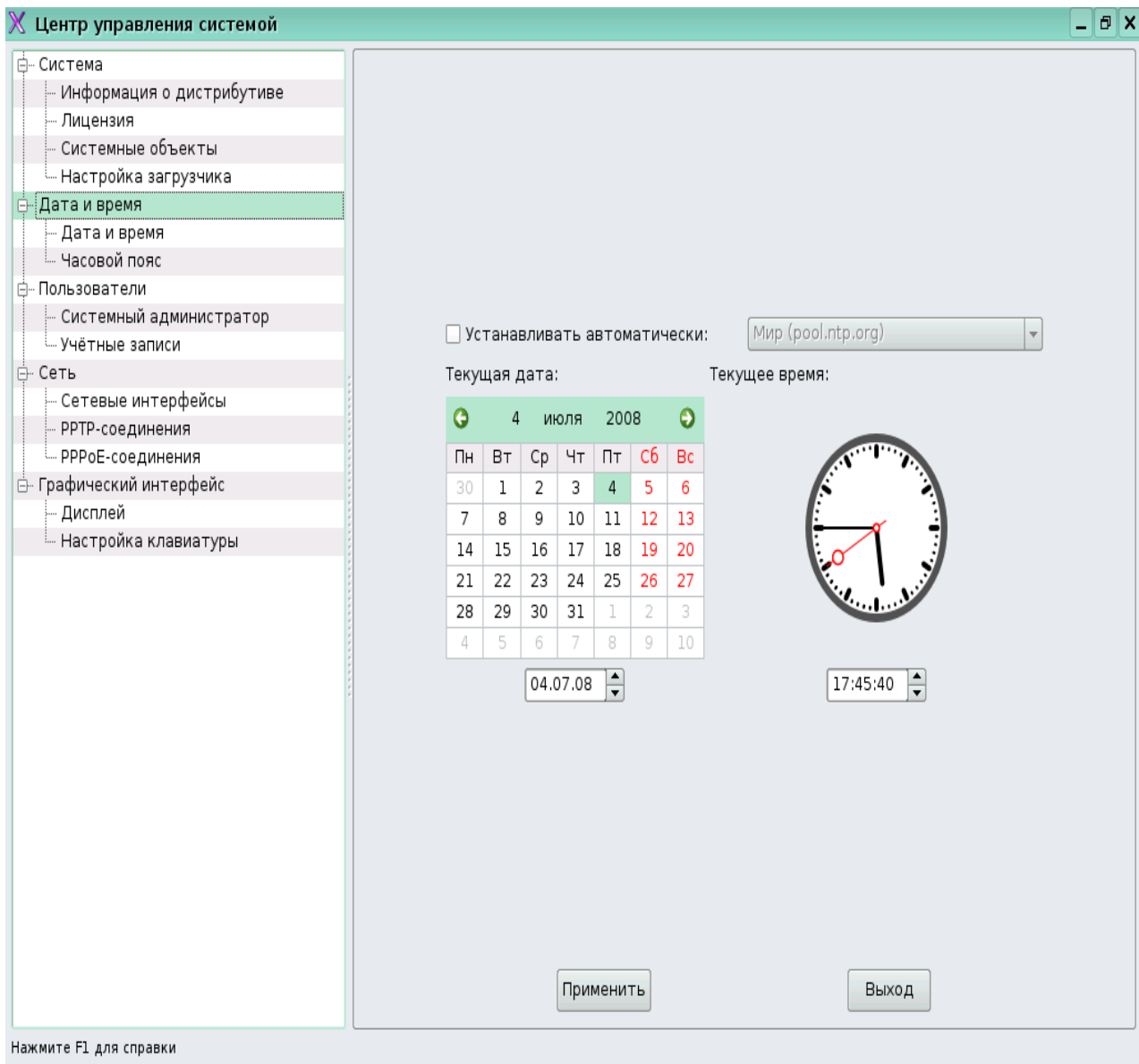
Нажмите F1 для справки

В лекциях несколько раз говорилось про последовательность загрузки. В этом пункте речь идёт о установке начального загрузчика LILO. Этот загрузчик по умолчанию устанавливается в главную загрузочную запись (MBR) диска для того, чтобы загрузчик LILO мог управлять загрузкой разных ОС. Изменить эту настройку нужно, если одновременно установлены несколько ОС и уже настроена их загрузка.

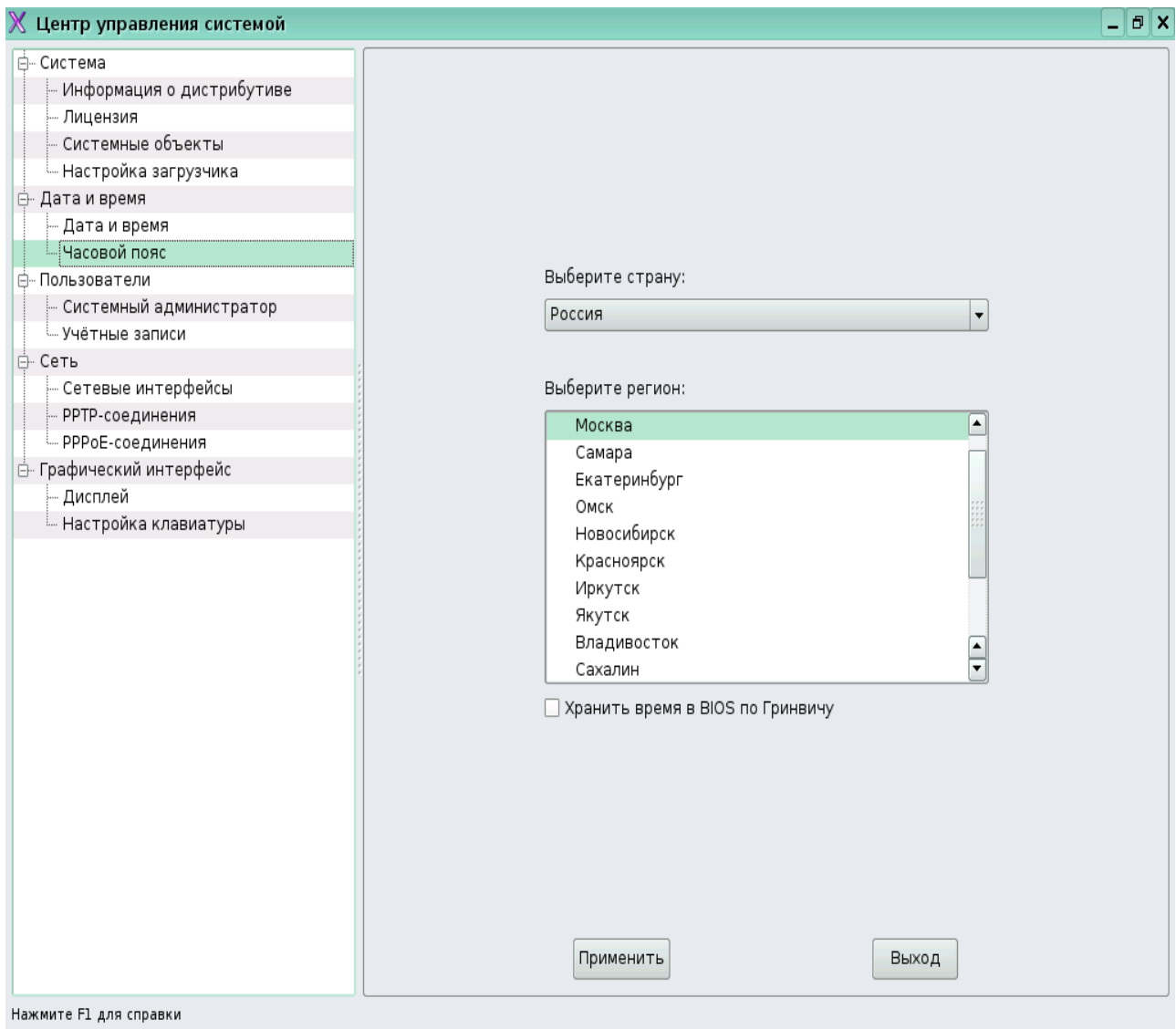
Экспертный конфигуризатор --- фактически, редактор файла lilo.conf. Для эффективного его использования нужно изучить документацию по загрузчику lilo.



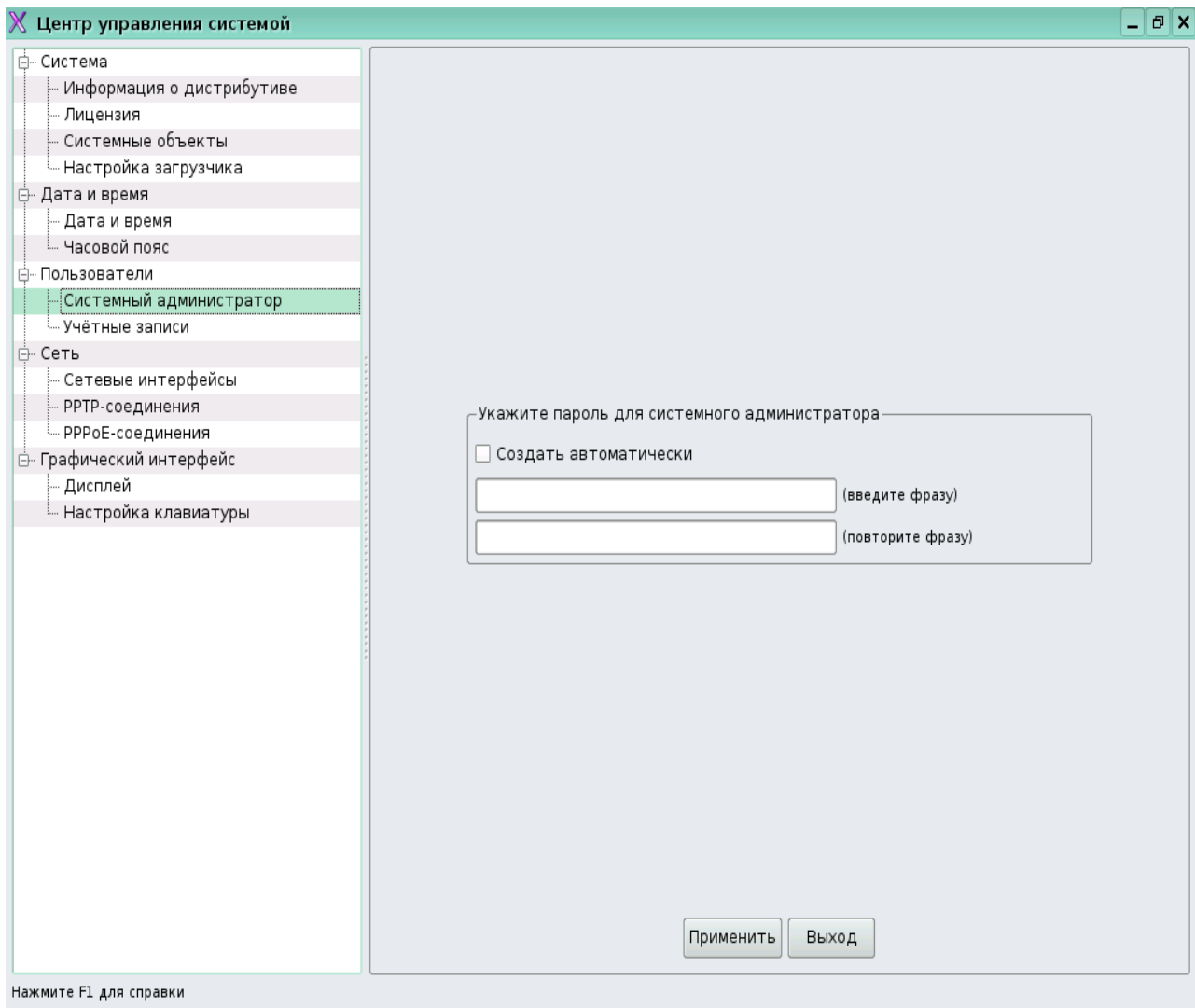
Пункт "Дата и время". Кроме собственно указания даты и времени здесь можно настроить автоматическую синхронизацию с одним из серверов времени в Интернете. Если у системы нет постоянного подключения к Интернету, то использовать эту настройку не нужно.



Пункт "Часовой пояс". Переключатель "Хранить время в BIOS по Гринвичу" устанавливает, будут ли внутренние часы компьютера установлены на время по Гринвичу. Обычно в Linux-системах это так и есть. У этого подхода есть преимущества по сравнению с хранением локального времени, например, при смене зимнего и летнего времени, единственное, что надо сделать --- прибавлять не 3 часа, а 4, и это может быть сделано без изменения времени в часах BIOS.



Пункт "Системный администратор". Предлагается задать пароль пользователя root, можно его сгенерировать. Сгенерированный пароль состоит из 3 английских слов, разделенных случайными знаками препинания. Такой пароль можно считать достаточно надёжным, поскольку его проблемно подобрать.



Пункт "Учётные записи". Здесь можно создавать и удалять пользователей, а также изменять информацию о пользователях. Системное имя (на английском оно называется login name) --- это имя пользователя, используемое при авторизации его в системе. Оно должно состоять из маленьких латинских букв, цифр и знаков подчеркивания. В поле "Комментарий" обычно вписывают полное имя пользователя. Пользоваться этим пунктом конфигуратора в большинстве случаев удобнее, чем консольными командами `adduser` или `useradd`, так как пользователь добавляется ещё в некоторые группы, например, `audio`.

- Система
 - Информация о дистрибутиве
 - Лицензия
 - Системные объекты
 - Настройка загрузчика
- Дата и время
 - Дата и время
 - Часовой пояс
- Пользователи
 - Системный администратор
 - Учётные записи
- Сеть
 - Сетевые интерфейсы
 - PPTP-соединения
 - PPPoE-соединения
- Графический интерфейс
 - Дисплей
 - Настройка клавиатуры

user

Системное имя: user

Комментарий:

Возможность переключения на администратора

Пароль:

Создать автоматически

(введите фразу)

(повторите фразу)

Редактировать

Новый

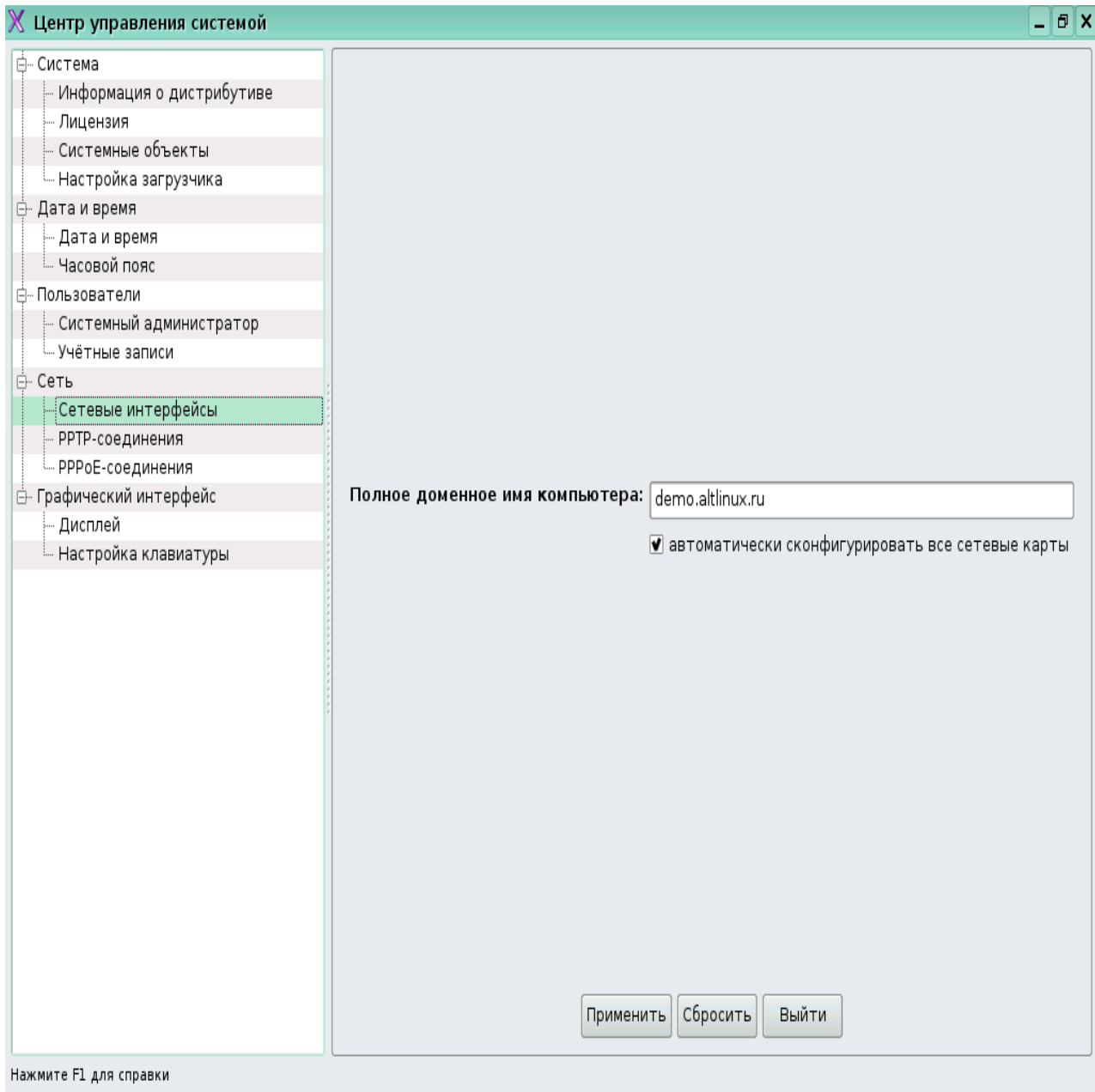
Удалить

Выход

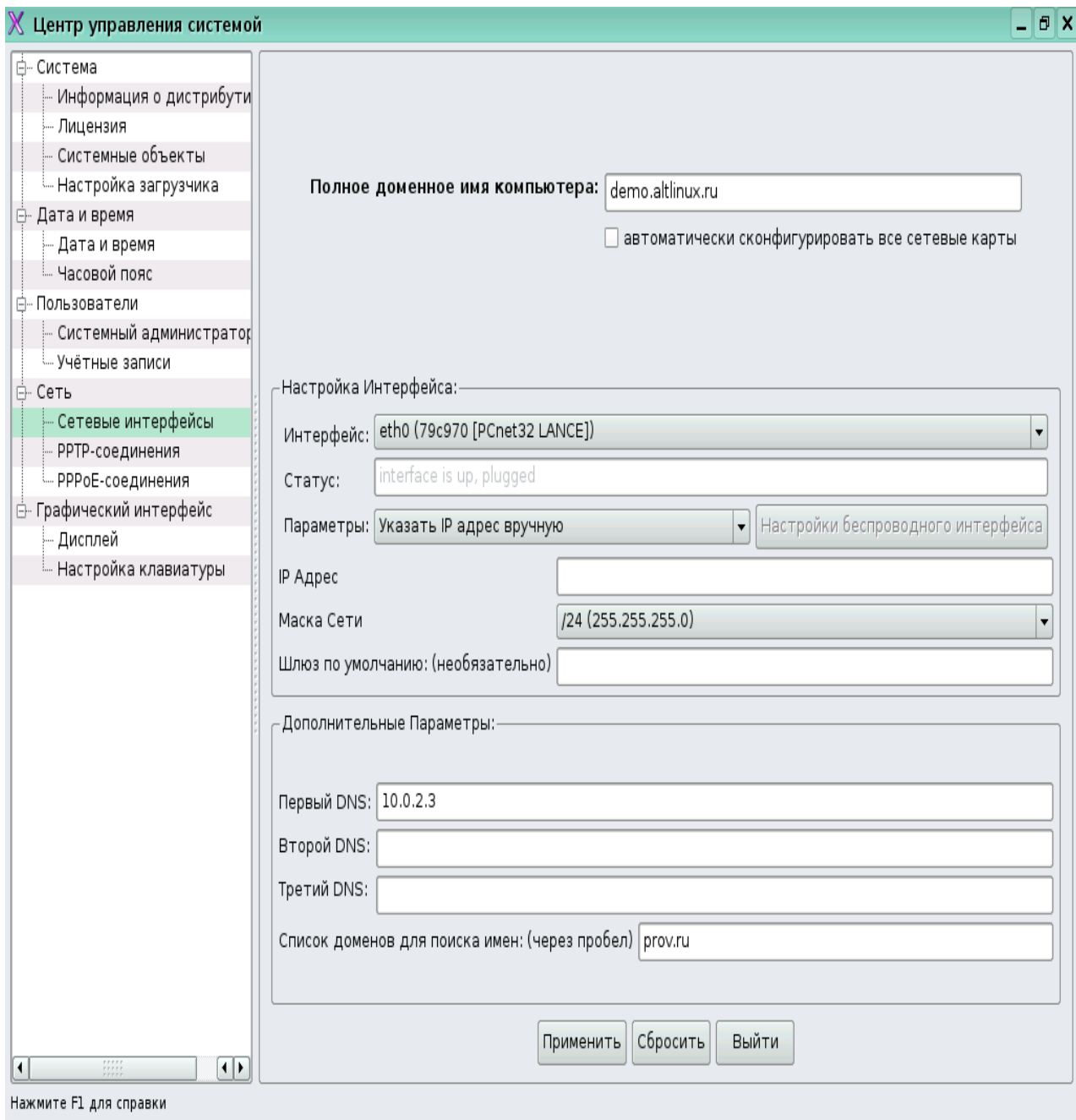
Настройка с использованием Центра управления: продолжение

Сетевые интерфейсы

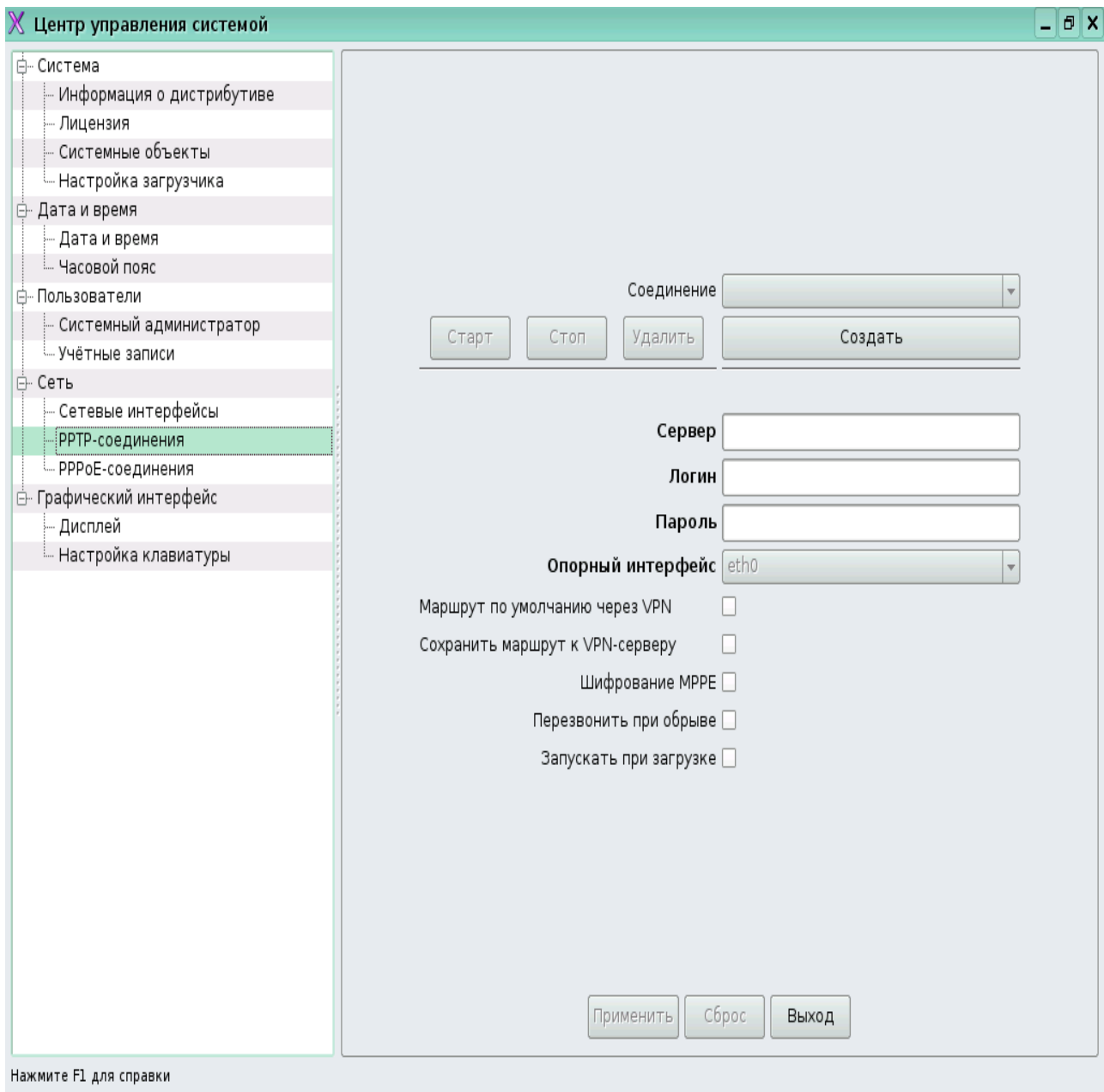
Существует переключатель автоматической настройки, то есть можно настроить сеть как автоматически (включить получение настроек по DHCP), так и вручную.



Дополнительные параметры. Фактически, это редактор нескольких файлов в `/etc/net`. Обратите внимание: `zerosnf` (один из вариантов выбора в "Настройка интерфейса -> Параметры") даёт такой IP-адрес, чтобы он не мешал другим компьютерам в сети (определяя его автоматически).



Стоит отметить, существует несколько разных мнений по поводу того, как называть то, что в Windows принято называть VPN (виртуальная частная сеть). Причина разногласий заключается в существовании очень большого количества способов организовать виртуальную частную сеть. Задача состоит в том, чтобы организовать защищённую среду передачи данных (СПД). В СПД разные компьютеры в интернете ведут себя так же, как машины, находящиеся в одной локальной сети. Тем не менее, до последнего времени так в Windows назывался исключительно PPTP с определёнными настройками. Этот шаблон в нашем дистрибутиве по умолчанию и реализован.



Рассмотрим архитектуру PPTP. Здесь используется всего два или три составляющих:

- GRE. Это протокол туннелирования любых пакетов в интернете на уровне IP. Когда на этом уровне передаётся пакет, про него ничего нельзя сказать, кроме того, что это пакет протокола GRE. Доставка его гарантирована. Тем самым мы организуем нечто вроде локальной сети: мы берём пакет, отправляем его на некоторый адрес, который находится где-то далеко. В результате того, что это виртуальная сеть, при отсылке пакет режется на куски, пакуется в GRE пакет и доходит по интернету до адресата, где он разворачивается и выясняется, что это, например, обычный IP пакет, который предназначен для данного хоста, и дальше с ним поступают как обычно.
- Данные, которые отправляются с одного конца GRE туннеля на другой, представляются как поток данных. Поверх этого запускается обычный PPP-сервер и PPP-клиент. После того, как выясняется, что пакеты GRE доставляются, организуются два виртуальных интерфейса --- это и делает PPP. Он организует IP в любой среде. Таким образом, мы создали СПД --- канал передачи информации с двумя концами. Существует также комплекс субпротоколов внутри PPP, что позволяет обмениваться данными, в том числе и настройками сети. Ограничивать доступ к данным для "посторонних" позволяет аутентификация. Это встроено в PPP по умолчанию.

- Можно использовать шифрование, чтобы передача данных было более надежной. Обычно в таких случаях используется TLS.

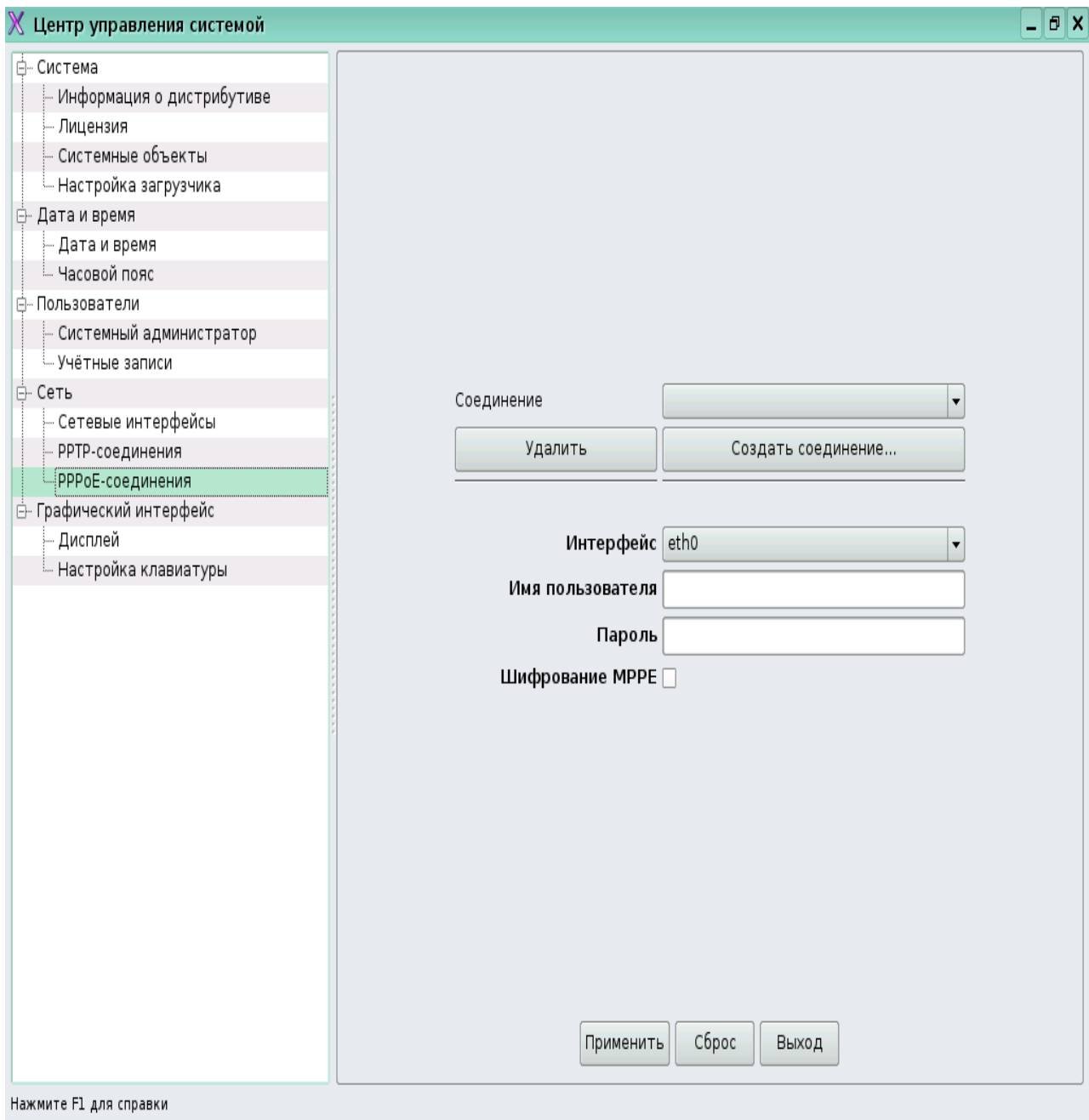
Однако PPTP имеет и недостатки. Во-первых, задача максимизировать надежность и одновременно заставить PPTP корректно работать с машинами с Windows представляется крайне сложной. Во-вторых, PPTP весьма разнообразный, и в GNU/Linux, пользователю зачастую приходится самостоятельно подбирать правильные настройки.

Следующий шаг --- задание дисциплины оформления туннеля. Итак, установлен туннель между двумя машинами, PPP подключение, сервер сообщает, что теперь маршрут по умолчанию проходит через него. Потом он организует это подключение с помощью GRE, тем самым создавая второй виртуальный интерфейс. На этом этапе сервер начинает всё посылать через виртуальный интерфейс. После чего функционировать перестанет всё, в том числе и GRE --- так как пропадут правила маршрутизации через тот интерфейс, через который создавался канал. Для того, чтобы избежать подобной ситуации, существует кнопка "сохранить маршрут". Однако это не является решением всех возникающих проблем, например, в случае, если Ваш интернет-провайдер вместо адреса VPN сервера выдает его доменное имя, а dns, скажем, находится в другом месте, куда нужно попасть вручную.

Последняя версия Windows рекомендует использовать вместо PPTP L2TP, поскольку он умеет всё вышеуказанное, но проблема заключается в том, что собственно под Windows оно не всегда работает корректно. Под свободными же системами L2TP работает нормально. Он хорош тем, что он - часть ipv6.

Рассмотрим ещё один организовать VPN. Существует openvpn: есть клиент и сервер. Под Windows он так же прост в настройке, как пресловутый "мастер подключения к интернету", за исключением того, что его (openvpn) надо ставить. Ещё один вариант - ipsec. Работает стабильно, особенно если VPN настраивается внутри большой сети, но не в интернете, потому что здесь его могут порезать, а внутри большой сети - нет. Кроме этого, существует много других способов.

Рассмотрим теперь архитектуру PPPoE. В локальных сетях, где сервер, осуществляющий связь с интернетом, находится в той же СПД, что и клиент, всё проще. Происходит раздача PPP через ethernet, то есть ещё на один уровень ниже. На каждое соединение создаётся интерфейс PPP с соответствующим номером. Главное, что надо знать --- оба протокола сводятся к тому, что каким-то одним способом организуется туннель, а потом поверх него поднимается PPP.



Дисплей

Еще пять лет назад одной из самых сложных составляющих общей настройки системы являлась настройка графической подсистемы. С тех пор она сильно упростилась, но одновременно с упрощением несколько уменьшилась вероятность успешного завершения процесса.

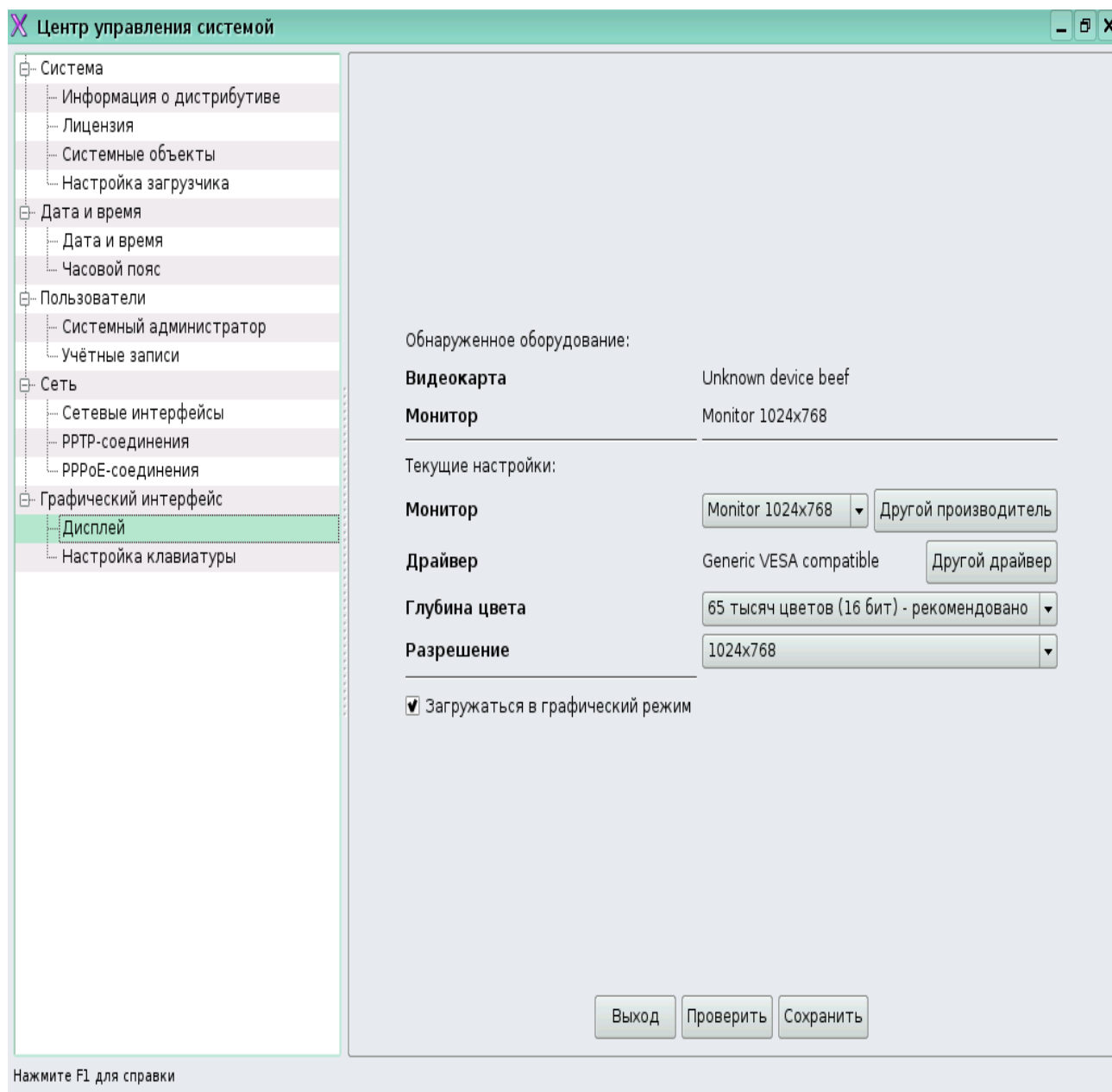
Графическая подсистема состоит из двух достаточно сложных в настройке устройств --- видеокарты и монитора. Настроить --- значит привести в соответствие их настройки. Когда речь идёт о способностях монитора (имеется в виду ЭЛТ, а не ЖК), различают три характеристики:

- dot clock - количество точек в секунду, которое можно отобразить на экране;
- частота развёртки - скорость, с которой рисуется одна линия на мониторе;
- частота кадров - частота мерцания монитора.

Эти три параметра друг от друга не зависят. Более того, у карточек тоже есть свой dot clock.

Зачастую эти частоты (как у карточек, так и у некоторых мониторов) фиксированные. Кроме того, есть несколько протоколов (например, EDID), согласно которым монитор сообщает видеокарте свои настройки, и эти протоколы правильно функционируют примерно в 80 % случаев. Наиболее сложной является ситуация, когда монитор сообщает карте завышенные данные о себе. Для таких случаев существует база данных по мониторам, в которой прописаны различные параметры, в частности:

- Разрешение --- количество пикселей на экране, в ширину и длину; есть несколько разных родов мониторов, которые умеют поддерживать то или иное разрешение.
- Глубина цветности. В зависимости от этого параметра один пиксель может представляться одним байтом, двумя байтами и так далее. Чем больше глубина цветности, тем больше требуется видеопамати. Стоит отметить, что современные видеокарты имеют очень много видеопамати, но если видеокарта не новая, то глубина цветности может быть ограничена возможностями видеокарты.

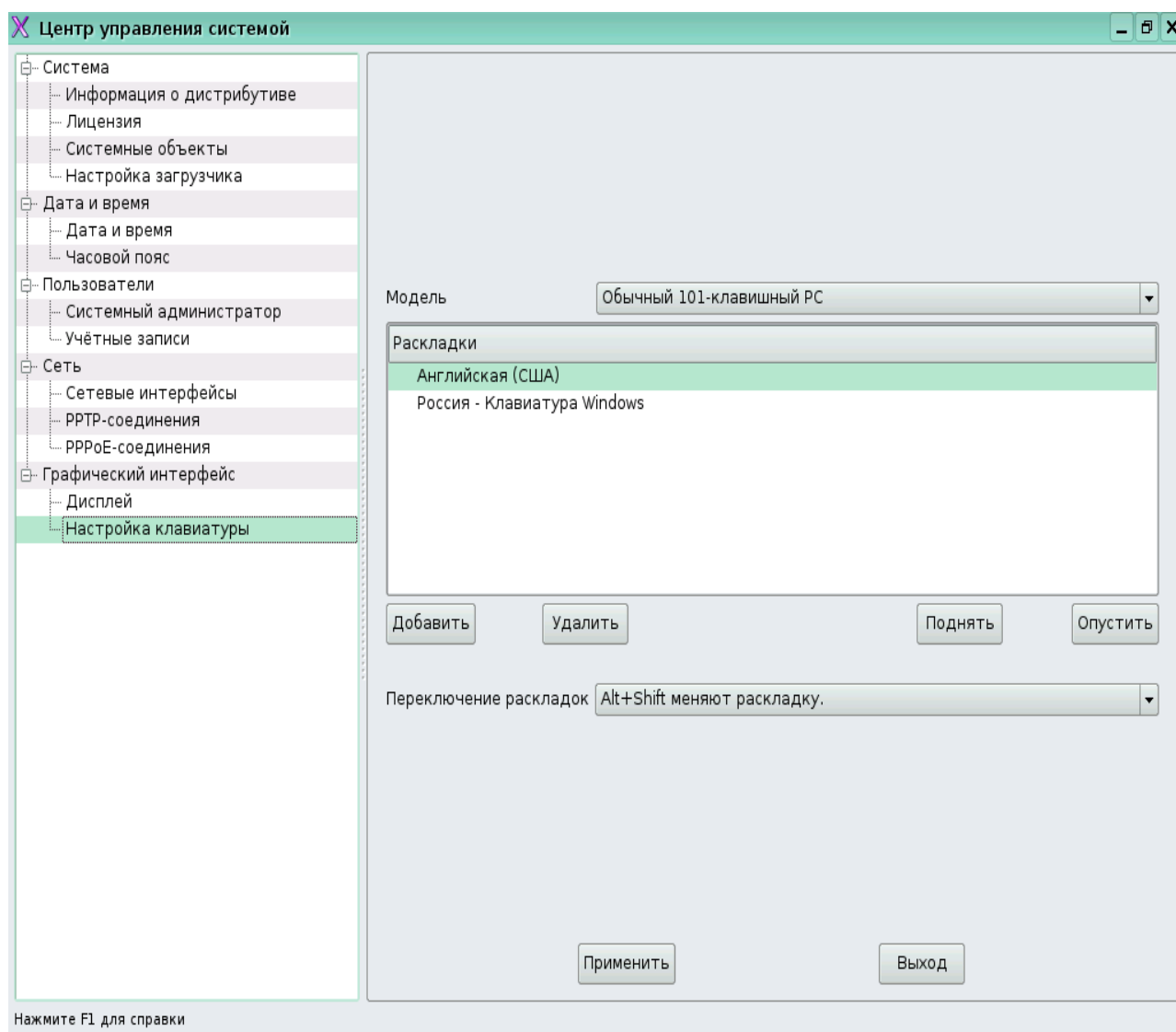


Кнопочка "загружаться в графический режим" включает/выключает kdm. Это необходимо для того, чтобы сделать из машины сервер, на котором нет надобности в графической подсистеме, и в случаях, когда графическая подсистема работает нестабильно.

Что касается драйверов, то они, как правило, определяются автоматически, однако в случае, если у вас очень старая, очень новая или непонятная системе видеокарта, нужно попробовать использовать VESA, однако функциональность в таком случае может быть ограничена. Добавим, что если у Вас очень новая видеокарта, то возможно, что после обновления видеокарта будет успешно определена системой.

Клавиатура

В нашем дистрибутиве есть поддержка разных клавиатур с дополнительными клавишами. По умолчанию можно выбрать ACPI. Проблема с power-sleep-wake (на многих дешёвых клавиатурах эти кнопки располагались так, что провоцировали своё случайное нажатие и случайное выключение компьютера, ибо функциональность этих кнопок было крайне трудно изменить), наблюдающаяся в Windows 2000, в нашем случае не возникает. Вообще, чем больше клавиш распознается, тем удобнее --- можно использовать их в личных целях. Проблема состоит в том, что не для всех клавиатур есть драйверы.



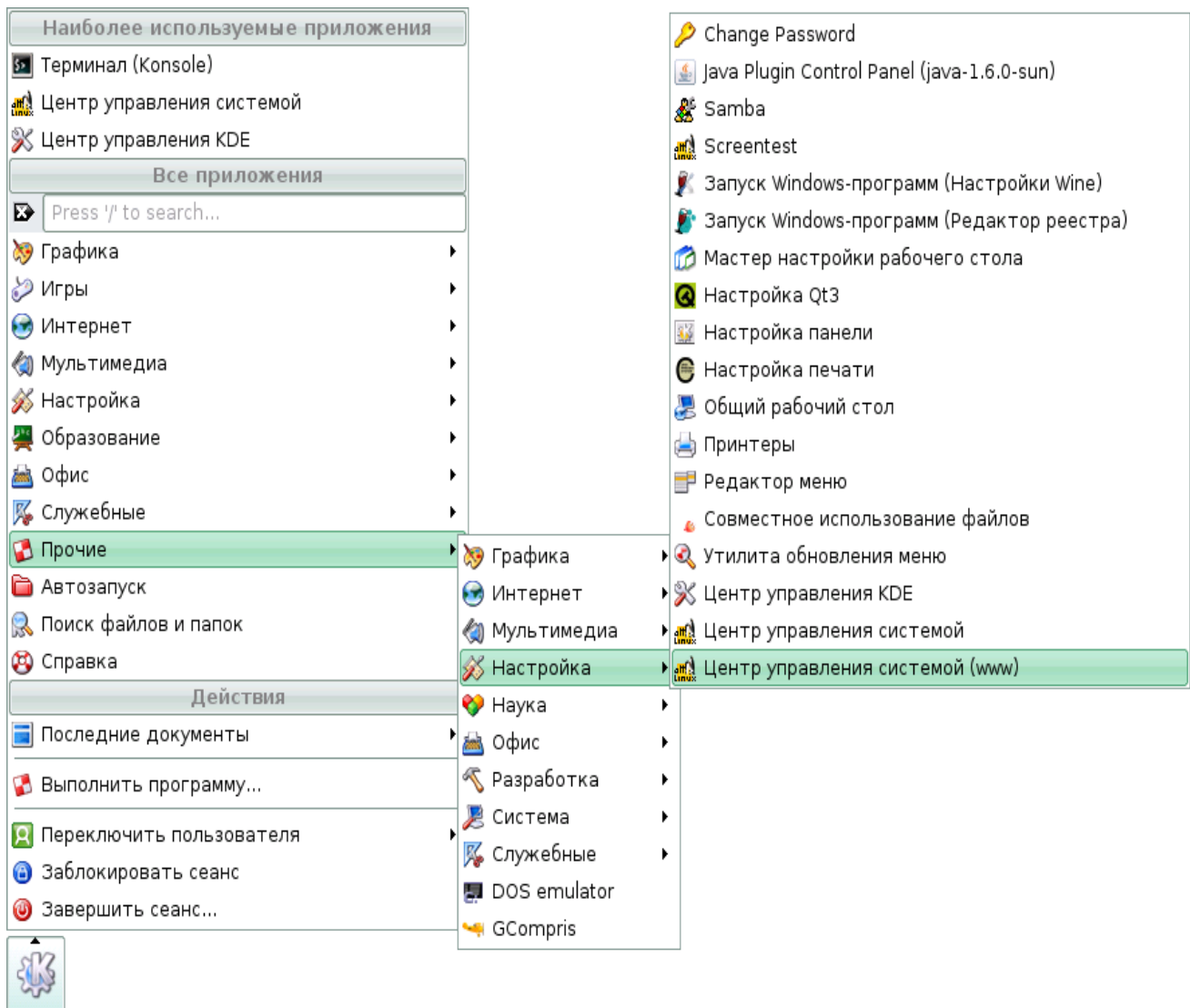
Что же касается переключения раскладок, то, в отличие от Windows предлагается много различных способов. Стоит учитывать, что стандартные alt+shift или ctrl+shift могут случайно сработать при нажатии других комбинаций клавиш с их использованием.

Web-интерфейс центра управления

Продолжим обзор настройки системы при помощи Alterator, но с другой стороны. У нас курс расширенного сетевого администрирования. По плану людей, которые его писали, это настройка служб. Если при нажатиях кнопок в том Alterator мы придерживались какой-то легенды --- оператор ЭВМ, недостаточная информированность, и так далее, то здесь подразумевается более глубокое знание предмета настройки. Прежде чем приступать к собственно изучению Web-интерфейса центра управления, рассмотрим применение Alterator в различных дистрибутивах. Alterator используется в полном объеме в Linux Master, также Web-интерфейс альтератора применяется в ALT Linux Server, наконец, он отсутствует в Junior. В ALT Linux Server, в целом, небольшое количество программных продуктов, управлять ими можно через web-интерфейс. В их число входят ряд web-служб. Напомним, что Alterator изначально был предназначен для настройки сервера.

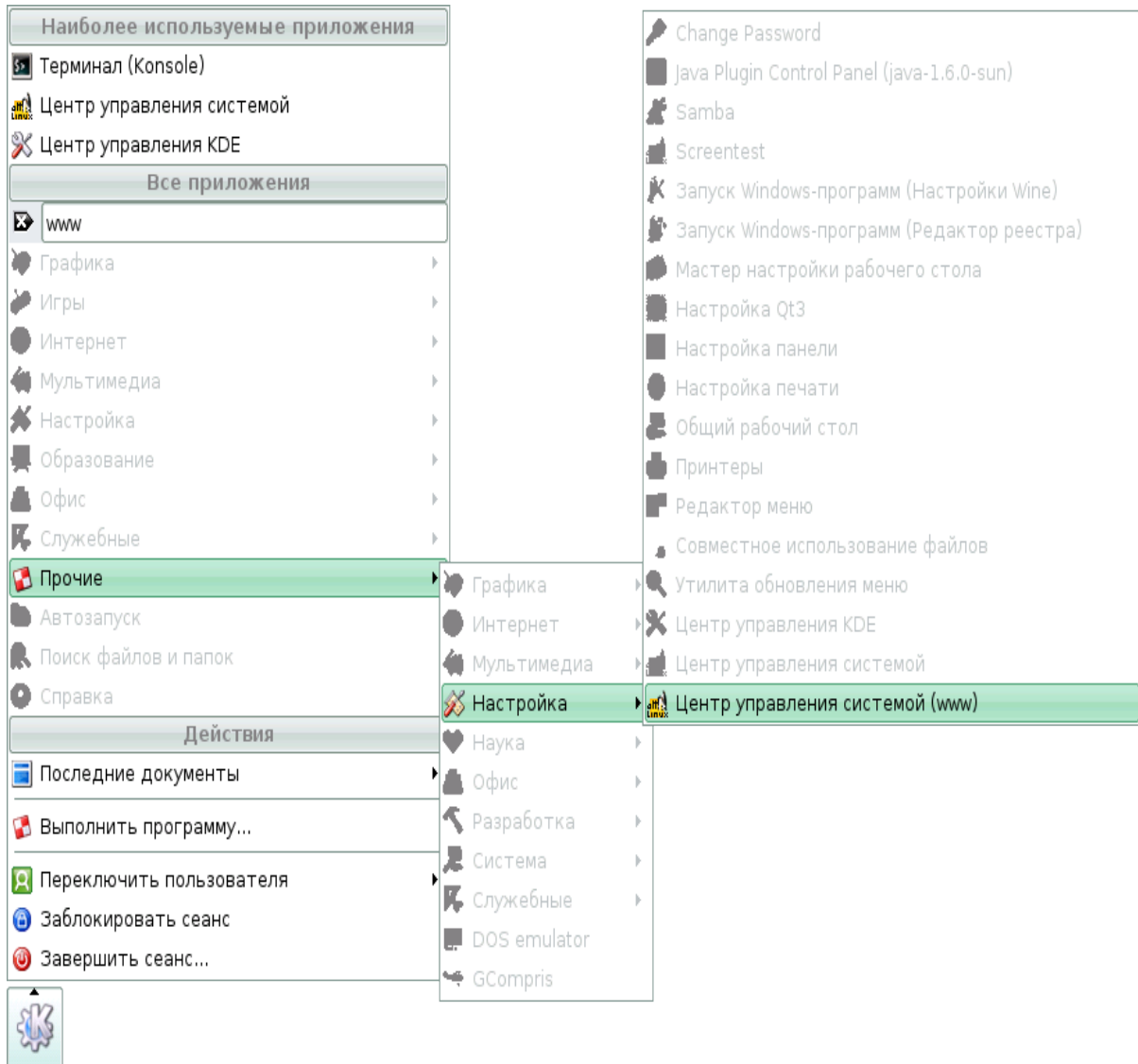
(Хотя в рамках данного проекта разрабатывается ПО для рабочей станции, не для класса, в ряд задач входят такие задачи, как фильтрация информации, межсетевые экраны и т. д.)

Итак, во-первых, определим расположение этого Web-интерфейса. Для доступа к Alterator через меню необходимо открыть *Меню* далее *Прочие* и наконец *Центр управления системой (www)* (или же, перейти в браузере по адресу <https://localhost:8080>).

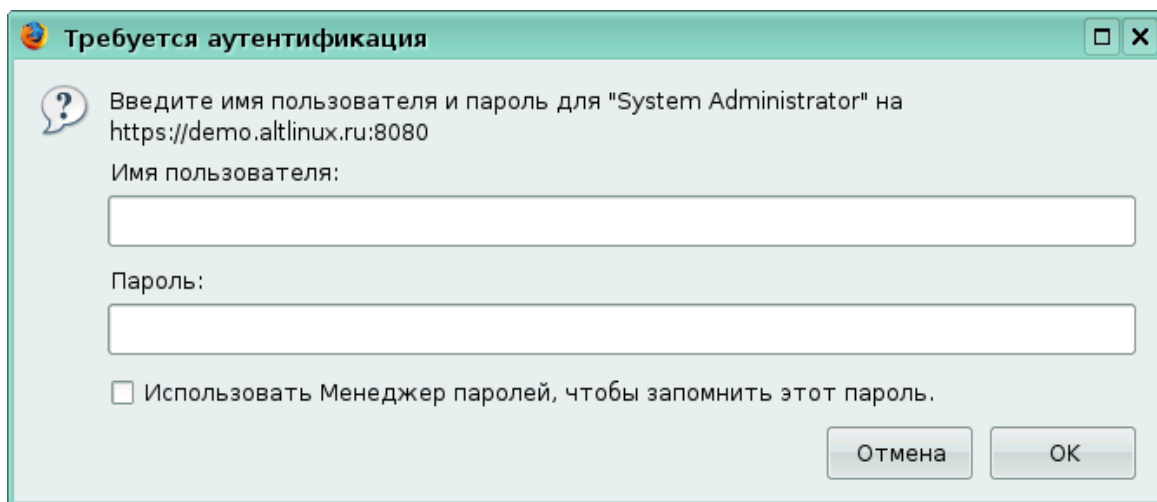


Одной из особенностей ПСПО является поиск по меню. Так, набрав в поиске "www", мы сразу увидим, как добраться до нужного пункта меню, а все остальные пункты меню будут

неактивны.

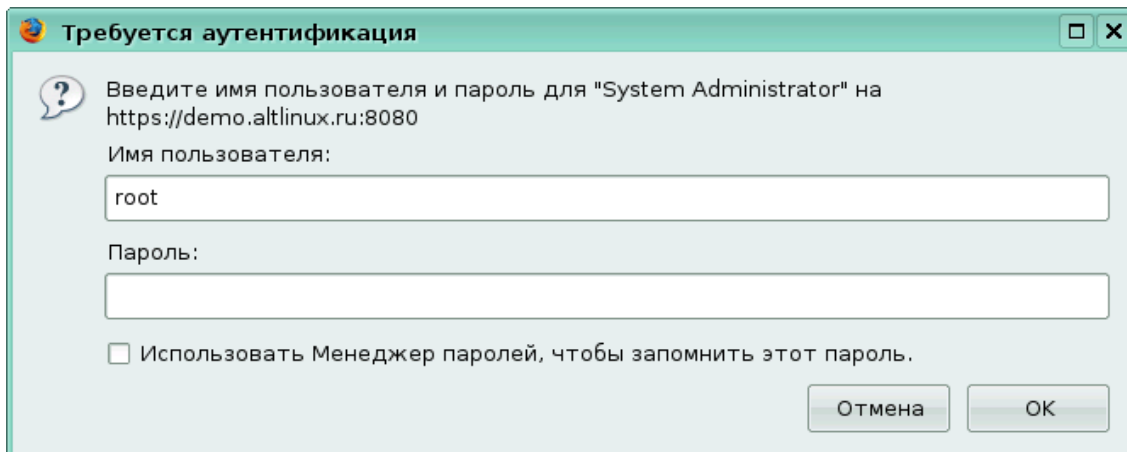


Обратите внимание, что работа с центром управления осуществляется по HTTPS.



Изначально эта технология была предназначена для управления сервером, причем не обязательно с той же машины, т.е. и по сети. Очевидно, что при входе в систему сервера по сети необходимо сделать ввод логина и пароля максимально безопасным, поскольку входить придется как суперпользователь. Для того, чтобы пароль не передавался по сети открытым

текстом, используется зашифрованное соединение. Кстати, сохранять пароль не стоит, также в целях повышения безопасности. Итак, вводим root в качестве имени пользователя и его пароль. Обратите внимание, что Центр Управления системой, который у нас запущен, работает на нестандартном порту --- 8080; некоторые серверы используют его для других целей, но и протокол там не http, а https . Логин и пароль здесь суперпользователя, а порт нестандартный, это означает, что, во-первых, попасть туда случайно очень трудно и, во-вторых, это не мешает работе других web-служб.



Требуется аутентификация

Введите имя пользователя и пароль для "System Administrator" на <https://demo.altlinux.ru:8080>

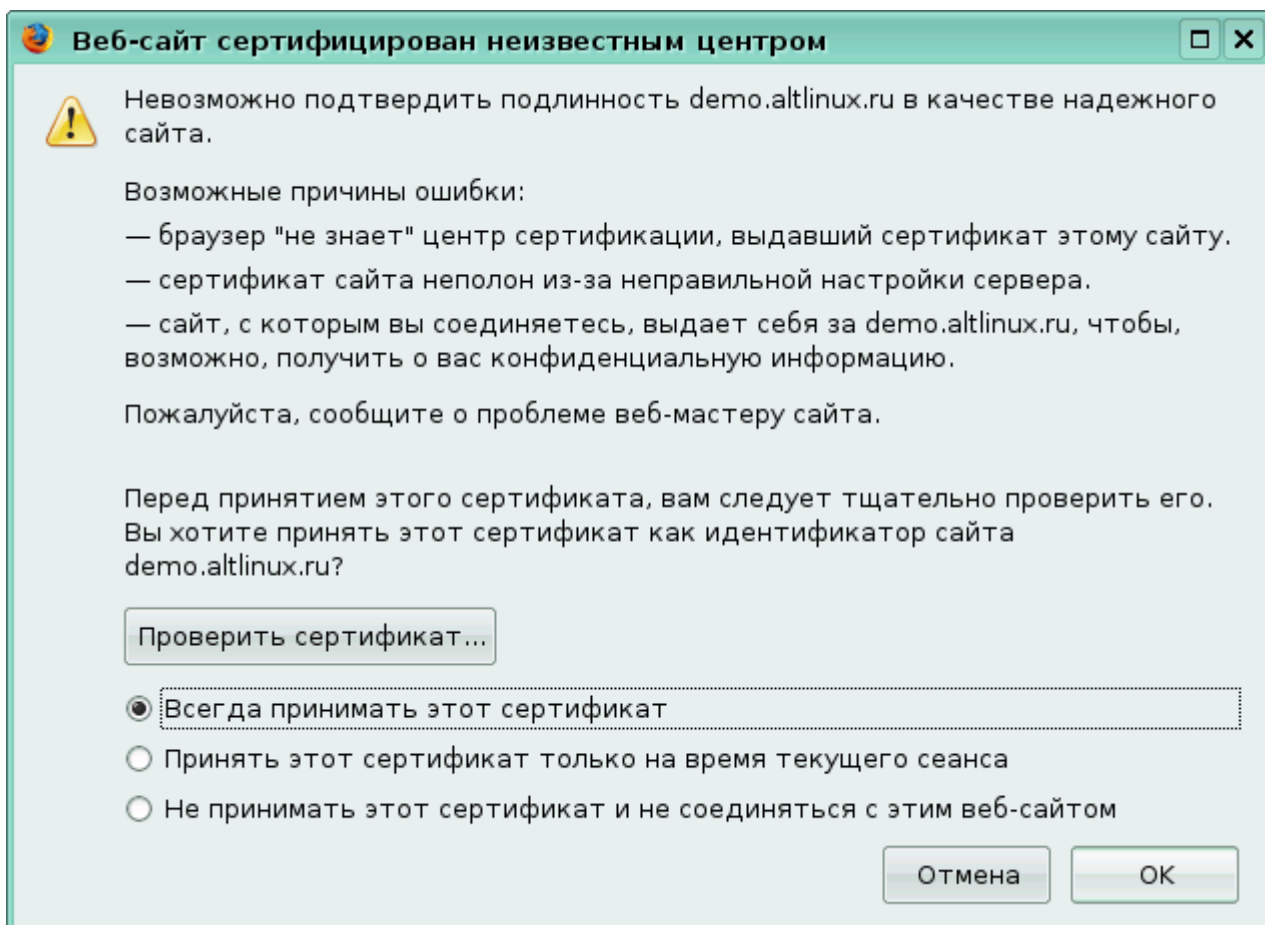
Имя пользователя:

Пароль:

Использовать Менеджер паролей, чтобы запомнить этот пароль.

Отмена ОК

Здесь появляется предупреждение о возможном нарушении безопасности, но бояться его не нужно. Это предупреждение о том, что сертификат этого компьютера, подтверждающий то, что он действительно этот компьютер, подписан им самим, а не в сертификационном центре.



Веб-сайт сертифицирован неизвестным центром

Невозможно подтвердить подлинность demo.altlinux.ru в качестве надежного сайта.

Возможные причины ошибки:

- браузер "не знает" центр сертификации, выдавший сертификат этому сайту.
- сертификат сайта неполон из-за неправильной настройки сервера.
- сайт, с которым вы соединяетесь, выдает себя за demo.altlinux.ru, чтобы, возможно, получить о вас конфиденциальную информацию.

Пожалуйста, сообщите о проблеме веб-мастеру сайта.

Перед принятием этого сертификата, вам следует тщательно проверить его. Вы хотите принять этот сертификат как идентификатор сайта demo.altlinux.ru?

Проверить сертификат...

Всегда принимать этот сертификат

Принять этот сертификат только на время текущего сеанса

Не принимать этот сертификат и не соединяться с этим веб-сайтом

Отмена ОК

Существуют значительные различия между сертификатом, который утверждён в сертификационном центре, и сертификатом, который не утверждён (самоподписанным). На самом деле, неизвестно, заходите вы на сайт свой или незнакомый, который может

подглядеть Ваш пароль. Чтобы быть полностью быть уверенными в том, что Вы заходите туда, куда надо, необходимо, чтобы сертификат был подписан в сертификационном центре. Однако сертификат стоит денег, и если кто-то действительно захочет взломать сеть при доступе к ней, то он сможет подменить корневые сертификаты (это не так просто сделать, но в общем случае возможно).

В нашем случае следует выбрать пункт "всегда принимать этот сертификат". Тогда при следующем подключении сам факт, что произошло соединение по https и предупреждение о возможном нарушении безопасности не возникает, говорит о том, что вы подключаетесь так же, как делали это раньше. Через некоторое время мы еще поговорим об этом.

ALT Linux 4.0 Server | demo.altlinux.ru [Справка](#) [en](#) [ru](#) [uk](#)

Система
[Информация о системе](#)
[Системные объекты](#)
[Обновления](#)
[Источники обновлений](#)

Дата и время
[Дата и время](#)
[Часовой пояс](#)

Пользователи
[Системный администратор](#)
[Учётные записи](#)

Сеть
[RTP-соединения](#)
[RPPoE-соединения](#)
[Сетевой экран](#)

Графический интерфейс
[Дисплей](#)
[Настройка клавиатуры](#)

Серверы
[Прокси-сервер](#)

Статистика
[Сетевой трафик](#)

Web-интерфейс
[Сервер](#)

Центр управления системы

Часть настроек здесь та же, очевидно, что для части модулей был написан другой пользовательский интерфейс. Стоит отметить, что некоторые из пунктов содержат больше настроек, чем в обычном Alterator. Например, в пункте "Информация о системе" можно узнать собственно некоторую информацию о системе:

- Система**
- [Информация о системе](#)
- [Системные объекты](#)
- [Обновления](#)
- [Источники обновлений](#)
- Дата и время**
- [Дата и время](#)
- [Часовой пояс](#)
- Пользователи**
- [Системный администратор](#)
- [Учётные записи](#)
- Сеть**
- [PPTP-соединения](#)
- [PPPoE-соединения](#)
- [Сетевой экран](#)
- Графический интерфейс**
- [Дисплей](#)
- [Настройка клавиатуры](#)
- Серверы**
- [Прокси-сервер](#)
- Статистика**
- [Сетевой трафик](#)
- Web-интерфейс**
- [Сервер](#)

Информация о системе

Версия ядра 2.6.18-std-smp-alt12.M40.1

Использование памяти

ОЗУ (всего):191448 kB ОЗУ (свободно):3388 kB
 Область подкачки (всего):262040 kB Область подкачки (свободно):261976 kB

Использование диска

точка монтирования	всего	свободно	используется	использование в процентах
/	5.7G	897M	4.5G	84%
/dev	5.0M	5.0M	60K	2%
/dev/shm	94M	94M	0	0%
/tmp	94M	94M	4.0K	1%
/media/cdrom	2.4G	0	2.4G	100%

В большинстве случаев настройки отличаются только внешним видом:

- Система**
- [Информация о системе](#)
- [Системные объекты](#)
- [Обновления](#)
- [Источники обновлений](#)
- Дата и время**
- [Дата и время](#)
- [Часовой пояс](#)
- Пользователи**
- [Системный администратор](#)
- [Учётные записи](#)
- Сеть**
- [PPTP-соединения](#)
- [PPPoE-соединения](#)
- [Сетевой экран](#)
- Графический интерфейс**
- [Дисплей](#)
- [Настройка клавиатуры](#)
- Серверы**
- [Прокси-сервер](#)
- Статистика**
- [Сетевой трафик](#)
- Web-интерфейс**
- [Сервер](#)

Часовой пояс

Хранить время в BIOS по Гринвичу

Страна

Часовой пояс

- Калининград
- Москва
- Самара
- Екатеринбург
- Омск
- Новосибирск
- Красноярск
- Иркутск
- Якутск
- Владивосток
- Сахалин
- Магадан
- Камчатка
- Анадырь

Обратите внимание, что здесь появилось несколько пунктов, которых не было в обычном Alterator. Начнем с пункта "Обновления". Перейдем туда и увидим:

- Система**
- [Информация о системе](#)
- [Системные объекты](#)
- Обновления**
- [Источники обновлений](#)
- Дата и время**
- [Дата и время](#)
- [Часовой пояс](#)
- Пользователи**
- [Системный администратор](#)
- [Учётные записи](#)
- Сеть**
- [PPTP-соединения](#)
- [PPPoE-соединения](#)
- [Сетевой экран](#)
- Графический интерфейс**
- [Дисплей](#)
- [Настройка клавиатуры](#)
- Серверы**
- [Прокси-сервер](#)
- Статистика**
- [Сетевой трафик](#)
- Web-интерфейс**
- [Сервер](#)

Обновления

Доступные обновления:

нет информации

После этого можно попробовать обновить систему:

- Система**
- [Информация о системе](#)
- [Системные объекты](#)
- Обновления**
- [Источники обновлений](#)
- Дата и время**
- [Дата и время](#)
- [Часовой пояс](#)
- Пользователи**
- [Системный администратор](#)
- [Учётные записи](#)
- Сеть**
- [PPTP-соединения](#)
- [PPPoE-соединения](#)
- [Сетевой экран](#)
- Графический интерфейс**
- [Дисплей](#)
- [Настройка клавиатуры](#)
- Серверы**
- [Прокси-сервер](#)
- Статистика**
- [Сетевой трафик](#)
- Web-интерфейс**
- [Сервер](#)

Обновления

Проверка обновлений...

Получение индексных файлов...

Однако в результате ничего не изменится. Итак, мы переходим к следующему пункту "Источники обновлений". В принципе, этот материал относится скорее к разделу

"Администрирование". Дело в том, что в разделе "Системное администрирование", мы рассказываем о том, как работает пакетный диспетчер, и где лежат его настроечные файлы и файлы, связанные с установкой и настройкой программного окружения. В принципе, никакой проблемы не составляет выполнить вручную команды `apt-get update` и `apt-get dist-upgrade`, однако в ситуации, когда большая часть системы управляется по веб-интерфейсу, то логично рассмотреть это сейчас.

■ ALT Linux 4.0 Server | demo.altlinux.ru Справка [en](#) [ru](#) [uk](#)

Система
 Информация о системе
 Системные объекты
 Обновления
Источники обновлений
 Дата и время
 Дата и время
 Часовой пояс
 Пользователи
 Системный администратор
 Учётные записи
 Сеть
 RPTP-соединения
 PPPoE-соединения
 Сетевой экран
 Графический интерфейс
 Дисплей
 Настройка клавиатуры
 Серверы
 Прокси-сервер
 Статистика
 Сетевой трафик
 Web-интерфейс
 Сервер

Источники обновлений

	Состояние	Тип	Подпись	Адрес	Архитектура	Компоненты	
<input type="checkbox"/>	включено	бинарные		cdrom:[ALT Linux 4.0 Junior DVD (HeyTeacher)]/	ALTLinux	base disk	ред
<input type="checkbox"/>	включено	бинарные	updates	ftp://updates.altlinux.org/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://mirror.yandex.ru/altlinux/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://mirror.yandex.ru/altlinux/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://mirror.yandex.ru/altlinux/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.linux.kiev.ua/pub/Linux/ALT/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.linux.kiev.ua/pub/Linux/ALT/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.linux.kiev.ua/pub/Linux/ALT/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/updates/4.0	i586	updates	ред

Выделенные:

Тип:

Архитектура:

Подпись:

Адрес:

Компоненты:

Используется схема, подобная той, что используется в Debian, с `/etc/apt/sources.list` и `/etc/apt/sources.list.d/`. То есть, имеется большой список источников обновлений, но большинство источников спрятаны в комментарии, в частности, там есть зеркала на Яндексе и многое другое. Среди них есть CD, с которого ставили систему --- он единственный включён. Если включить какие-то из других источников обновлений, например, `updates.altlinux.org/4.0.`, этот интерфейс всего лишь удалит комментарий из соответствующего файла и больше ничего, но, с другой стороны, просмотреть весь список всего и выбрать, что нужно, в web-интерфейсе гораздо удобнее.

Существуют сервера разного типа: школьные сервера (с пометкой `school`) и общие для ALT Linux (`updates`).

Речь идёт не только об обновлении системы, но и о выборе дополнительного хранилища пакетов. Мы можем установить в качестве дополнительного хранилища, откуда мы будем брать пакеты с программами, которые у нас не установлены (и обновленные пакеты) `branch`. Стоит отметить, что в хранилище, которое использовалось для подготовки дистрибутива ПСПО, существенно больше пакетов, чем в любом дистрибутиве ПСПО. Это обусловлено тем, что существует некий набор ПО, за который компания ALT Linux отвечаем перед заказчиком, есть некий набор, который мы посчитали нужным включить, чтобы обеспечить комфорт пользователя, а есть некий набор ПО, который включать или не включать --- личное дело каждого. Но при этом обеспечивать поддержку одновременно всех пакетов невозможно. С другой стороны, не хотелось включать пакеты, которые не вполне понятны.

Но все-таки давайте попробуем подключить хранилище пакетов. Выберем один или

несколько серверов и отметим их, выберем "включить" и нажмем *OK*:

- Система
 - Информация о системе
 - Системные объекты
 - Обновления
 - Источники обновлений**
- Дата и время
 - Дата и время
 - Часовой пояс
- Пользователи
 - Системный администратор
 - Учётные записи
- Сеть
 - PPTP-соединения
 - PPPoE-соединения
 - Сетевой экран
- Графический интерфейс
 - Дисплей
 - Настройка клавиатуры
- Серверы
 - Прокси-сервер
- Статистика
 - Сетевой трафик
- Web-интерфейс
 - Сервер

Источники обновлений

<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.altlinux.org/pub/distributions/ALTLinux/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.altlinux.org/pub/distributions/ALTLinux/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/branch	i586	classic	ред
<input checked="" type="checkbox"/>	выключено	бинарные	updates	ftp://updates.altlinux.org/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	ftp://mirror.yandex.ru/altlinux/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	ftp://mirror.yandex.ru/altlinux/4.0/branch	noarch	classic	ред

Выделенные:

Тип:

Архитектура:

Подпись:

Адрес:

Компоненты:

Выбранные сервера станут включенными, они отобразятся в верхней части списка. Теперь можно попробовать обновиться. Появились доступные обновления, теперь можно нажать "обновить систему".

- Система
- Информация о системе
- Системные объекты
- Обновления**
- Источники обновлений
- Дата и время
- Дата и время
- Часовой пояс
- Пользователи
- Системный администратор
- Учётные записи
- Сеть
- PPTP-соединения
- PPPoE-соединения
- Сетевой экран
- Графический интерфейс
- Дисплей
- Настройка клавиатуры
- Серверы
- Прокси-сервер
- Статистика
- Сетевой трафик
- Web-интерфейс
- Сервер

Обновления

Доступные обновления:

Следующие пакеты будут обновлены:
 cups libcups libsmclient samba-client samba-client-cups samba-common

Можно было бы конечно выбрать и branch (так в ALT Linux называются основное хранилище для текущей ветки дистрибутива).

- Система
- Информация о системе
- Системные объекты
- Обновления
- Источники обновлений**
- Дата и время
- Дата и время
- Часовой пояс
- Пользователи
- Системный администратор
- Учётные записи
- Сеть
- PPTP-соединения
- PPPoE-соединения
- Сетевой экран
- Графический интерфейс
- Дисплей
- Настройка клавиатуры
- Серверы
- Прокси-сервер
- Статистика
- Сетевой трафик
- Web-интерфейс
- Сервер

Источники обновлений

<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.linux.kiev.ua/pub/Linux/ALT/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.linux.kiev.ua/pub/Linux/ALT/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/branch	i586	classic	ред
<input checked="" type="checkbox"/>	выключено	бинарные	updates	http://ftp.altlinux.org/pub/distributions/ALTlinux/4.0/branch	noarch	classic	ред
<input checked="" type="checkbox"/>	выключено	бинарные	updates	http://ftp.altlinux.org/pub/distributions/ALTlinux/4.0/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/updates/4.0	i586	updates	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	updates	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/branch	i586	classic	ред

Выделенные:

Тип:

Архитектура:

Подпись:

Адрес:

Компоненты:

Однако надо понимать, что подавляющее количество пользователей этого дистрибутива --- учителя, причём не только учителя информатики, и дети. А кому надо, тот может поставить и сам любой пакет. Единственная возможная проблема --- в отсутствии качественного интернета. Даже самый большой дистрибутив, который называется Мастер, всё равно не включает полного комплекта всех пакетов.

Также существует школьный бранч (он же school).

ALT Linux 4.0 Server | demo.altlinux.ru Справка [en](#) [ru](#) [uk](#)

Система
[Информация о системе](#)
[Системные объекты](#)
[Обновления](#)
Источники обновлений
[Дата и время](#)
[Дата и время](#)
[Часовой пояс](#)
Пользователи
[Системный администратор](#)
[Учётные записи](#)
Сеть
[PPTP-соединения](#)
[PPPoE-соединения](#)
[Сетевой экран](#)
Графический интерфейс
[Дисплей](#)
[Настройка клавиатуры](#)
Серверы
[Прокси-сервер](#)
Статистика
[Сетевой трафик](#)
Web-интерфейс
[Сервер](#)

Источники обновлений

<input type="checkbox"/>	выключено	бинарные	school	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/school/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	http://ftp.heanet.ie/mirrors/ftp.altlinux.org/4.0/school/branch	i586	classic	ред
<input checked="" type="checkbox"/>	выключено	бинарные	school	http://ftp.altlinux.org/pub/distributions/ALTlinux/4.0/school/branch	noarch	classic	ред
<input checked="" type="checkbox"/>	выключено	бинарные	school	http://ftp.altlinux.org/pub/distributions/ALTlinux/4.0/school/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/school/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	http://distrib-coffee.ipsl.jussieu.fr/pub/linux/altlinux/4.0/school/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	ftp://mirror.yandex.ru/altlinux/4.0/school/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	ftp://mirror.yandex.ru/altlinux/4.0/school/branch	i586	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	ftp://linux4u.jinr.ru/pub/ALTlinux/4.0/school/branch	noarch	classic	ред
<input type="checkbox"/>	выключено	бинарные	school	ftp://linux4u.jinr.ru/pub/ALTlinux/4.0/school/branch	i586	classic	ред

Выделенные:

Тип:

Архитектура:

Подпись:

Адрес:

Компоненты:

ALTlinux имеет обязанность некое время поддерживать школьную ветку. В школьный branch включено меньше программ, чем в основной. Школьный бранч был отделен от 4.0, поскольку там делаются изменения, которые в 4.0 делать не будут. Этот бранч является основой для создания школьных дистрибутивов. Вообще говоря, его и следует использовать для компьютеров в школах.