

Альт СП Сервер для облачной среды

Документация

Руководство пользователя

Редакция март, 2023



Аннотация

Добро пожаловать в документацию облачного дистрибутива Альт СП Сервер. Руководство описывает процедуру настройки и использования системы.

Названия компаний и продуктов, встречающихся в руководстве, могут являться торговыми знаками соответствующих компаний.

Данное руководство соответствует текущему состоянию сведений, но какие-либо окончательные правки могли не попасть в него. В случае обнаружения ошибок и неточностей в руководство вносятся изменения.

I. Что такое Альт СП Сервер?

1. Что такое Альт СП Сервер
2. Что такое Linux
3. Что такое системы Альт

II. Начало работы

4. Подключение к серверу по SSH
5. Получение прав суперпользователя

III. Установка пакетов

Введение

6. Источники программ (репозитории)

7. Поиск пакетов

8. Установка или обновление пакета

9. Удаление установленного пакета

10. Обновление системы

11. Единая команда управления пакетами (rpm)

IV. Корпоративная инфраструктура

12. Установка LEMP сервера (Linux, Nginx, MySQL, PHP)

V. Настройка системы

13. Центр управления системой

14. Обслуживание сервера

15. Соединение удалённых офисов (OpenVPN-сервер)

16. Доступ к службам сервера из сети Интернет

17. Статистика

18. Прочие возможности ЦУС

19. Права доступа к модулям

VI. Основы администрирования Linux

20. Общие принципы работы ОС

21. Режим суперпользователя

22. Команда sudo

23. Управление пользователями

24. Система инициализации systemd и sysvinit

25. Документация

VII. Техническая поддержка продуктов «Базальт СПО»

26. Покупателям нашей продукции

27. Пользователям нашей продукции

Часть I. Что такое Альт СП Сервер?

В этой части рассматривается что такое Linux и Альт СП Сервер.

Содержание

1. Что такое Альт СП Сервер

2. Что такое Linux

3. Что такое системы Альт

Глава 1. Что такое Альт СП Сервер

Альт СП Сервер — операционная система для серверов со встроенными программными средствами защиты информации, сертифицированная ФСТЭК России.

Операционная система Альт СП Сервер предназначена для:

- обеспечения выполнения программ в защищенной среде;
- работы со средствами виртуализации;
- применения в государственных информационных системах 1 класса защищенности;
- применения в автоматизированных системах управления производственными и технологическими процессами 1 класса защищенности;
- применения в информационных системах персональных данных при необходимости обеспечения 1 уровня защищенности персональных данных;
- применения в информационных системах общего пользования II класса;
- применения в государственных информационных системах 1 класса защищенности;
- применения в автоматизированных системах управления производственными и технологическими процессами 1 класса защищенности.

Преимущества Альт СП Сервер:

- наличие средств виртуализации;
- наличие средств для управления контейнерами;
- возможность установки на компьютеры с 64-разрядными процессорами с поддержкой EFI (без отключения SecureBoot);
- поддержка доменов Samba-DC (Active Directory), FreeIPA и OpenLDAP/MIT Kerberos;
- поддержка групповых политик для интеграции в инфраструктуру Active Directory;
- набор серверных служб для корпоративной инфраструктуры;
- модульная платформа конфигурирования с графическим и веб-интерфейсом (Alterator);
- возможность развернуть сервер сетевой установки и сервер обновлений с помощью веб-интерфейса;
- совместимость с отечественным программным обеспечением;
- оперативный выпуск обновлений по безопасности.

Глава 2. Что такое Linux

2.1. Свободные программы

2.2. Разработка Linux

2.3. Защищённость

2.4. Дистрибутивы Linux

2.5. Новичку

2.1. Свободные программы

Операционная система (далее — ОС) Linux — ядро, основные компоненты системы и большинство её пользовательских приложений — свободные программы. Свободные программы можно:

- запускать на любом количестве компьютеров;
- распространять бесплатно или за деньги без каких-либо ограничений;
- получать исходные тексты этих программ и вносить в них любые изменения.

Свобода программ обеспечила их широкое использование и интерес к ним со стороны тысяч разработчиков. Основные программы для Linux выходят под лицензией GNU General Public License (далее — GPL). Лицензия GNU не только гарантирует свободу, но и защищает её. Она допускает дальнейшее распространение программ только под той же лицензией, поэтому исходный код ядра Linux, компиляторов, библиотеки glibc, пользовательских графических оболочек не может быть использован для создания приложений с закрытым кодом. В этом принципиальное отличие Linux от свободных ОС семейства BSD (FreeBSD, NetBSD, OpenBSD), фрагменты которых вошли в Microsoft Windows и даже стали основой OS X. Linux включает в себя многие разработки BSD, но его компиляторы и системные библиотеки разработаны в рамках проекта GNU (<http://www.gnu.org/home.ru.html>).

2.2. Разработка Linux

В отличие от распространённых несвободных ОС, Linux не имеет географического центра разработки. Нет фирмы, которая владела бы этой ОС, нет и единого координационного центра. Программы для Linux — результат работы тысяч проектов. Большинство из них объединяет программистов из разных стран, связанных друг с другом только перепиской. Лишь некоторые проекты централизованы и сосредоточены в фирмах. Создать свой проект или присоединиться к уже существующему может любой программист, и, в случае успеха, результаты этой работы станут известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую. Это позволяет за короткий срок добавлять в программное обеспечение новые возможности, оперативно находить ошибки и исправлять их.

Именно гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность Linux. Низкая стоимость свободных разработок, отлаженные механизмы тестирования и распространения, привлечение независимых специалистов, обладающих индивидуальным, самостоятельным видением проблем, защита исходного текста программ лицензией GPL — всё это стало причиной успеха свободных программ.

Такая высокая эффективность разработки не могла не заинтересовать крупные фирмы. Они стали создавать свои свободные проекты, основывающиеся на тех же принципах. Так появились Mozilla, LibreOffice, свободный клон Interbase, SAP DB. IBM способствовала переносу Linux на свои мейнфреймы.

Открытый код программ значительно снизил себестоимость разработки закрытых систем для Linux и позволил снизить цену решения для пользователя. Вот почему Linux стала платформой, часто рекомендуемой для таких продуктов, как Oracle, DB2, Informix, Sybase, SAP ERP, Lotus Domino.

2.3. Защищённость

ОС Linux унаследовала от UNIX надёжность и отличную систему защиты. Система разграничения доступа к файлам позволяет не бояться вирусов. Но всё же, программ без ошибок не бывает, и Linux не исключение. Благодаря открытости исходного кода программ, аудит системы может осуществить любой специалист без подписок о неразглашении и без необходимости работы в стенах нанявшей его компании. Сообщества разработчиков и пользователей свободных программ создали множество механизмов оповещения об ошибках и их исправления. Сообщить об ошибке и принять участие в её исправлении независимому программисту или пользователю так же просто, как специалисту фирмы-разработчика или автору проекта. Благодаря этому ошибки защиты эффективно выявляются и быстро исправляются.

2.4. Дистрибутивы Linux

Большинство пользователей для установки Linux используют дистрибутивы. Дистрибутив — это не просто набор программ, а готовое решение для выполнения различных задач пользователя, обладающее идентичностью установки, управления, обновления, а также едиными системами настройки и поддержки.

2.5. Новичку

Linux — самостоятельная операционная система. Все операционные системы разные: Linux — не Windows, не OS X и не FreeBSD. В Linux свои правила, их необходимо изучить и к ним необходимо привыкнуть. Терпение и настойчивость в изучении Linux обернётся значительным повышением эффективности и безопасности вашей работы. То, что сегодня кажется странным и непривычным, завтра понравится и станет нормой.

Не стесняйтесь задавать вопросы, ведь самый простой способ найти ответ — совет опытного специалиста. Взаимопомощь и общение — традиция в мире Linux. Всегда можно обратиться за помощью к сообществу пользователей и разработчиков Linux. Большинство вопросов повторяются, поэтому для начала стоит поискать ответ на свой вопрос в документации, затем в сети Интернет. Если вы не нашли ответа в перечисленных источниках, не стесняйтесь, пишите на форум или в списки рассылки так, как писали бы своим друзьям, и вам обязательно помогут.

Глава 3. Что такое системы Альт

3.1. ALT Linux Team

3.2. Сизиф

3.1. ALT Linux Team

Команда ALT Linux (http://www.altlinux.org/ALT_Linux_Team) — это интернациональное сообщество, насчитывающее более 200 разработчиков свободного программного обеспечения.

3.2. Сизиф

Sisyphus (<https://packages.altlinux.org>) — наш ежедневно обновляемый банк программ (часто называемый репозиторий). Поддерживаемая ALT Linux Team целостность Sisyphus, оригинальная технология сборки программ, утилита **apt-get** и её графическая оболочка **synaptic** позволяют пользователям легко обновлять свои системы и быть в курсе актуальных новостей мира свободных программ.

Ежедневно изменяющийся репозиторий содержит самое новое программное обеспечение со всеми его преимуществами и недостатками (иногда ещё неизвестными). Поэтому, перед обновлением вашей системы из Sisyphus, мы советуем взвесить преимущества новых возможностей, реализованных в последних версиях программ, и вероятность возникновения неожиданностей в работе с ними (http://www.altlinux.org/Sisyphus_changes).

Разработка Sisyphus полностью доступна. У нас нет секретных изменений кода и закрытого тестирования с подписками о неразглашении. То, что мы сделали сегодня, завтра вы найдёте в сети. По сравнению с другими аналогичными банками программ (Debian unstable, Mandriva Cooker, PLD, Fedora), в Sisyphus есть немало самобытного. Особое внимание уделяется защите системы, локализации на русский язык, полноте и корректности зависимостей.

Название Sisyphus (Сизиф) заимствовано из греческой мифологии. С кропотливым Сизифом, непрерывно закатывающим в гору камни, команду ALT Linux Team объединяет постоянная работа над усовершенствованием технологий, заложенных в репозиторий.

Sisyphus, в первую очередь, — открытая лаборатория решений. Если вам это интересно, если вы хотите дополнить Sisyphus новыми решениями, если вы считаете, что можете собрать какую-то программу лучше — присоединяйтесь к проекту ALT Linux Team (<http://www.altlinux.org/Join>).

Часть II. Начало работы

Содержание

4. Подключение к серверу по SSH

5. Получение прав суперпользователя

Глава 4. Подключение к серверу по SSH

4.1. Создание пары ключей SSH

4.2. Подключение к серверу

4.3. Файл конфигурации

4.1. Создание пары ключей SSH

Чтобы создать пару ключей на локальном устройстве, достаточно запустить команду **ssh-keygen**. По умолчанию будет создан ключ RSA.

Также можно запустить команду **ssh-keygen** с дополнительными параметрами:

- **-t** — указывает тип ключа;
- **-b** — указывает размер ключа.

Ключ Ed25519 считается безопасным. Размер ключа фиксирован и не может быть изменён. Команда генерации Ed25519 ключа:

```
$ ssh-keygen -t ed25519
```

Размер ключа RSA по умолчанию составляет 3072 бита и не должен быть меньше 1024 бит. Для повышения безопасности можно установить размер ключа равным 4096 бит:

```
$ ssh-keygen -t rsa -b 4096
```

После запуска команды для создания ключа SSH, будет задан вопрос:

```
Generating public/private <тип_ключа> key pair.  
Enter file in which to save the key (/home/<пользователь>/.ssh/id_<тип_ключа>):
```

По умолчанию ключ будет сохранён в файл `~/.ssh/id_<тип_ключа>`. Можно нажать **ENTER**, чтобы согласиться с этим или указать другой файл.

Затем будет предложено установить парольную фразу:

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Парольная фраза еще больше повышает безопасность. Если установить парольную фразу, она будет запрашиваться каждый раз, когда используется ключ. Если нажать **ENTER** без ввода парольной фразы, SSH-ключ будет сгенерирован без пароля.

```
Терминал - user@host-15: /home/user
Файл Правка Вид Терминал Вкладки Справка
user@host-15 ~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/user/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_ed25519.
Your public key has been saved in /home/user/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:DHA3QrRTqb2K05bRYvEXfvw5VRsax67LUvZvuWy5enQ user@host-15
The key's randomart image is:
+--[ED25519 256]--+
|      .o= +.      |
|      o =..      |
|      +o         |
|      o+. . . +. |
|      +So o  = +  |
|      + + o o+ +E |
|      + = . .o.+o+ |
|      o =      ...=* |
|      o        .+*=* |
+-----[SHA256]-----+
user@host-15 ~ $
```

В результате выполнения команды **ssh-keygen** будут созданы два ключа:

- »<имя_ключа> — закрытый ключ (никогда не должен передаваться кому-либо ещё);
- »<имя_ключа>.pub — открытый ключ (часть, которую необходимо передать на удалённое устройство, к которому будет осуществляться подключение. На удалённом устройстве открытый ключ будет сохранен в файле **author_keys**).

Чтобы просмотреть открытый ключ, можно использовать команду **cat**:

```
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMnCtC+IJ0Bbw480ql79hijTPNFN9QFM+sLM19n+BBMu
user@host-15
```

Выделите текст и скопируйте его в буфер обмена. Этот ключ необходимо указать при создании ВМ в поле **SSH-ключ**.

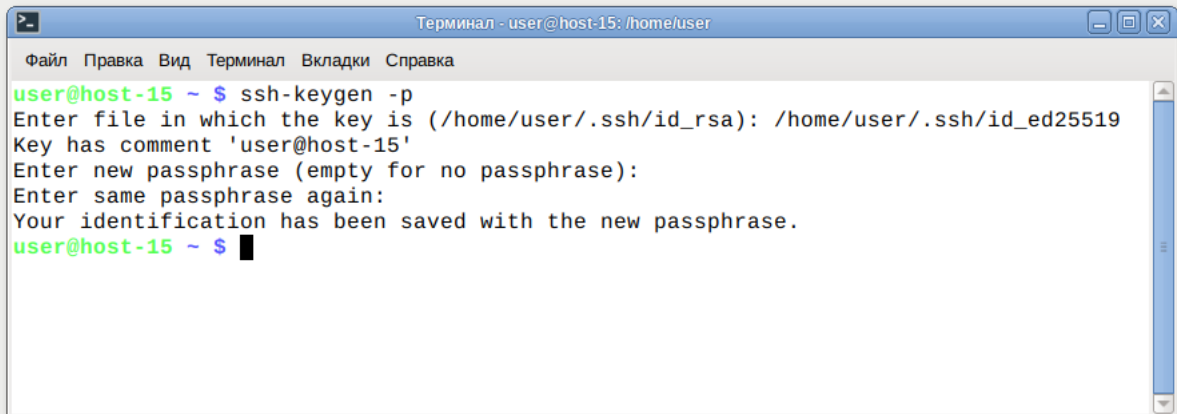


Примечание

Можно добавить, изменить и удалить парольную фразу для ключа SSH на локальном устройстве:

```
$ ssh-keygen -p
```

Обратите внимание, что вы должны знать свою текущую парольную фразу. Если вы забыли парольную фразу, то больше не сможете использовать свой SSH-ключ.



```
Терминал - user@host-15: /home/user
Файл  Правка  Вид  Терминал  Вкладки  Справка
user@host-15 ~ $ ssh-keygen -p
Enter file in which the key is (/home/user/.ssh/id_rsa): /home/user/.ssh/id_ed25519
Key has comment 'user@host-15'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
user@host-15 ~ $
```

4.2. Подключение к серверу

После того как сервер создан, к нему можно подключиться по SSH, используя интерфейс командной строки. VM должна находиться в статусе *RUNNING*.

Для подключения к серверу необходимо указать публичный адрес виртуальной машины.



Примечание

В Альт СП Сервер, предоставляемым Yandex Cloud, возможность подключения по протоколу SSH с использованием логина и пароля по умолчанию отключена.

Подключиться к серверу можно, выполнив команду:

```
$ ssh <логин>@<IP-адрес>
```

Например:

```
$ ssh user@158.160.0.157
The authenticity of host '158.160.0.157 (158.160.0.157)' can't be established.
ED25519 key fingerprint is SHA256:4+0X7yelFgZ0NfcLCXAv0hUVTs8GBIqfjmNHP5L65tyr.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.160.0.157' (ED25519) to the list of known hosts.
```



Примечание

При первом подключении к серверу, появится предупреждение о подлинности хоста, в ответ на которое, следует ввести `yes`.

Отпечаток ключа (key fingerprint), показанный в предупреждении, будет сохранён в файле `~/.ssh/known_hosts` и автоматически распознан при следующих подключениях.

Далее, если для SSH-ключа, не был установлен пароль произойдёт автоматическое подключение. Если пароль был установлен, необходимо ввести его в ответ на запрос пароля:

```
Enter passphrase for key '/home/user/.ssh/id_ed25519':
```

Обратите внимание, что вводимый пароль не отображается в интерфейсе командной строки.



Примечание

Если у вас несколько закрытых ключей, следует указать нужный:

```
$ ssh -i <файл_ключа> <логин>@<IP-адрес>
```

Например:

```
$ ssh -i .ssh/id_ed25519 user@158.160.0.157
```

В случае успешного прохождения процедуры аутентификации и идентификации будет выполнен вход в систему.

4.3. Файл конфигурации

После успешного подключения к серверу, можно добавить запись о подключении в файл конфигурации на локальном устройстве. Наличие такой записи позволит для подключения к серверу использовать команду:

```
$ ssh <уникальное-имя>
```

ВМЕСТО:

```
$ ssh -i <ключ> <логин>@<IP-адрес>
```

В файл конфигурации (`~/.ssh/config`) новые записи добавляются в следующем формате:

```
Host <уникальное-имя>
  HostName <IP-адрес>
  User <логин>
  IdentityFile <ключ>
```

Например:

```
Host altserver;  
    HostName 158.160.0.157  
    User user  
    IdentityFile ~/.ssh/id_ed25519
```

Для проверки того, работает ли запись, следует подключиться к серверу:

```
$ ssh altserver
```

Глава 5. Получение прав суперпользователя

5.1. Платформа Yandex Cloud

Команды, помеченные в данном руководстве символом #, необходимо выполнять с правами суперпользователя. Получить же такие права можно несколькими способами, в том числе через **sudo**. За подробностями следует обратиться к документации облачной платформы.

5.1. Платформа Yandex Cloud

Пользователь, указанный при создании ВМ, входит в группу sudo. Для повышения прав до root следует использовать команду **sudo**.

По умолчанию, у root нет пароля, поэтому достаточно подключиться к ВМ по SSH (см. [Подключение к серверу по SSH](#)) и ввести команду:

```
$ sudo su -
```

Также можно выполнять различные команды, не переключаясь на root. Например, для перезагрузки ВМ из гостевой ОС можно выполнить команду:

```
$ sudo reboot
```

Часть III. Установка пакетов

Содержание

[Введение](#)

[6. Источники программ \(репозитории\)](#)

[7. Поиск пакетов](#)

[8. Установка или обновление пакета](#)

[9. Удаление установленного пакета](#)

[10. Обновление системы](#)

[11. Единая команда управления пакетами \(dpkg\)](#)

Введение

В современных системах на базе Linux существует огромное число общих ресурсов: разделяемых библиотек, содержащих стандартные функции, исполняемые файлы, сценарии и стандартные утилиты и т.д. Этими общими ресурсами пользуются сразу несколько программ. Удаление или изменение версии одного из составляющих систему компонентов может повлечь неработоспособность других, связанных с ним компонентов, или может привести к выводу из строя всей системы. В контексте системного администрирования проблемы такого рода называют нарушением *целостности системы*. Задача администратора — обеспечить наличие в системе согласованных версий всех необходимых программных компонентов (обеспечение целостности системы).

Для установки, удаления и обновления программ, а также поддержания целостности системы в Linux в первую очередь стали использоваться программы *менеджеры пакетов* (например, такие, как **rpm**). С точки зрения менеджера пакетов программное обеспечение представляет собой набор компонентов — программных *пакетов*. Пакеты содержат в себе набор исполняемых программ и вспомогательных файлов, необходимых для корректной работы программного обеспечения. Менеджеры пакетов облегчают установку программ: они позволяют проверить наличие необходимого для работы устанавливаемой программы компонента подходящей версии непосредственно в момент установки. Менеджеры пакетов производят необходимые процедуры для регистрации программы во всех операционных средах пользователя: сразу после установки программа становится доступна пользователю из командной строки и появляется, если это было предусмотрено, в меню приложений всех графических оболочек.

Часто компоненты, используемые различными программами, выделяют в отдельные пакеты и помечают, что для работы ПО, предоставляемого пакетом А, необходимо установить пакет В. В таком случае говорят, что пакет А *зависит* от пакета В или между пакетами А и В существует *зависимость*.

Отслеживание зависимостей между такими пакетами представляет собой важную задачу для любого дистрибутива. Некоторые компоненты пакетов могут быть взаимозаменяемыми, т.е. может обнаружиться несколько пакетов, предлагающих затребованный ресурс.

Ещё более сложной является задача контроля целостности и непротиворечивости установленного в системе ПО. Представим, что некие программы А и В требуют наличия в системе компонентов С версии 1.0. Обновление версии пакета А, требующее обновления компонентов С до новой версии (например, до версии 2.0, использующей новый интерфейс доступа), влечёт за собой обязательное обновление и программы В.

На практике менеджеры пакетов оказались неспособны эффективно устранить нарушения целостности системы и предотвратить все коллизии при установке или удалении программ. Особенно остро этот недостаток сказался на обновлении систем из централизованного репозитория, в котором пакеты непрерывно обновляются, дробятся на более мелкие и т.п. Именно этот недостаток стимулировал создание систем управления программными пакетами и поддержания целостности ОС.

Для автоматизации и контроля описанных выше процессов стала применяться усовершенствованная система управления программными пакетами **APT** (от англ. Advanced Packaging Tool). Автоматизация и контроль достигаются путём создания одного или нескольких внешних репозиториях. В них хранятся доступные для установки пакеты программ.

В распоряжении **APT** находятся две базы данных: одна описывает установленные в системе пакеты, вторая — внешний репозиторий. **APT** отслеживает целостность установленной системы и, в случае обнаружения противоречий в зависимостях пакетов, разрешает конфликты, находит пути их корректного устранения, руководствуясь сведениями из внешних репозиториях.

Система **APT** состоит из нескольких утилит. Чаще всего используется утилита управления пакетами **apt-get**. Она автоматически определяет зависимости между пакетами и строго следит за её соблюдением при выполнении любой из следующих операций: установка, удаление или обновление пакетов.

Глава 6. Источники программ (репозитории)

6.1. Редактирование репозиториях

Отличие репозиториях, с которыми работает **APT**, от простого набора пакетов — наличие метаинформации. В ней содержится индекс находящихся в репозитории пакетов и сведения о них. Поэтому, чтобы получить всю информацию о репозитории, **APT** достаточно получить его индексы.

APT может пользоваться любым количеством репозиториях одновременно, формируя единую информационную базу обо всех содержащихся в них пакетах. При установке пакетов **APT** обращает внимание только на название пакета, его версию и зависимости. Для **APT** не имеет значения расположение пакета в том или ином репозитории.



Важно

Для одновременного подключения нескольких репозиториях необходимо отслеживать их совместимость друг с другом, т.е. их пакетная база должна отражать один определённый этап разработки. Совместное использование репозиториях, относящихся к разным дистрибутивам, или смешивание стабильного репозитория с нестабильной веткой разработки (Sisyphus) может привести к различным неожиданностям и трудностям при обновлении пакетов.

APT осуществляет взаимодействие с репозиториями при помощи различных протоколов доступа. Наиболее популярные — HTTP и FTP.

Для того чтобы **APT** мог использовать тот или иной репозиторий, информацию о нём необходимо поместить в файл **/etc/apt/sources.list**, либо в любой файл **.list** (например, **mysources.list**) в каталоге **/etc/apt/sources.list.d/**. Описания репозиториях заносятся в эти файлы в следующем виде:

```
rpm [подпись] метод:путь база название
rpm-src [подпись] метод:путь база название
```

Здесь:

- » rpm или rpm-src — тип репозитория (скомпилированные программы или исходные тексты);
- » [подпись] — необязательная строка-указатель на электронную подпись разработчиков. Наличие этого поля подразумевает, что каждый пакет из данного репозитория должен быть подписан соответствующей электронной подписью. Подписи описываются в файле **/etc/apt/vendor.list**;

- метод — способ доступа к репозиторию: ftp, http, file, rsh, ssh, cdrom, copy;
- путь — путь к репозиторию в терминах выбранного метода;
- база — относительный путь к базе данных репозитория;
- название — название репозитория.

Непосредственно после установки дистрибутива Альт СП Сервер в файлах **/etc/apt/sources.list.d/*.list** обычно указывается интернет-репозиторий, совместимый с установленным дистрибутивом.

После редактирования списка репозитория в **sources.list**, необходимо обновить локальную базу данных **APT** о доступных пакетах. Это делается командой **apt-get update**.

Если в **sources.list** присутствует репозиторий, содержимое которого может изменяться (например, постоянно разрабатываемый репозиторий или репозиторий обновлений по безопасности), то прежде чем работать с **APT**, необходимо синхронизировать локальную базу данных с удалённым сервером командой **apt-get update**. Локальная база данных создаётся заново при каждом изменении в репозитории: добавлении, удалении или переименовании пакета.

При установке определённого пакета **APT** производит поиск самой новой версии этого пакета во всех известных ему репозиториях вне зависимости от способа доступа к ним. Так, если в репозитории, доступном в сети Интернет, обнаружена более новая в сравнении с компакт-диск версия программы, то **APT** начнёт загружать соответствующий пакет из сети Интернет. Поэтому, если подключение к сети Интернет отсутствует или ограничено низкой пропускной способностью канала или высокой стоимостью, то следует закомментировать строки (добавить в начало строки символ #) в **/etc/apt/sources.list**, относящиеся к ресурсам в сети Интернет.

6.1. Редактирование репозитория

6.1.1. Утилита apt-repo

Для редактирования репозитория можно воспользоваться утилитой **apt-repo**:

- просмотреть список активных репозитория:

```
apt-repo
```

- добавить репозиторий в список активных репозитория:

```
apt-repo add репозиторий
```

- удалить или выключить репозиторий:

```
apt-repo rm репозиторий
```

- обновить информацию о репозиториях:

```
apt-repo update
```

- справка о команде **apt-repo**:

```
man apt-repo
```

или

```
apt-repo --help
```



Примечание

Для выполнения большинства команд необходимы права администратора.

6.1.2. Добавление репозиториев вручную

Для изменения списка репозиториев можно отредактировать в любом текстовом редакторе файлы из каталога `/etc/apt/sources.list.d/`.



Примечание

Для изменения этих файлов необходимы права администратора.

В файле **`altsp-C.list`** может содержаться такая информация:

```
# update.altsp.su (IVK, Moscow)

# ALT Certified 8
#rpm [cert8] ftp://update.altsp.su/pub/distributions/ALTLinux CF2/branch/x86_64
classic
#rpm [cert8] ftp://update.altsp.su/pub/distributions/ALTLinux CF2/branch/x86_64-
i586 classic
#rpm [cert8] ftp://update.altsp.su/pub/distributions/ALTLinux CF2/branch/noarch
classic

rpm [cert8] http://update.altsp.su/pub/distributions/ALTLinux CF2/branch/x86_64
classic
rpm [cert8] http://update.altsp.su/pub/distributions/ALTLinux CF2/branch/x86_64-
i586 classic
rpm [cert8] http://update.altsp.su/pub/distributions/ALTLinux CF2/branch/noarch
classic
```

По сути, каждая строка соответствует некому репозиторию. Не активные репозитории — строки, начинающиеся со знака `#`. Для добавления нового репозитория, достаточно дописать его в этот или другой файл.

После обновления списка репозиториев следует обновить информацию о них (выполнить команду **`apt-get update`** или **`apt-repo update`**).

Глава 7. Поиск пакетов

Если точное название пакета неизвестно, то для его поиска можно воспользоваться утилитой **`apt-cache`**. Данная утилита позволяет искать пакет не только по имени, но и по его описанию.

Команда **apt-cache search подстрока** позволяет найти все пакеты, в именах или описании которых присутствует указанная подстрока. Например:

```
$ apt-cache search dictionary
stardict-wn - GCIDE - The Collaborative International Dictionary of English
firefox-ru - Russian (RU) Language Pack for Firefox
gnome-dictionary-applet - GNOME panel applet for gnome-dictionary
gnome-utils - Utilities for the GNOME 2.0 desktop
libgdict - GNOME Dictionary Library.
stardict-mueller7 - V.K. Mueller English-Russian Dictionary, 7 Edition: stardict
format
stardict-slovnyk_be-en - Dictionary: Slovnyk Belarusian-English
stardict-slovnyk_be-ru - Dictionary: Slovnyk Belarusian-Russian
stardict-slovnyk_be-uk - Dictionary: Slovnyk Belarusian-Ukrainian
stardict-slovnyk_cs-ru - Dictionary: Slovnyk Czech-Russian
stardict-slovnyk_en-be - Dictionary: Slovnyk English-Belarusian
stardict-slovnyk_en-ru - Dictionary: Slovnyk English-Russian
stardict-slovnyk_en-uk - Dictionary: Slovnyk English-Ukrainian
stardict-slovnyk_es-ru - Dictionary: Slovnyk Spanish-Russian
stardict-slovnyk_ru-be - Dictionary: Slovnyk Russian-Belarusian
stardict-slovnyk_ru-cs - Dictionary: Slovnyk Russian-Czech
stardict-slovnyk_ru-en - Dictionary: Slovnyk Russian-English
stardict-slovnyk_ru-es - Dictionary: Slovnyk Russian-Spanish
stardict-slovnyk_ru-uk - Dictionary: Slovnyk Russian-Ukrainian
stardict-slovnyk_uk-be - Dictionary: Slovnyk Ukrainian-Belarusian
stardict-slovnyk_uk-en - Dictionary: Slovnyk Ukrainian-English
stardict-slovnyk_uk-ru - Dictionary: Slovnyk Ukrainian-Russian
words - A dictionary of English words for the /usr/share/dict directory
```

Для того чтобы подробнее узнать информацию о найденном пакете и получить его подробное описание, воспользуйтесь командой **apt-cache show**:

```
$ apt-cache show stardict-mueller7
Package: stardict-mueller7
Section: Text tools
Installed Size: 3094848
Maintainer: Anton V. Boyarshinov <boyarsh@altlinux.ru>
Version: 1.0-alt8@1338342590
Pre-Depends: rpmlib(PayloadIsLzma)
Depends: stardict (>= 2.4.2)
Provides: stardict-mueller7 (= 1.0-alt8)
Architecture: noarch
Size: 3134862
MD5Sum: 54f9e085c1fc67084253b3ba72a0c482
Filename: stardict-mueller7-1.0-alt8.noarch.rpm
Description: V.K. Mueller English-Russian Dictionary, 7th Edition, for stardict
Electronic version of V.K. Mueller English-Russian Dictionary,
7th Edition, in stardict format, for use with a stardict client.
```

При поиске с помощью **apt-cache** можно использовать русскую подстроку. В этом случае будут найдены пакеты, имеющие описание на русском языке. К сожалению, описание на русском языке в настоящее время есть не у всех пакетов, но наиболее актуальные описания переведены.

Глава 8. Установка или обновление пакета



Важно

Для установки пакетов требуются привилегии администратора.

Установка пакета с помощью APT выполняется командой **apt-get install имя_пакета**.



Важно

Перед установкой и обновлением пакетов необходимо выполнить команду обновления индексов пакетов:

```
# apt-get update
```

apt-get позволяет устанавливать в систему пакеты, требующие для работы наличие других, пока ещё не установленных пакетов. В этом случае он определяет, какие пакеты необходимо установить. **apt-get** устанавливает их, пользуясь всеми доступными репозиториями.

Установка пакета *stardict-mueller7* командой **apt-get install stardict-mueller7** приведёт к следующему диалогу с **APT** (если пакет еще не установлен):

```
# apt-get install stardict-mueller7
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие дополнительные пакеты будут установлены:
  icon-theme-hicolor libgtk+2 libgtk+2-locales libsigc++2 stardict
Следующие НОВЫЕ пакеты будут установлены:
  icon-theme-hicolor libgtk+2 libgtk+2-locales libsigc++2 stardict stardict-
mueller7
0 будет обновлено, 6 новых установлено, 0 пакетов будет удалено и 24 не будет
обновлено.
Необходимо получить 9629kB архивов.
После распаковки потребуется дополнительно 36,2MB дискового пространства.
Продолжить? [Y/n] y
Совершаем изменения...
Подготовка... ##### [100%]
Обновление / установка...
1: libsigc++2-2.10.2-alt1 ##### [ 17%]
2: libgtk+2-locales-2.24.32-alt3 ##### [ 33%]
3: icon-theme-hicolor-0.17-alt2 ##### [ 50%]
4: libgtk+2-2.24.32-alt3 ##### [ 67%]
5: stardict-3.0.6-alt1 ##### [ 83%]
6: stardict-mueller7-1.0-alt8 ##### [100%]
Завершено.
```

Команда

```
apt-get install имя_пакета
```

используется также и для обновления уже установленного пакета или группы пакетов. В этом случае **apt-get** дополнительно проверяет, есть ли обновлённая, в сравнении с установленной в системе, версия пакета в репозитории.

Например, если пакет *stardict-mueller7* установлен и в репозитории нет обновлённой версии этого пакета, то вывод команды **apt-get install stardict-mueller7** будет таким:

```
# apt-get install stardict-mueller7
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Последняя версия stardict-mueller7 уже установлена.
0 будет обновлено, 0 новых установлено, 0 пакетов будет удалено и 24 не будет
обновлено.
```

При помощи **APT** можно установить и отдельный rpm- пакет, не входящий в состав репозитория (например, полученный из сети Интернет). Для этого достаточно выполнить команду

```
# apt-get install /путь/к/файлу.rpm
```

При этом **APT** проведёт стандартную процедуру проверки зависимостей и конфликтов с уже установленными пакетами.

Иногда в результате операций с пакетами без использования **APT** целостность системы нарушается, и **apt-get** отказывается выполнять операции установки, удаления или обновления. В этом случае необходимо повторить операцию, задав опцию **-f**, заставляющую **apt-get** исправить нарушенные зависимости, удалить или заменить конфликтующие пакеты. В этом случае необходимо внимательно следить за сообщениями, выводимыми **apt-get**. Любые действия в этом режиме обязательно требуют подтверждения со стороны пользователя.

Глава 9. Удаление установленного пакета

Для удаления пакета используется команда **apt-get remove имя_пакета**. Для того чтобы не нарушать целостность системы, будут удалены и все пакеты, зависящие от удаляемого. В случае удаления пакета, который относится к базовым компонентам системы, **apt-get** потребует дополнительное подтверждение с целью предотвращения возможной случайной ошибки.



Важно

Для удаления пакетов требуются привилегии администратора.

При попытке с помощью **apt-get** удалить базовый компонент системы, вы увидите следующий запрос на подтверждение операции:

```
# apt-get remove filesystem
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
...
ВНИМАНИЕ: Будут удалены важные для работы системы пакеты
Обычно этого делать не следует. Вы должны точно понимать возможные последствия!
...
0 будет обновлено, 0 новых установлено, 2648 пакетов будет удалено и 0 не будет
```

обновлено.
Необходимо получить 0В архивов.
После распаковки будет освобождено 8994МВ дискового пространства.
Вы делаете нечто потенциально опасное!
Введите фразу 'Yes, do as I say!' чтобы продолжить.



Предупреждение

Каждую ситуацию, в которой **APT** выдаёт такой запрос, необходимо рассматривать отдельно. Вероятность того, что после выполнения этой команды система окажется неработоспособной, очень велика.

Глава 10. Обновление системы

10.1. Обновление всех установленных пакетов

10.2. Обновление ядра

10.1. Обновление всех установленных пакетов

Для обновления всех установленных пакетов необходимо выполнить команды:

```
# apt-get update
# apt-get dist-upgrade
```

Первая команда (**apt-get update**) обновит индексы пакетов. Вторая команда (**apt-get dist-upgrade**) позволяет обновить только те установленные пакеты, для которых в репозиториях, перечисленных в `/etc/apt/sources.list`, имеются новые версии.



Примечание

Несмотря на то, что команда **apt-get upgrade** существует, использовать её следует осторожно, либо не использовать вовсе.

Она позволяет обновить только те установленные пакеты, для которых в репозиториях, перечисленных в `/etc/apt/sources.list`, имеются новые версии.

Никакие другие пакеты при этой операции из системы удалены не будут. Этот способ полезен при работе со стабильными пакетами приложений, относительно которых известно, что они при смене версии изменяются несущественно.

Иногда, однако, происходит изменение в наименовании пакетов или изменение их зависимостей. Такие ситуации не обрабатываются командой **apt-get upgrade**, в результате чего происходит нарушение целостности системы: появляются неудовлетворённые зависимости. Для разрешения этой проблемы существует режим обновления в масштабе дистрибутива — **apt-get dist-upgrade**.

В случае обновления всего дистрибутива **APT** проведёт сравнение системы с репозиторием и удалит устаревшие пакеты, установит новые версии присутствующих в системе пакетов, отследит ситуации с переименованиями пакетов или изменения зависимостей между старыми и новыми версиями программ. Всё, что потребуется поставить (или удалить) дополнительно к уже имеющемуся в системе, будет указано в отчёте **apt-get**, которым **APT** предварит само обновление.



Примечание

Команда **apt-get dist-upgrade** обновит систему, но ядро ОС не будет обновлено.

10.2. Обновление ядра

Для обновления ядра ОС необходимо выполнить команду:

```
# update-kernel
```



Примечание

Если индексы сегодня еще не обновлялись перед выполнением команды **update-kernel** необходимо выполнить команду **apt-get update**.

Если необходимо обновить/установить другой тип ядра, необходимо выполнить команду:

```
# update-kernel -t <новый тип ядра>
```

где <новый тип ядра> — std-def, un-def и т.п.



Примечание

Ключ **-t** и тип ядра (std-def, un-def и т.п.) следует указывать только если необходимо обновить ядро другого типа, так как по умолчанию обновляется текущий тип ядра. Узнать версию загруженного ядра можно, выполнив команду:

```
$ uname -r
```

Команда **update-kernel** обновляет и модули ядра, если в репозитории обновилось что-то из модулей без обновления ядра.

Установка/обновление модулей ядра выполняется командой:

```
# apt-get install kernel-modules-<модуль>-<тип ядра>
```

Новое ядро загрузится только после перезагрузки системы, которую рекомендуется выполнить немедленно.

После успешной загрузки на обновленном ядре можно удалить старое, выполнив команду:

Глава 11. Единая команда управления пакетами (epm)

Основное назначение единой команды управления пакетами — унифицировать управление пакетами в дистрибутивах с разными пакетными менеджерами. Утилита **epm** упрощает процедуру управления пакетами, может использоваться в скриптах и установщиках, сервисных программах, в повседневном администрировании различных систем. В **epm** добавлены типовые операции, которые в случае использования apt, потребовали бы ввода более одной команды.

Единая команда управления пакетами включает в себя следующую функциональность:

- управление пакетами (установка/удаление/поиск);
- управление репозиториями (добавление/удаление/обновление/список);
- управление системными сервисами (включение/выключение/список).

Список поддерживаемых форматов пакетов: rpm, deb, tgz, tbz, tbz2, apk (alpine), pkg.gz.



Примечание

Установка утилиты **epm**, если она еще не установлена, выполняется командой:

```
# apt-get install eepm
```

Подробную информацию об утилите **epm** и её опциях можно получить, выполнив команду:

```
$ epm --help
```

Ниже описаны лишь некоторые возможности утилиты **epm**.

Установка пакета из репозитория или из локального файла в систему:

```
# epm install <имя_пакета>
```



Важно

Если пакет создан сторонним поставщиком, то при его установке командой **epm install** не будут выполнены установочные скрипты из пакета. Это предохраняет систему от повреждения, но может привести к тому, что пакет не заработает. Вернуть стандартное поведение можно добавлением **--scripts**:

```
# epm install --scripts <имя_пакета>
```



Примечание

Для некоторых сторонних rpm-пакетов, написаны дополнительные правила для перепакетки (при перепакетке пакета создаётся пакет, учитывающий, что нужно для работы исходного пакета). Установить такие пакеты можно, выполнив команду:

```
# rpm install --repack <имя_пакета>
```

Для deb-пакетов ключ **--repack** применяется автоматически.

Удаление пакета из системы:

```
# rpm remove <имя_пакета>
```

Поиск пакета в репозитории:

```
# rpm search <текст>
```

Получить список установленных пакетов:

```
$ rpm list
```

Удалить пакеты, от которых не зависят какие-либо другие пакеты, установленные в системе:

```
# rpm autoremove
```

Обновить все установленные пакеты и ядро ОС:

```
# rpm full-upgrade
```

Часть IV. Корпоративная инфраструктура

Содержание

[12. Установка LEMP сервера \(Linux, Nginx, MySQL, PHP\)](#)

Глава 12. Установка LEMP сервера (Linux, Nginx, MySQL, PHP)

[12.1. Установка веб-сервера Nginx](#)

[12.2. Установка MySQL](#)

[12.3. Установка php](#)

[12.4. Конфигурирование Nginx для использования php](#)

[12.5. Тестирование подключения к базе данных из PHP](#)

LEMP — это набор (комплекс) серверного программного обеспечения, используемый для разработки и развертывания веб-приложений. LEMP состоит из следующих компонентов: Linux — операционная система; Nginx — веб-сервер; MariaDB/MySQL — СУБД; PHP — язык программирования.

LEMP — это аналог стека LAMP (Linux, Apache, MySQL и PHP).



Примечание

В этом разделе в качестве примера доменного имени используется домен test.alt.

12.1. Установка веб-сервера Nginx

Установить пакет *nginx*:

```
# apt-get install nginx
```

Запустить Nginx и добавить его в автозагрузку:

```
# systemctl enable --now nginx
```

Создать корневой каталог для домена:

```
# mkdir -p /var/www/test.alt
```

Создать файл конфигурации **/etc/nginx/sites-available.d/test.alt.conf** на основе **/etc/nginx/sites-available.d/default.conf**:

```
# cp /etc/nginx/sites-available.d/default.conf /etc/nginx/sites-available.d/test.alt.conf
```

Внести изменения в файл конфигурации **/etc/nginx/sites-available.d/test.alt.conf**:

```
server {
    listen *:80;
    server_name test.alt www.test.alt;

    location / {
        root /var/www/test.alt;
    }

    access_log /var/log/nginx/test.alt-access.log;
}
```

Активировать конфигурацию:

```
# ln -s /etc/nginx/sites-available.d/test.alt.conf /etc/nginx/sites-enabled.d/
```

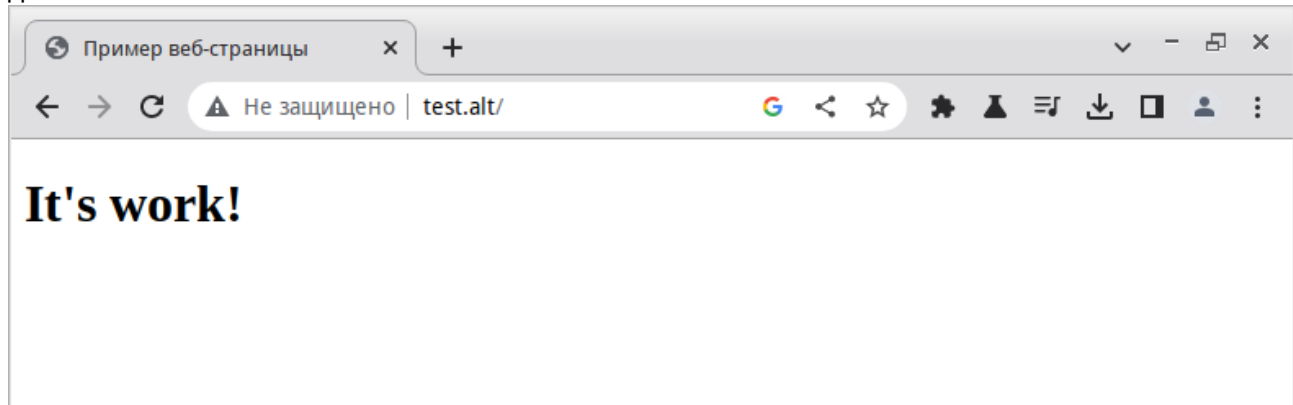
Перезапустить Nginx:

```
# systemctl reload nginx
```

Создать файл `/var/www/test.alt/index.html` со следующим содержимым:

```
<html>
<head>
  <title>Пример веб-страницы</title>
</head>
<body>
  <h1>It's work!</h1>
</body>
</html>
```

Для проверки работы веб-сервера, в веб-браузере открыть страницу `http://IP-сервера` или `http://домен`:



Администрирование сервера Nginx в основном заключается в настройке и поддержке его файлов конфигурации:

- ▀ **`/etc/nginx/nginx.conf`** — главный файл конфигурации Nginx;
- ▀ **`/etc/nginx/sites-available`** — каталог с конфигурациями виртуальных хостов (каждый файл, находящийся в этом каталоге, содержит информацию о конкретном сайте);
- ▀ **`/etc/nginx/sites-enabled`** — в этом каталоге содержатся конфигурации активных сайтов, обслуживаемых Nginx.

12.2. Установка MySQL

MySQL — это система управления базами данных (СУБД), которая используется для организации и получения контента. MariaDB имеет те же возможности, что и MySQL, и при этом является полностью бесплатной.

Установить MySQL (MariaDB):

```
# apt-get install mariadb
```

Запустить и добавить в автозагрузку сервер MySQL:

```
# systemctl enable --now mariadb.service
```

Задать пароль root для MySQL и настройки безопасности:

```
# mysql_secure_installation
```


Далее можно проверить возможность входа в консоль MySQL:

```
$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.4.20-MariaDB-alt1 (ALT c9f2)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
$
```



Примечание

Для подключения к MySQL в качестве пользователя root можно не указывать пароль, даже если он был определен при запуске скрипта **mysql_secure_installation**. Это связано с тем, что метод аутентификации по умолчанию для пользователя root в MySQL — `unix_socket`. Пользователями, которым разрешено входить в систему в качестве пользователя root MySQL, являются системные пользователи с привилегиями `sudo`, подключающиеся из консоли или через приложение, работающее с одинаковыми привилегиями.

```
# mysql
MariaDB [(none)]> exit
#
```

Для повышения безопасности лучше иметь выделенные учетные записи пользователей с менее широкими привилегиями, настроенными для каждой базы данных, особенно если на сервере планируется размещать несколько баз данных.

12.3. Установка php



Примечание

Начиная с версии php8.0, пакеты модулей именуются следующим образом: `php<мажорная>.<минорная>.<номер сборки>-<имя модуля>`. Из репозитория можно установить и эксплуатировать в одной системе одновременно разные версии php. В данном руководстве в качестве примера используется php8.1.

Установить пакеты:

```
# apt-get install php8.1-fpm-fcgi php8.1-mysqlnd php8.1-mysqlnd-mysqli
```

Запустить `php8.1-fpm` и добавить его в автозагрузку:

```
# systemctl enable --now php8.1-fpm
```



Примечание

Чтобы узнать версию PHP, которая установлена, можно воспользоваться командой:

```
$ php -v
```

12.4. Конфигурирование Nginx для использования php

Внести изменения в файл конфигурации `/etc/nginx/sites-available.d/test.alt.conf`:

```
server {
    listen *:80;
    # can't use wildcards in first server_name
    server_name test.alt www.test.alt;
    index      index.php;
    root /var/www/test.alt;
    location / {
        try_files $uri =404;
    }

    location ~ /\.php$ {
        try_files $uri =404;
        include /etc/nginx/fastcgi_params;
        fastcgi_pass unix:/var/run/php8.1-fpm/php8.1-fpm.sock;
        fastcgi_param SCRIPT_FILENAME /var/www/test.alt/$fastcgi_script_name;
    }
    access_log /var/log/nginx/test.alt-access.log;
}
```

Описание директив из файла конфигурации:

- **listen** — задаёт адрес и порт, на которых сервер будет принимать запросы. Можно указать адрес и порт, либо только адрес или только порт;
- **root** — определяет корневой каталог, в котором хранятся файлы, обслуживаемые данным веб-сайтом;
- **index** — определяет файлы, которые будут использоваться в качестве индекса. В имени файла можно использовать переменные. Наличие файлов проверяется в порядке их перечисления;
- **server_name** — определяет имя виртуального сервера (в этой директиве необходимо указать доменное имя сервера или общедоступный IP-адрес);
- **location /** — первый блок location включает директиву **try_files**, которая проверяет наличие файлов или каталогов, соответствующих запросу URI. Если Nginx не может найти подходящий ресурс, он вернет ошибку 404;
- **location ~ /\.php\$** — этот блок location включает фактическую обработку PHP, указывая Nginx на файл конфигурации `fastcgi-php.conf` и файл `php8.1-fpm.sock`, который объявляет, какой сокет связан с `php-fpm`.

Проверить конфигурацию на наличие синтаксических ошибок, можно выполнив команду:

```
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

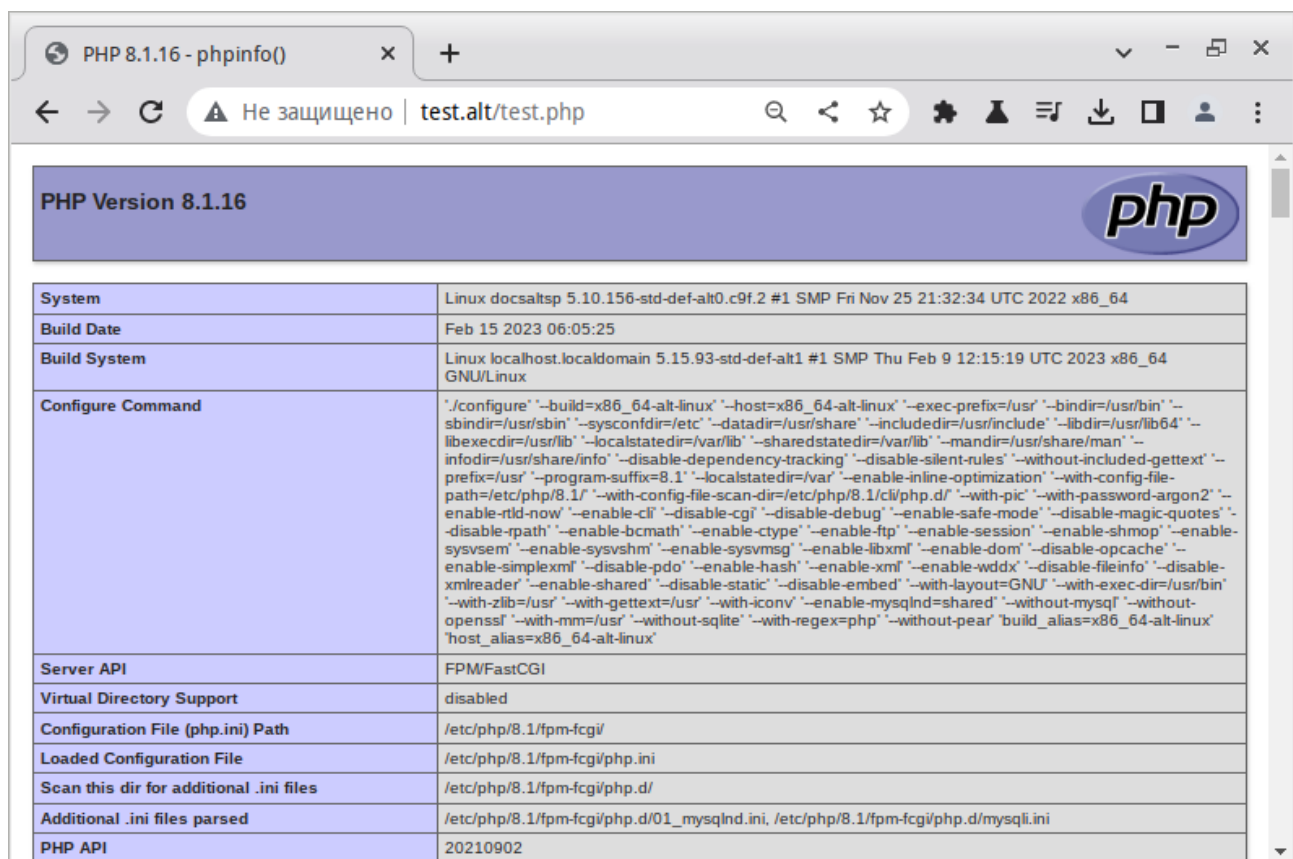
Если ошибки отсутствуют, необходимо перезагрузить Nginx, чтобы применить изменения:

```
# systemctl reload nginx.service
```

Для проверки того что Nginx правильно передаёт файлы .php PHP-процессору можно создать тестовый файл `/var/www/test.alt/test.php` со следующим содержимым:

```
<?php phpinfo(); ?>
```

При открытии этого файла по URL `http://test.alt/test.php` будет доступна информация о PHP:



PHP Version 8.1.16	
System	Linux docsaltsp 5.10.156-std-def-alt0.c9f.2 #1 SMP Fri Nov 25 21:32:34 UTC 2022 x86_64
Build Date	Feb 15 2023 06:05:25
Build System	Linux localhost.localdomain 5.15.93-std-def-alt1 #1 SMP Thu Feb 9 12:15:19 UTC 2023 x86_64 GNU/Linux
Configure Command	./configure '--build=x86_64-alt-linux' '--host=x86_64-alt-linux' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib64' '--libexecdir=/usr/lib' '--localstatedir=/var/lib' '--sharedstatedir=/var/lib' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--disable-dependency-tracking' '--disable-silent-rules' '--without-included-gettext' '--prefix=/usr' '--program-suffix=8.1' '--localstatedir=/var' '--enable-inline-optimization' '--with-config-file-path=/etc/php/8.1/' '--with-config-file-scan-dir=/etc/php/8.1/cli/php.d/' '--with-pic' '--with-password-argon2' '--enable-rtld-now' '--enable-cli' '--disable-cgi' '--disable-debug' '--enable-safe-mode' '--disable-magic-quotes' '--disable-rpath' '--enable-bcmath' '--enable-ctype' '--enable-ftp' '--enable-session' '--enable-shmop' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--enable-libxml' '--enable-dom' '--disable-openssl' '--enable-simplexml' '--disable-pdo' '--enable-hash' '--enable-xml' '--enable-wddx' '--disable-fileinfo' '--disable-xmlreader' '--enable-shared' '--disable-static' '--disable-embed' '--with-layout=GNU' '--with-exec-dir=/usr/bin' '--with-zlib=/usr' '--with-gettext=/usr' '--with-iconv' '--enable-mysqlnd=shared' '--without-mysql' '--without-openssl' '--with-mm=/usr' '--without-sqlite' '--with-regex=php' '--without-pear' '--build_alias=x86_64-alt-linux'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/fpm-fcgi
Loaded Configuration File	/etc/php/8.1/fpm-fcgi/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/fpm-fcgi/php.d/
Additional .ini files parsed	/etc/php/8.1/fpm-fcgi/php.d/01_mysqlnd.ini, /etc/php/8.1/fpm-fcgi/php.d/mysql.ini
PHP API	20210902

После проверки информации о PHP-сервере лучше всего удалить созданный файл, поскольку он содержит информацию о среде PHP и сервере:

```
# rm /var/www/test.alt/test.php
```

12.5. Тестирование подключения к базе данных из PHP

Для тестирования подключения к базе данных из PHP создадим базу данных с именем `example_database` и пользователя с именем `example_user`:

```
# mysql
```

Создать новую БД:

```
MariaDB [(none)]> CREATE DATABASE example_db;  
Query OK, 1 row affected (0,000 sec)
```

Создать нового пользователя и предоставить ему полные права доступа к базе данных example_database:

```
MariaDB [(none)]> CREATE USER 'example_user'@'%' IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0,002 sec)
```

Предоставить пользователю example_user все права на базу example_db:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON example_db.* TO 'example_user'@'%;  
Query OK, 0 rows affected (0,002 sec)  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0,000 sec)
```

Выйти из оболочки MySQL:

```
MariaDB [(none)]> exit;
```

Проверить, есть ли у нового пользователя надлежащие разрешения, снова войдя в консоль MySQL, с использованием пользовательских учетных данных пользователя:

```
$ mysql -u example_user -p  
Enter password:  
MariaDB [(none)]>
```

После входа в консоль MySQL убедитесь, что у пользователя example_user есть доступ к базе данных example_database:

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| example_db |  
| information_schema |  
+-----+  
2 rows in set (0,000 sec)
```

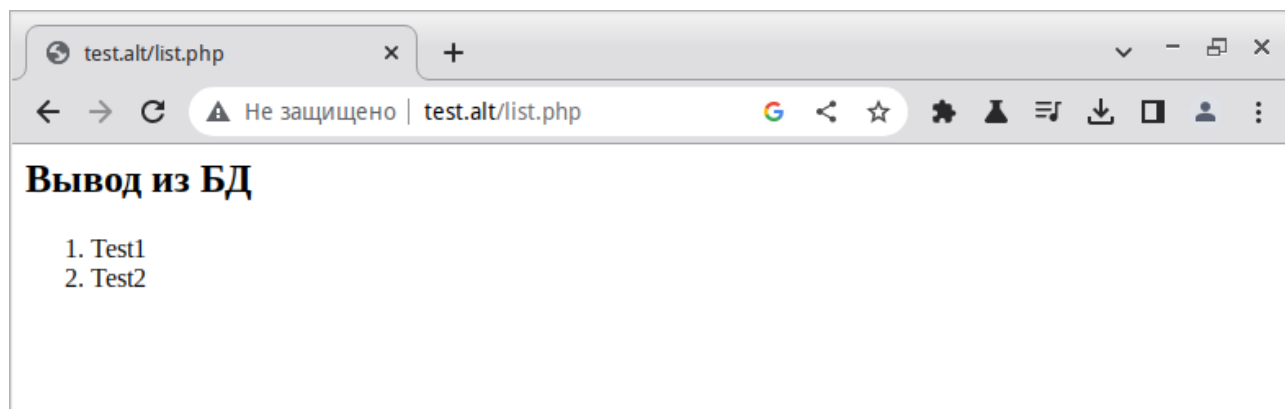
Создать таблицу и заполнить её тестовыми данными:

```
MariaDB [(none)]> USE example_db;  
MariaDB [(none)]> CREATE TABLE test (item_id INT AUTO_INCREMENT, content  
VARCHAR(255), PRIMARY KEY(item_id));  
MariaDB [(none)]> INSERT INTO test (content) VALUES ("Test1");  
MariaDB [(none)]> INSERT INTO test (content) VALUES ("Test2");  
MariaDB [(none)]> EXIT
```

Создать скрипт PHP (**/var/www/test.alt/list.php**), который будет подключаться к MySQL и запрашивать контент, со следующим содержанием:

```
<?php
$user = "example_user";
$password = "password";
$database = "example_db";
$table = "test";
$conn = mysqli_connect("localhost", $user, $password, $database);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "<h2>Вывод из БД</h2><ol>";
foreach($conn->query("SELECT content FROM $table") as $row) {
    echo "<li>" . $row['content'] . "</li>";
}
echo "</ol>";
mysqli_close($conn);
?>
```

Открыть этот файл по URL **<http://test.alt/list.php>**:



Часть V. Настройка системы

Если ВМ планируется использовать как полноценный сервер и требуется удобная настройка разных сервисов, то для управления настройками установленной системы можно воспользоваться **Центром управления системой**.

Центр управления системой (ЦУС) представляет собой удобный интерфейс для выполнения наиболее востребованных административных задач: добавление и удаление пользователей, управление системными службами, просмотр информации о состоянии системы и т.п.

Содержание

- [13. Центр управления системой](#)
- [14. Обслуживание сервера](#)
- [15. Соединение удалённых офисов \(OpenVPN-сервер\)](#)
- [16. Доступ к службам сервера из сети Интернет](#)
- [17. Статистика](#)
- [18. Прочие возможности ЦУС](#)

Глава 13. Центр управления системой

13.1. Установка и запуск веб-интерфейса ЦУС

13.2. Применение центра управления системой

13.1. Установка и запуск веб-интерфейса ЦУС

ЦУС имеет веб-ориентированный интерфейс, позволяющий управлять данным компьютером с любого другого компьютера сети.

Необходимо установить пакет *alterator-fbi*:

```
# apt-get install alterator-fbi
```



Примечание

Для работоспособности ЦУС необходимо задать пароль root:

```
# passwd -u root; passwd root
```

Для запуска веб-ориентированного интерфейса следует запустить сервисы *ahttpd* и *alteratord*:

```
# systemctl enable --now ahttpd
# systemctl enable --now alteratord
```

Работа с ЦУС может происходить из любого веб-браузера. Для начала работы необходимо перейти по адресу **https://ip-адрес:8080/**.

При запуске центра управления системой необходимо ввести в соответствующие поля имя пользователя (**root**) и пароль пользователя:

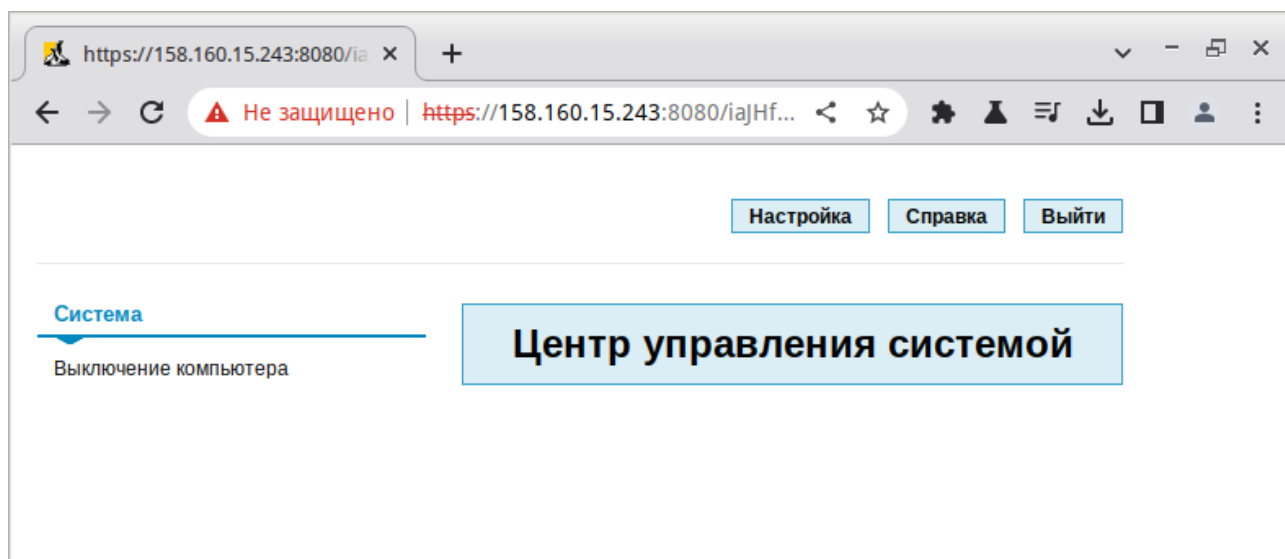
Пожалуйста, зарегистрируйтесь

Учётная запись:

Пароль:

Язык интерфейса:

После этого будут доступны все возможности ЦУС на той машине, к которой было произведено подключение через веб-интерфейс.



ЦУС состоит из независимых диалогов-модулей. Каждый модуль отвечает за настройку определённой функции или свойства системы.

Модули можно дополнительно загружать и удалять как обычные программы:

```
# apt-get install alterator-datetime
# apt-get remove alterator-datetime
```

Установленные пакеты, которые относятся к ЦУС, можно посмотреть, выполнив команду:

```
$ rpm -qa | grep alterator*
```

Прочие пакеты для ЦУС можно найти, выполнив команду:

```
$ apt-cache search alterator*
```

Веб-интерфейс ЦУС можно настроить (кнопка **Настройка**), выбрав один из режимов:

- » основной режим;
- » режим эксперта.

Выбор режима влияет на количество отображаемых модулей. В режиме эксперта отображаются все модули, а в основном режиме только наиболее используемые.

Центр управления системой содержит справочную информацию по включённым в него модулям. Об использовании самого интерфейса системы управления можно прочитать, нажав на кнопку **Справка** на начальной странице центра управления системой.



Предупреждение

После работы с центром управления системой, в целях безопасности, не оставляйте открытым браузер. Обязательно выйдите, нажав на кнопку **Выйти**.

13.2. Применение центра управления системой

Вы можете использовать ЦУС для разных целей, например:

- » Настройки **Даты и времени** ([datetime](#));
- » Управления выключением и перезагрузкой компьютера ([ahttpd-power](#));
- » Управления **Системными службами** ([services](#));
- » Просмотра **Системных журналов** ([logs](#));
- » Настройки **OpenVPN-подключений** ([openvpn-server](#) и [net-openvpn](#));
- » Создания, удаления и редактирования учётных записей **Пользователей** ([users](#));
- » Настройки ограничения **Использования диска (квоты)** ([quota](#)).

Дальнейшие разделы описывают некоторые возможности использования Альт СП Сервер, настраиваемые в ЦУС.

Глава 14. Обслуживание сервера

14.1. Мониторинг состояния системы

14.2. Системные службы

14.3. Обновление системы

14.4. Обновление систем, не имеющих выхода в Интернет

14.5. Локальные учётные записи

14.6. Администратор системы

14.7. Дата и время

14.8. Агент наблюдения

14.9. Ограничение использования диска

14.10. Выключение и перезагрузка компьютера

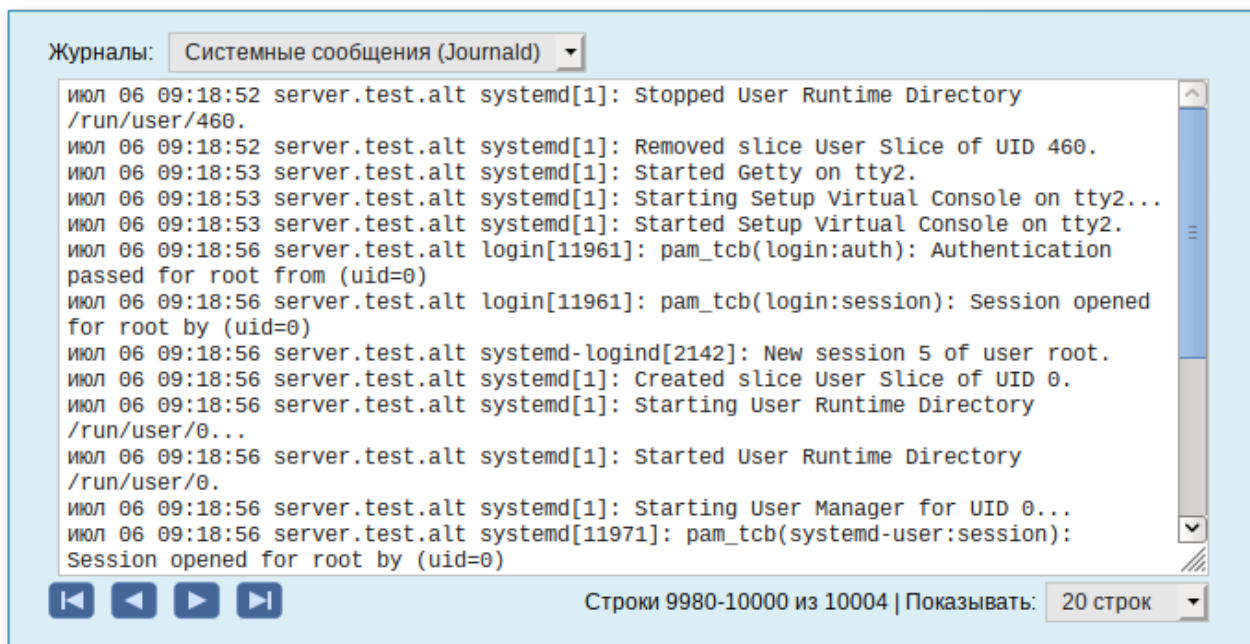
Для безотказной работы всего домена очень важно следить за корректной работой его центрального звена — сервера под управлением Альт СП Сервер. Регулярный мониторинг состояния сервера, своевременное резервное копирование, обновление установленного ПО являются важной частью комплекса работ по обслуживанию сервера.

14.1. Мониторинг состояния системы

Для обеспечения бесперебойной работы сервера крайне важно производить постоянный мониторинг его состояния. Все события, происходящие с сервером, записываются в *журналы*, анализ которых помогает избежать сбоев в работе сервера и предоставляет возможность разобраться в причинах некорректной работы сервера.

Для просмотра журналов предназначен модуль ЦУС **Системные журналы** (пакет *alterator-logs*) из раздела **Система**). Интерфейс позволяет просмотреть различные типы журналов с возможностью перехода к более старым или более новым записям.

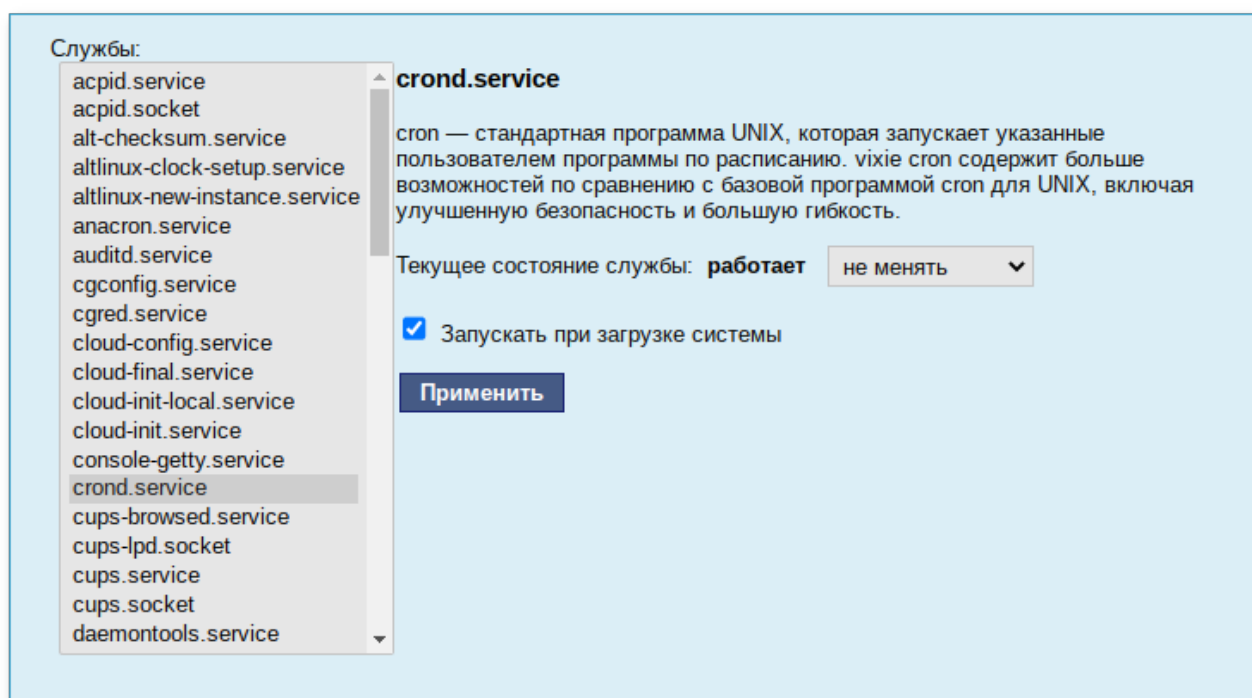
Различные журналы могут быть выбраны из списка **Журналы**.



Каждый журнал может содержать довольно большое количество сообщений. Уменьшить либо увеличить количество выводимых строк можно, выбрав нужное значение в списке **Показывать**.

14.2. Системные службы

Для изменения состояния служб можно использовать модуль ЦУС **Системные службы** (пакет *alterator-services*) из раздела **Система**. Интерфейс позволяет изменять текущее состояние службы и, если необходимо, применить опцию запуска службы при загрузке системы.



После выбора названия службы из списка отображается описание данной службы, а также текущее состояние: Работает/Остановлена/Неизвестно.

14.3. Обновление системы

После установки системы крайне важно следить за обновлениями ПО. Обновления для Альт СП Сервер могут содержать как исправления, связанные с безопасностью, так и новый функционал или просто улучшение и ускорение алгоритмов. В любом случае настоятельно рекомендуется регулярно обновлять систему для повышения надёжности работы сервера.

Для автоматизации процесса установки обновлений предусмотрен модуль ЦУС **Обновление системы** (пакет *alterator-updates*) из раздела **Система**. Здесь можно включить автоматическое обновление через Интернет с одного из предлагаемых серверов или задать собственные настройки.

☐ Не обновлять систему

☒ Обновление системы управляемое сервером

☐ Обновлять систему автоматически из Интернет

Источник:

Репозитории: ☒ Репозиторий обновлений для Альт 8 СП

Расписание обновлений

☒ Ежедневно

☐ Еженедельно в:

☐ Ежемесячно в день:

Время:

Источник обновлений указывается явно (при выбранном режиме **Обновлять систему автоматически из сети Интернет**) или вычисляется автоматически (при выбранном режиме **Обновление системы управляемое сервером** и наличии в локальной сети настроенного сервера обновлений).



Примечание

Рабочие станции «видят» локальный сервер обновлений, при выборе режима **Обновление системы управляемое сервером**, если они находятся в домене (при этом сервер обновлений должен быть настроен на **Опубликовать как репозиторий для автоматических обновлений**).

Процесс обновления системы будет запускаться автоматически согласно заданному расписанию.



Примечание

Чтобы указать в качестве сервера обновлений локально настроенный источник, необходимо выбрать режим **Обновлять систему автоматически из Интернет**, выбрать в списке **Другой адрес** и указать адрес локального сервера обновлений, например `http://<ip сервера>/mirror`:

☒ Обновлять систему автоматически из Интернет

Источник: Другой адрес ▾

Репозитории: ☒ Репозиторий обновлений для Альт 8 СП

14.4. Обновление систем, не имеющих выхода в Интернет

Для систем, не имеющих прямого выхода в Интернет, рекомендуется установка отдельного сервера обновлений на базе ОС Альт СП Сервер, находящегося вне защищенного контура и организация ограниченного доступа к этому серверу.

Модуль ЦУС **Сервер обновлений** (пакет *alterator-mirror*) из раздела **Серверы** предназначен для зеркалирования репозитория и публикации их для обновлений рабочих станций и серверов. Репозиторий выбирается в соответствии с выбранной веткой для соответствующего дистрибутива.

Сервер обновлений — технология, позволяющая настроить автоматическое обновление программного обеспечения, установленного на клиентских машинах (рабочих местах), работающих под управлением Альт Рабочая станция.

Репозиторий	Источник	Архитектуры	Локальное зеркало	Опубликовано
Стабильная ветка ALT Linux 5.1			<input type="checkbox"/>	<input type="checkbox"/>
Репозиторий обновлений для Альт 8 СП - 2021	update.altsp.su	x86_64	<input checked="" type="checkbox"/> (31 Гб)	<input type="checkbox"/>
Репозиторий обновлений для Альт 8 СП			<input type="checkbox"/>	<input type="checkbox"/>
Пятая платформа			<input type="checkbox"/>	<input type="checkbox"/>
Шестая платформа			<input type="checkbox"/>	<input type="checkbox"/>
Седьмая платформа			<input type="checkbox"/>	<input type="checkbox"/>
Восьмая платформа			<input type="checkbox"/>	<input type="checkbox"/>
Девятая платформа			<input type="checkbox"/>	<input type="checkbox"/>
Девятая платформа (mipsel)			<input type="checkbox"/>	<input type="checkbox"/>
ALT Linux Sisyphus			<input type="checkbox"/>	<input type="checkbox"/>
ALT Linux Sisyphus (mipsel)			<input type="checkbox"/>	<input type="checkbox"/>
ALT Linux Sisyphus (riscv64)			<input type="checkbox"/>	<input type="checkbox"/>
Публичный бранч TEAM t6			<input type="checkbox"/>	<input type="checkbox"/>
Публичный бранч TEAM t7			<input type="checkbox"/>	<input type="checkbox"/>

Свободное место: 11 Гб

Предупреждение: зеркалирование потребует наличия большого количества места на диске.

- ☐ Отключить зеркалирование
☒ Зеркалировать ежедневно
☐ Зеркалировать еженедельно в:
☐ Зеркалировать ежемесячно в день:

Время:

Применить

Сбросить

На странице модуля можно выбрать, как часто выполнять загрузку пакетов, можно выставить время, когда начинать зеркалирование.

Здесь также можно выбрать репозитории, локальные срезы которых необходимы. При нажатии на название репозитория, появляются настройки этого репозитория. Необходимо выбрать источник (сайт, откуда будет скачиваться репозиторий), архитектуру процессора (если их несколько, то стоит выбрать соответствующие).



Примечание

При выборе любой архитектуры также будет добавлен источник с noarch.

Репозиторий: Репозиторий обновлений для Альт 8 СП - 2021

Источник:

Архитектуры: ☐ i586
☒ x86_64
☐ x86_64-i586

☒ Локальное зеркало репозитория
☐ Опубликовать как репозиторий для автоматических обновлений

Исключить каталоги и файлы (каждый шаблон в отдельной строке)

```
SRPMS  
RPMS.debuginfo  
*-debuginfo-*
```

Сервер обновлений предоставляет возможность автоматически настроить обновление клиентских машин в нужном режиме:

» Локальное зеркало репозитория

В этом режиме на сервере создаётся копия удалённого репозитория. Загрузка ПО клиентскими машинами может производиться с локального сервера по протоколам HTTP, HTTPS, FTP, rsync (для каждого протокола нужно настроить соответствующие службы, ниже приведён пример настройки HTTP- и FTP-сервера). Наличие на локальном сервере зеркала репозитория при большом количестве машин в сети позволяет существенно сэкономить трафик.



Важно

Зеркалирование потребует наличия большого количества места на диске.

Уменьшить размер скачиваемых файлов и занимаемое репозиторием место на диске можно, указав имена каталогов и файлов, которые будут исключены из синхронизации. Например, не скачивать пакеты с исходным кодом и пакеты с отладочной информацией:

```
SRPMS
*-debuginfo-*
```

Шаблоны указываются по одному в отдельной строке. Символ «*» используется для подстановки любого количества символов.

» Публикация репозитория

В этом случае публикуется или URL внешнего сервера, содержащего репозиторий или, если включено локальное зеркало репозитория, адрес этого сервера. Такая публикация позволяет клиентским машинам автоматически настроить свои менеджеры пакетов на использование внешнего или локального репозитория.

Со стороны клиентских машин, в этом случае, необходимо настроить модуль [Обновление системы](#), отметив в нём **Обновление системы управляемое сервером**.

Настройка локального репозитория заканчивается нажатием на кнопку **Применить**.



Примечание

По умолчанию локальное зеркало репозитория находится в **/srv/public/mirror**. Для того чтобы зеркалирование происходило в другую папку, необходимо эту папку примонтировать в папку **/srv/public/mirror**. Для этого в файл **/etc/fstab** следует вписать строку:

```
/media/disk/localrepo /srv/public/mirror none rw,bind,auto 0 0
```

где **/media/disk/localrepo** — папка-хранилище локального репозитория.



Примечание

Если в каталогах **/srv/public/mirror/<репозиторий>/branch/<архитектура>/base/** нет файлов **pkglist.*** значит зеркалирование не закончено (т.е. не все файлы загружены на ваш сервер).

14.4.1. Настройка веб-сервера

Установить веб-сервер (в данном примере nginx):

```
# apt-get install nginx
```

Создать файл конфигурации сервера в **/etc/nginx/sites-available.d/repo.conf**:

```
server {
    listen 80;
    server_name localhost .local <ваш ip>;

    access_log /var/log/nginx/repo-access.log;
    error_log /var/log/nginx/repo-error.log;

    location /mirror {
        root /srv/public;
        autoindex on;
    }
}
```

Сделать ссылку в **/etc/nginx/sites-enabled.d/**:

```
# ln -s /etc/nginx/sites-available.d/repo.conf /etc/nginx/sites-enabled.d/
repo.conf
```

Запустить nginx и добавить его в автозагрузку:

```
# systemctl enable --now nginx
```

На клиентских машинах необходимо настроить репозитории. Сделать это можно в программе управления пакетами **Synaptic** (**Параметры** → **Репозитории**) или в командной строке:

```
# apt-repo rm all
# apt-repo add http://<ip сервера>/mirror/c9f2branch
```

Проверить правильность настройки репозитория:

```
# apt-repo
rpm http://10.129.0.24/mirror c9f2/branch/x86_64 classic
rpm http://10.129.0.24/mirror c9f2/branch/noarch classic
```

14.4.2. Настройка FTP-сервера

Установить пакеты *vsftpd*, *lftp*, если они еще не установлены:

```
# apt-get install vsftpd lftp
```

Настроить параметры использования vsftpd в файле **/etc/xinetd.d/vsftpd**:

```
# default: off
# description: The vsftpd FTP server.
service ftp
{
    disable = no # включает службу
    socket_type = stream
    protocol = tcp
    wait = no
```



```
user = root
nice = 10
rlimit_as = 200M
server = /usr/sbin/vsftpd
only_from = 0/0 # предоставить доступ для всех IP
}
```

Перезапустить xinetd:

```
# systemctl restart xinetd
```

Изменить настройку прав доступа в файле **/etc/vsftpd/conf**:

```
local_enable=YES
```

Создать каталог **/var/ftp/mirror**:

```
# mkdir -p /var/ftp/mirror
```

Примонтировать каталог **/srv/public/mirror** в **/var/ftp/mirror** с опцией **--bind**:

```
# mount --bind /srv/public/mirror /var/ftp/mirror
```



Примечание

Для автоматического монтирования каталога **/srv/public/mirror** при загрузке системы необходимо добавить следующую строку в файл **/etc/fstab**:

```
/srv/public/mirror /var/ftp/mirror none defaults,bind 0 0
```

На клиентских машинах необходимо настроить репозитории:

```
# apt-repo rm all
# apt-repo add ftp://<ip сервера>/mirror/c9f2/branch
# apt-repo
rpm ftp://10.129.0.24/mirror c9f2/branch/x86_64 classic
rpm ftp://10.129.0.24/mirror c9f2/branch/noarch classic
```

14.5. Локальные учётные записи

Модуль **Локальные учётные записи** (пакет *alterator-users*) из раздела **Пользователи** предназначен для администрирования системных пользователей.

Новая учётная запись: **Создать**

user

Комментарий:

Домашний каталог:

Интерпретатор команд:

☐ Входит в группу администраторов

☐ Создать автоматически

Пароль: (введите фразу)

(повторите фразу)

Применить **Удалить пользователя**

Для создания новой учётной записи необходимо ввести имя новой учётной записи и нажать кнопку **Создать**, после чего имя отобразится в списке слева.

Для дополнительных настроек необходимо выделить добавленное имя, либо, если необходимо изменить существующую учётную запись, выбрать её из списка.

14.6. Администратор системы

В модуле **Администратор системы** (пакет *alterator-root*) из раздела **Пользователи** можно изменить пароль суперпользователя (root), заданный при начальной настройке системы.

В данном модуле (только в веб-интерфейсе) можно добавить публичную часть ключа RSA или DSA для доступа к серверу по протоколу SSH.

Пароль системного администратора:

☐ Создать автоматически

(введите фразу)

(повторите фразу)

Сменить пароль

Разрешённые ssh ключи:

SHA256:h5ldexZzlBaqCHl6Nr4enxJlt9XQc1a5lnoJG+VSvo

Удалить ключ

Новый ключ: **Обзор...** **Файл не выбран.** **Добавить**

14.7. Дата и время

В модуле **Дата и время** (пакет *alterator-datetime*) из раздела **Система** можно изменить дату и время на сервере, сменить часовой пояс, а также настроить автоматическую синхронизацию часов на самом сервере по протоколу NTP и предоставление точного времени по этому протоколу для рабочих станций локальной сети.

☒ Получать точное время с NTP-сервера:

☐ Работать как NTP-сервер

Текущая дата: < Февраль 2023 >

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

2023-02-10

Текущее время:

10:30:58

☒ Хранить время в BIOS по Гринвичу

Часовой пояс: Европа/Калининград

Выбрать источник сигналов времени:

Системное время зависит от следующих факторов:

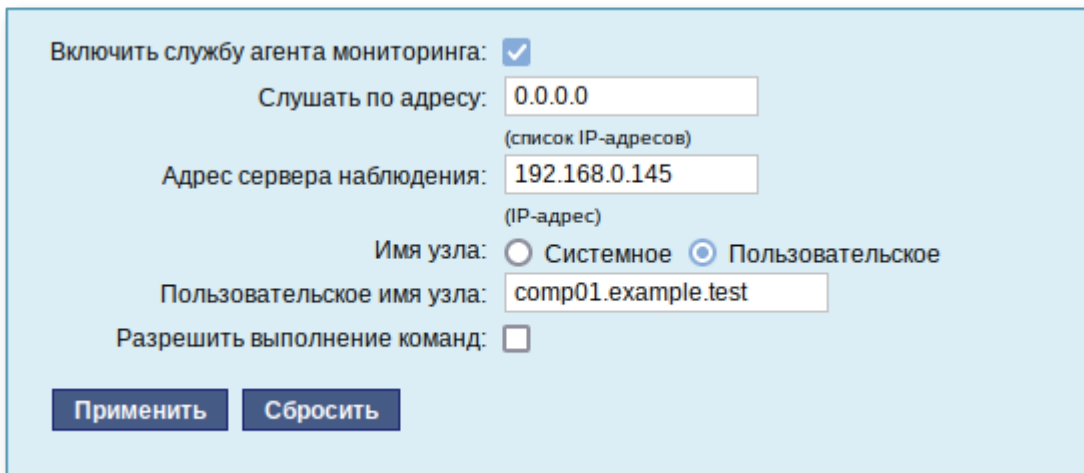
- » часы в BIOS — часы, встроенные в компьютер. Они работают, даже если он выключен;
- » системное время — часы в ядре операционной системы. Во время работы системы все процессы пользуются именно этими часами;
- » часовые пояса — регионы Земли, в каждом из которых принято единое местное время.

При запуске системы происходит активация системных часов и их синхронизация с аппаратными, кроме того, в определённых случаях учитывается значение часового пояса. При завершении работы системы происходит обратный процесс.

Если настроена синхронизация времени с NTP-сервером, то сервер сможет сам работать как сервер точного времени. Для этого достаточно отметить соответствующий пункт **Работать как NTP-сервер**.

14.8. Агент наблюдения

Модуль **Агент наблюдения** (пакет *alterator-zabbix-agent*) из раздела **Система** позволяет настроить клиентскую часть системы мониторинга Zabbix.



The screenshot shows a configuration window for the Zabbix agent. It contains the following fields and controls:

- Включить службу агента мониторинга:** A checked checkbox.
- Слушать по адресу:** A text input field containing "0.0.0.0". Below it, in smaller text, is "(список IP-адресов)".
- Адрес сервера наблюдения:** A text input field containing "192.168.0.145". Below it, in smaller text, is "(IP-адрес)".
- Имя узла:** Two radio buttons: "Системное" (unselected) and "Пользовательское" (selected).
- Пользовательское имя узла:** A text input field containing "comp01.example.test".
- Разрешить выполнение команд:** An unchecked checkbox.
- At the bottom, there are two buttons: "Применить" and "Сбросить".

На странице модуля можно задать следующие параметры:

- » **Слушать по адресу** — IP-адрес, который агент должен прослушивать;
- » **Адрес сервера наблюдения** — адрес сервера Zabbix, которому разрешено обращаться к агенту;
- » **Имя узла** — при выборе пункта **Системное**, узел будет добавлен на сервер Zabbix под системным именем, при выборе пункта **Пользовательское**, имя узла можно указать в поле **Пользовательское имя узла**;
- » **Пользовательское имя узла** — имя узла мониторинга, которое будет указано на сервере Zabbix.



Примечание

Параметр **Разрешить выполнение команд** использовать не рекомендуется.

Чтобы применить настройки и запустить Zabbix-агент, следует отметить пункт **Включить службу агента мониторинга** и нажать кнопку **Применить**.

14.9. Ограничение использования диска

Модуль **Использование диска** (пакет *alterator-quota*) в разделе **Пользователи** позволяет ограничить использование дискового пространства пользователями, заведёнными на сервере в модуле **Пользователи**.

Файловая система: / Текущее использование диска: 0 КБ

Включено: ☐ Мягкое ограничение: 0 КБ

Пользователь: user Жесткое ограничение: 0 КБ

Количество файлов: 0

Мягкое ограничение: 0

Жесткое ограничение: 0

Применить Сбросить

Модуль позволяет задать ограничения (квоты) для пользователя при использовании определённого раздела диска. Ограничить можно как суммарное количество килобайт, занятых файлами пользователя, так и количество этих файлов.

Для управления квотами файловая система должна быть подключена с параметрами **usrquota**, **grpquota**. Для этого следует выбрать нужный раздел в списке **Файловая система** и установить отметку в поле **Включено**:

Файловая система: /home Текущее использование диска: 567320 КБ

Включено: ☒ Мягкое ограничение: 0 КБ

Пользователь: user Жесткое ограничение: 0 КБ

Количество файлов: 1143

Мягкое ограничение: 100

Жесткое ограничение: 100

Применить Сбросить

Для того чтобы задать ограничения для пользователя, необходимо выбрать пользователя в списке **Пользователь**, установить ограничения и нажать кнопку **Применить**.

При задании ограничений различают жёсткие и мягкие ограничения:

- » **Мягкое ограничение:** нижняя граница ограничения, которая может быть временно превышена. Временное ограничение — одна неделя.
- » **Жёсткое ограничение:** использование диска, которое не может быть превышено ни при каких условиях.

Значение 0 при задании ограничений означает отсутствие ограничений.

14.10. Выключение и перезагрузка компьютера

Иногда, в целях обслуживания или по организационным причинам необходимо корректно выключить или перезагрузить сервер. Для этого можно воспользоваться модулем ЦУС **Выключение компьютера** в разделе **Система**.

☒ Продолжить работу

☐ Выключить компьютер сейчас

☐ Перезагрузить компьютер сейчас

☐ Приостановить компьютер сейчас

☐ Погрузить компьютер в сон сейчас

☒ Выключать компьютер каждый день в: 19:45:00

☐ Перезагружать компьютер каждый день в: 23:00:00

☐ Приостанавливать компьютер каждый день в: 23:00:00

☐ Погружать компьютер в сон каждый день в: 23:00:00

☐ При изменении состояния системы отправлять электронное письмо по адресу:

Модуль **Выключение компьютера** позволяет:

- »выключить компьютер;
- »перезагрузить компьютер;
- »приостановить работу компьютера;
- »погрузить компьютер в сон.

Возможна настройка ежедневного применения данных действий в заданное время.

Так как выключение и перезагрузка — критичные для функционирования компьютера операции, то по умолчанию настройка выставлена в значение **Продолжить работу**. Для выключения, перезагрузки или перехода в энергосберегающие режимы нужно отметить соответствующий пункт и нажать **Применить**.

Для ежедневного автоматического выключения компьютера, перезагрузки, а также перехода в энергосберегающие режимы необходимо отметить соответствующий пункт и задать желаемое время. Например, для выключения компьютера следует отметить пункт **Выключать компьютер каждый день в**, задать время выключения в поле ввода слева от этого флажка и нажать кнопку **Применить**.



Примечание

Для возможности настройки оповещений на e-mail, должен быть установлен пакет *state-change-notify-postfix*:

```
# apt-get install state-change-notify-postfix
```

Для настройки оповещений необходимо отметить пункт **При изменении состояния системы отправлять электронное письмо по адресу**, ввести e-mail адрес и нажать кнопку **Применить**:

☒ Продолжить работу
☐ Выключить компьютер сейчас
☐ Перезагрузить компьютер сейчас
☐ Приостановить компьютер сейчас
☐ Погрузить компьютер в сон сейчас

☐ Выключать компьютер каждый день в: 23:00:00
☒ Перезагружать компьютер каждый день в: 11:22:00
☐ Приостанавливать компьютер каждый день в: 23:00:00
☐ Погружать компьютер в сон каждый день в: 23:00:00

☒ При изменении состояния системы отправлять электронное письмо по адресу:
user_freeipa@example.test

Применить **Сбросить**

По указанному адресу, при изменении состоянии системы будут приходить электронные письма. Например, при включении компьютера, содержание письма будет следующее:

```
Tue Jun 16 11:46:59 EET 2022: The server.test.alt is about to start.
```

При выключении:

```
Tue Jun 16 12:27:02 EET 2022: The server.test.alt is about to shutdown.
```

Кнопка **Сбросить** возвращает сделанный выбор к безопасному значению по умолчанию: **Продолжить работу**, перечитывает расписания и выставляет отметки для ежедневного автоматического действия в соответствие с прочитанным.

Глава 15. Соединение удалённых офисов (OpenVPN-сервер)

15.1. Настройка OpenVPN-сервера

15.2. Настройка клиентов

Альт СП Сервер предоставляет возможность безопасного соединения удалённых офисов используя технологию VPN (англ. Virtual Private Network — виртуальная частная сеть), которая позволяет организовать безопасные зашифрованные соединения через публичные сети (например, Интернет) между удалёнными офисами или локальной сетью и удалёнными пользователями. Таким образом, вы можете связать два офиса организации, что, делает работу с документами, расположенными в сети удалённого офиса, более удобной.

Помимо соединения целых офисов, также существует возможность организовать доступ в офисную сеть для работы в ней извне. Это означает, например, что сотрудник может работать в своём привычном окружении, даже находясь в командировке или просто из дома.

15.1. Настройка OpenVPN-сервера

Для организации VPN соединения на стороне сервера предусмотрен модуль ЦУС **OpenVPN-сервер** (пакет *alterator-openvpn-server*) из раздела **Серверы**.

☐ Включить службу OpenVPN

Тип: Маршрутизируемое (TUN)

Сети сервера: 192.168.0.0/255.255.255.0 Удалить

Новая сеть:

Маска сети: /24 (255.255.255.0)

Добавить

VPN сеть: 10.8.0.0

Маска сети: /24 (255.255.255.0)

Алгоритм шифрования: default

Алгоритм шифрования TLS: default

Алгоритм хэширования: default

☐ Отключить согласование алгоритмов шифрования (NCP)

Порт: 1194

☐ Сжатие LZO

☐ Использовать соединение TCP

Сертификат и ключ SSL...

Положить сертификат УЦ: Обзор... Файл не выбран. Положить

Сети клиентов...

Применить Сбросить

Используя модуль **OpenVPN-сервер** можно:

- » включить/отключить OpenVPN-сервер;
- » настроить параметры сервера: тип, сети сервера, использование сжатия и т.д.;
- » управлять сертификатами сервера;
- » настроить сети клиентов.

Особое внимание при планировании и настройке подключений следует обратить на используемые сети. Они *не должны* пересекаться.

Для создания соединения необходимо установить флажок **Включить службу OpenVPN**, выбрать тип подключения: маршрутизируемое (используется TUN) или через мост (используется TAP), и проверить открываемую по соединению сеть (обычно это локальная сеть в виде IP-адреса и маски подсети).

Для настройки сертификата и ключа ssl необходимо нажать на кнопку **Сертификат и ключ ssl**... Откроется окно модуля **Управление ключами SSL** (пакет *alterator-sslkey*):

Настройки SSL

Общее имя (CN):
(имя компьютера для сервера или что-либо другое для клиента)

Страна (C):
(двухбуквенный код страны)

Местоположение (L):
(название города или области, написанное латинскими буквами)

Организация (O):
(название организации, написанное латинскими буквами)

Подразделение (OU):
(название подразделения, написанное латинскими буквами)

E-mail адрес:
(ваш адрес электронной почты)

☒ (Пере)создать ключ и запрос на подпись **Подтвердить**

Здесь нужно заполнить поле **Общее имя (CN)** и поле **Страна (C)** (прописными буквами), отметить пункт **(Пере)создать ключ и запрос на подпись** и нажать кнопку **Подтвердить**. После чего станет активной кнопка **Забрать запрос на подпись**:

Подпись

Забрать запрос на подпись

Положить сертификат, подписанный УЦ: Файл не выбран.

Положить

Если нажать на кнопку **Забрать запрос на подпись**, появится диалоговое окно с предложением сохранить файл **openvpn-server.csr**. Необходимо сохранить этот файл на диске.

В модуле **Управление ключами SSL** появился новый ключ *openvpn-server* (Нет сертификата):

SSL ключи:

ahhttpd (истекает: 09.02.2024)

openvpn-server (Нет сертификата)

postfix (истекает: 09.02.2024)

Новый...

Изменить...

Чтобы подписать сертификат, необходимо перейти в модуль **Удостоверяющий Центр** → **Управление сертификатами**, нажать кнопку **Обзор**, указать путь до полученного файла **openvpn-server.csr** и загрузить запрос:

Подписать сертификат

Обзор...

openvpn-server.csr

Загрузить запрос

В результате на экране появится две группы цифр и кнопка **Подписать**. Необходимо нажать на кнопку **Подписать** и сохранить файл **output.pem** (подписанный сертификат).

Certificate Request:

Data:

Version: 1 (0x0)

Subject: CN = openvpn-server, C = RU, L = Kaliningrad

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

00:b3:62:2a:d4:f8:e1:db:5d:e6:49:ce:b3:79:29:
bf:dc:f2:19:ba:63:3a:7e:52:30:23:3e:13:83:92:
d0:ae:9d:cb:82:2a:44:f2:1c:d2:e6:92:47:86:07:
16:cf:03:5a:be:80:58:b0:d9:4a:e4:de:c2:b7:68:
20:23:2a:9e:e1:a2:50:52:61:99:79:5a:af:42:9f:
78:2d

Exponent: 65537 (0x10001)

Attributes:

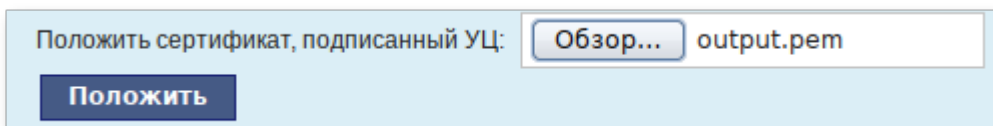
a0:00

Signature Algorithm: sha256WithRSAEncryption

7c:2f:14:8f:80:e6:96:cd:ab:93:16:d0:a9:9d:59:b9:e9:80:
6c:a7:29:bf:b9:ca:15:89:55:8e:3c:78:03:55:21:77:97:4d:
0d:43:95:14:13:72:f3:2e:69:e0:f7:6d:5e:a2:ca:c9:34:b2:
8b:bd:4b:6d:d9:9f:4e:ce:6d:09:65:f6:7d:bd:b7:4a:02:d2:
21:40:97:5a

Подписать

Далее в разделе **Управление ключами SSL**, необходимо выделить ключ *openvpn-server (Нет сертификата)* и нажать кнопку **Изменить**. В появившемся окне, в пункте **Положить сертификат, подписанный УЦ** нужно нажать кнопку **Обзор**, указать путь до файла **output.pem** и нажать кнопку **Положить**:

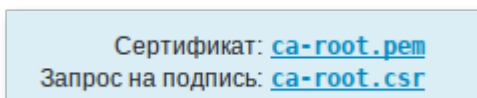


Положить сертификат, подписанный УЦ:

Положить

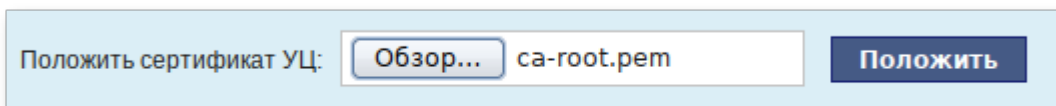
В модуле **Управление ключами SSL**, видно, что изменился ключ *openvpn-server* (*истекает_u_data*). Ключ создан и подписан.

Для того чтобы положить сертификат УЦ, необходимо найти его в модуле **Удостоверяющий Центр**, нажать на ссылку **Управление УЦ** и забрать сертификат, нажав на ссылку **Сертификат: ca-root.pem**:



Сертификат: [ca-root.pem](#)
Запрос на подпись: [ca-root.csr](#)

В модуле **OpenVPN-сервер**, в графе **Положить сертификат УЦ**: при помощи кнопки **Обзор** указать путь к файлу **ca-root.pem** и нажать кнопку **Положить**:



Положить сертификат УЦ:

Положить

Появится сообщение: «Сертификат УЦ успешно загружен».

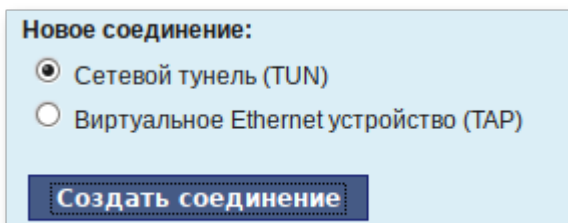
Для включения OpenVPN необходимо отметить пункт **Включить службу OpenVPN** и нажать кнопку **Применить**.

Если необходимо организовать защищённое соединение между двумя локальными сетями, воспользуйтесь модулем **OpenVPN-соединения** (раздел **Сеть**).

15.2. Настройка клиентов

Со стороны клиента соединение настраивается в модуле ЦУС **OpenVPN-соединения** (пакет *alterator-net-openvpn*) из раздела **Сеть**. Доступ к настроенной приватной сети могут получить пользователи, подписавшие свои ключи и получившие сертификат в удостоверяющем центре на том же сервере.

Для создания нового соединения необходимо отметить пункт **Сетевой туннель (TUN)** или **Виртуальное Ethernet устройство (TAP)** и нажать кнопку **Создать соединение**. Должен быть выбран тот же тип, что и на стороне сервера.



Новое соединение:

☒ Сетевой туннель (TUN)
☐ Виртуальное Ethernet устройство (TAP)

Создать соединение

Обратите внимание, что на стороне клиента, должен быть выбран тот же тип виртуального устройства, что и на стороне сервера. Для большинства случаев подходит маршрутизируемое подключение.

Помимо этого нужно подписать ключ **openvpn** в модуле **Удостоверяющий Центр** (пакет *alterator-ca*) из раздела **Система**.

В результате станут доступны настройки соединения. На клиенте в модуле OpenVPN-соединение необходимо указать:

- » **Состояние** — «запустить»;
- » **Сервер** — IP адрес сервера или домен;
- » **Порт** — 1194;
- » **Ключ** — выбрать подписанный на сервере ключ.

Для применения настроек, нажать кнопку **Применить**. Состояние с **Выключено** должно поменяться на **Включено**.

tun1

Состояние: выключено запустить

Сервер: 192.168.0.122

Порт: 1194

Ключ: openvpn

Управление ключами...

☐ Запускать при загрузке

☐ Маршрут по умолчанию через VPN

☐ Сжатие LZO

☐ Использовать соединение TCP

Алгоритм шифрования: default

Алгоритм шифрования TLS: default

Алгоритм хэширования: default

☒ Отключить согласование алгоритмов шифрования (NCP)

Применить Сбросить Удалить соединение

Положить сертификат УЦ: Обзор... ca-root.pem Положить

Сертификат УЦ успешно загружен

Проверить, появилось ли соединение с сервером можно командой

```
ip addr
```

должно появиться новое соединение tun1. При обычных настройках это может выглядеть так:

```
tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 100
    link/[none]
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
```

Глава 16. Доступ к службам сервера из сети Интернет

16.1. Внешние сети

16.2. Список блокируемых хостов

16.1. Внешние сети

Сервер предоставляет возможность организовать доступ к своим службам извне. Например, можно предоставить доступ к корпоративному веб-сайту из сети Интернет. Для обеспечения такой возможности необходимо разрешить входящие соединения на внешних интерфейсах. По умолчанию такие соединения блокируются.

Для разрешения внешних и внутренних входящих соединений предусмотрен раздел ЦУС **Брандмауэр**. В списке **Разрешить входящие соединения на внешних интерфейсах** модуля **Внешние сети** (пакет *alterator-net-iptables*) перечислены наиболее часто используемые службы, отметив которые, вы делаете их доступными для соединений на внешних сетевых интерфейсах. Если вы хотите предоставить доступ к службе, отсутствующей в списке, задайте используемые этой службой порты в соответствующих полях.

Версия IP: ☒ Включить брандмауэр

Выберите режим работы:

Выберите внешние интерфейсы: ☐ enp0s3 (Intel Corporation 82540EM Gigabit Ethernet Controller) 192.168.0.91/24

Разрешить входящие соединения на внешних интерфейсах:

Службы:

- ☒ Центр управления системой (www)
- ☐ Система печати CUPS
- ☐ DHCP
- ☐ DNS
- ☐ Передача файлов (FTP)
- ☐ Почтовый сервер (IMAP)
- ☐ LDAP
- ☒ OpenVPN
- ☐ Почтовый сервер (POP3)
- ☐ Прокси-сервер
- ☐ Файловый сервер (Samba)

Можно выбрать один из двух режимов работы:

- »Роутер. В этом режиме перенаправление пакетов между сетевыми интерфейсами происходит без трансляции сетевых адресов.

- » Шлюз (NAT). В этом режиме будет настроена трансляция сетевых адресов (NAT) при перенаправлении пакетов на внешние интерфейсы. Использование этого режима имеет смысл, если у вас настроен, по крайней мере, один внешний и один внутренний интерфейс.



Примечание

В любом режиме включено только перенаправление пакетов с внутренних интерфейсов. Перенаправление пакетов с внешних интерфейсов всегда выключено.



Примечание

Все внутренние интерфейсы открыты для любых входящих соединений.

За дополнительной информацией по настройке обращайтесь к встроенной справке модуля ЦУС.

16.2. Список блокируемых хостов

Модуль ЦУС **Список блокируемых хостов** (пакет *alterator-net-iptables*) предназначен для блокирования любого трафика с указанными узлами. Данный модуль позволяет блокировать любой сетевой трафик с указанных в списке узлов (входящий, исходящий и пересылаемый).

Блокирование трафика с указанных в списке узлов начинается после установки флажка **Использовать чёрный список**.

The screenshot shows a configuration window titled "Чёрный список:" (Blacklist:). It contains a dropdown menu for "Версия IP:" (IP version) set to "IPv4". Below it is a checkbox labeled "Использовать чёрный список" (Use blacklist), which is currently unchecked. To the right of the checkbox is a "Удалить" (Delete) button. Below the checkbox is a large empty text area for listing IP addresses. At the bottom, there is a label "Добавить IP адрес сети или хоста:" (Add IP address of network or host:) followed by an input field and a "Добавить" (Add) button.

Для добавления блокируемого узла необходимо ввести IP-адрес в поле **Добавить IP адрес сети или хоста** и нажать кнопку **Добавить**.

Для удаления узла из списка выберите его и нажмите кнопку **Удалить**.

Глава 17. Статистика

17.1. Сетевой трафик

17.1. Сетевой трафик



Все входящие и исходящие с сервера сетевые пакеты могут подсчитываться, и выводиться по запросу для анализа.


Модуль **Сетевой трафик** (пакет *alterator-ulogd*) из раздела **Статистика** предназначен для просмотра статистики входящих и исходящих с сервера сетевых пакетов. Данный модуль позволяет оценить итоговый объём полученных и переданных данных за всё время работы сервера, за определённый период времени и по каждой службе отдельно.

Для включения сбора данных необходимо установить флажок **Включить сбор данных**, и нажать кнопку **Применить**.




☐ Включить сбор данных

Применить

Период с: 1970-01-01  по 1970-01-01 

Интерфейс: enp0s3 - 192.168.0.91 

Показать

Служба 	Входящий трафик(Кб) 	Исходящий трафик(Кб) 

Для просмотра статистики укажите период (в виде начальной и конечной дат). Дата указывается в формате YYYY-MM-DD (год-месяц-день) или выбирается из календаря справа от поля ввода даты. Из списка доступных сетевых интерфейсов необходимо выбрать интересующий и нажать на кнопку **Показать**.

☒ Включить сбор данных

Применить

Период с: по

Интерфейс:

Показать

Служба	Входящий трафик(Кб)	Исходящий трафик(Кб)
Центр управления системой (www)	0.0	0.0
Система печати CUPS	0.0	0.0
DHCP	0.0	0.0
DNS	0.0	0.0
Передача файлов (FTP)	0.0	0.0
Почтовый сервер (IMAP)	0.0	0.0
LDAP	0.0	0.0
OpenVPN	0.0	0.0
Почтовый сервер (POP3)	0.0	0.0
Прокси-сервер	0.0	0.0
Файловый сервер (Samba)	0.0	0.0
Почтовый сервер (SMTP)	0.0	0.0

Трафик на указанном интерфейсе за заданный период показывается в виде:

- служба (название протокола);
- входящий трафик в килобайтах;
- исходящий трафик в килобайтах.

17.2. Прокси-сервер

Пересылка каждого запроса во внешнюю сеть фиксируется прокси-сервером в специальном журнале. На основании этих данных автоматически формируются отчёты о статистике использования ресурсов сети, в том числе потраченного времени и количества переданных данных (трафика).

Статистика не собирается по умолчанию. Для включения сбора статистики и просмотра отчётов воспользуйтесь модулем ЦУС **Прокси-сервер** (пакет *alterator-squidmill*) из раздела **Статистика**.

Включить сбор данных прокси-сервера: ☐ [Применить](#)

Общий объем трафика принятый за [сегодня](#) [▼](#)
[всеми пользователями](#) [▼](#)
[со всех сайтов](#) [▼](#)
составляет **0.00 Б**

[Обновить](#)

Список сайтов, набравших [любой объем](#) [▼](#) данных

UID/IP-адрес	Количество	Сайт/домен	Время последнего запроса
--------------	------------	------------	--------------------------

Для включения сбора статистики прокси-сервера установите флажок **Включить сбор данных прокси-сервера**.

В том случае, если на прокси-сервере производилась аутентификация пользователей, отчёты будут содержать данные об обращениях каждого пользователя. Иначе отчёты будут формироваться только на основании адресов локальной сети.

Для показа отчёта задайте условия фильтра и нажмите кнопку **Показать**. Данные в таблице будут отсортированы по объёму трафика в порядке убывания.

Для учёта пользователей в статистике необходимо добавить хотя бы одно правило. Самое очевидное правило — запрет неаутентифицированных пользователей. Только после этого в статистике начнут показываться пользователи.

Глава 18. Прочие возможности ЦУС

Возможности Альт СП Сервер не ограничиваются только теми, что были описаны выше. Вы всегда можете поискать другие модули, предоставляющие прочие возможности для настройки системы в веб-интерфейсе.

Глава 19. Права доступа к модулям

Администратор системы (root) имеет доступ ко всем модулям, установленным в системе, и может назначать права доступа для пользователей к определенным модулям.

Для разрешения доступа пользователю к конкретному модулю, администратору в веб-интерфейсе ЦУС необходимо выбрать нужный модуль и нажать ссылку **Параметры доступа к модулю**, расположенную в нижней части окна модуля:

Брандмауэр

Внешние сети

Перенаправление портов

Список блокируемых хостов

Внутренние сети

Ручной режим управления

[Параметры доступа к модулю...](#)

В открывшемся окне, в списке **Новый пользователь** необходимо выбрать пользователя, который получит доступ к данному модулю, и нажать кнопку **Добавить**.

Параметры доступа к модулю

Следующие пользователи имеют доступ:

user	Удалить
------	---------

Новый пользователь:

	Добавить
--	----------

Замечание: Все ваши изменения вступят в силу после перезапуска HTTP сервера.

Перезапустить HTTP-сервер

Для сохранения настроек необходимо перезапустить HTTP-сервер, для этого достаточно нажать кнопку **Перезапустить HTTP-сервер**.

Для удаления доступа пользователя к определенному модулю, администратору, в окне этого модуля необходимо нажать ссылку **Параметры доступа к модулю**, в открывшемся окне в списке пользователей которым разрешен доступ, должен выбрать пользователя, нажать кнопку **Удалить** и перезапустить HTTP-сервер.

Системный пользователь, пройдя процедуру аутентификации, может просматривать и вызывать модули, к которым он имеет доступ.

Часть VI. Основы администрирования Linux

Содержание

- 20. Общие принципы работы ОС
- 21. Режим суперпользователя
- 22. Команда sudo
- 23. Управление пользователями
- 24. Система инициализации systemd и sysvinit
- 25. Документация

Глава 20. Общие принципы работы ОС

20.1. Процессы и файлы

20.2. Работа с наиболее часто используемыми компонентами

20.3. Стыкование команд в системе Linux

20.1. Процессы и файлы

ОС Альт СП Сервер является многопользовательской интегрированной системой. Это значит, что она разработана в расчете на одновременную работу нескольких пользователей.

Пользователь может либо сам работать в системе, выполняя некоторую последовательность команд, либо от его имени могут выполняться прикладные процессы.

Пользователь взаимодействует с системой через командный интерпретатор. Командный интерпретатор представляет собой прикладную программу, которая принимает от пользователя команды или набор команд и транслирует их в системные вызовы к ядру системы. Интерпретатор позволяет пользователю просматривать файлы, передвигаться по дереву файловой системы, запускать прикладные процессы. Все командные интерпретаторы UNIX имеют развитый командный язык и позволяют писать достаточно сложные программы, упрощающие процесс администрирования системы и работы с ней.

20.1.1. Процессы функционирования ОС

Все программы, которые выполняются в текущий момент времени, называются процессами. Процессы можно разделить на два основных класса: системные процессы и пользовательские процессы.

Системные процессы — программы, решающие внутренние задачи ОС, например, организацию виртуальной памяти на диске или предоставляющие пользователям те или иные сервисы (процессы-службы).

Пользовательские процессы — процессы, запускаемые пользователем из командного интерпретатора для решения задач пользователя или управления системными процессами. Linux изначально разрабатывался как многозадачная система. Он использует технологии, опробованные и отработанные другими реализациями UNIX, которые существовали ранее.

Фоновый режим работы процесса — режим, когда программа может работать без взаимодействия с пользователем. В случае необходимости интерактивной работы с пользователем (в общем случае) процесс будет «остановлен» ядром, и работа его продолжается только после перевода его в «нормальный» режим работы.

20.1.2. Файловая система ОС

В ОС использована файловая система Linux, которая, в отличие от файловых систем DOS и Windows[™], является единым деревом. Корень этого дерева — каталог, называемый root (рут) и обозначаемый /.

Части дерева файловой системы могут физически располагаться в разных разделах разных дисков или вообще на других компьютерах — для пользователя это прозрачно. Процесс присоединения файловой системы раздела к дереву называется монтированием, удаление — размонтированием. Например, файловая система CD-ROM в дистрибутиве монтируется по умолчанию в каталог **/media/cdrom** (путь в дистрибутиве обозначается с использованием */*, а не **, как в DOS/Windows).

Текущий каталог обозначается **.**.

20.1.3. Структура каталогов

Корневой каталог **/**:

- **/bin** — командные оболочки (shell), основные утилиты;
- **/boot** — содержит ядро системы;
- **/dev** — псевдофайлы устройств, позволяющие работать с устройствами напрямую. Файлы в **/dev** создаются сервисом **udev**
- **/etc** — общесистемные конфигурационные файлы для большинства программ в системе;
- **/etc/rc?.d**, **/etc/init.d**, **/etc/rc.boot**, **/etc/rc.d** — каталоги, где расположены командные файлы, выполняемые при запуске системы или при смене её режима работы;
- **/etc/passwd** — база данных пользователей, в которой содержится информация об имени пользователя, его настоящем имени, личном каталоге, его зашифрованный пароль и другие данные;
- **/etc/shadow** — теневая база данных пользователей. При этом информация из файла **/etc/passwd** перемещается в **/etc/shadow**, который недоступен для чтения всем, кроме пользователя **root**. В случае использования альтернативной схемы управления теневыми паролями (TCB), все теневые пароли для каждого пользователя располагаются в каталоге **/etc/tcb/имя пользователя/shadow**;
- **/home** — домашние каталоги пользователей;
- **/lib** — содержит файлы динамических библиотек, необходимых для работы большей части приложений, и подгружаемые модули ядра;
- **/lost+found** — восстановленные файлы;
- **/media** — подключаемые носители (каталоги для монтирования файловых систем сменных устройств);
- **/mnt** — точки временного монтирования;
- **/opt** — вспомогательные пакеты;
- **/proc** — виртуальная файловая система, хранящаяся в памяти компьютера при загруженной ОС. В данном каталоге расположены самые свежие сведения обо всех процессах, запущенных на компьютере.
- **/root** — домашний каталог администратора системы;
- **/run** — файлы состояния приложений;

- » **/sbin** — набор программ для административной работы с системой (системные утилиты);
- » **/selinux** — виртуальная файловая система SELinux;
- » **/srv** — виртуальные данные сервисных служб;
- » **/sys** — файловая система, содержащая информацию о текущем состоянии системы;
- » **/tmp** — временные файлы.
- » **/usr** — пользовательские двоичные файлы и данные, используемые только для чтения (программы и библиотеки);
- » **/var** — файлы для хранения изменяющихся данных (рабочие файлы программ, очереди, журналы).

Каталог **/usr**:

- » **/usr/bin** — дополнительные программы для всех учетных записей;
- » **/usr/sbin** — команды, используемые при администрировании системы и не предназначенные для размещения в файловой системе root;
- » **/usr/local** — место, где рекомендуется размещать файлы, установленные без использования пакетных менеджеров, внутренняя организация каталогов практически такая же, как и корневого каталога;
- » **/usr/man** — каталог, где хранятся файлы справочного руководства **man**;
- » **/usr/share** — каталог для размещения общедоступных файлов большей части приложений.

Каталог **/var**:

- » **/var/log** — место, где хранятся файлы аудита работы системы и приложений;
- » **/var/spool** — каталог для хранения файлов, находящихся в очереди на обработку для того или иного процесса (очереди печати, непочитанные или не отправленные письма, задачи cron т.д.).

20.1.4. Организация файловой структуры

Система домашних каталогов пользователей помогает организовывать безопасную работу пользователей в многопользовательской системе. Вне своего домашнего каталога пользователь обладает минимальными правами (обычно чтение и выполнение файлов) и не может нанести ущерб системе, например, удалив или изменив файл.

Кроме файлов, созданных пользователем, в его домашнем каталоге обычно содержатся персональные конфигурационные файлы некоторых программ.

Маршрут (путь) — это последовательность имён каталогов, представляющая собой путь в файловой системе к данному файлу, где каждое следующее имя отделяется от предыдущего наклонной чертой (слешем). Если название маршрута начинается со слеша, то путь в искомый файл начинается от корневого каталога всего дерева системы. В обратном случае, если название маршрута начинается непосредственно с имени файла, то путь к искомому файлу должен начинаться от текущего каталога (рабочего каталога).

Имя файла может содержать любые символы за исключением косой черты (/). Однако следует избегать применения в именах файлов большинства знаков препинания и непечатаемых символов. При выборе имен файлов рекомендуется ограничиться следующими символами:

- **строчные и ПРОПИСНЫЕ** буквы. Следует обратить внимание на то, что регистр всегда имеет значение;
- символ подчеркивания (_);
- точка (.).

Для удобства работы точку можно использовать для отделения имени файла от расширения файла. Данная возможность может быть необходима пользователям или некоторым программам, но не имеет значение для shell.

20.1.5. Имена дисков и разделов

Все физические устройства вашего компьютера отображаются в каталог **/dev** файловой системы дистрибутива (об этом — ниже). Диски (в том числе IDE/SATA/SCSI/SAS жёсткие диски, USB-диски) имеют имена:

- **/dev/sda** — первый диск;
- **/dev/sdb** — второй диск;
- и т.д.

Диски обозначаются **/dev/sdX**, где *X* — a, b, c, d, e, ... в зависимости от порядкового номера диска на шине.

Раздел диска обозначается числом после его имени. Например, **/dev/sdb4** — четвертый раздел второго диска.

20.1.6. Разделы, необходимые для работы ОС

Для работы ОС на жестком диске (дисках) должны быть созданы, по крайней мере, два раздела: корневой (то есть тот, который будет содержать каталог /) и раздел подкачки (swp). Размер последнего, как правило, составляет от однократной до двукратной величины оперативной памяти компьютера. Если на диске много свободного места, то можно создать отдельные разделы для каталогов **/usr**, **/home**, **/var**.

20.2. Работа с наиболее часто используемыми компонентами

20.2.1. Командные оболочки (интерпретаторы)

Для управления ОС используются командные интерпретаторы (shell).

Зайдя в систему, Вы увидите приглашение — строку, содержащую символ «\$» (далее этот символ будет обозначать командную строку). Программа ожидает ваших команд. Роль командного интерпретатора — передавать ваши команды операционной системе. По своим функциям он соответствует **command.com** в DOS, но несравненно мощнее. При помощи командных интерпретаторов можно писать небольшие программы — сценарии (скрипты). В Linux доступны следующие командные оболочки:

- **bash** — самая распространенная оболочка под linux. Она ведет историю команд и предоставляет возможность их редактирования;
- **pdksh** — клон korn shell, хорошо известной оболочки в UNIX™ системах.

Проверить, какая оболочка используется в данный момент можно, выполнив команду:

```
$ echo $SHELL
```

Оболочкой по умолчанию является Bash (Bourne Again Shell) — самая распространённая оболочка под Linux, которая ведет историю команд и предоставляет возможность их редактирования. В дальнейшем описании работы с Альт СП Сервер будут использоваться примеры с использованием этой оболочки.

20.2.2. Командная оболочка Bash

В Bash имеется несколько приемов для работы со строкой команд. Например, можно использовать следующие сочетания:

- **Ctrl+A** — перейти на начало строки;
- **Ctrl+U** — удалить текущую строку;
- **Ctrl+C** — остановить текущую задачу.

Для ввода нескольких команд одной строкой можно использовать разделитель «;». По истории команд можно перемещаться с помощью клавиш ↑ («вверх») и ↓ («вниз»).

Чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, можно нажать **Ctrl+R** и начать вводить символы ранее введенной команды.

Для просмотра истории команд можно воспользоваться командой **history**. Команды, присутствующие в истории, отображаются в списке пронумерованными. Чтобы запустить конкретную команду необходимо набрать:

```
!номер команды
```

Если ввести:

```
!!
```

запустится последняя из набранных команд.

В Bash имеется возможность самостоятельного завершения имен команд из общего списка команд, что облегчает работу при вводе команд, в случае, если имена программ и команд слишком длинны. При нажатии клавиши **Tab** Bash завершает имя команды, программы или каталога, если не существует нескольких альтернативных вариантов. Например, чтобы использовать программу декомпрессии **gunzip**, можно набрать следующую команду:


```
gu
```

Затем нажать клавишу **Tab**. Так как в данном случае существует несколько возможных вариантов завершения команды, то необходимо повторно нажать клавишу **Tab**, чтобы получить список имен, начинающихся с **gu**.

В предложенном примере можно получить следующий список:

```
$ gu
guile gunzip gupnp-binding-tool
```

Если набрать: **n** (**gunzip** — это единственное имя, третьей буквой которого является «n»), а затем нажать клавишу **Tab**, то оболочка самостоятельно дополнит имя. Чтобы запустить команду нужно нажать **Enter**.

Программы, вызываемые из командной строки, Bash ищет в каталогах, определяемых в системной переменной **\$PATH**. По умолчанию в этот перечень каталогов не входит текущий каталог, обозначаемый **./** (точка слеш) (если только не выбран один из двух самых слабых уровней защиты). Поэтому, для запуска программы из текущего каталога, необходимо использовать команду (в примере запускается команда **prog**):

```
./prog
```

20.2.3. Команда

Простейшая команда состоит из одного «слова», например, команда **cal**, выводящая календарь на текущий месяц.

```
$ cal
      Март 2023
Пн Вт Ср Чт Пт Сб Вс
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

20.2.4. Команда и параметры

```
$ cal 1 2024
      Январь 2024
Пн Вт Ср Чт Пт Сб Вс
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Команда **cal 1 2024** состоит из двух частей — собственно команды **cal** и «остального». То, что следует за командой называется *параметрами* (или аргументами) и они вводятся для изменения поведения команды. В большинстве случаев, первое слово считается именем команды, а остальные — её параметрами.

20.2.5. Команда и ключи

Для решения разных задач одни и те же действия необходимо выполнять по-разному. Например, для синхронизации работ в разных точках земного шара лучше использовать единое для всех время (по Гринвичу), а для организации собственного рабочего дня — местное время (с учётом сдвига по часовому поясу и разницы зимнего и летнего времени). И то, и другое время показывает команда **date**, только для работы по Гринвичу ей нужен дополнительный параметр **-u** (он же **--universal**).

```
$ date
Пн мар 6 10:42:35 MSK 2023
$ date -u
Пн мар 6 07:43:07 UTC 2023
```

Такого рода параметры называются *ключами* или *модификаторами выполнения*. Ключ принадлежит данной конкретной команде и сам по себе смысла не имеет. Этим он отличается от других параметров (например, имён файлов, чисел), имеющих собственный смысл, не зависящий ни от какой команды. Каждая команда может распознавать некоторый набор ключей и соответственно изменять своё поведение. Один и тот же ключ может определять для разных команд совершенно разные значения.

Для формата ключей нет жёсткого стандарта, однако существуют договорённости:

- Если ключ начинается на -, то это *однобуквенный ключ*. За -, как правило, следует один символ, чаще всего буква, обозначающая действие или свойство, которое этот ключ придаёт команде. Так проще отличать ключи от других параметров.
- Если ключ начинается на --, то он называется *полнословным ключом*. Полнословный формат ключа начинается на два знака --, за которыми следует полное имя обозначаемого этим ключом содержания.

Некоторые ключи имеют и однобуквенный, и полнословный формат, а некоторые — только полнословный.

Информацию о ресурсах каждой команды можно получить, используя ключ **--help**. К примеру, получить подсказку о том, что делает команда **rm**, можно, набрав в терминале **rm --help**.

20.2.6. Обзор основных команд системы

Все команды, приведенные ниже, могут быть запущены в режиме консоли. Для получения более подробной информации используйте команду **man**. Пример:

```
$ man ls
```



Примечание

Параметры команд обычно начинаются с символа «-», и обычно после одного символа «-» можно указать сразу несколько опций. Например, вместо команды **ls -l -F** можно ввести команду **ls -lF**

Учетные записи пользователей

Команда **su**

Команда **su** позволяет изменить «владельца» текущего сеанса (сессии) без необходимости завершать сеанс и открывать новый.

Синтаксис:

```
su [ОПЦИИ...] [ПОЛЬЗОВАТЕЛЬ]
```

Команду можно применять для замены текущего пользователя на любого другого.

Команда **id**

Команда **id** выводит информацию о пользователе и группах, в которых он состоит для заданного пользователя или о текущем пользователе (если ничего не указано).

Синтаксис:

```
id [ОПЦИИ...] [ПОЛЬЗОВАТЕЛЬ]
```

Команда **passwd**

Команда **passwd** меняет (или устанавливает) пароль, связанный с входным_именем пользователя.

Обычный пользователь может менять только пароль, связанный с его собственным входным_именем.

Команда запрашивает у обычных пользователей старый пароль (если он был), а затем дважды запрашивает новый. Новый пароль должен соответствовать техническим требованиям к паролям, заданным администратором системы.

Основные операции с файлами и каталогами

Команда **ls**

Команда **ls** (list) печатает в стандартный вывод содержимое каталогов.

Синтаксис:

```
ls [ОПЦИИ...] [ФАЙЛ...]
```

Основные опции:

- » **-a** — просмотр всех файлов, включая скрытые;
- » **-l** — отображение более подробной информации;
- » **-R** — выводить рекурсивно информацию о подкаталогах.

Команда **cd**

Команда **cd** предназначена для смены каталога. Команда работает как с абсолютными, так и с относительными путями. Если каталог не указан, используется значение переменной окружения **\$HOME** (домашний каталог пользователя). Если каталог задан полным маршрутным именем, он становится текущим. По отношению к новому каталогу нужно иметь право на выполнение, которое в данном случае трактуется как разрешение на поиск.

Синтаксис:

```
cd [-L|-P] [КАТАЛОГ]
```

Если в качестве аргумента задано «-», то это эквивалентно **\$OLDPWD**. Если переход был осуществлен по переменной окружения **\$CDPATH** или в качестве аргумента был задан «-» и смена каталога была успешной, то абсолютный путь нового рабочего каталога будет выведен на стандартный вывод.

Пример. Находясь в домашнем каталоге перейти в его подкаталог **docs/** (относительный путь):

```
cd docs/
```

Сделать текущим каталог **/usr/bin** (абсолютный путь):

```
cd /usr/bin/
```

Сделать текущим родительский каталог:

```
cd ..
```

Вернуться в предыдущий каталог:

```
cd -
```

Сделать текущим домашний каталог:

```
cd
```

Команда pwd

Команда **pwd** выводит абсолютный путь текущего (рабочего) каталога.

Синтаксис:

```
pwd [-L|-P]
```

Опции:

- » **-P** — не выводить символические ссылки;
- » **-L** — выводить символические ссылки.

Команда rm

Команда **rm** служит для удаления записей о файлах. Если заданное имя было последней ссылкой на файл, то файл уничтожается.



Предупреждение

Удалив файл, вы не сможете его восстановить!

Синтаксис:

```
rm [ОПЦИИ...] <ФАЙЛ>
```

Основные опции:

- » **-f** — никогда не запрашивать подтверждения;
- » **-i** — всегда запрашивать подтверждение;
- » **-r, -R** — рекурсивно удалять содержимое указанных каталогов.

Пример. Удалить все файлы **html** в каталоге **~/html**:

```
rm -i ~/html/*.html
```

Команда **mkdir**

mkdir — команда для создания новых каталогов.

Синтаксис:

```
mkdir [-p] [-m права] <КАТАЛОГ...>
```

Команда **rmdir**

Команда **rmdir** удаляет каталоги из файловой системы. Каталог должен быть пуст перед удалением.

Синтаксис:

```
rmdir [ОПЦИИ] <КАТАЛОГ...>
```

Основные опции:

- » **-p** — удалить каталог и его потомки.

Команда **rmdir** часто заменяется командой **rm -rf**, которая позволяет удалять каталоги, даже если они не пусты.

Команда **cp**

Команда **cp** предназначена для копирования файлов из одного в другие каталоги.

Синтаксис:

```
cp [-fip] [ИСХ_ФАЙЛ...] [ЦЕЛ_ФАЙЛ...]
```

```
cp [-fip] [ИСХ_ФАЙЛ...] [КАТАЛОГ]
```

```
cp [-R] [[-H] | [-L] | [-P]] [-fip] [ИСХ_ФАЙЛ...] [КАТАЛОГ]
```

Основные опции:

- » **-p** — сохранять по возможности времена изменения и доступа к файлу, владельца и группу, права доступа;

- » **-i** — запрашивать подтверждение перед копированием в существующие файлы;
- » **-r, -R** — рекурсивно копировать содержимое каталогов.

Команда **mv**

Команда **mv** предназначена для перемещения файлов.

Синтаксис:

```
mv [-fi] [ИСХ_ФАЙЛ...] [ЦЕЛ_ФАЙЛ...]
```

```
mv [-fi] [ИСХ_ФАЙЛ...] [КАТАЛОГ]
```

В первой синтаксической форме, характеризующейся тем, что последний операнд не является ни каталогом, ни символической ссылкой на каталог, **mv** перемещает **исх_файл** в **цел_файл** (происходит переименование файла).

Во второй синтаксической форме **mv** перемещает исходные файлы в указанный каталог под именами, совпадающими с краткими именами исходных файлов.

Основные опции:

- » **-f** — не запрашивать подтверждения перезаписи существующих файлов;
- » **-i** — запрашивать подтверждение перезаписи существующих файлов.

Команда **cat**

Команда **cat** последовательно выводит содержимое файлов.

Синтаксис:

```
cat [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- » **-n, --number** — нумеровать все строки при выводе;
- » **-E, --show-ends** — показывать \$ в конце каждой строки.

Если файл не указан, читается стандартный ввод. Если в списке файлов присутствует имя «-», вместо этого файла читается стандартный ввод.

Команда **head**

Команда **head** выводит первые 10 строк каждого файла на стандартный вывод.

Синтаксис:

```
head [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- » **-n, --lines=[-]K** — вывести первые K строк каждого файла, а не первые 10;
- » **-q, --quiet** — не печатать заголовки с именами файлов.

Команда **less**

Команда **less** позволяет постранично просматривать текст (для выхода необходимо нажать **q**).

Синтаксис:

```
less ФАЙЛ
```

Команда **grep**

Команда **grep** имеет много опций и предоставляет возможности поиска символьной строки в файле.

Синтаксис:

```
grep [шаблон_поиска] ФАЙЛ
```

Команда **chmod**

Команда **chmod** предназначена для изменения прав доступа файлов и каталогов.

Синтаксис:

```
chmod [ОПЦИИ] РЕЖИМ[ , РЕЖИМ] ... <ФАЙЛ>
```

```
chmod [ОПЦИИ] --reference=ИФАЙЛ <ФАЙЛ>
```

Основные опции:

- » **-R** — рекурсивно изменять режим доступа к файлам, расположенным в указанных каталогах;
- » **--reference=ИФАЙЛ** — использовать режим файла ИФАЙЛ.

chmod изменяет права доступа каждого указанного файла в соответствии с правами доступа, указанными в параметре режим, который может быть представлен как в символьном виде, так и в виде восьмеричного, представляющего битовую маску новых прав доступа.

Формат символьного режима следующий:

```
[ugoa...][[+|=][разрешения...]]...
```

Здесь разрешения — это ноль или более букв из набора «**rwXst**» или одна из букв из набора «**ugo**».

Каждый аргумент — это список символьных команд изменения прав доступа, разделены запятыми. Каждая такая команда начинается с нуля или более букв «**ugoa**», комбинация которых указывает, чьи права доступа к файлу будут изменены: пользователя, владеющего файлом (**u**), пользователей, входящих в группу, к которой принадлежит файл (**g**), остальных пользователей (**o**) или всех пользователей (**a**). Если не задана ни одна буква, то автоматически будет использована буква «**a**», но биты, установленные в **umask**, не будут затронуты.

Оператор «+» добавляет выбранные права доступа к уже имеющимся у каждого файла, «-» удаляет эти права. «=» присваивает только эти права каждому указанному файлу.

Буквы «rwxXst» задают биты доступа для пользователей: «r» — чтение, «w» — запись, «x» — выполнение (или поиск для каталогов), «X» — выполнение/поиск только если это каталог или же файл с уже установленным битом выполнения, «s» — задать ID пользователя и группы при выполнении, «t» — запрет удаления.

Примеры. Позволить всем выполнять файл **f2**:

```
chmod +x f2
```

Запретить удаление файла **f3**:

```
chmod +t f3
```

Команда **chown**

Команда **chown** изменяет владельца и/или группу для каждого заданного файла.

Синтаксис:

```
chown [КЛЮЧ]...[ВЛАДЕЛЕЦ] [: [ГРУППА]] <ФАЙЛ>
```

Изменить владельца может только владелец файла или суперпользователь. Владелец не изменяется, если он не задан в аргументе. Группа также не изменяется, если не задана, но если после символического ВЛАДЕЛЬЦА стоит символ «:», подразумевается изменение группы на основную группу текущего пользователя. Поля ВЛАДЕЛЕЦ и ГРУППА могут быть как числовыми, так и символическими.

Примеры. Поменять владельца каталога **/u** на пользователя **test**:

```
chown test /u
```

Поменять владельца и группу каталога **/u**:

```
chown test:staff /u
```

Поменять владельца каталога **/u** и вложенных файлов на **test**:

```
chown -hR test /u
```

Поиск файлов

Команда **find**

Команда **find** предназначена для поиска всех файлов, начиная с корневого каталога. Поиск может осуществляться по имени, типу или владельцу файла.

Синтаксис:

```
find [-H] [-L] [-P] [-Oуровень] [-D help|tree|search|stat|rates|opt|exec]  
[ПУТЬ...] [ВЫРАЖЕНИЕ]
```


Ключи для поиска:

- » **-name** — поиск по имени файла;
- » **-type** — поиск по типу f=файл, d=каталог, l=ссылка(lnk);
- » **-user** — поиск по владельцу (имя или UID).

Когда выполняется команда **find**, можно выполнять различные действия над найденными файлами. Основные действия:

- » **-exec команда \;** — выполнить команду. Запись команды должна заканчиваться экранированной точкой с запятой. Строка «{» заменяется текущим маршрутным именем файла;
- » **execdir команда \;** — то же самое что и **-exec**, но команда вызывается из подкаталога, содержащего текущий файл;
- » **-ok команда** — эквивалентно **-exec** за исключением того, что перед выполнением команды запрашивается подтверждение (в виде сгенерированной командной строки со знаком вопроса в конце) и она выполняется только при ответе: y;
- » **-print** — вывод имени файла на экран.

Путем по умолчанию является текущий подкаталог. Выражение по умолчанию **-print**.

Примеры. Найти в текущем каталоге обычные файлы (не каталоги), имя которых начинается с символа «~»:

```
find . -type f -name "~*" -print
```

Найти в текущем каталоге файлы, измененные позже, чем файл **file.bak**:

```
find . -newer file.bak -type f -print
```

Удалить все файлы с именами **a.out** или ***.o**, доступ к которым не производился в течение недели:

```
find / \( -name a.out -o -name '*.o' \) \ -atime +7 -exec rm {} \;
```

Удалить из текущего каталога и его подкаталогов все файлы нулевого размера, запрашивая подтверждение:

```
find . -size 0c -ok rm {} \;
```

Команда **whereis**

whereis сообщает путь к исполняемому файлу программы, ее исходным файлам (если есть) и соответствующим страницам справочного руководства.

Синтаксис:

```
whereis [ОПЦИИ] <ИМЯ>
```

Опции:

- » **-b** — вывод информации только об исполняемых файлах;
- » **-m** — вывод информации только о страницах справочного руководства;
- » **-s** — вывод информации только об исходных файлах.

Мониторинг и управление процессами

Команда **ps**

Команда **ps** отображает список текущих процессов.

Синтаксис:

```
ps [ОПЦИИ]
```

По умолчанию выводится информация о процессах с теми же действующим UID и управляющим терминалом, что и у подающего команду пользователя.

Основные опции:

- » **-a** — вывести информацию о процессах, ассоциированных с терминалами;
- » **-f** — вывести «полный» список;
- » **-l** — вывести «длинный» список;
- » **-p список** — вывести информацию о процессах с перечисленными в списке PID;
- » **-u список** — вывести информацию о процессах с перечисленными идентификаторами или именами пользователей.

Команда **kill**

Команда **kill** позволяет прекратить исполнение процесса или передать ему сигнал.

Синтаксис:

```
kill [-s] [сигнал] [идентификатор] [...]
```

```
kill [-l] [статус_завершения]
```

```
kill [-номер_сигнала] [идентификатор] [...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предварённый знаком «%».

Основные опции:

- » **-l** — вывести список поддерживаемых сигналов;
- » **-s сигнал, -сигнал** — послать сигнал с указанным именем.

Если обычная команда **kill** не дает желательного эффекта, необходимо использовать команду **kill** с параметром **-9** (**kill -9 PID_номер**).

Команда **df**

Команда **df** показывает количество доступного дискового пространства в файловой системе, в которой содержится файл, переданный как аргумент. Если ни один файл не указан, показывается доступное место на всех смонтированных файловых системах. Размеры по умолчанию указаны в блоках по 1КБ.

Синтаксис:

```
df [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- » **--total** — подсчитать общий объем в конце;
- » **-h, --human-readable** — печатать размеры в удобочитаемом формате (например, 1K, 234M, 2G).

Команда **du**

Команда **du** подсчитывает использование диска каждым файлом, для каталогов подсчет происходит рекурсивно.

Синтаксис:

```
du [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- » **-a, --all** — выводить общую сумму для каждого заданного файла, а не только для каталогов;
- » **-c, --total** — подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;
- » **-d, --max-depth=N** — выводить объем для каталога (или файлов, если указано **--all**) только если она на N или менее уровней ниже аргументов командной строки;
- » **-S, --separate-dirs** — выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;
- » **-s, --summarize** — отобразить только сумму для каждого аргумента.

Команда **which**

Команда **which** отображает полный путь к указанным командам или сценариям.

Синтаксис:

```
which [ОПЦИИ] <ФАЙЛ...>
```

Основные опции:

- » **-a, --all** — выводит все совпавшие исполняемые файлы по содержимому в переменной окружения **\$PATH**, а не только первый из них;

- » **-c, --total** — подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;
- » **-d, --max-depth=N** — выводить объем для каталога (или файлов, если указано **--all**) только если она на N или менее уровней ниже аргументов командной строки;
- » **-S, --separate-dirs** — выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;
- » **--skip-dot** — пропускает все каталоги из переменной окружения **\$PATH**, которые начинаются с точки.

Использование многозадачности

Альт СП Сервер — это многозадачная система.

Для того, чтобы запустить программу в фоновом режиме, необходимо набрать «&» после имени программы. После этого оболочка даст возможность запустить другие приложения.

Так как некоторые программы интерактивны — их запуск в фоновом режиме бессмысленен. Подобные программы просто остановятся, если их запустить в фоновом режиме.

Можно также запускать нескольких независимых сеансов. Для этого в консоли необходимо набрать **Alt** и одну из клавиш, находящихся в интервале от **F1** до **F6**. На экране появится новое приглашение системы, и можно открыть новый сеанс. Этот метод также позволяет вам работать на другой консоли, если консоль, которую вы использовали до этого, не отвечает или вам необходимо остановить зависшую программу.

Команда **bg**

Команда **bg** позволяет перевести задание на задний план.

Синтаксис:

```
bg [ИДЕНТИФИКАТОР ...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предварённый знаком «%».

Команда **fg**

Команда **fg** позволяет перевести задание на передний план.

Синтаксис:

```
fg [ИДЕНТИФИКАТОР ...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предварённый знаком «%».

Сжатие и упаковка файлов

Команда **tar**

Сжатие и упаковка файлов выполняется с помощью команды **tar**, которая преобразует файл или группу файлов в архив без сжатия (tarfile).

Упаковка файлов в архив чаще всего выполняется следующей командой:

```
tar -cf [имя создаваемого файла архива] [упаковываемые файлы и/или каталоги]
```

Пример использования команды упаковки архива:

```
tar -cf moi_dokumenti.tar Docs project.tex
```

Распаковка содержимого архива в текущий каталог выполняется командой:

```
tar -xf [имя файла архива]
```

Для сжатия файлов используются специальные программы сжатия: **gzip**, **bzip2** и **7z**.

20.3. Стыкование команд в системе Linux

20.3.1. Стандартный ввод и стандартный вывод

Многие команды системы имеют так называемые стандартный ввод (standard input) и стандартный вывод (standard output), часто сокращаемые до **stdin** и **stdout**. Ввод и вывод здесь — это входная и выходная информация для данной команды. Программная оболочка делает так, что стандартным вводом является клавиатура, а стандартным выводом — экран монитора.

Пример с использованием команды **cat**. По умолчанию команда **cat** читает данные из всех файлов, которые указаны в командной строке, и посылает эту информацию непосредственно в стандартный вывод (**stdout**). Следовательно, команда:

```
cat history-final masters-thesis
```

выведет на экран сначала содержимое файла **history-final**, а затем — файла **masters-thesis**.

Если имя файла не указано, программа **cat** читает входные данные из **stdin** и возвращает их в **stdout**. Пример:

```
cat  
Hello there.  
Hello there.  
Bye.  
Bye.  
Ctrl-D
```

Каждую строку, вводимую с клавиатуры, программа **cat** немедленно возвращает на экран. При вводе информации со стандартного ввода конец текста сигнализируется вводом специальной комбинации клавиш, как правило, **Ctrl+D**. Сокращённое название сигнала конца текста — **EOT** (end of text).

20.3.2. Перенаправление ввода и вывода

При необходимости можно перенаправить стандартный вывод, используя символ **>**, и стандартный ввод, используя символ **<**.

Фильтр (filter) — программа, которая читает данные из стандартного ввода, некоторым образом их обрабатывает и результат направляет на стандартный вывод. Когда применяется перенаправление, в качестве стандартного ввода и вывода могут выступать файлы. Как указывалось выше, по умолчанию, `stdin` и `stdout` относятся к клавиатуре и к экрану соответственно. Программа **sort** является простым фильтром — она сортирует входные данные и посылает результат на стандартный вывод. Совсем простым фильтром является программа **cat** — она ничего не делает с входными данными, а просто пересылает их на выход.

20.3.3. Использование состыкованных команд

Стыковку команд (pipelines) осуществляет командная оболочка, которая `stdout` первой команды направляет на `stdin` второй команды. Для стыковки используется символ `|`. Направить `stdout` команды **ls** на `stdin` команды **sort**:

```
ls | sort -r
notes
masters-thesis
history-final
english-list
```

Вывод списка файлов частями:

```
ls /usr/bin | more
```

Если необходимо вывести на экран последнее по алфавиту имя файла в текущем каталоге, можно использовать следующую команду:

```
ls | sort -r | head -1 notes
```

где команда **head -1** выводит на экран первую строку получаемого ей входного потока строк (в примере поток состоит из данных от команды **ls**), отсортированных в обратном алфавитном порядке.

20.3.4. Недеструктивное перенаправление вывода

Эффект от использования символа `>` для перенаправления вывода файла является деструктивным; т.е, команда

```
ls > file-list
```

уничтожит содержимое файла **file-list**, если этот файл ранее существовал, и создаст на его месте новый файл. Если вместо этого перенаправление будет сделано с помощью символов `>>`, то вывод будет приписан в конец указанного файла, при этом исходное содержимое файла не будет уничтожено.



Примечание

Перенаправление ввода и вывода и стыкование команд осуществляется командными оболочками, которые поддерживают использование символов `>`, `>>` и `|`. Сами команды не способны воспринимать и интерпретировать эти символы.

Глава 21. Режим суперпользователя

21.1. Какие бывают пользователи?

21.2. Для чего может понадобиться режим суперпользователя?

21.3. Как получить права суперпользователя?

21.4. Как перейти в режим суперпользователя?

21.1. Какие бывают пользователи?

Linux — система многопользовательская, а потому пользователь — ключевое понятие для организации всей системы доступа в Linux. Файлы всех пользователей в Linux хранятся отдельно, у каждого пользователя есть собственный домашний каталог, в котором он может хранить свои данные. Доступ других пользователей к домашнему каталогу пользователя может быть ограничен.

Суперпользователь в Linux — это выделенный пользователь системы, на которого не распространяются ограничения прав доступа. Именно суперпользователь имеет возможность произвольно изменять владельца и группу файла. Ему открыт доступ на чтение и запись к любому файлу или каталогу системы.

Среди учётных записей Linux всегда есть учётная запись суперпользователя — **root**. Поэтому вместо «суперпользователь» часто говорят «root». Множество системных файлов принадлежат **root**, множество файлов только ему доступны для чтения или записи. Пароль этой учётной записи — одна из самых больших драгоценностей системы. Именно с её помощью системные администраторы выполняют самую ответственную работу.

21.2. Для чего может понадобиться режим суперпользователя?

Системные утилиты, например, такие, как **Центр управления системой** или **Программа управления пакетами Synaptic** требуют для своей работы привилегий суперпользователя, потому что они вносят изменения в системные файлы. При их запуске выводится диалоговое окно с запросом пароля системного администратора.

21.3. Как получить права суперпользователя?

Для опытных пользователей, умеющих работать с командной строкой, существует два различных способа получить права суперпользователя.

Первый — это зарегистрироваться в системе под именем **root**.

Второй способ — воспользоваться специальной утилитой **su** (shell of user), которая позволяет выполнить одну или несколько команд от лица другого пользователя. По умолчанию эта утилита выполняет команду **sh** от пользователя **root**, то есть запускает командный интерпретатор. Отличие от предыдущего способа в том, что всегда известно, кто именно запускал **su**, а значит, ясно, кто выполнил определённое административное действие.

В некоторых случаях удобнее использовать не **su**, а утилиту **sudo**, которая позволяет выполнять только заранее заданные команды.



Важно

Для того чтобы воспользоваться командами **su** и **sudo**, необходимо быть членом группы **wheel**. Пользователь, созданный при установке системы, по умолчанию уже включён в эту группу.

В дистрибутивах Альт для управления доступом к важным службам используется подсистема **control**. **control** — механизм переключения между неким набором фиксированных состояний для задач, допускающих такой набор.

Команда **control** доступна только для суперпользователя (**root**). Для того, чтобы посмотреть, что означает та или иная политика **control** (разрешения выполнения конкретной команды, управляемой **control**), надо запустить команду с ключом **help**:

```
# control su help
```

Запустив **control** без параметров, можно увидеть полный список команд, управляемых командой (**facilities**) вместе с их текущим состоянием и набором допустимых состояний.

21.4. Как перейти в режим суперпользователя?

Для перехода в режим суперпользователя наберите в терминале команду **su -**.

Если воспользоваться командой **su** без ключа, то происходит вызов командного интерпретатора с правами **root**. При этом значение переменных окружения, в частности **\$PATH**, остаётся таким же, как у пользователя: в переменной **\$PATH** не окажется каталогов **/sbin**, **/usr/sbin**, без указания полного имени будут недоступны команды **route**, **shutdown**, **mkswap** и другие. Более того, переменная **\$HOME** будет указывать на каталог пользователя, все программы, запущенные в режиме суперпользователя, сохраняют свои настройки с правами **root** в каталоге пользователя, что в дальнейшем может вызвать проблемы.

Чтобы избежать этого, следует использовать **su -**. В этом режиме **su** запустит командный интерпретатор в качестве **login shell**, и он будет вести себя в точности так, как если бы в системе зарегистрировался **root**.

Глава 22. Команда sudo

22.1. Настройка control для работы sudo

22.2. Настройка sudo

Команда **sudo** может использоваться для выполнения пользователем какой-либо команды, требующей права суперпользователя (**root**), то есть получение прав **root** для выполнения какой-либо команды на время её выполнения.

22.1. Настройка control для работы sudo

В дистрибутивах Альт для управления доступом к важным службам используется подсистема control. control — механизм переключения между неким набором фиксированных состояний для задач, допускающих такой набор.

Для того, чтобы посмотреть, что означает та или иная политика **control** (разрешения выполнения конкретной команды, управляемой **control**), надо запустить команду с ключом help:

```
$ sudo control sudo help
```

Запустив **control** без параметров, можно увидеть полный список команд, управляемых командой (facilities) вместе с их текущим состоянием и набором допустимых состояний.

Штатное состояние политики sudo:

```
$ sudo control sudo  
wheelonly
```

Означает что пользователь из группы wheel имеет право запускать саму команду **sudo**, но не означает, что он через **sudo** может выполнить какую-то команду с правами root.

22.2. Настройка sudo

Для того, чтобы настроить работу **sudo**, необходимо с применением административных привилегий отредактировать файл **/etc/sudoers** при помощи специальной команды **visudo**.

Настройка по умолчанию:

```
WHEEL_USERS ALL=(ALL) NOPASSWD: ALL
```

Позволяет пользователям входящим в группу **wheel** выполнять через **sudo** любую команду с любого компьютера, не запрашивая их пароль.

Пример настройки:

```
user ALL = (ALL) ALL  
user ALL = NOPASSWD: /usr/bin/apt-get update
```

Позволяет пользователю **user** запускать все приложения через **sudo** с правами суперпользователя (root) с запросом пароля, а при выполнении команды **sudo apt-get update** пароль запрашиваться не будет.

Глава 23. Управление пользователями

23.1. Общая информация

23.2. Команда passwd

23.3. Добавления нового пользователя

23.4. Модификация пользовательских записей

23.1. Общая информация

Пользователи и группы внутри системы обозначаются цифровыми идентификаторами — UID и GID, соответственно.

Пользователь может входить в одну или несколько групп. По умолчанию он входит в группу, совпадающую с его именем. Чтобы узнать, в какие еще группы входит пользователь, введите команду **id**, вывод её может быть примерно следующим:

```
uid=500(test) gid=500(test) группы=500(test),16(rpm)
```

Такая запись означает, что пользователь *test* (цифровой идентификатор 500) входит в группы *test* и *rpm*. Разные группы могут иметь разный уровень доступа к тем или иным каталогам; чем в большее количество групп входит пользователь, тем больше прав он имеет в системе.



Примечание

В связи с тем, что большинство привилегированных системных утилит в дистрибутивах Альт имеют не SUID-, а SGID-бит, будьте предельно внимательны и осторожны в переназначении групповых прав на системные каталоги.

23.2. Команда passwd

Команда **passwd** поддерживает традиционные опции **passwd** и утилит **shadow**.

Синтаксис:

```
passwd [ОПЦИИ...] [ИМЯ ПОЛЬЗОВАТЕЛЯ]
```

Возможные опции:

- » **-d --delete** — удалить пароль для указанной записи;
- » **-f, --force** — форсировать операцию;
- » **-k, --keep-tokens** — сохранить не устаревшие пароли;
- » **-l, --lock** — заблокировать указанную запись;
- » **--stdin** — прочитать новые пароли из стандартного ввода;
- » **-S, --status** — дать отчет о статусе пароля в указанной записи;
- » **-u, --unlock** — разблокировать указанную запись;
- » **-, --help** — показать справку и выйти;
- » **--usage** — дать короткую справку по использованию;
- » **-V, --version** — показать версию программы и выйти.

Код выхода: при успешном завершении **passwd** заканчивает работу с кодом выхода 0. Код выхода 1 означает, что произошла ошибка. Текстовое описание ошибки выводится на стандартный поток ошибок.

Пользователь может в любой момент поменять свой пароль. Единственное, что требуется для смены пароля — знать текущий пароль.

Только суперпользователь может обновить пароль другого пользователя.

23.3. Добавления нового пользователя

Для добавления нового пользователя используйте команды **useradd** и **passwd**:

```
# useradd test1

# passwd test1
passwd: updating all authentication tokens for user test1.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters,
digits, and other characters. You can use an 8 character long
password with characters from at least 3 of these 4 classes, or
a 7 character long password containing characters from all the
classes. An upper case letter that begins the password and a
digit that ends it do not count towards the number of character
classes used.

A passphrase should be of at least 3 words, 11 to 40 characters
long, and contain enough different characters.

Alternatively, if no one else can see your terminal now, you can
pick this as your password: "holder5dinghy-Arm".

Enter new password:
```

В результате описанных действий в системе появился пользователь *test1* с некоторым паролем. Если пароль оказался слишком слабым с точки зрения системы, она об этом предупредит (как в примере выше). Пользователь в дальнейшем может поменять свой пароль при помощи команды **passwd** — но если он попытается поставить слабый пароль, система откажет ему (в отличие от *root*) в изменении.

В Альт СП Сервер для проверки паролей на слабость используется модуль PAM *passwdqc*.

Программа **useradd** имеет множество параметров, которые позволяют менять её поведение по умолчанию. Например, можно принудительно указать, какой будет UID или какой группе будет принадлежать пользователь.

23.4. Модификация пользовательских записей

Для модификации пользовательских записей применяется утилита **usermod**:

```
# usermod -G audio,rpm,test1 test1
```

Такая команда изменит список групп, в которые входит пользователь *test1* — теперь это *audio*, *rpm*, *test1*.

```
# usermod -l test2 test1
```

Будет произведена смена имени пользователя с *test1* на *test2*.

Команды **usermod -L test2** и **usermod -U test2** соответственно временно блокируют возможность входа в систему пользователю *test2* и возвращают всё на свои места.

Изменения вступят в силу только при следующем входе пользователя в систему.

При неинтерактивной смене или задании паролей для целой группы пользователей используйте утилиту **chpasswd**. На стандартный вход ей следует подавать список, каждая строка которого будет выглядеть как **имя:пароль**.

23.5. Удаление пользователей

Для удаления пользователей используйте **userdel**.

Команда **userdel test2** удалит пользователя *test2* из системы. Если будет дополнительно задан параметр **-r**, то будет уничтожен и домашний каталог пользователя. Нельзя удалить пользователя, если в данный момент он еще работает в системе.

Глава 24. Система инициализации systemd и sysvinit

24.1. Запуск операционной системы

24.2. Системы инициализации systemd и sysvinit

24.3. Примеры команд управления службами, журнал в systemd

24.4. Журнал в systemd

24.1. Запуск операционной системы

24.1.1. Запуск системы

Алгоритм запуска компьютера приблизительно такой:

1. BIOS компьютера.
2. Загрузчик системы (например, LILO, GRUB или другой). В загрузчике вы можете задать параметры запуска системы или выбрать систему для запуска.
3. Загружается ядро Linux.
4. Запускается на выполнение первый процесс в системе — **init**.

Ядром запускается самая первая программа в системе **init**. Её задачей является запуск новых процессов и повторный запуск завершившихся. Вы можете посмотреть, где расположился **init** в иерархии процессов вашей системы, введя команду **pstree**.

От конфигурации **init** зависит, какая система инициализации будет использована.

24.1.2. Система инициализации

Система инициализации — это набор скриптов, которые будут выполнены при старте системы.

Существуют разные системы инициализации, наиболее популярные системы: **sysvinit** (и её модификации) и **systemd**.

В Альт СП Сервер используется **systemd**.

24.2. Системы инициализации systemd и sysvinit

24.2.1. sysvinit

System V — классическая схема инициализации, на которой базируются многие дистрибутивы. Привычна и довольно проста для понимания: **init** описывает весь процесс загрузки в своем конфигурационном файле **/etc/inittab**, откуда вызываются другие программы и скрипты на определенном этапе запуска.

24.2.2. systemd

systemd является альтернативной системой инициализации Linux, вобравшей в себя достоинства классического **System V init** и более современных **launchd** (OS X), **SMF** (Solaris) и **Upstart** (Ubuntu, Fedora), но при этом лишенной многих их недостатков. Он разрабатывался для обеспечения лучшего выражения зависимостей между службами, что позволяет делать одновременно больше работы при загрузке системы, и уменьшить время загрузки системы.

systemd (system daemon) реализует принципиально новый подход к инициализации и контролю работы системы. Одним из ключевых новшеств этого подхода является высокая степень параллелизации запуска служб при инициализации системы, что в перспективе позволяет добиться гораздо более высокой скорости, чем традиционный подход с последовательным запуском взаимозависимых служб. Другим важным моментом является контроль над точками монтирования (не-жизненно-важные файловые системы можно монтировать только при первом обращении к ним, не тратя на это время при инициализации системы) и устройствами (можно запускать и останавливать определенные службы и при появлении или удалении заданных устройств). Для отслеживания групп процессов используется механизм cgroups, который также может быть использован для ограничения потребляемых ими системных ресурсов.

Удобство **systemd** особенно заметно на компьютерах для домашнего пользования — когда пользователи включают и перезагружают компьютер ежедневно. В отличие от **sysvinit**, подвисание при запуске одного сервиса не приведет к остановке всего процесса загрузки.

24.3. Примеры команд управления службами, журнал в systemd

Обратите внимание, что команды **service** и **chkconfig** продолжают работать в мире **systemd** практически без изменений. Тем не менее, в этой таблице показано как выполнить те же действия с помощью встроенных утилит **systemctl**.

Таблица 24.1. Команды управления службами

Команды Sysvinit	Команды Systemd	Примечания
service frobozz start	systemctl start frobozz.service	Используется для запуска службы (не перезагружает постоянные)
service frobozz stop	systemctl stop frobozz.service	Используется для остановки службы (не перезагружает постоянные)
service frobozz restart	systemctl restart frobozz.service	Используется для остановки и последующего запуска службы
service frobozz reload	systemctl reload frobozz.service	Если поддерживается, перезагружает файлы конфигурации без прерывания незаконченных операций
service frobozz condrestart	systemctl condrestart frobozz.service	Перезапускает службу, если она уже работает
service frobozz status	systemctl status frobozz.service	Сообщает, запущена ли уже служба
ls /etc/rc.d/init.d/	systemctl list-unit-files --type=service (preferred) ls /lib/systemd/system/*.service /etc/systemd/system/*.service	Используется для отображения списка служб, которые можно запустить или остановить. Используется для отображения списка всех служб.
chkconfig frobozz on	systemctl enable frobozz.service	Включает службу во время следующей перезагрузки, или любой другой триггер
chkconfig frobozz off	systemctl disable frobozz.service	Выключает службу во время следующей перезагрузки, или любой другой триггер
chkconfig frobozz	systemctl is-enabled frobozz.service	Используется для проверки, сконфигурирована ли служба для запуска в текущем окружении
chkconfig --list	systemctl list-unit-files --type=service(preferred) ls /etc/systemd/system/*.wants/	Выводит таблицу служб. В ней видно, на каких уровнях загрузки они (не)запускаются
chkconfig frobozz --list	ls /etc/systemd/system/*.wants/frobozz.service	Используется, для отображения на каких уровнях служба (не)запускается
chkconfig frobozz --add	systemctl daemon-reload	Используется, когда вы создаете новую службу или модифицируете любую конфигурацию

24.4. Журнал в systemd

В **systemd** включена возможность ведения системного журнала. Для чтения журнала следует использовать команду **journalctl**. По умолчанию, больше не требуется запуск службы **syslog**.

Вы можете запускать **journalctl** с разными ключами:

- **journalctl -b** — покажет сообщения только с текущей загрузки;
- **journalctl -f** — покажет только последние сообщения.

Так же вы можете посмотреть сообщения определенного процесса:

- **journalctl _PID=1** — покажет сообщения первого процесса (init).

Для ознакомления с прочими возможностями, читайте руководство по **journalctl**. Для этого используйте команду **man journalctl**.

Глава 25. Документация

25.1. Экранная документация

25.2. Документация по пакетам

Каждый объект системы Linux обязательно сопровождается документацией, описывающей их назначение и способы использования. От пользователя системы не требуется заучивать все возможные варианты взаимодействия с ней. Достаточно понимать основные принципы её устройства и уметь находить справочную информацию.

Не пренебрегайте чтением документации: она поможет вам избежать многих сложностей, сэкономят массу времени и усилий при установке, настройке и администрировании системы, поможет найти нужное для работы приложение и быстро разобраться в нём.

25.1. Экранная документация

Почти все системы семейства UNIX, включая систему Linux, имеют экранную документацию. Её тексты содержат документацию по системным командам, ресурсам, конфигурационным файлам и т. д., а также могут быть выведены на экран в процессе работы.

25.1.1. man

Для доступа к экранной документации используется команда **man** (сокращение от manual). Каждая страница руководства посвящена одному объекту системы. Для того чтобы прочесть страницу руководства по программе, необходимо набрать **man название_программы**. К примеру, если вы хотите узнать, какие опции есть у команды **date**, вы можете ввести команду:

```
$ man date
```

Большинство экранной документации написано для пользователей, имеющих некоторое представление о том, что делает данная команда. Поэтому большинство текстов экранной документации содержит исключительно технические детали команды без особых пояснений. Тем не менее, экранная документация оказывается очень ценной в том случае, если вы помните название команды, но её синтаксис просто выпал у вас из памяти.

Поиск по описаниям **man** осуществляется командой **apropos**. Если вы точно не знаете, как называется необходимая вам программа, то поиск осуществляется по ключевому слову, к примеру, **apropos date** или при помощи ввода слова, обозначающего нужное действие, после команды **man -k** (например, **man -k сору**). Слово, характеризующее желаемое для вас действие, можно вводить и на русском языке. При наличии русского перевода страниц руководства **man** результаты поиска будут выведены на запрашиваемом языке.

«Страница руководства» занимает, как правило, больше одной страницы экрана. Для того чтобы читать было удобнее, **man** запускает программу постраничного просмотра текстов. Страницы перелистывают пробелом, для выхода из режима чтения описания команд **man** необходимо нажать на клавиатуре **q**. Команда **man man** выдаёт справку по пользованию самой командой **man**.

Документация в подавляющем большинстве случаев пишется на простом английском языке. Необходимость писать на языке, который будет более или менее понятен большинству пользователей, объясняется постоянным развитием Linux. Дело не в том, что страницу руководства нельзя перевести, а в том, что её придётся переводить всякий раз, когда изменится описываемый ею объект! Например, выход новой версии программного продукта сопровождается изменением его возможностей и особенностей работы, а следовательно, и новой версией документации.

Тем не менее, некоторые наиболее актуальные руководства существуют в переводе на русский язык. Свежие версии таких переводов на русский язык собраны в пакете *man-pages-ru*. Установив этот пакет, вы добавите в систему руководства, для которых есть перевод, и **man** по умолчанию будет отображать их на русском языке.

25.1.2. info

Другой источник информации о Linux и составляющих его программах — справочная подсистема *info*. Страница руководства, несмотря на обилие ссылок различного типа, остаётся «линейным» текстом, структурированным только логически. Документ *info* — это настоящий гипертекст, в котором множество небольших страниц объединены в дерево. В каждом разделе документа *info* всегда есть оглавление, из которого можно перейти к нужному подразделу, а затем вернуться обратно (ссылки для перемещения по разделам текста помечены *). Для получения вспомогательной информации о перемещении по тексту используйте клавишу **h**. Полное руководство *info* вызывается командой **info info**. Команда **info**, введённая без параметров, предлагает пользователю список всех документов *info*, установленных в системе.

25.2. Документация по пакетам

Дополнительным источником информации об интересующей вас программе, в основном на английском языке, является каталог **/usr/share/doc** — место хранения разнообразной документации.

Каждый пакет также содержит поставляемую вместе с включённым в него ПО документацию, располагающуюся обычно в каталоге **/usr/share/doc/имя_пакета**. Например, документация к пакету *foo-1.0-alt1* находится в **/usr/share/doc/foo-1.0-alt1**. Для получения полного списка файлов документации, относящихся к пакету, воспользуйтесь командой **rpm -qd имя_установленного_пакета**.

В документации к каждому пакету вы можете найти такие файлы как **README**, **FAQ**, **TODO**, **ChangeLog** и другие. В файле **README** содержится основная информация о программе — имя и контактные данные авторов, назначение, полезные советы и пр. **FAQ** содержит ответы на часто задаваемые вопросы; этот файл стоит прочитать в первую очередь, если у вас возникли

проблемы или вопросы по использованию программы, поскольку большинство проблем и сложностей типичны, вполне вероятно, что в **FAQ** вы тут же найдёте готовое решение. В файле **TODD** записаны планы разработчиков на реализацию той или иной функциональности. В файле **ChangeLog** записана история изменений в программе от версии к версии.

Для поиска внешней информации о программе, например, адреса сайта программы в сети Интернет можно использовать команду **rpm -qi имя_установленного_пакета**. В информационном заголовке соответствующего пакета, среди прочей информации, будет выведена искомая ссылка.

Возможно, будет полезно знать расположение собрания практических рекомендаций по самым различным вопросам, связанным с использованием Linux. Файлы **HOWTO** в формате HTML (от англ. how to — «как сделать») каталога **/usr/share/doc/HOWTO/** (при условии их наличия в системе) содержат многообразную информацию о работе Linux-систем.

Часть VII. Техническая поддержка продуктов «Базальт СПО»

Содержание

26. Покупателям нашей продукции

27. Пользователям нашей продукции

Глава 26. Покупателям нашей продукции

Право на получение консультационной и технической поддержки вы приобретаете при покупке большинства продуктов торговой марки Альт. Сроки и объём помощи указаны в талоне технической поддержки. Техническая поддержка дистрибутива может быть расширена в зависимости от потребностей пользователя.

Для получения информации об условиях технической поддержки облачного дистрибутива следует обратиться к своему облачному провайдеру.

Глава 27. Пользователям нашей продукции

Задавать вопросы или обсуждать их с сообществом пользователей дистрибутивов «Альт» можно на форуме или в списках рассылки.

Помощь сообщества:

- » Форум: <http://forum.altlinux.org>
- » Списки рассылки: <http://lists.altlinux.org/>
- » Сообщить об ошибке: <http://bugs.altlinux.org/>
- » Репозиторий: <http://packages.altlinux.org/>
- » Сборочная среда: <http://git.altlinux.org/>

Ресурсы компании «Базальт СПО»:

- Сайт компании: <http://www.basealt.ru/>
- Контакты: <http://basealt.ru/about/contacts/>
- Новости обновлений безопасности: <http://cve.basealt.ru/>

Форум и списки рассылки читают опытные пользователи, профессиональные системные администраторы и разработчики «Базальт СПО». Сообщество пользователей и специалистов окажет содействие в поиске ответа на ваш вопрос или посоветует выход из сложной ситуации. При обращении к данному виду помощи у вас нет гарантии на полноту и своевременность ответа, но мы стараемся не оставлять без ответа вопросы, задаваемые в списках.