УТВЕРЖДЕН

ЛКНВ.11100-01 90 02-ЛУ

ОПЕРАЦИОННАЯ СИСТЕМА АЛЬТ 8 СП

(ОС Альт 8 СП)

Руководство администратора

ЛКНВ.11100-01 90 02

Листов 324

2019

Литера О

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Настоящий документ содержит инструкции по установке и эксплуатации программного изделия (ПИ) «Операционная система Альт 8 СП» (ОС Альт 8 СП) на архитектуре **Эльбрус**.

Версия документа **1.1**.

Документ предназначен для администратора ОС Альт 8 СП и содержит общие сведения об ОС Альт 8 СП, ее общей структуре, настройке, проверке, контрольных характеристиках развертывания и сообщениях администратору.

Также в документе приведены сведения, необходимые для выполнения операций администрирования:

- установки и начального конфигурирования ОС Альт 8 СП;
- конфигурирования параметров даты и времени, графической среды, средств ввода и вывода;
- конфигурирования сетей и сетевых служб;
- управления учетными записями и правами доступа пользователей;
- управления системными сервисами и служебными программами;
- настройки специализированного программного обеспечения;
- обновления программного обеспечения;
- просмотра системных журналов;
- управления автозапуском приложений;
- управления параметрами печати;
- работы с носителями информации;
- работы с руководствами, различными документами и дополнительными средствами.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ ОБ ОС АЛЬТ 8 СП

1.1. Назначение и функции ОС Альт 8 СП

ОС Альт 8 СП, представляет собой совокупность интегрированных программ, созданных на основе операционной системы Linux.

ОС Альт 8 СП предназначено для группового и корпоративного использования, автоматизации информационных, конструкторских и производственных процессов предприятий (организаций, учреждений) всех возможных типов и направлений.

ОС Альт 8 СП поддерживает клиент-серверную архитектуру и может обслуживать процессы как в пределах одной компьютерной системы, так и процессы на других персональных электронных вычислительных машинах (далее – ПЭВМ) через каналы передачи данных или сетевые соединения.

ОС Альт 8 СП обладает следующими функциональными характеристиками:

- обеспечивает возможность обработки, хранения и передачи информации в защищенной программной среде;
- обеспечивает возможность запуска пользовательского программного обеспечения в сертифицированном окружении;
- обеспечивает возможность функционирования в многозадачном режиме (одновременное выполнение множества процессов);
- обеспечивает возможность масштабирования системы: возможна эксплуатация ОС как на одной ПЭВМ, так и в информационных системах различной архитектуры;
- обеспечивает многопользовательский режим эксплуатации;
- обеспечивает поддержку мультипроцессорных систем;
- обеспечивает сетевую обработку данных, в том числе разграничение доступа к сетевым пакетам.

Для поддержки выполнения описанных функций в ОС Альт 8 СП реализованы следующие возможности:

- управление процессами и информационными ресурсами;
- управление системными ресурсами;
- управление памятью;
- управление файлами и внешними устройствами;
- управление доступом к обрабатываемой информации;
- защита хранимых, обрабатываемых и передаваемых информационных ресурсов комплексом средств защиты (далее – КСЗ) операционной системы (далее – ОС);
- администрирование;
- поддержка интерфейса прикладного программирования;
- поддержка пользовательского интерфейса.

Дистрибутив ОС Альт 8 СП поставляется на DVD-диске. Дополнительно прилагается диск с документацией.

1.2. Уровень подготовки администратора

Администратор ОС Альт 8 СП должен иметь базовые знания в областях:

- принципы построения и функционирования современных вычислительных систем, механизмов защиты информации;
- работа с ОС семейства Linux;
- администрирование общесистемного и прикладного программного обеспечения (далее – ПО);
- настройка средств защиты, используемых в составе ОС Альт 8 СП;
- конфигурирование проводных подключений.

2. СТРУКТУРА ОС АЛЬТ 8 СП

ОС Альт 8 СП состоит из набора компонентов, предназначенных для реализации функциональных задач необходимых пользователям (должностным лицам для выполнения определенных должностными инструкциями, повседневных действий). ПИ ОС Альт 8 СП поставляется в виде дистрибутива и комплекта эксплуатационной документации.

Структура ОС Альт 8 СП представлена на рис. 1.

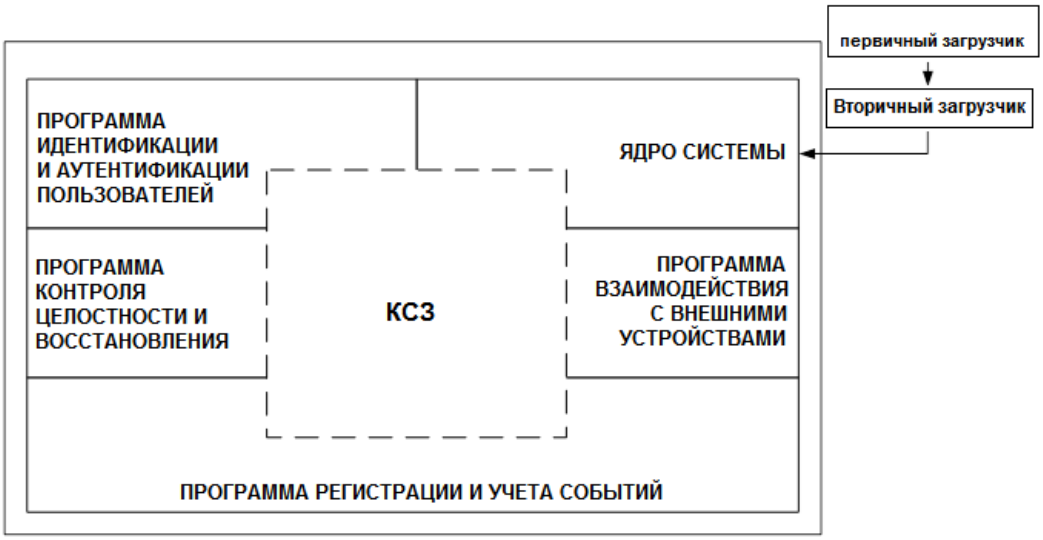

Рис. 1 – Структура ОС Альт 8 СП

В состав ОС Альт 8 СП входят следующие компоненты:

- «Ядро системы»;
- «Программа идентификации и аутентификации пользователей»;
- «Программа контроля целостности и восстановления»;
- «Программа взаимодействия с внешними устройствами»;
- «Программа регистрации и учета событий».

В структуре компонентов ОС Альт 8 СП выделены следующие функциональные элементы:

- ядро ОС;
- КСЗ;
- системные библиотеки;
- серверные программы;
- программы веб-серверов;
- прочие серверные программы;
- интерактивные рабочие среды;

- командные интерпретаторы;
- графическая оболочка MATE;
- системы управления базами данных;
- электронные справочники.

Взаимодействие и обмен информацией в ОС Альт 8 СП контролируются КСЗ, предназначенным для защиты ОС от несанкционированного доступа к обрабатываемой (хранящейся) информации на ПЭВМ.

2.1. Ядро ОС Альт 8 СП

Ядро ОС Альт 8 СП управляет доступом к оперативной памяти, сети, дисковым и прочим внешним устройствам. Оно запускает и регистрирует процессы, управляет разделением времени между ними, реализует разграничение прав и определяет политику безопасности, обойти которую, не обращаясь к нему, нельзя.

Ядро работает в режиме «супервизора», позволяющем ему иметь доступ сразу ко всей оперативной памяти и аппаратной таблице задач. Процессы запускаются в «режиме пользователя»: каждый жестко привязан ядром к одной записи таблицы задач, в которой, в числе прочих данных, указано, к какой именно части оперативной памяти этот процесс имеет доступ. Ядро постоянно находится в памяти, выполняя системные вызовы – запросы от процессов на выполнение этих подпрограмм.

2.2. КСЗ

КСЗ представляет собой набор специальных программных пакетов, в том числе из состава ядра ОС Альт 8 СП и системных библиотек, и предназначенных для реализации механизмов безопасности и контроля функционирования ОС Альт 8 СП в целом.

КСЗ включает в себя следующие программные пакеты:

- acl – утилиты, предназначенные для администрирования списков контроля доступа Access Control Lists, которые используются для более точного задания прав доступа к файлам и директориям;
- audit – утилиты для хранения и поиска записей аудита, генерируемых подсистемой аудита;
- bash – командная оболочка Bourne-Again Shell;
- cyrus-sasl2 – слой простой аутентификации и (Simple Authentication and Security Layer) – механизм, позволяющий добавлять поддержку аутентификации в протоколы, связанные с соединениями. SASL протокол включает команду идентификации и аутентификации пользователя на сервере и опционально, защиту последующего взаимодействия сторон в рамках протокола;
- glibc – основная библиотека C, в библиотеке находятся базовые процедуры распределения памяти, поиска в директориях, открытия и закрытия файлов,

чтения и записи файлов, обработки строк, сравнения по образцам, арифметических операций и так далее;
- iptables – используется для настройки, обслуживания и проверки, находящихся в ядре Linux таблиц правил фильтрации пакетов IP;
- kernel-image* – ядро операционной системы Linux, используется для загрузки и запуска системы;
- krb5 – система сетевой аутентификации Kerberos;
- libcap – библиотека для получения и установки возможностей POSIX.1e;
- libcap-ng – альтернативная библиотека возможности POSIX;
- libgcc1 – разделяемая версия вспомогательной библиотеки внутренних подпрограмм, используемых GCC для преодоления недостатков устаревших машин и специальных требований некоторых языков;
- libpam0 – подключаемые модули аутентификации Pluggable Authentication Modules (PAM). Они позволяют администратору выбрать, каким образом в приложениях будет осуществляться авторизация пользователей;
- libshell – библиотека часто используемых функций для POSIX shell, выделенная в отдельный пакет для увеличения переиспользования кода;
- libssh2 – библиотека реализации протокола SSH2;
- libstdc++-6 – дополнительная библиотека времени исполнения, необходимая программам, написанным на языке C++ и собранным при помощи компилятора GNU;
- libXau – библиотека, реализующая протокол авторизации X11. Используется для ограничения доступа клиентов к дисплею;
- logrotate – осуществляет автоматическую ротацию, сжатие, удаление и рассылку файлов журналов;
- msulogin – программа входа однопользовательского режима (sulogin);
- nss – набор библиотек, предназначенных для поддержки кросс-платформенной разработки безопасных клиентских и серверных приложений;
- openldap – реализация протокола LDAP (Lightweight Directory Access Protocol) с открытым исходным кодом. LDAP представляет собой набор протоколов для доступа к службам каталогов через Интернет. Пакет включает в себя: автономный сервер LDAP (Slapd), библиотеки для реализации протокола LDAP, утилиты, инструменты и образцы клиентов;
- osec – программный комплекс проверки целостности, предназначенный для обнаружения различий между двумя состояниями системы, а также для поиска потенциально опасных файлов, например, файлов с установленными битами прав смены идентификаторов пользователя (suid), группы (sgid) и с общедоступной записью;
- pam-config – инструменты системы безопасности, позволяющие администраторам устанавливать политику аутентификации без необходимости повторной компиляции программ проверки подлинности. Этот пакет содержит общесистемные конфигурационные файлы PAM. Этот пакет также содержит общие файлы и каталоги, используемые совместно с другими реализациями PAM;

- pam0_mktemp – поддержка личных каталогов `/tmp` для интерактивных (Shell) сессий;
- pam0_userpass – модуль аутентификации PAM для использования специфических услуг, реализующих не интерактивные протоколы и желающих проверить пару имя пользователя/пароль;
- passwd – утилита для установки/смены паролей с использованием PAM;
- passwdqc – набор инструментов для контроля сложности паролей и парольных фраз, включающий PAM-модуль, программы и библиотеку;
- procps – содержит программы для мониторинга и завершения системных процессов. Procps получает информацию о процессах из директории `/proc`;
- psmisc –утилиты для управления процессами в системе;
- qemu – быстрый эмулятор процессора, использующий динамическую трансляцию для достижения хорошей скорости эмуляции;
- rootfiles – набор базовых файлов конфигурации системы, таких как `/root/.bashrc`, представляющих собой переконфигурированное окружение пользователя root;
- seabios – реализация, устаревших BIOS с открытым исходным кодом, который может быть использован в качестве Coreboot полезной нагрузки;
- setup – начальный набор конфигурационных файлов;
- shadow – усиливает безопасность системных паролей;
- syslogd – программы, записывающие в журнальные файлы сообщения, например, те, которые выдаются ядром системы в случае возникновения непредусмотренных ситуаций;
- systemd – менеджер системы и служб для Linux, совместимый со скриптами инициализации SysV и LSB, использует, предлагает запуск демонов по необходимости, отслеживает процессы, поддерживает мгновенные снимки и восстановление состояния системы, монтирование и точки монтирования, а также внедряет основанную на зависимостях логику контроля процессов сложных транзакций;
- tcb – библиотеки и инструменты, реализующие схему сокрытия паролей tcb;
- tcp_wrappers – инструмент безопасности, предназначенный для разграничения доступа к сетевым сервисам;
- udev_static-addon – набор статических узлов устройств для виртуальной файловой системы UDev;
- util-linux – коллекция основных системных утилит;
- xinetd – является демоном дополнительных сервисов сети интернет (eXtended InterNET) и его можно использовать в качестве безопасной замены вместе inetd. Xinetd имеет механизмы контроля доступа, расширенные возможности регистрации, возможность сделать услуги доступными в зависимости от времени и может разместить ограничения по количеству серверов, которые могут быть запущены.

2.3. Системные библиотеки

Системные библиотеки – наборы программ (пакетов программ), выполняющие различные функциональные задачи и предназначенные для динамического подключения к работающим программам, которым необходимо выполнение этих задач.

2.4. Серверные программы и приложения

Серверные программы и приложения предоставляют пользователю специализированные услуги (почтовые службы, хранилище файлов, веб-сервер, система управления базой данных, обеспечение документооборота, хранилище данных пользователей и так далее) в локальной или глобальной сети и обеспечивают их выполнение.

В состав ОС Альт 8 СП включены следующие серверные программы и приложения:

- приложения, обеспечивающие поддержку сетевого протокола DHCP (Dynamic Host Configuration Protocol);
- приложения, обеспечивающие поддержку протокола аутентификации LDAP (Lightweight Directory Access Protocol);
- приложения, обеспечивающие поддержку протоколов FTP, SFTP, SSHD;
- программы, обеспечивающие работу системы управления баз данных MySQL;
- программы, обеспечивающие работу SMB-сервера (Сервер файлового обмена);
- программы почтового сервера Postfix;
- программы прокси-сервера Squid;
- программы веб-сервера Apache2;
- программы DNS-сервера;
- программы FreeNX-сервера.

2.5. Прочие системные приложения

Прочие системные приложения – приложения (программы), оказывающие пользователю дополнительные системные услуги при работе с ОС.

В состав ОС Альт 8 СП включены следующие дополнительные системные приложения:

- архиваторы;
- приложения для управления RPM-пакетами;
- приложения резервного копирования;
- приложения мониторинга системы;
- приложения для работы с файлами;
- приложения для настройки системы;
- настройка параметров загрузки;
- настройка оборудования;
- настройка сети.

2.6. Программы веб-серверов

Программы веб-серверов участвуют в организации доступа пользователей к сети Интернет. Доступ организуется с помощью клиент-серверной архитектуры.

Клиент, которым обычно является веб-браузер, передает программе веб-сервера запросы на получение ресурсов. В качестве ресурсов могу выступать HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту. В ответ веб-сервер передает клиенту запрошенные данные. Обмен происходит по протоколу HTTP.

В состав ОС Альт 8 СП включены программы веб-сервера Apache.

2.7. Интерактивные рабочие среды

Интерактивные рабочие среды – программы (пакеты программ), предназначенные для работы пользователя в ОС Альт 8 СП и предоставляющие ему удобный интерфейс для общения с ней.

2.8. Командные интерпретаторы

Командные интерпретаторы – специальные программы (терминалы), предназначенные для выполнения различных команд пользователей при работе с ОС Альт 8 СП.

2.9. Графическая оболочка MATE

Графическая оболочка MATE – набор программ и технологий, предназначенных для управления ОС Альт 8 СП и предоставляющих пользователю графический интерфейс для работы.

2.10. Системы управления базами данных

Системы управления базами данных (далее – СУБД) – приложения, предназначенные для работы с данными, представленными в виде набора записей. СУБД осуществляет поиск, обработку и хранение данных в виде специальных таблиц, являющихся базой данных.

2.11. Электронные справочники

Электронные справочники – наборы внутрисистемных справочных страниц, описывающих работу команд и приложений, которые выполнены в виде примеров HOWTOs и справки `man`.

3. ПОДГОТОВИТЕЛЬНЫЕ ПРОЦЕДУРЫ

3.1. Процедуры приемки

3.1.1. Процедура верификации ПИ

Проверка поставленного потребителю дистрибутива производится путем подсчета контрольной суммы с использованием программы фиксации и контроля исходного состояния программного комплекса «ФИКС» (версия 2.0.1) (сертификат № 913, выдан ФСТЭК России 28 мая 2004 года, действителен до 01 июня 2019 года) по алгоритму «Уровень-3» и сравнения ее с контрольной суммой, указанной в приложении документа «Формуляр. ЛКНВ.11100-01 30 01» и на этикетке ПИ.

3.1.2. Минимальные системные требования для безопасной установки

Технические требования к конфигурации компьютера для работы ОС Альт 8 СП:

- процессор архитектуры Эльбрус-4С/8С;
- RAM (оперативная память) – минимум 512 Мбайт (рекомендуется от 1 Гбайт и более);
- наличие свободного места на HDD (жестком диске) – не менее 30 Гбайт;
- наличие DVD-ROM (привода dvd дисков) для инсталляции дистрибутива.

3.1.3. Информация о требованиях для среды функционирования

Для среды функционирования ОС Альт 8 СП предъявляются следующие требования:

- ОС Альт 8 СП должен быть совместим с СВТ (ИС), в котором (которой) он функционирует;
- должны быть обеспечены установка, конфигурирование и управление ОС Альт 8 СП в соответствии с эксплуатационной документацией;
- должна быть обеспечена защита от осуществления действий, направленных на нарушение физической целостности СВТ, на котором функционирует ОС Альт 8 СП;
- должна быть обеспечена доверенная загрузка ОС (блокирование попыток несанкционированной загрузки, контроль доступа субъектов доступа к процессу загрузки, контроль целостности компонентов загружаемой операционной среды);
- должны быть обеспечены необходимые ресурсы для выполнения функциональных возможностей безопасности операционной системы, хранения резервных копий, создаваемых операционной системой, а также

защищенное хранение данных операционной системы и защищаемой информации;

- должно быть обеспечено ограничение на установку программного обеспечения и его компонентов, не задействованных в технологическом процессе обработки информации;
- должен обеспечиваться доверенный маршрут между ОС и пользователями ОС (администраторами, пользователями);
- должен обеспечиваться доверенный канал передачи данных между ОС и средствами вычислительной техники, на которых происходит обработка информации, а также с которых происходит их администрирование;
- должна быть обеспечена невозможность отключения (обхода) компонентов ОС;
- должны быть реализованы меры, препятствующие несанкционированному копированию информации, содержащейся в ОС, на съемные машинные носители информации (или за пределы информационной системы). В том числе должен осуществляться контроль вноса (выноса) в (из) контролируемую зону (контролируемой зоны) съемных машинных носителей информации;
- должна осуществляться проверка целостности внешних модулей уровня ядра, получаемых от заявителя (разработчика, производителя), перед их установкой в операционную систему;
- должно быть обеспечено выделение вычислительных ресурсов для процессов в соответствии с их приоритетами;
- персонал, ответственный за функционирование ОС Альт 8 СП, должен обеспечивать функционирование ОС Альт 8 СП, в точности руководствуясь эксплуатационной документацией;
- лица, ответственные за эксплуатацию ОС Альт 8 СП, должны обеспечить, чтобы аутентификационная информация для каждой учетной записи пользователя ОС содержались в тайне и были недоступны лицам, не уполномоченным использовать данную учетную запись;
- должна обеспечиваться возможность генерации аутентификационной информации соответствующей метрике качества.

3.2. Настройка опций безопасности

Во время установки ОС Альт 8 СП администратор должен обеспечить выполнение следующих условий:

- задать пользователя с паролем, отвечающим требованиям безопасности;
- задать пароль администратора, отвечающего требованиям безопасности.

Перед началом эксплуатации ОС Альт 8 СП администратор должен обеспечить выполнение следующих условий:

1) настроить параметры входа пользователя (порядок действий приведен в документе «Руководство по комплексу средств защиты. ЛКНВ.11100-01 99 02»):
   - время ограничения сеанса;
   - время засыпания;
2) настроить параметры пароля пользователя (порядок действий приведен в документе «Руководство по комплексу средств защиты. ЛКНВ.11100-01 99 02»);
   - сложность пароля;
   - время действия;
3) настроить параметры проверки целостности (порядок действий в документе «Руководство по комплексу средств защиты. ЛКНВ.11100-01 99 02»);
4) настроить параметры запрета удаления файлов (порядок действий приведен п. 7.3 настоящего документа);
5) настроить разрешенные сервисы;
6) настроить аудит:
   - правила аудита (порядок действий приведен в документе «Руководство по комплексу средств защиты. ЛКНВ.11100-01 99 02»);
   - настроить экспорт аудита на другой компьютер (порядок действий приведен п. 7.15 настоящего документа);
7) настроить подключение оповещений администратора (порядок действий приведен п. 7.16 настоящего документа);
8) для защиты от атаки подбора пароля (brute force) – внести изменения в файл `/etc/pam.d/sshd` – добавить строку:
   ```
   auth required pam_tally2.so deny=3 unlock_time=19
   ```

3.3. Описание механизмов устранения идентифицированных скрытых каналов

Механизмы защиты, направленные на ограничение, мониторинг, полное или частичное устранение идентифицированных скрытых каналов, которые могут возникнуть в информационных (автоматизированных) системах вследствие использования в них ОО:

1) Исключение возможности работы с общими каталогами с правом записи для пользователей, имеющих разные полномочия доступа.

2) Для противодействия атакам на каналы передачи по времени и памяти необходимо администратором безопасности исключить наличие в системе общих для пользователей файловых ресурсов, где размещаются файлы с разными ДРД, в

частности исключить размещение в каталогах файлов, доступ к которым полностью закрыт для конкретных пользователей данного каталога. Также можно монтировать файловую систему без учета времени доступа: `mount -noatime –nodiratime`.

3) На уровне ядра запретить процессам создавать слушающие сокеты, кроме тех, что им действительно необходимы, в том числе запрещать слушать на фиксированном порту, а также контролировать частоту создания сокета.

4) Монтировать подсистему `/proc` с флагом `hidepid=2` или `1`. При этом имена процессов других пользователей и другие данные таких процессов будут недоступны вызывающему непривилегированному пользователю.

5) Организовать маскирующие процессы, имитирующие постоянную загрузку процессора. Использовать механизмы ограничения CPU для процессов, гарантирующий время выполнения, одинаковое для всех процессов, такой как cgroups.

6) Для предотвращения Timestamp Evaluation – отключить отметки времени TCP в ОС Альт 8 СП. Для этого выполнить следующие команды:

```
# echo 0 > /proc/sys/net/ipv4/tcp_timestamps
To make that change permenant though, you need to add the
following line to /etc/sysctl.conf:
net.ipv4.tcp_timestamps = 0
```

также можно настроить правила iptables:

```
iptables -A INPUT -p icmp --icmp-type timestamp-request -j DROP
iptables -A OUTPUT -p icmp --icmp-type timestamp-reply -j DROP
```

7) Для предотвращения ISN Evaluation (оценка временной отметки) – использовать TCP/IP прокси (socks).

8) Для предотвращения TCP URG Pointer (указателя TCP URG) – настроить правила iptables:

```
iptables -N BADFLAGS
iptables -A BADFLAGS -j LOG --log-prefix "BADFLAGS: "
iptables -A BADFLAGS -j DROP
iptables -N TCP_FLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ACK,FIN FIN                 -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ACK,PSH PSH                 -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ACK,URG URG                 -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags FIN,RST FIN,RST             -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags SYN,FIN SYN,FIN             -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags SYN,RST SYN,RST             -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ALL     ALL                 -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ALL     NONE                -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ALL     FIN,PSH,URG         -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ALL     SYN,FIN,PSH,URG     -j BADFLAGS
iptables -A TCP_FLAGS -p tcp --tcp-flags ALL     SYN,RST,ACK,FIN,URG -j BADFLAGS
```

9) Для предотвращения IP ToS Evaluation (Оценки IP-ToS) – настроить способ обслуживания для telnet, ftp-control и ftp-data – выполнить команды:

```
# iptables -A PREROUTING -t mangle -p tcp --sport telnet \
  -j TOS --set-tos Minimize-Delay
# iptables -A PREROUTING -t mangle -p tcp --sport ftp \
  -j TOS --set-tos Minimize-Delay
# iptables -A PREROUTING -t mangle -p tcp --sport ftp-data \
  -j TOS --set-tos Maximize-Throughput
```

Эти правила прописываются на удаленном хосте и воздействуют на входящие, по отношению к компьютеру, пакеты. Для пакетов, отправляемых в обратном направлении, эти флаги устанавливаются автоматически. Настроить их можно, прописав следующие правила:

```
# iptables -A OUTPUT -t mangle -p tcp --dport telnet \
  -j TOS --set-tos Minimize-Delay
# iptables -A OUTPUT -t mangle -p tcp --dport ftp \
  -j TOS --set-tos Minimize-Delay
# iptables -A OUTPUT -t mangle -p tcp --dport ftp-data \
  -j TOS --set-tos Maximize-Throughput
```

Для противодействия данной атаке необходимо в командной строке выполнить следующие команды:

```
# Разрешить главные типы протокола ICMP
iptables -A OUTPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 4 -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 11 -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 12 -j ACCEPT
```

Типы ICMP-сообщений:

- 0 – echo reply (echo-ответ, пинг);
- 3 – destination unreachable (адресат недосягаем);
- 4 – source quench (подавление источника, просьба посылать пакеты медленнее);
- 5 – redirect (редирект);
- 8 – echo request (echo-запрос, ping);
- 9 – router advertisement (объявление маршрутизатора);
- 10 – router solicitation (ходатайство маршрутизатора);
- 11 – time-to-live exceeded (истечение срока жизни пакета);
- 12 – IP header bad (неправильный IP заголовок пакета);
- 13 – timestamp request (запрос значения счетчика времени);
- 14 – timestamp reply (ответ на запрос значения счетчика времени);
- 15 – information request (запрос информации);
- 16 – information reply (ответ на запрос информации);
- 17 – address mask request (запрос маски сети);
- 18 – address mask reply (ответ на запрос маски сети).

10) Для предовращения Initial Sequence Number hijacking and spoofing (урона и подделки исходного кода последовательности) – настроить правила iptables:

```
# Защита от спуфинга
iptables -I INPUT -m conntrack --ctstate NEW,INVALID -p tcp --
tcp-flags SYN,ACK SYN,ACK -j REJECT --reject-with tcp-reset


# Защита от SYN-флуда
iptables -A INPUT -p tcp --syn -m limit --limit 10/s --limit-
burst 50 -j ACCEPT
iptables -A INPUT -p udp -m limit --limit 10/s --limit-burst 50 -
j ACCEPT
iptables -A INPUT -p icmp -m limit --limit 10/s --limit-burst 50
-j ACCEPT
iptables -A INPUT -j DROP
# Отбрасывать ошибочные пакеты
iptables -A INPUT -m state --state INVALID -j DROP
iptables -I INPUT -m conntrack --ctstate INVALID -j DROP
# Отбрасывать фрагментированные пакеты
iptables -A INPUT -f -j DROP
# Защита от попытки открыть входящее соединение TCP не через SYN
iptables -I INPUT -m conntrack --ctstate NEW -p tcp ! --syn -j
DROP
# Защита от Ping of death
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --
limit 10/s --limit-burst 50 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
# Защита от некорректных ICMP
iptables -I INPUT -p icmp -f -j DROP
# Отбросить ошибочные пакеты
iptables -A FORWARD -m state --state INVALID -j DROP
iptables -I FORWARD -m conntrack --ctstate INVALID -j DROP
# Отбросить фрагментированные пакеты
iptables -A FORWARD -f -j DROP
# Сбрасывать фрагментированные пакеты
iptables -A OUTPUT -f -j DROP
```

Дополнительно необходимо внести правки в `/etc/sysctl.conf`:

```
$ sudo vi /etc/systcl.conf
# Отбросить ICMP-редиректы (против атак типа MITM)
net.ipv4.conf.all.accept_redirects=0
net.ipv6.conf.all.accept_redirects=0
# Включить механизм TCP syncookies
net.ipv4.tcp_syncookies=1
# Различные улучшения (защита от спуфинга
# увеличение очереди «полуоткрытых» TCP-соединений и далее):
net.ipv4.tcp_timestamps=0
net.ipv4.conf.all.rp_filter=1
net.ipv4.tcp_max_syn_backlog=1280
kernel.core_uses_pid=1
```

4. ФУНКЦИИ И ЗАДАЧИ АДМИНИСТРИРОВАНИЯ ОС АЛЬТ 8 СП

4.1. Режимы работы

Имеется два режима работы:

- обычный режим (пользовательский);
- режим администрирования.

Все обычные действия пользователей выполняются в обычном режиме. В обычном режиме недоступны операции создания и удаления пользователей, а также операция изменения меток защиты. В обычном режиме выполняться все свойства безопасности.

В режиме администрирования защиты могут выполняться обычные операции от имени администратора, а также могут создаваться и удаляться пользователи.

При переходе в обычный режим администратор отвечает за проверку выполнения и соблюдения свойств безопасности. Администратор имеет возможность определять и модифицировать режим выполнения функций безопасности, связанных с: аудитом безопасности, идентификацией и аутентификацией, управлением доступа, обеспечением надежного функционирования, контролем целостности, с защитой памяти. Списки контроля доступа модифицируются только администратором в режиме администрирования.

4.2. Функции администратора

Основными функциями администратора при эксплуатации ОС Альт 8 СП являются:

- ввод в эксплуатацию и эксплуатация в соответствии с указаниями, приведенными в документе «Формуляр. ЛКНВ.11100-01 30 01»;
- соблюдение подготовительных процедур (см. раздел 3);
- установка и настройка ОС Альт 8 СП;
- управление и поддержка функционирования персональной электронной вычислительной машины (далее – ПЭВМ).

4.3. Задачи администрирования

В состав основных задач администрирования входят следующие:

- установка ОС Альт 8 СП и назначение параметров системы;
- создание загрузочных носителей информации;
- конфигурирование параметров даты и времени, графической среды, средств ввода и вывода;
- настройка и управление системными сервисами и служебными программами;
- настройка и управление работой системы управления пакетами Advanced Packaging Tool (далее – APT);
- обновление ОС и прикладного ПО из ее состава;
- настройка и управление учетными записями и правами доступа пользователей;
- конфигурирование сети `/etc/net` и проверка ее работоспособности;
- настройка FTP-северов;
- настройка служб DNS;
- настройка серверов электронной почты postfix;
- настойка и управление кэширующими прокси-серверами;
- настройка серверного и клиентского программного обеспечения (далее – ПО) Samba для осуществления связи UNIX-машин с сетями Microsoft и LanManager;
- настройка и управление печатью;
- настройка и управление базами данных.

5. НАСТРОЙКА ОС АЛЬТ 8 СП

5.1. Подготовка к установке ОС Альт 8 СП

Для установки ПИ ОС Альт 8 СП необходимо выбрать способ первоначальной загрузки компьютера и перейти в режим установки.

5.1.1. Способы первоначальной загрузки

Инструкция для создания и загрузки с USB-flash (аналогично можно создать загрузочный жесткий диск).

Для создания загрузочной USB-flash, на ней должен быть один раздел, отформатированный в файловой системе ext2.

Далее необходимо скопировать на нее все содержимое загрузочного DVD-диска, включая каталог (`.disk`).

5.1.1.1.1. Загрузка с USB-flash

1) При загрузке компьютера дождаться появления сообщения `Autoboot Press space to disable it`, нажать пробел. Появится меню редактирования загрузки.

2) Для просмотра подключенных дисков необходимо нажать клавишу `d`. На экран выведется информация о подключенных дисках. В списке дисков необходимо найти подключенную USB-flash и запомнить ее номер диска (Drv [10]) и номер раздела на ней (Partition [0]).

3) Для редактирования параметров загрузки нажмите клавишу `c`. В поле `Enter drive number` введите номер диска USB-flash (10), в поле `Enter Partition number` введите номер раздела на ней. Остальные вопросы можно пропустить, нажав клавишу `Esc`.

4) Для дальнейшей загрузки нажмите `b`. Появится командный интерфейс загрузчика (boot#), нажмите клавишу `Enter`. Начнется загрузка системы.

5) Дождаться появления окна выбора загрузочного диска и нажать кнопку «Cancel».

6) В окне выбора метода установки выбрать метод `Hard disk` и нажать кнопку «Ok».

7) В окне выбора загрузочного диска выбрать диск с USB-flash (sdb Ultra USB 3.0) и нажать кнопку «Ok».

8) В окне выбора загрузочного раздела выбрать первый раздел (sdb1) и нажать клавишу «Ok».

9) В окне выбора каталога, содержащего установочные файлы, ввести корневой каталог «/» и нажать кнопку «Ok».

10) Дождаться начала установки.

5.1.2. Загрузка с DVD

1) При загрузке компьютера дождаться появления сообщения `Autoboot Press space to disable it`, нажать пробел. Появится меню редактирования загрузки.

2) Для просмотра подключенных дисков необходимо нажать клавишу `d`. На экран выведется информация о подключенных дисках. В списке дисков необходимо найти подключенный DVD и запомнить его номер диска (Drv [10]).

3) Для редактирования параметров загрузки нажмите клавишу `c`. В поле `Enter drive number` введите номер диска DVD (10). Остальные вопросы можно пропустить, нажав клавишу «Esc».

4) Для дальнейшей загрузки нажмите клавишу `b`. Появится командный интерфейс загрузчика (boot#), нажмите клавишу `Enter`. Начнется загрузка системы.

5.2. Установка системы

5.2.1. Последовательность установки

До того, как будет произведена установка базовой системы на жесткий диск, программа установки работает с образом системы, загруженным в оперативную память компьютера.

Если инициализация оборудования завершилась успешно, будет запущен графический интерфейс программы-установщика. Процесс установки разделен на шаги; каждый шаг посвящен настройке или установке определенного свойства системы. Шаги нужно проходить последовательно, переход к следующему шагу происходит по нажатию кнопки «Далее». При помощи кнопки «Назад» при необходимости можно вернуться к уже пройденному шагу и изменить настройки.

Однако на этом этапе установки возможность перехода к предыдущему шагу ограничена теми шагами, где нет зависимости от данных, введенных ранее.

В случае необходимости отмены установки, необходимо нажать кнопку <Reset> на корпусе системного блока компьютера.

П р и м е ч а н и е . Совершенно безопасно выполнить отмену установки только до шага «Подготовка диска», поскольку до этого момента не производится никаких изменений на жестком диске. Если прервать установку между шагами «Подготовка диска» и «Сохранение настроек», существует вероятность, что после этого с жесткого диска не сможет загрузиться ни одна из установленных ОС (если такие имеются).

Технические сведения о ходе установки можно посмотреть, нажав <Ctrl>+<Alt>+<F1>, вернуться к программе установки – <Ctrl>+<Alt>+<F7>. По нажатию <Ctrl>+<Alt>+<F2> откроется отладочная виртуальная консоль.

Каждый шаг сопровождается краткой справкой, которую можно вызвать, нажав <F1>.

Во время установки системы выполняются следующие шаги:

- язык;
- подтверждение согласия;
- дата и время;
- подготовка диска;
- установка системы;
- сохранение настроек;
- настройка сети;
- администратор системы;
- системный пользователь;
- в случае создания LUKS разделов – этап установки пароля на LUKS разделы;
- завершение установки.

5.2.2. Язык

Установка начинается с выбора основного языка – языка интерфейса программы установки и устанавливаемой системы (рис. 2).

Также на данном этапе выбирается вариант переключения раскладки клавиатуры. Раскладка клавиатуры – это привязка букв, цифр и специальных символов к клавишам на клавиатуре. Переключение между раскладками осуществляется при помощи специально зарезервированных для этого клавиш.

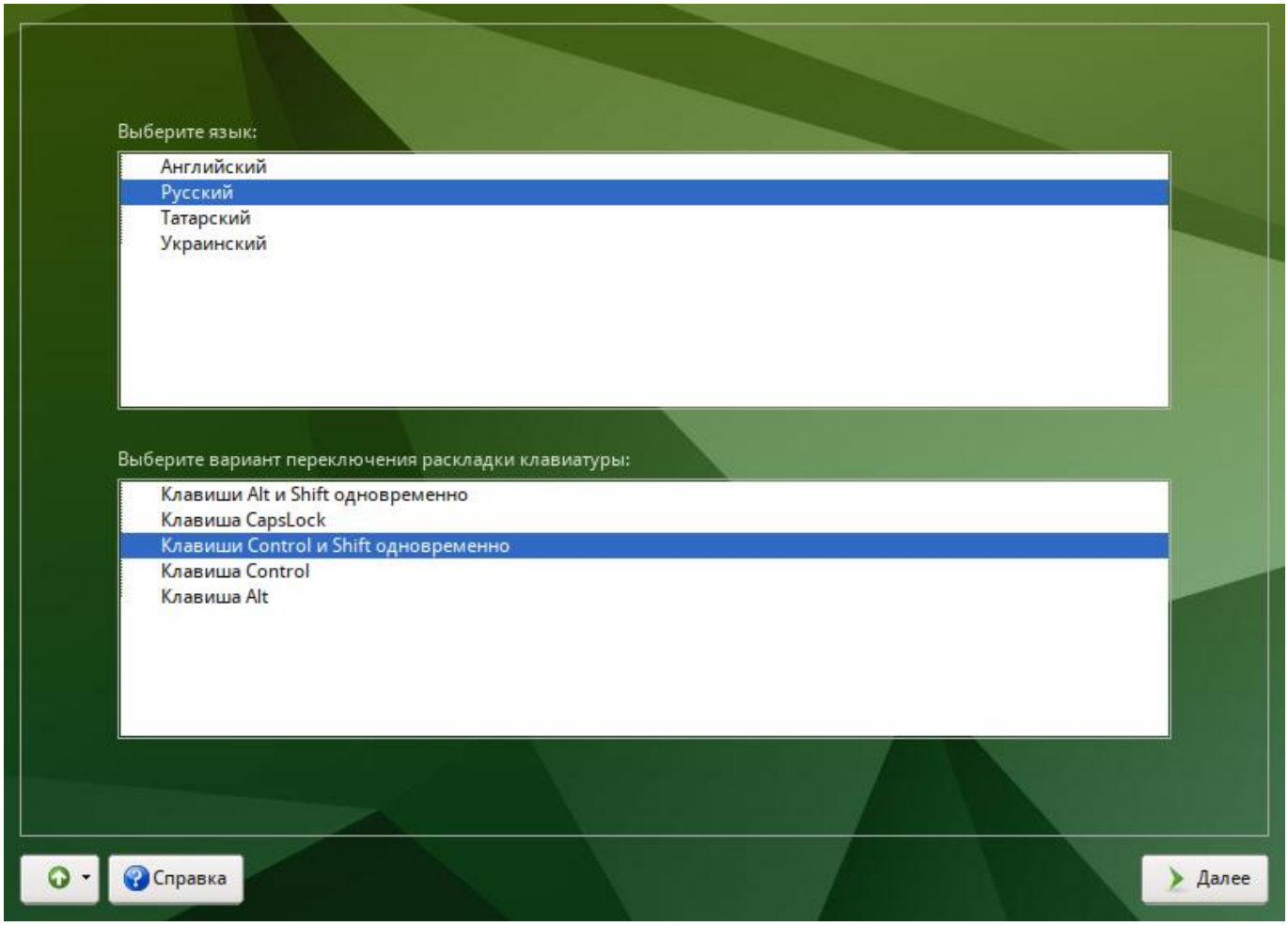

Рис. 2 – Установка. Выбор языка

Для настройки варианта переключения раскладки клавиатуры в пункте «Выберите вариант переключения раскладки клавиатуры:» необходимо установить одно из следующих значений (доступно при выборе русского языка, в качестве основного):

- «клавиши Alt и Shift одновременно»;
- «клавиша CapsLock»;
- «клавиши Control и Shift одновременно»;
- «клавиша Control»;
- «клавиша Alt».

Если выбранный основной язык имеет всего одну раскладку (например, при выборе английского языка в качестве основного), эта единственная раскладка будет принята автоматически.

После завершения настройки основного языка и варианта переключения раскладки клавиатуры необходимо нажать кнопку «Далее».

5.2.3. Подтверждение согласия

После окна выбора языковых параметров ОС Альт 8 СП программа установки переходит к окну «Подтверждение согласия» (рис. 3).

Перед продолжением установки следует внимательно прочитать условия, регулирующие права владельца экземпляра дистрибутива ОС Альт 8 СП на использование дистрибутива, а также включенных в состав дистрибутива отдельных программ для ЭВМ в установленных условиями пределах.

Для подтверждения согласия, необходимо отметить пункт «Да, я согласен с условиями» и нажать кнопку «Далее».

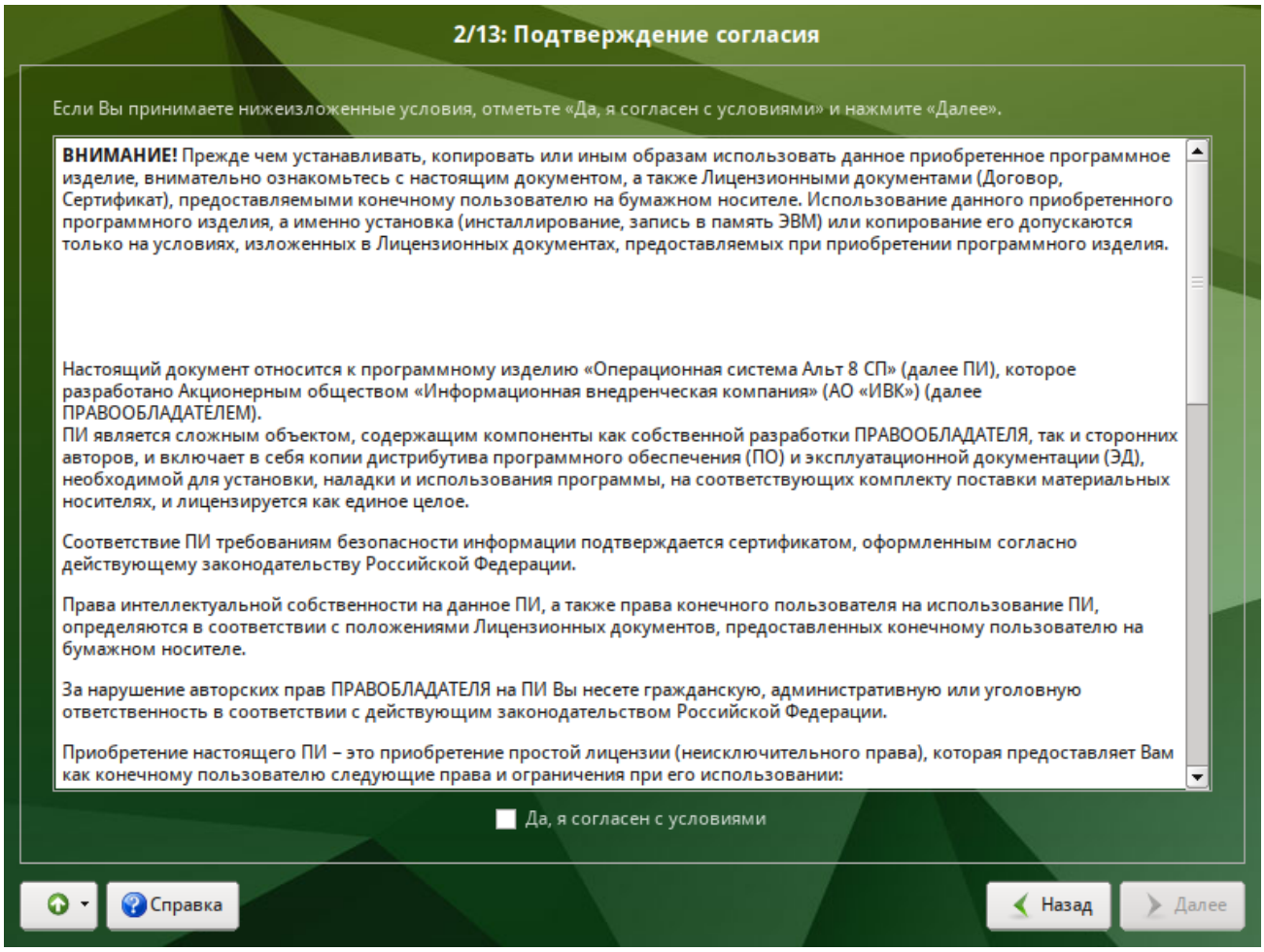

Рис. 3 – Установка. Подтверждение согласия

5.2.4. Дата и время

После выбора языка системы ОС Альт 8 СП программа установки переходит к окну «Дата и время». На данном этапе выполняется выбор страны и города, по которым будет определен часовой пояс и установлены системные часы (рис. 4).

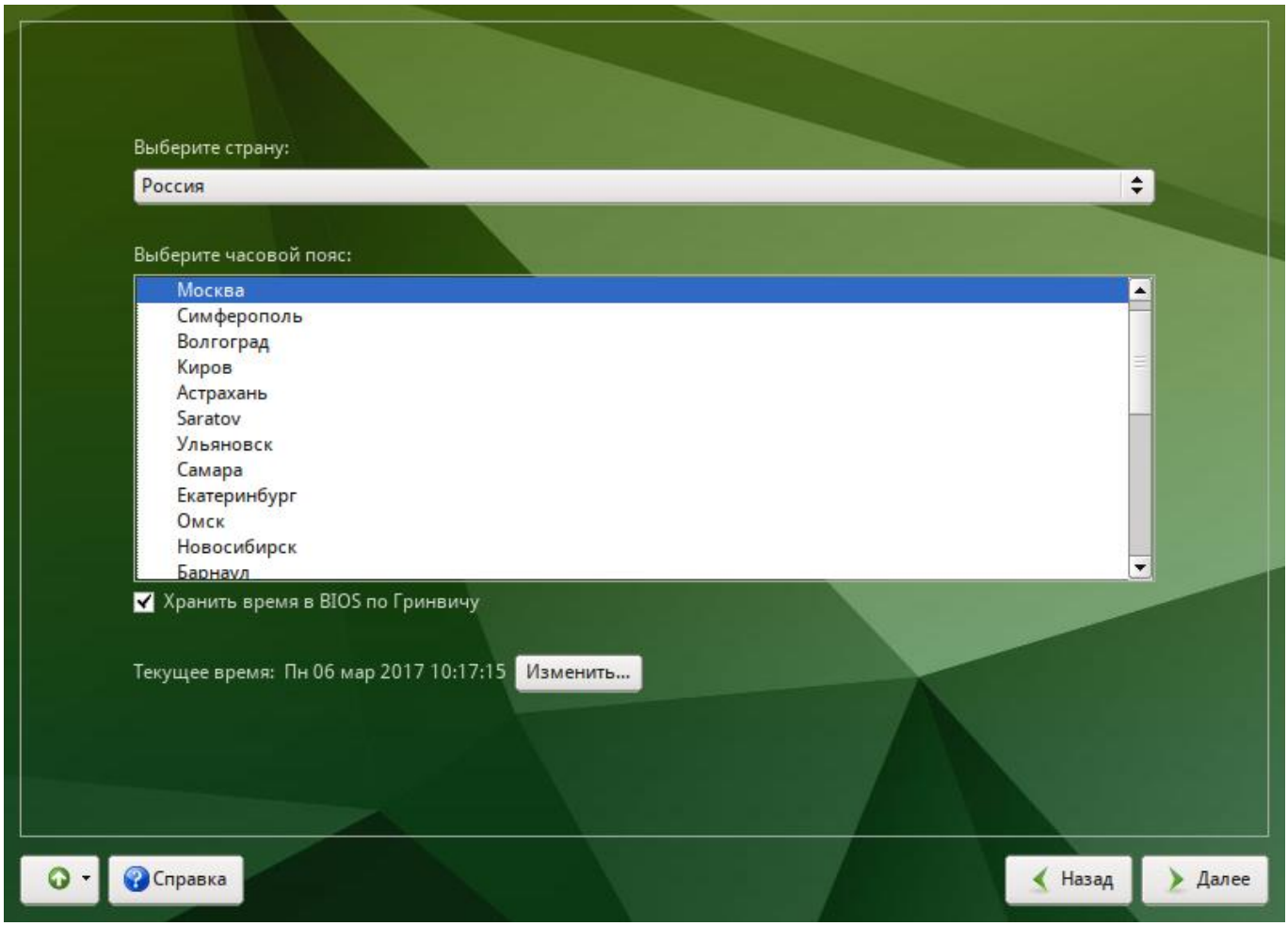

Рис. 4 – Установка. Выбор часового пояса

На этом шаге следует выбрать часовой пояс, по которому нужно установить часы. Для этого в соответствующих списках выберите страну, а затем регион. Поиск по списку можно ускорить, набирая на клавиатуре первые буквы искомого слова.

Пункт «Хранить время в BIOS по Гринвичу» выставляет настройки даты и времени в соответствии с часовыми поясами, установленными по Гринвичу, и добавляет к местному времени часовую поправку для выбранного региона.

После выбора часового пояса будут предложены системные дата и время по умолчанию.

Для ручной установки текущих даты и времени нужно нажать кнопку «Изменить». Откроется окно ручной настройки системных параметров даты и времени (рис. 5).

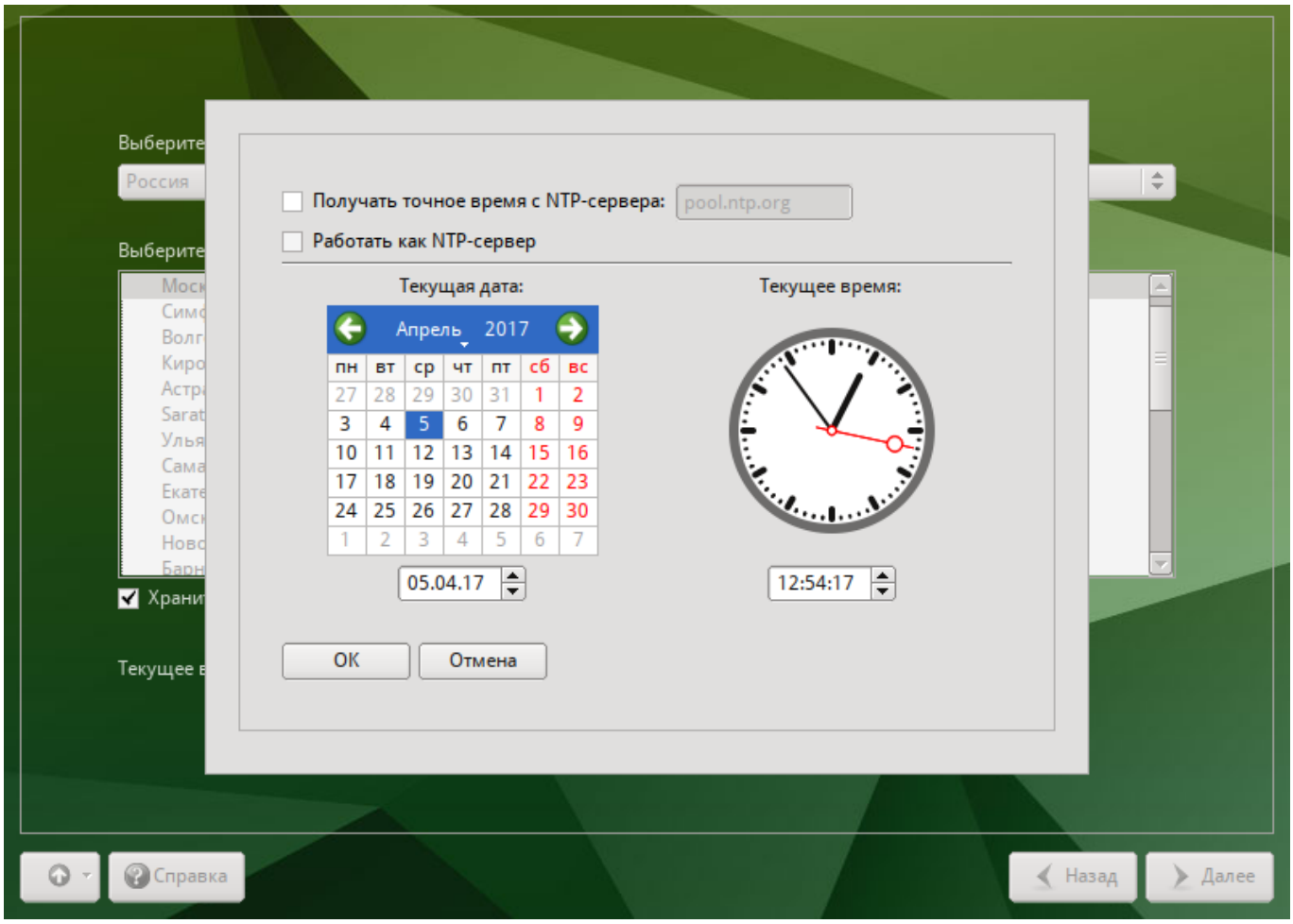

Рис. 5 – Установка. Настройка времени

Для синхронизации системных часов (NTP) с удаленным сервером по локальной сети или по сети Интернет нужно отметить пункт «Получать точное время с NTP-сервера» и указать предпочитаемый NTP-сервер. В большинстве случаев можно указать сервер `pool.ntp.org`.

Для работы компьютера в качестве сервера точного времени внутри локальной сети нужно отметить пункт «Работать как NTP-сервер».

Для сохранения настроек и продолжения установки системы в окне ручной установки даты и времени необходимо нажать кнопку «ОК» и затем в окне «Дата и время» нажать кнопку «Далее».

Примечание. В случае если ОС Альт 8 СП устанавливается как вторая ОС, необходимо снять отметку с пункта «Хранить время в BIOS по Гринвичу», иначе время в уже установленной ОС может отображаться некорректно.

5.2.5. Подготовка диска

На этом этапе программа установки подготавливает площадку для установки ОС Альт 8 СП, в первую очередь – выделяется свободное место на диске.

Переход к этому шагу может занять некоторое время – период ожидания может быть разным и зависит от производительности компьютера, объема жесткого диска, количества разделов на нем и других параметров.

5.2.5.1. Выбор профиля разбиения диска

После завершения первичной конфигурации загрузочного носителя откроется окно «Подготовка диска» (рис. 6). В списке разделов перечислены уже существующие на жестких дисках разделы (в том числе здесь могут оказаться съемные USB-носители, подключенные к компьютеру в момент установки).

В списке «Выберите метод установки:» перечислены доступные профили разбиения диска. Профиль – это шаблон распределения места на диске для установки ОС. Можно выбрать один из профилей:

- использовать неразмеченное пространство;
- удалить все разделы и создать разделы автоматически;
- подготовить разделы вручную.

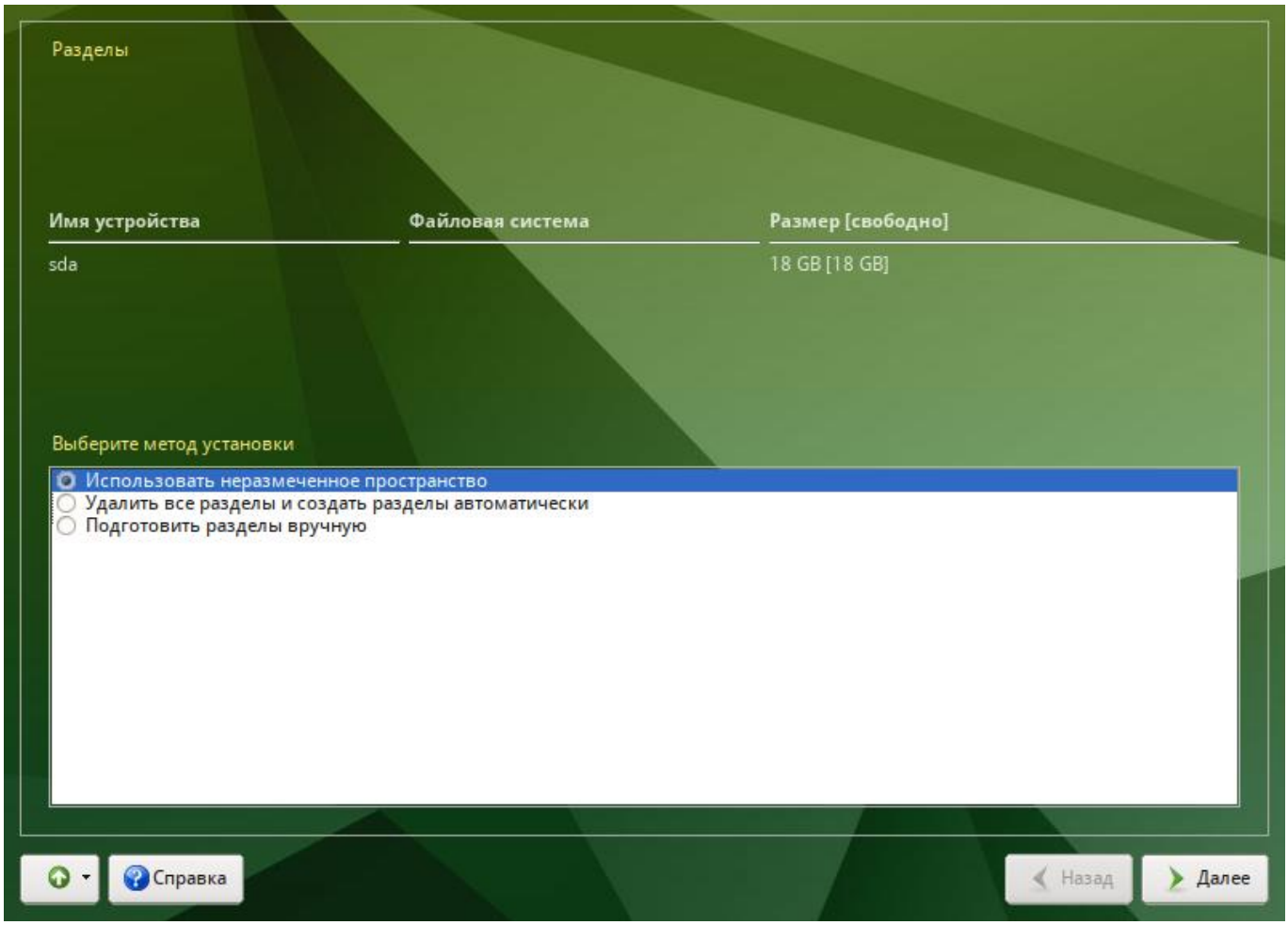

Рис. 6 – Установка. Выбор профиля разбиения диска(ов)

5.2.5.2. Автоматические профили разбиения диска

Первые два профиля предполагают автоматическое разбиение диска. Выбор автоматического профиля разбиения диска влияет также и на предлагаемый по умолчанию профиль устанавливаемого программного обеспечения.

Если происходит установка сервера, то при разбиении диска будут выделены отдельные разделы для подкачки и для корневой файловой системы. Оставшееся место будет отведено под файловую систему, содержащую данные серверных приложений `/var`.

Если происходит установка рабочей станции, то при разбиении диска будут выделены отдельные разделы для подкачки и для корневой файловой системы. Оставшееся место будет отведено под файловую систему, содержащую домашние каталоги пользователей `/home`.

П р и м е ч а н и е . При использовании автоматических профилей разбиения дисков, соответствующие изменения на диске происходят сразу же по нажатию кнопки «Далее» – будет произведена запись новой таблицы разделов на НЖМД и форматирование разделов.

Если при применении одного из профилей автоматического разбиения диска доступного места на диске окажется недостаточно, то на экран будет выведено сообщение об ошибке:

```
Невозможно применить профиль, недостаточно места на диске
```

Если данное сообщение появилось после попытки применить профиль «Использовать неразмеченное пространство», то можно полностью очистить место на диске, применив профиль «Удалить все разделы и создать разделы автоматически».

Если сообщение о недостатке места на диске появляется и при применении профиля «Удалить все разделы и создать разделы автоматически», то это связано с недостаточным для использования автоматических методов разметки объемом всего диска. В этом случае следует воспользоваться профилем разбиения «Подготовить разделы вручную».

П р и м е ч а н и е . При применении профиля «Удалить все разделы и создать разделы автоматически» будут удалены все данные со всех дисков (включая внешние USB-носители) без возможности восстановления. Рекомендуется использовать эту возможность при полной уверенности в том, что диски не содержат никаких ценных данных.

5.2.5.3. Ручной профиль разбиения диска

При необходимости освободить часть дискового пространства следует воспользоваться профилем разбиения «Подготовить разделы вручную». Можно удалить некоторые из существующих разделов или содержащиеся в них файловые системы. После этого можно создать необходимые разделы самостоятельно или вернуться к шагу выбора профиля и применить один из автоматических профилей. Выбор этой возможности требует знаний об устройстве диска и технологиях его разбиения.

По нажатию «Далее» будет произведена запись новой таблицы разделов на диск и форматирование разделов. Разделы, только что созданные на диске программой установки, пока не содержат данных и поэтому форматируются без предупреждения. Уже существовавшие, но измененные разделы, которые будут отформатированы, помечаются специальным значком в колонке «Файловая система» слева от названия. При уверенности в том, что подготовка диска завершена, подтвердите переход к следующему шагу нажатием кнопки «OK».

5.2.5.4. Дополнительные возможности разбиения диска

Ручной профиль разбиения диска позволяет установить ОС на программный RAID-массив, разместить разделы в томах LVM и использовать маскирование на разделах. Данные возможности требуют от пользователя понимания принципов функционирования указанных технологий.

5.2.5.4.1. Создание программного RAID-массива

Избыточный массив независимых дисков RAID (redundant array of independent disks) – технология виртуализации данных, которая объединяет несколько НЖМД в логический элемент для избыточности и повышения производительности.

Для создания RAID-массива необходимо два и более жестких диска. Программа установки поддерживает создание программных RAID-массивов следующих типов:

- RAID 1;
- RAID 0;
- RAID 4/5/6;
- RAID 10.

Процесс подготовки к установке на RAID условно можно разбить на следующие шаги:

- создание разделов на жестких дисках;
- создание RAID-массивов на разделах жесткого диска;
- создание файловых систем на RAID-массиве.

Для настройки параметров нового раздела из состава RAID-массива необходимо выбрать неразмеченный диск в окне профиля разбивки пространства «Подготовить разделы вручную» и нажать кнопку «Создать раздел». После этого откроется окно (рис. 7).

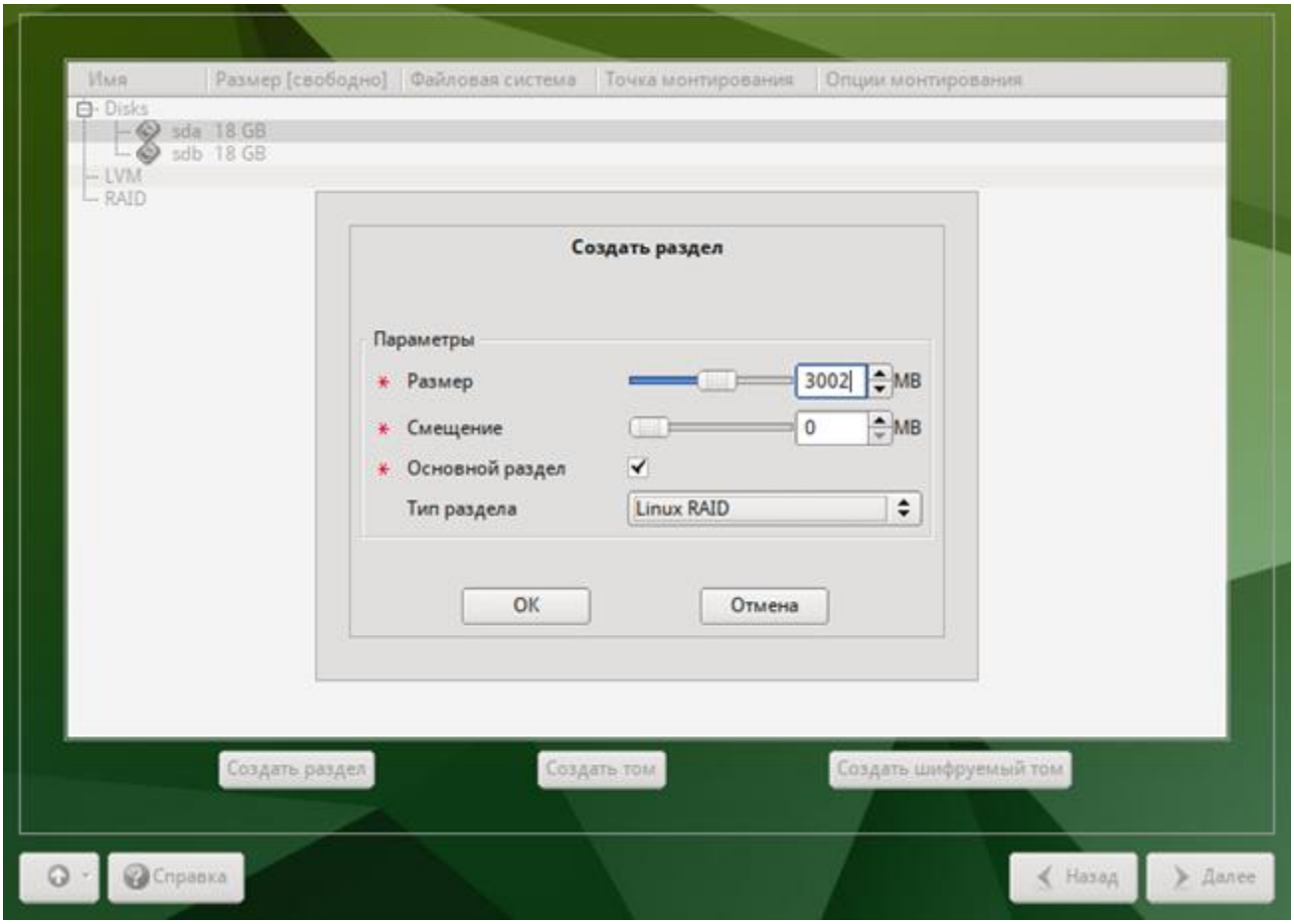

Рис. 7 – Установка. Создание раздела Linux RAID

В этом окне необходимо настроить следующие параметры:

- «Размер» –в поле необходимо указать размер будущего раздела в Мбайт;
- «Смещение» – в поле необходимо указать смещение начала данных на диске в Мбайт;
- «Основной раздел» – необходимо отметить пункт, если раздел является основным для установки ОС;
- «Тип раздела» – в выпадающем поле нужно выбрать значение «Linux RAID» для последующего включения раздела в RAID-массивы.

Примечание. Объем результирующего массива может зависеть от размера, включенных в него разделов жесткого диска.

После создания разделов на дисках можно переходить к организации самих RAID-массивов. Для этого в списке следует выбрать пункт «RAID», после чего нажать кнопку «Создать RAID». Далее мастер предложит выбрать тип массива и указать его участников (рис. 8, рис. 9).

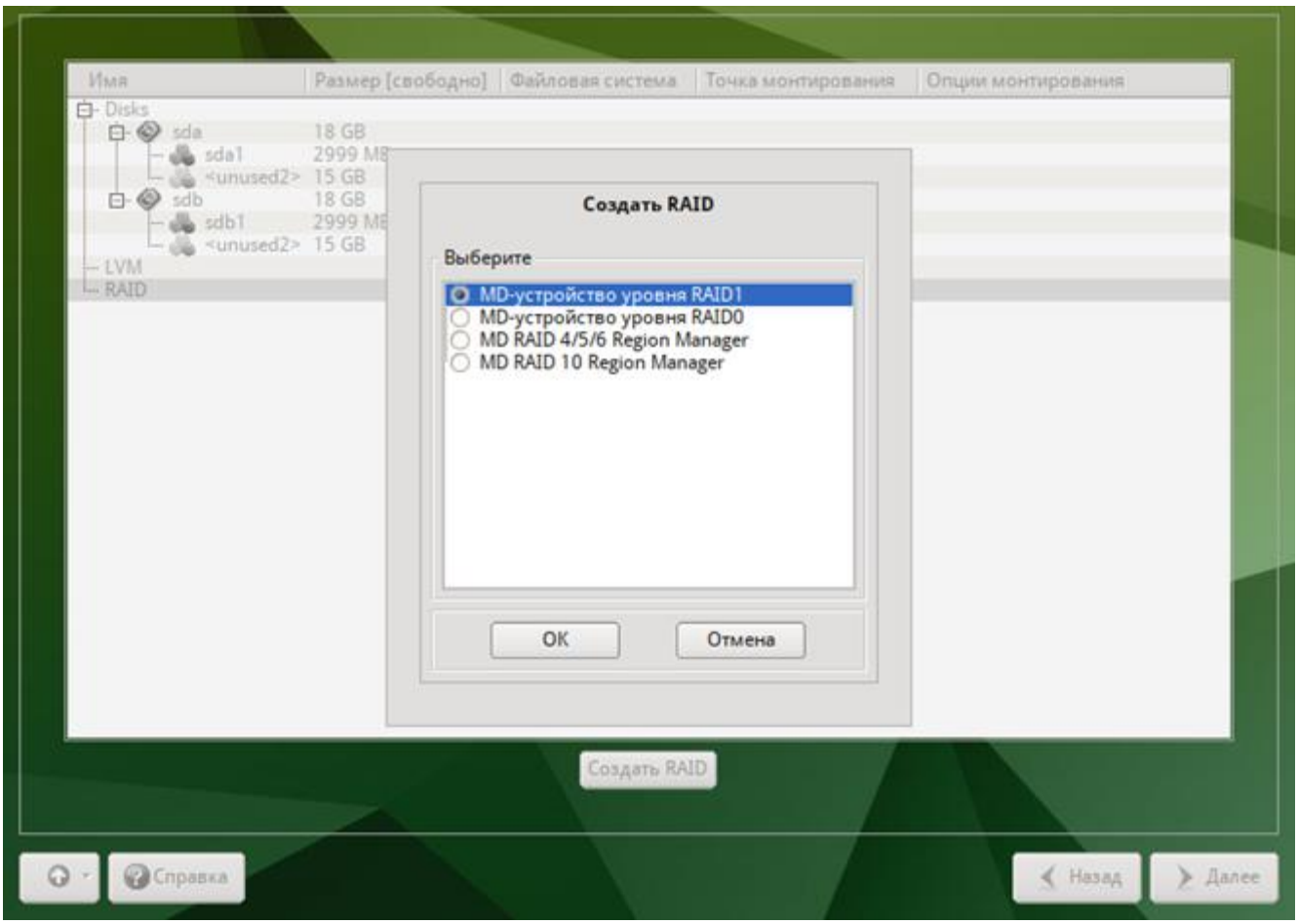

Рис. 8 – Установка. Выбор типа RAID-массива

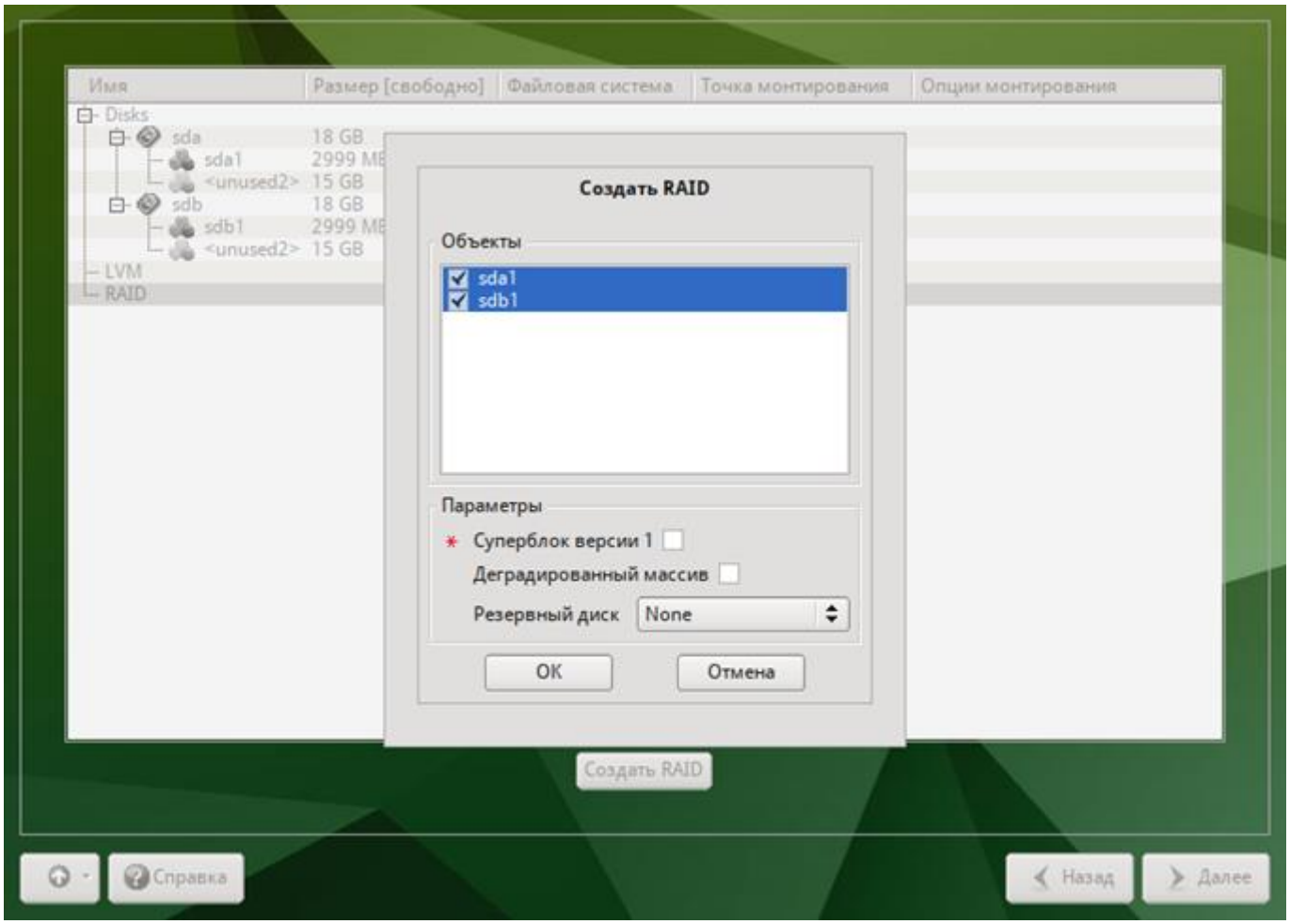

Рис. 9 – Установка. Выбор участников RAID-массива

После создания RAID-массивов их можно использовать как обычные разделы на жестких дисках, то есть, на них можно создавать файловые системы или же, например, включать их в LVM-тома.

5.2.5.4.2. Создание LVM-томов

Менеджер логических дисков LVM (Logical Volume Manager) – средство гибкого управления дисковым пространством, позволяющее создавать поверх физических разделов (либо неразбитых дисков) логические тома, которые в самой системе будут видны как обычные блочные устройства с данными (обычные разделы).

Процесс подготовки к установке на LVM можно разбить на следующие шаги:

- создание группы томов LVM;
- создание томов LVM;
- создание файловых систем на томах LVM.

Примечание. Для создания группы томов LVM может потребоваться предварительно удалить таблицу разделов с жесткого диска.

Для создания группы томов LVM в списке следует выбрать пункт «LVM», после чего нажать кнопку «Создать группу томов» (рис. 10).

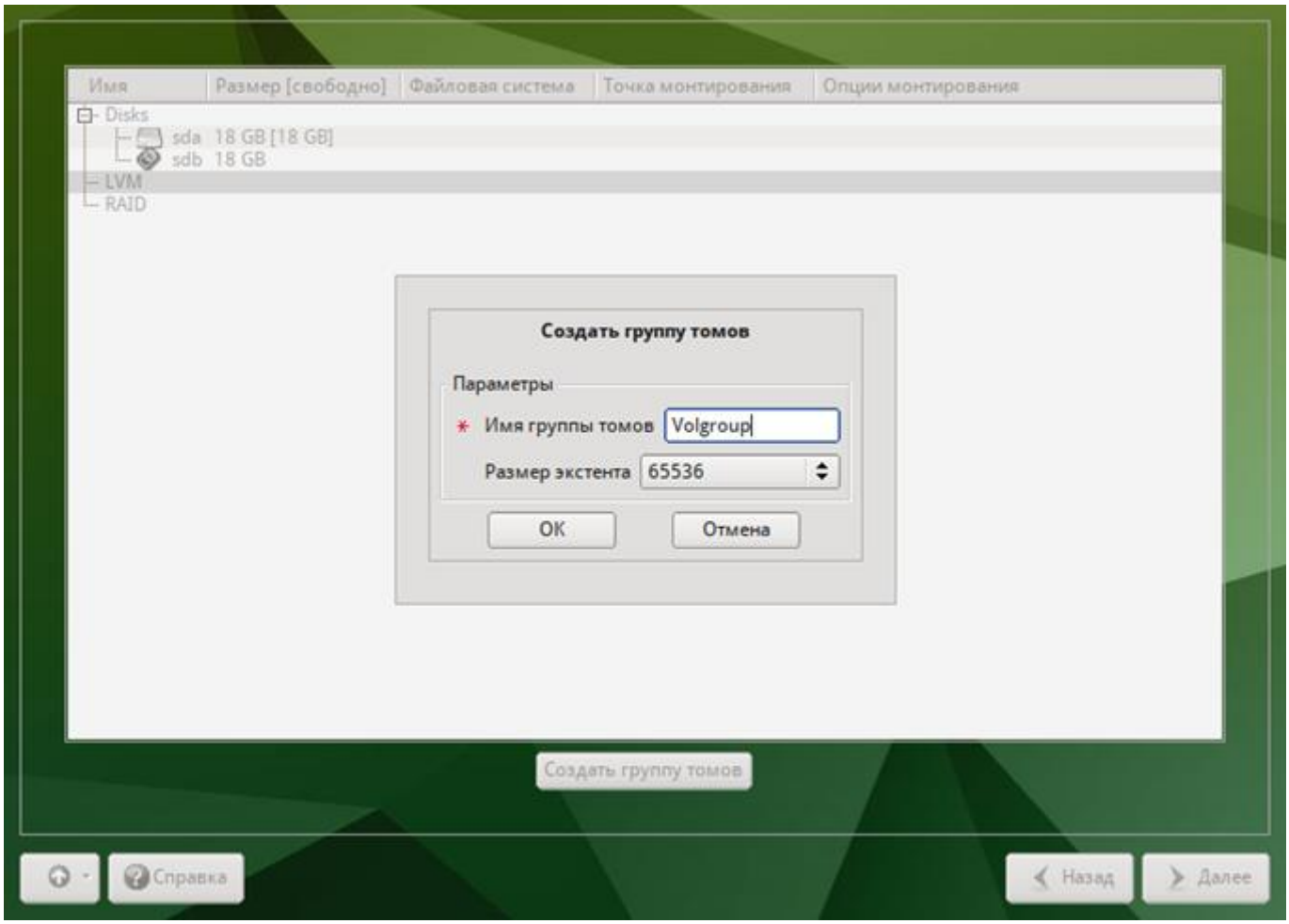

Рис. 10 – Установка. Создание группы томов LVM

После создания группы томов LVM ее можно использовать как обычный жесткий диск, то есть внутри группы томов можно создавать тома (аналог раздела на физическом жестком диске) и файловые системы внутри томов (рис. 11).

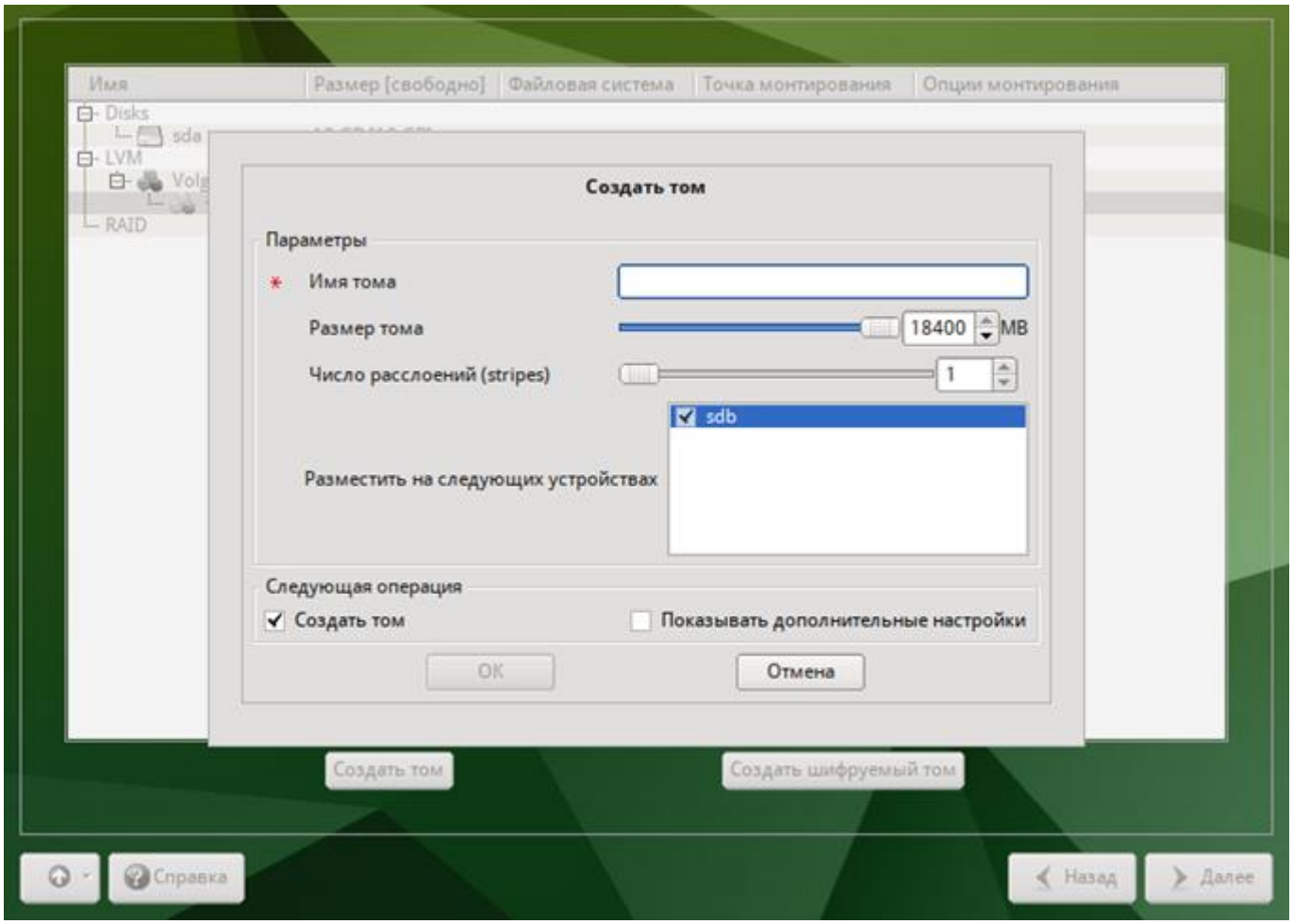

Рис. 11 – Установка. Создание тома

5.2.5.4.3. Создание кодированных разделов

Программа установки ОС Альт 8 СП позволяет создавать кодированные разделы с использованием встроенных средств маскирования.

Для создания кодированного раздела и выполнения дальнейшей разметки нужно выбрать требуемый диск и нажать кнопку «Создать шифруемый раздел».

В открывшемся окне доступны следующие настройки (рис. 12):

- «Размер» – общий размер шифрованного тома;
- «Смещение» – настройка осуществляется с помощью ползунка либо путем ввода значения с клавиатуры (в поле необходимо указать смещение начала данных на диске в Мбайт);
- «Основной раздел» – необходимо отметить пункт, если раздел является основным для установки ОС;
- «Тип раздела» – в выпадающем поле нужно выбрать значение «Linux»;

- «Создать шифруемый том» – отметить пункт для автоматического перехода к настройке файловой системы на данном разделе;
- «Показывать дополнительные настройки» – отметить пункт для отображения дополнительных настроек при последующей работе с разделом.

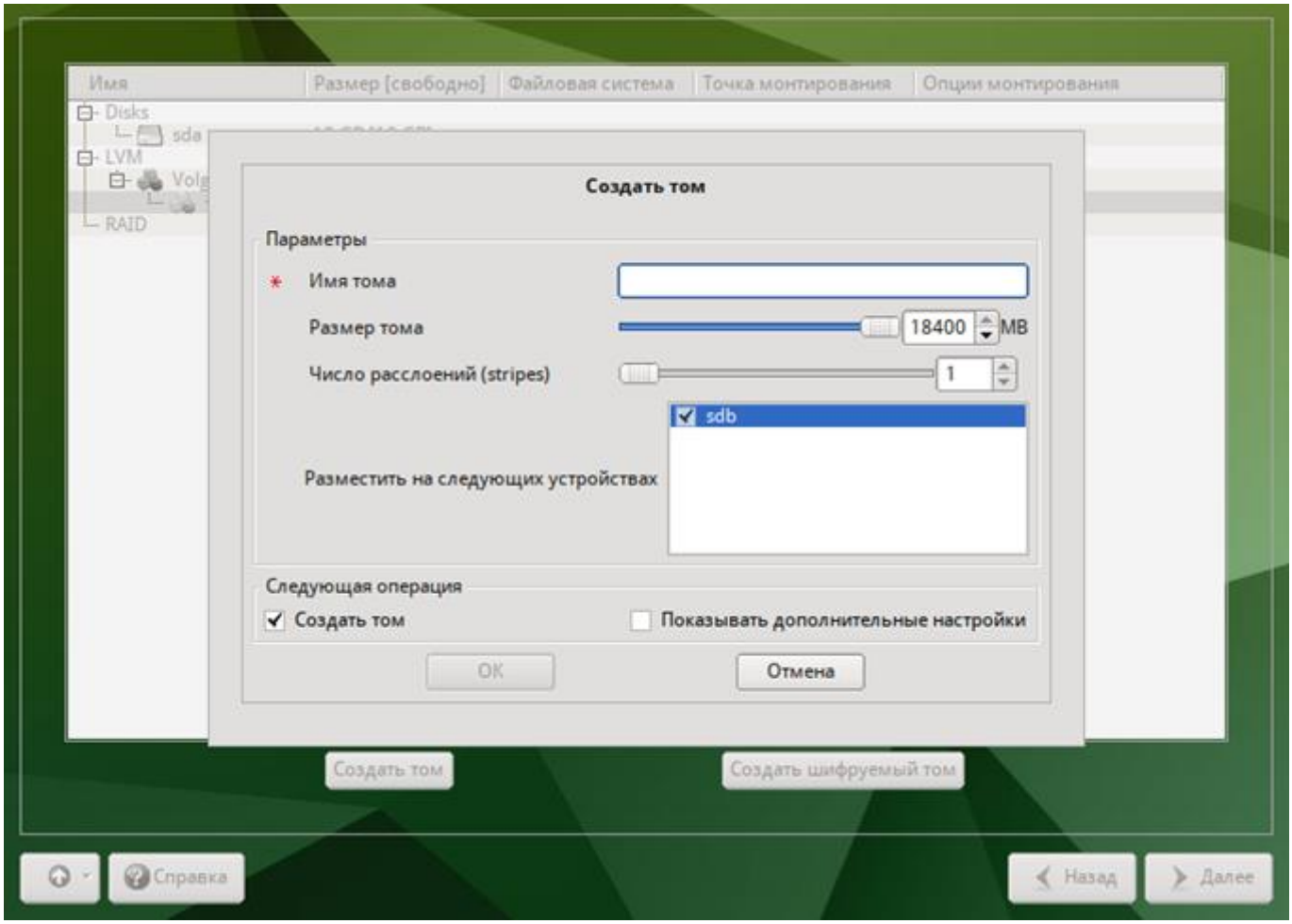

Рис. 12 – Установка. Создание кодируемого раздела

После создания кодированного раздела мастер, как и при создании обычного раздела, предложит создать на нем файловую систему и при необходимости потребует указать точку монтирования.

Для сохранения всех внесенных настроек и продолжения установки в окне «Подготовка диска» нужно нажать кнопку «Далее».

5.2.6. Установка системы

На данном этапе происходят распаковка ядра и установка набора программ, необходимых для работы ОС Альт 8 СП.

Программа установки предлагает выбрать дополнительные пакеты программ, которые будут включены в состав ОС Альт 8 СП и установлены вместе с ней на диск (рис. 13).

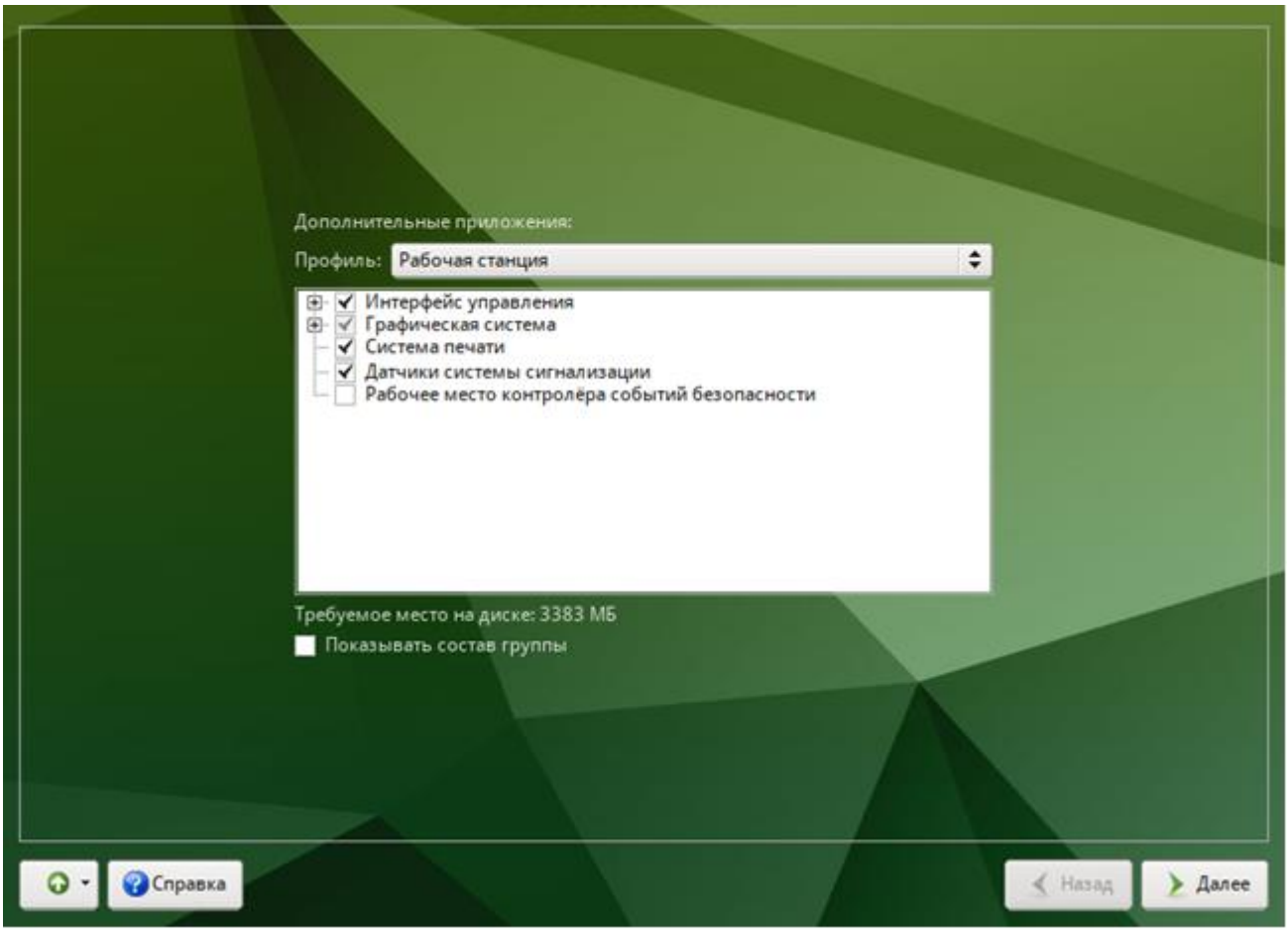

Рис. 13 – Установка. Выбор групп пакетов

При установке рабочей станции доступны следующие профили:

- «Рабочая станция» – для установки будет предложена графическая оболочка MATE и набор офисных графических приложений;
- «Минимальная установка» – дополнительное ПО в состав устанавливаемых пакетов включаться не будет.

При установке сервера доступны следующие профили:

- «Офисный сервер» – для установки будут предложены группы пакетов с серверными приложениями;
- «Сервер Samba-DC (контроллер AD)» – для установки будут предложена группа пакетов для конфигурации сервера в качестве контроллера AD;

- «Минимальная установка» – дополнительное ПО в состав устанавливаемых пакетов включаться не будет.

После выбора профиля можно изменить состав устанавливаемых пакетов.

Под списком групп на экране отображается информация об объеме дискового пространства, которое будет занято после установки пакетов, входящих в выбранные группы.

Необходимо учитывать, что пакеты «Рабочее место контролера событий безопасности» и «Датчики системы сигнализации» конфликтуют между собой – допускается выбор только одного из пакетов.

Опция «Показать состав группы» выводит список программных пакетов, входящих в состав той или иной группы пакетов (рис. 14).

Если была отмечена для установки группа «Среда MATE» (по умолчанию в профиле «Рабочая станция»), то графическая оболочка MATE будет автоматически запускаться при загрузке операционной системы автоматически.

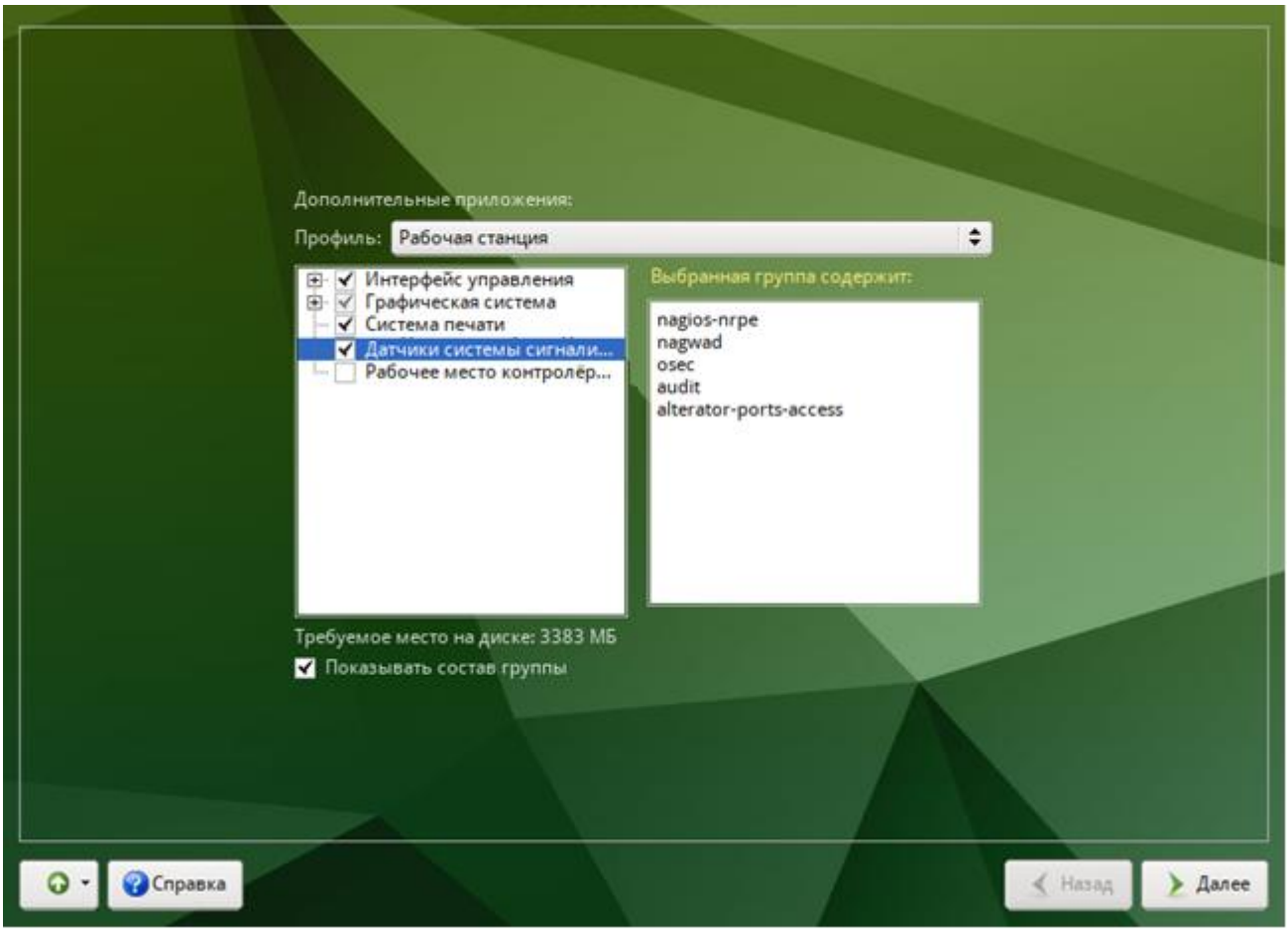

Рис. 14 – Установка. Состав группы пакетов

Выбрав профиль и группы пакетов, следует нажать «Далее», после чего начнется установка пакетов (рис. 15).

Установка происходит автоматически в два этапа:

- получение пакетов;

- установка пакетов.

Получение пакетов осуществляется с источника, выбранного на этапе начальной загрузки. При сетевой установке (по протоколу FTP или HTTP) время выполнения этого шага будет зависеть от скорости соединения.

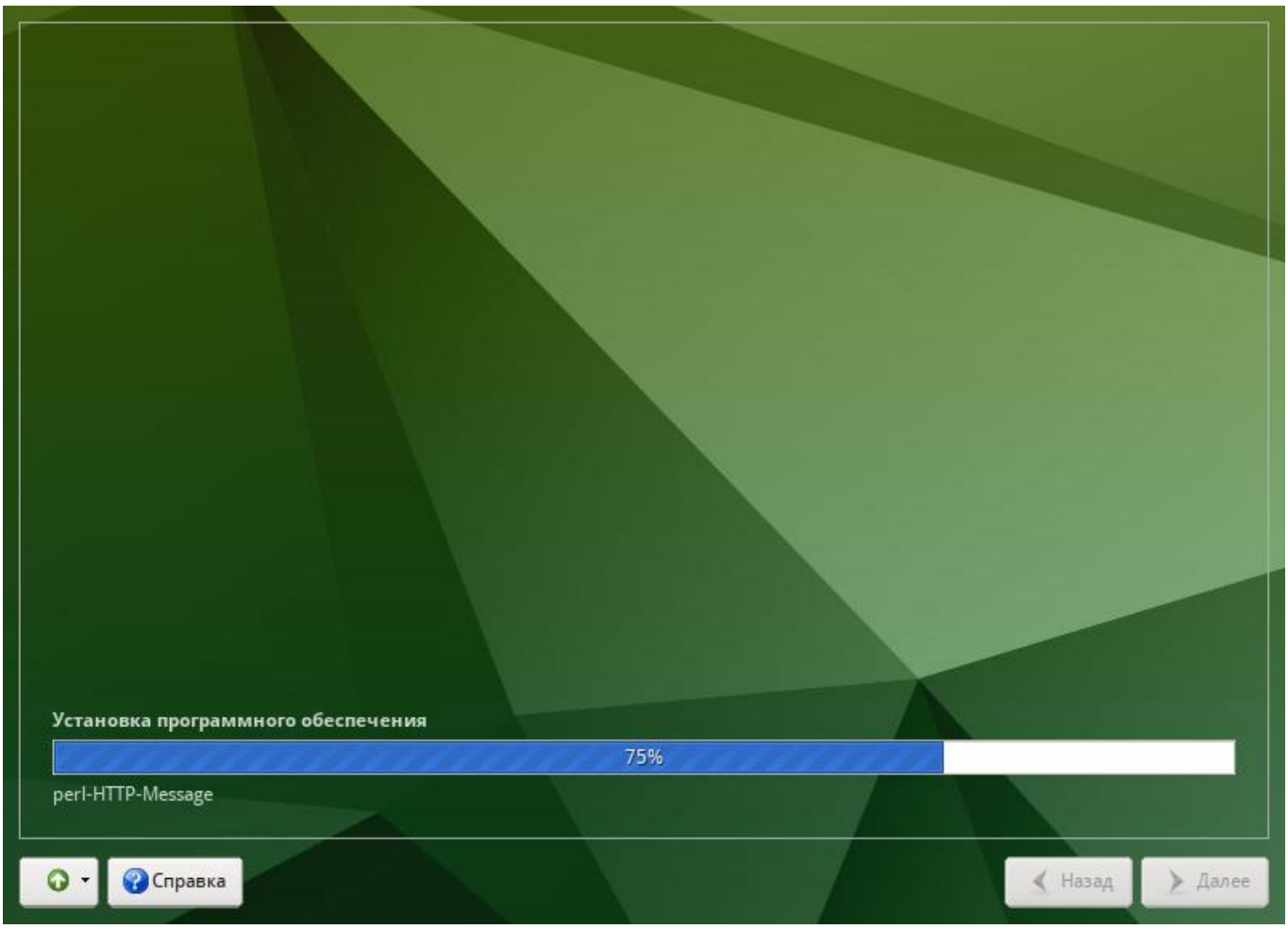

Рис. 15 – Установка. Установка пакетов

5.2.7. Сохранение настроек

Начиная с данного этапа, программа установки работает с файлами только что установленной базовой системы. Все последующие изменения можно будет совершить после завершения установки посредством редактирования

соответствующих конфигурационных файлов или при помощи модулей управления, включенных в дистрибутив.

После завершения установки базовой системы выполняется шаг сохранения настроек (рис. 16). Он проходит автоматически и не требует вмешательства пользователя, на экране отображается индикатор выполнения.

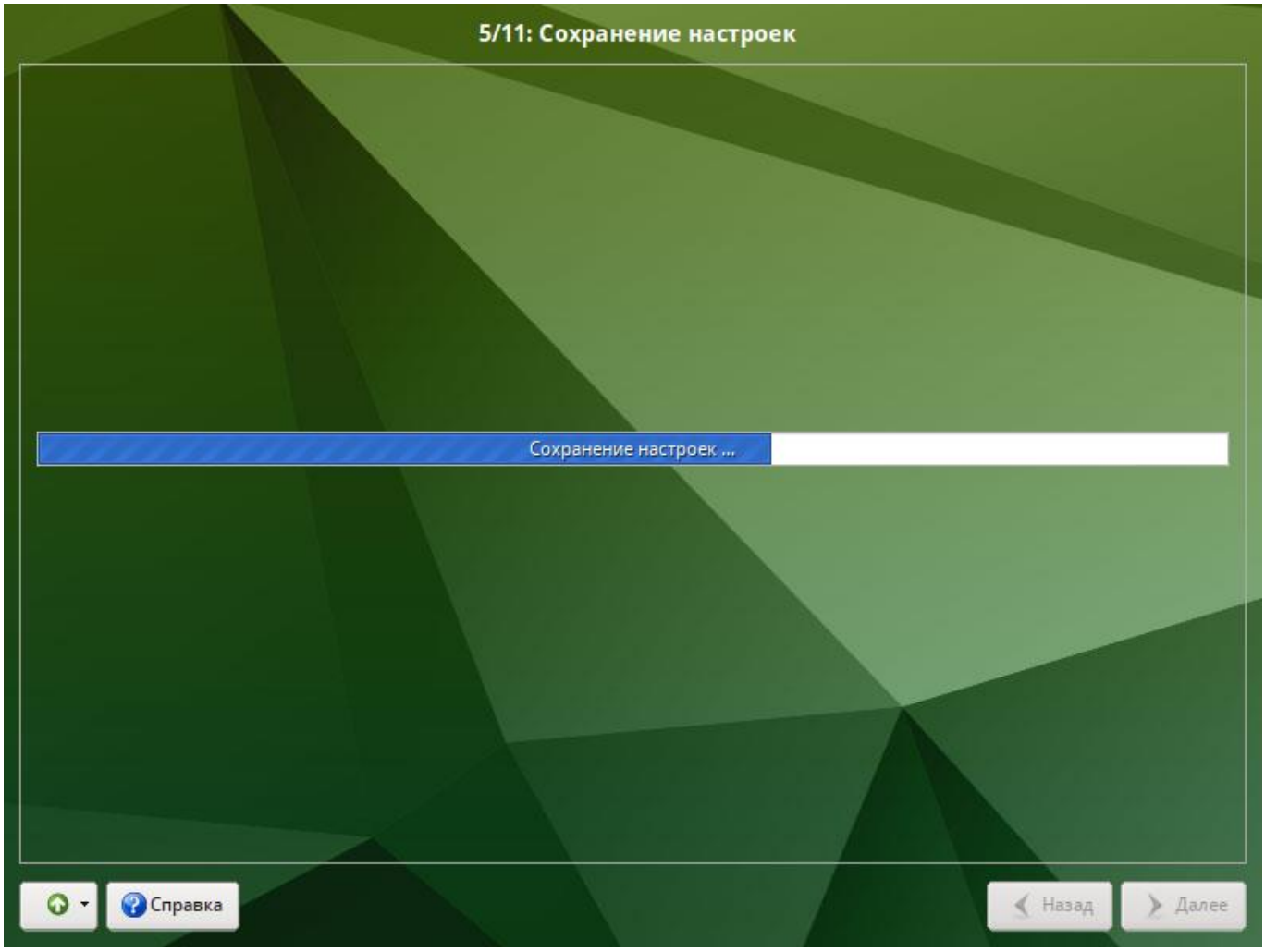

Рис. 16 – Установка. Сохранение настроек

На данном этапе производится перенос настроек, выполненных на первых шагах установки, в установленную базовую систему. Также производится запись информации о соответствии разделов жесткого диска смонтированным на них файловым системам (заполняется конфигурационный файл `/etc/fstab`). В список доступных источников программных пакетов добавляется репозиторий, находящийся на установочном лазерном диске – выполняется команда

`apt-cdrom add`, осуществляющая запись в конфигурационный файл `/etc/apt/sources.list`.

После сохранения настроек осуществляется автоматический переход к следующему шагу.

5.2.8. Настройка сети

На этом этапе в окне «Настройка сети» необходимо задать параметры работы сетевой карты и настройки сети (рис. 17):

- «Имя компьютера:» – указать сетевое имя ПЭВМ в поле для ввода имени компьютера;
- «Интерфейсы:» – выбрать доступный сетевой интерфейс, для которого будут выполняться настройки;
- «Версия протокола IP:» – указать в выпадающем списке версию используемого протокола IP (IPv4 либо IPv6) и убедиться, что пункт «Включить», обеспечивающий поддержку работы протокола, отмечен;
- «Конфигурация:» – выбрать способ назначения IP-адресов (службы DHCP, Zeroconf либо вручную);
- «IP-адреса:» – пул назначенных IP-адресов из поля «IP:», выбранные адреса можно удалить нажатием кнопки «Удалить»;
- «IP:» – ввести IP-адрес вручную и выбрать в выпадающем поле предпочтительную маску сети, затем нажать кнопку «Добавить» для переноса адреса в пул поля «IP-адреса:»;
- «Шлюз по умолчанию:» – в поле для ввода необходимо ввести адрес шлюза, который будет использоваться сетью по умолчанию;
- «DNS-серверы:» – в поле для ввода необходимо ввести список предпочтительных DNS-серверов, которые будут получать информацию о доменах, выполнять маршрутизацию почты и управлять обслуживающими узлами для протоколов в домене;
- «Домены поиска:» – в поле для ввода необходимо ввести список предпочтительных доменов, по которым будет выполняться поиск.

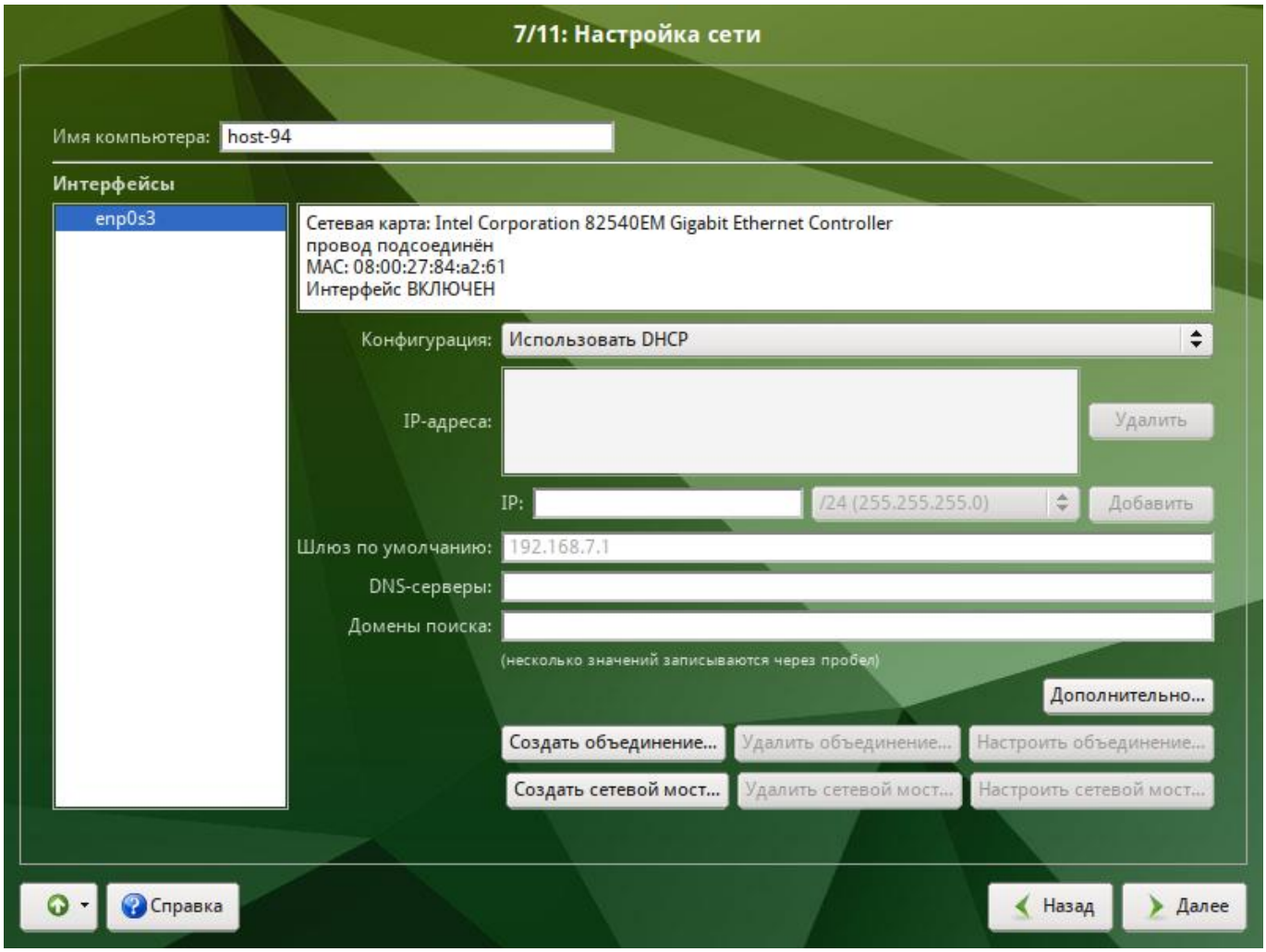

Рис. 17 – Установка. Настройка сети

Конкретные значения будут зависеть от используемого сетевого окружения. Ручного введения настроек можно избежать, если в сети уже есть настроенный DHCP-сервер. В этом случае все необходимое сетевые настройки будут получены автоматически.

Для сохранения настроек сети и продолжения работы программы установки необходимо нажать кнопку «Далее».

5.2.9. Администратор системы

На данном этапе создается учетная запись администратора (рис. 18). В открывшемся окне необходимо ввести пароль учетной записи администратора (root). Чтобы исключить опечатки при вводе пароля, пароль учетной записи вводится дважды.

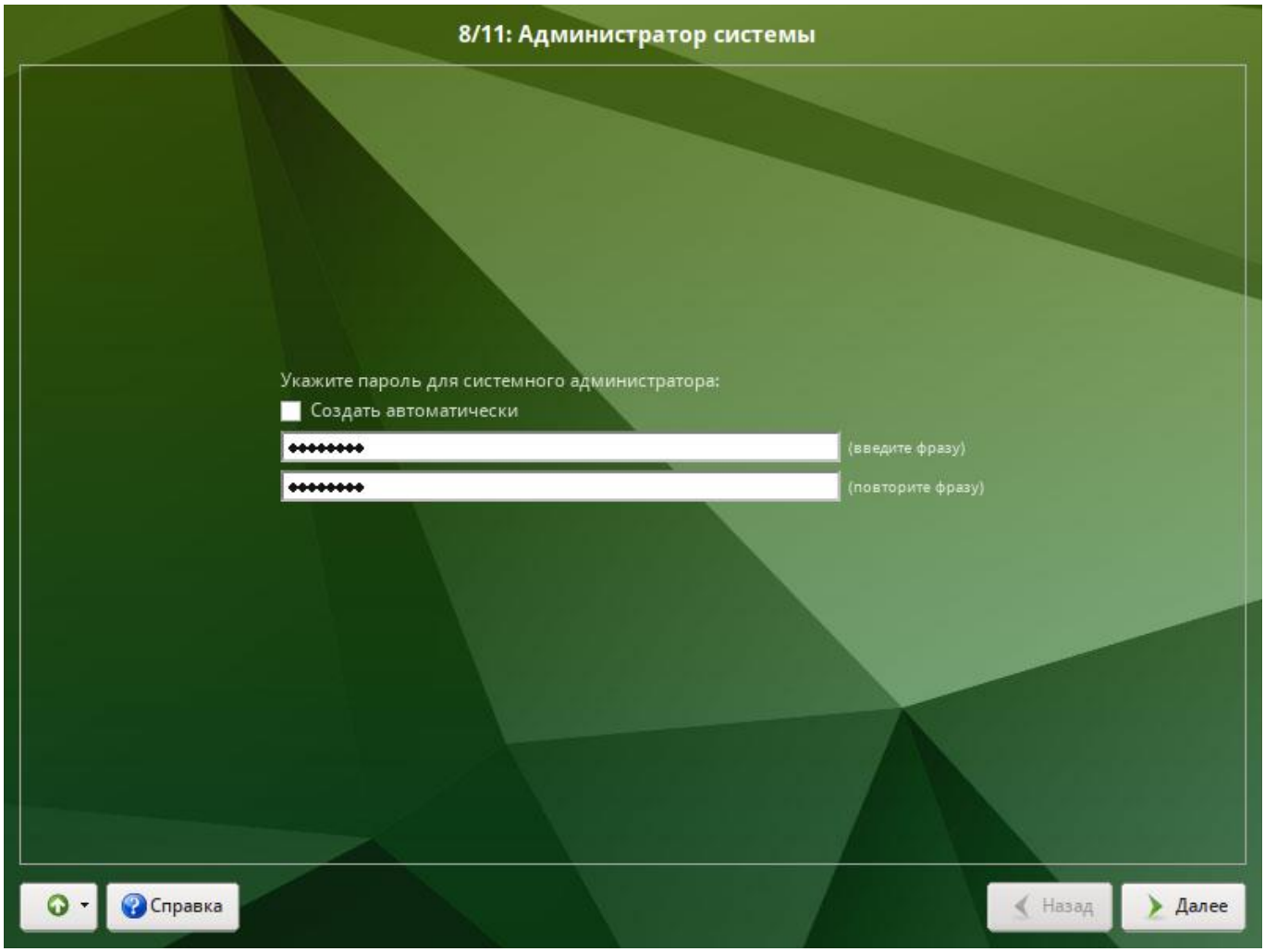

Рис. 18 – Установка. Задание пароля администратора

Для автоматической генерации пароля необходимо отметить пункт «Создать автоматически». Система предложит пароль, сгенерированный автоматическим образом в соответствии с требованиями по стойкости паролей.

Подтверждение введенного (или сгенерированного) пароля учетной записи администратора (root) и продолжение работы программы установки выполняется нажатием кнопки «Далее».

5.2.10. Системный пользователь

На данном этапе программа установки создает учетную запись системного пользователя (пользователя) ОС Альт 8 СП (рис. 19).

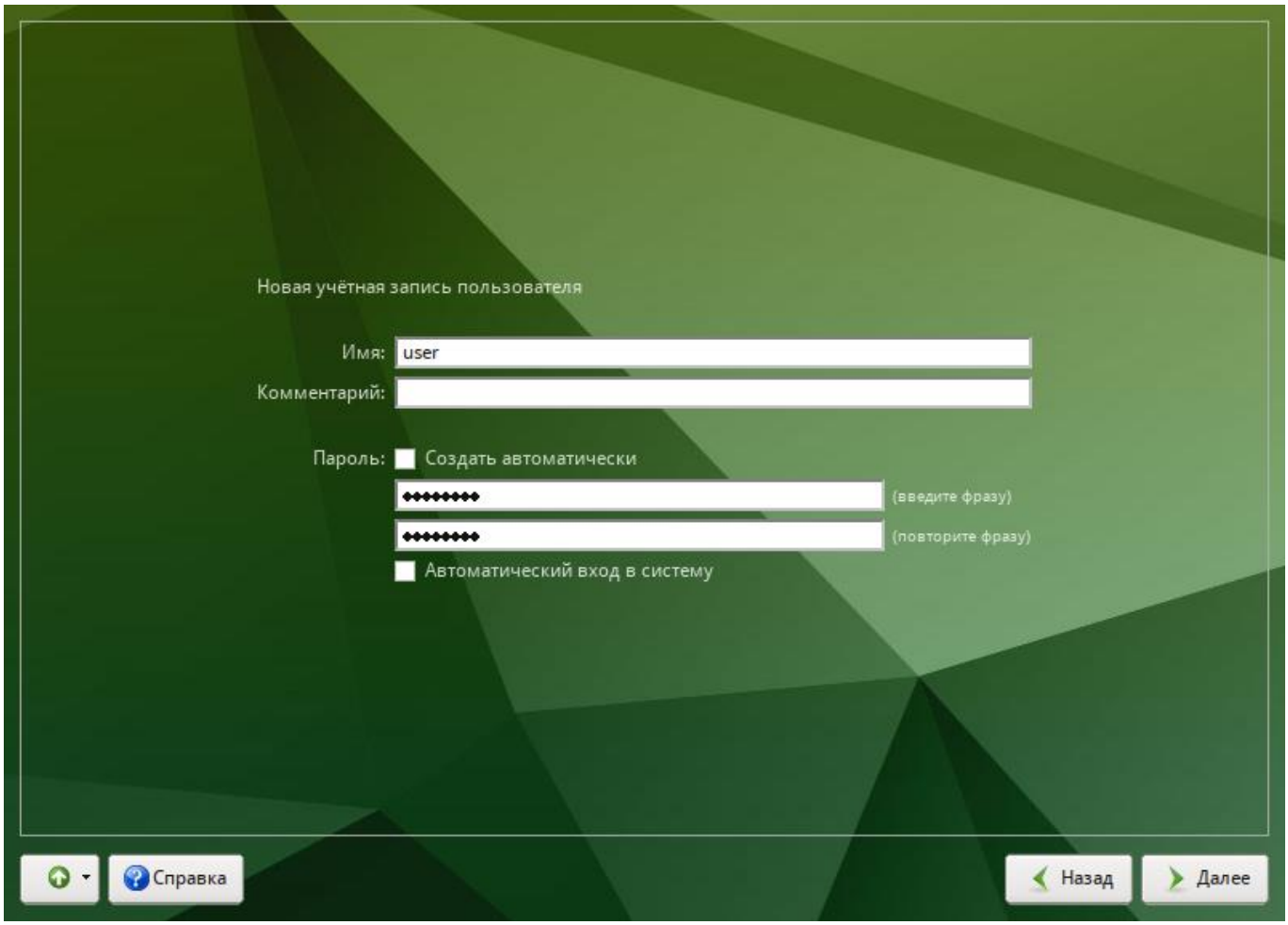

Рис. 19 – Установка. Создание пользователя

В окне «Системный пользователь» необходимо заполнить следующие поля:

- «Имя:» – имя учетной записи пользователя ОС Альт 8 СП (слово, состоящее только из строчных латинских букв, цифр и символа подчеркивания «_», причем цифра и символ «_» не могут стоять в начале слова);
- «Комментарий:» – любой комментарий к имени учетной записи;
- «Пароль:» – пароль учетной записи пользователя (чтобы исключить опечатки при вводе пароля, пароль пользователя вводится дважды).

Для автоматического создания пароля необходимо отметить пункт «Создать автоматически». Система предложит пароль, сгенерированный автоматическим образом в соответствии с требованиями по стойкости паролей.

В процессе установки предлагается создать только одну учетную запись пользователя – чтобы от его имени администратор мог выполнять задачи, которые не требуют привилегий администратора (root). Учетные записи для всех прочих пользователей системы можно будет создать в любой момент после ее установки.

5.2.11. Установка пароля на LUKS-разделы

Если на этапе подготовки диска был создан LUKS-раздел, на данном этапе необходимо ввести пароль для обращения к этому разделу (рис. 20).

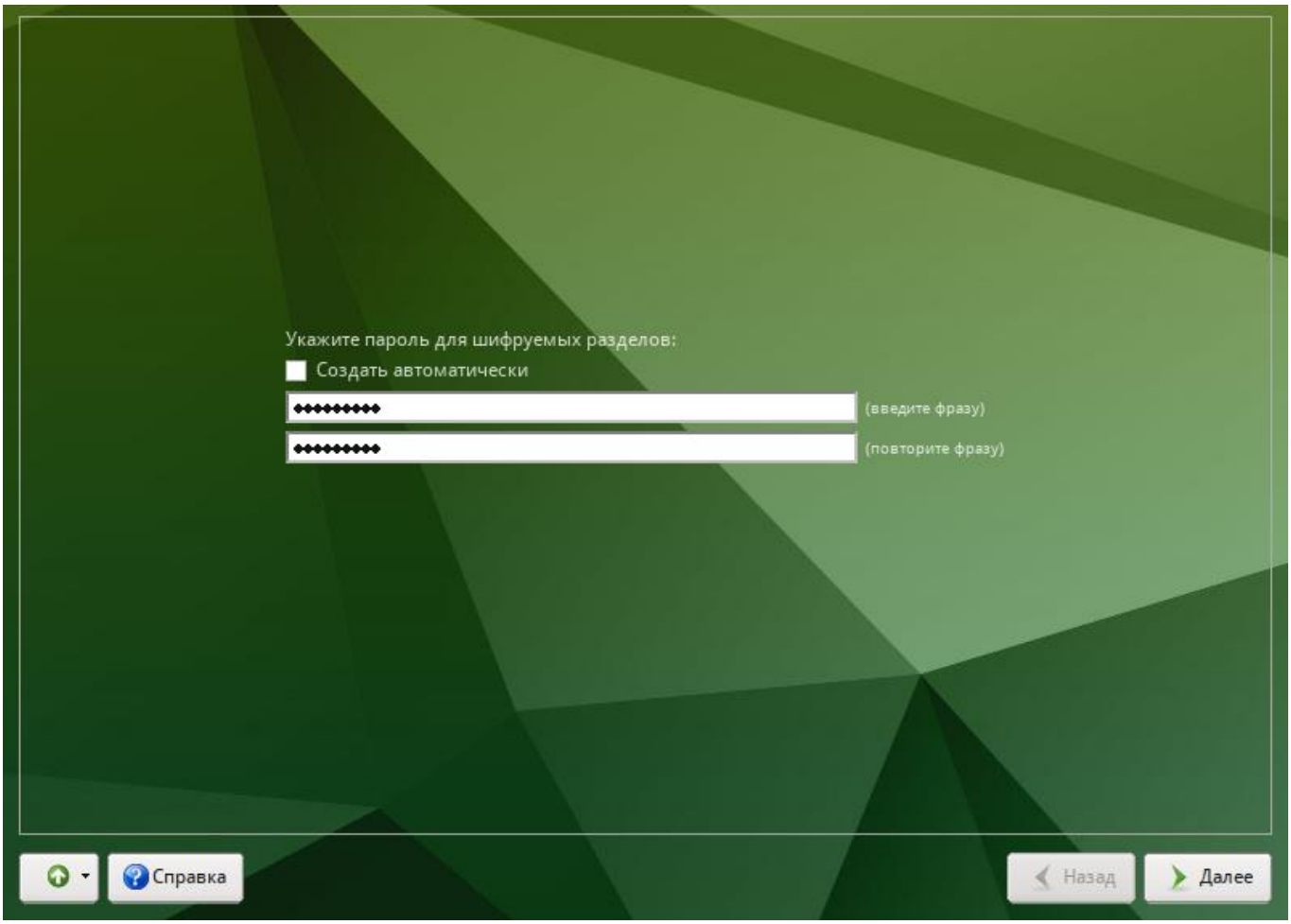

Рис. 20 – Установка. Установка пароля на LUKS-разделы

Установленный пароль потребуется вводить для получения доступа к информации на данных разделах.

П р и м е ч а н и е . Если кодируемые разделы, не создавались, этот шаг пропускается автоматически.

LUKS надо устанавливать при разметке вручную, удаляя и пересоздавая каждый раздел. LUKS будет требовать пароля при загрузке для каждого раздела.

5.2.12. Завершение установки

На экране последнего этапа установки отображается информация о завершении установки ОС Альт 8 СП (рис. 21).

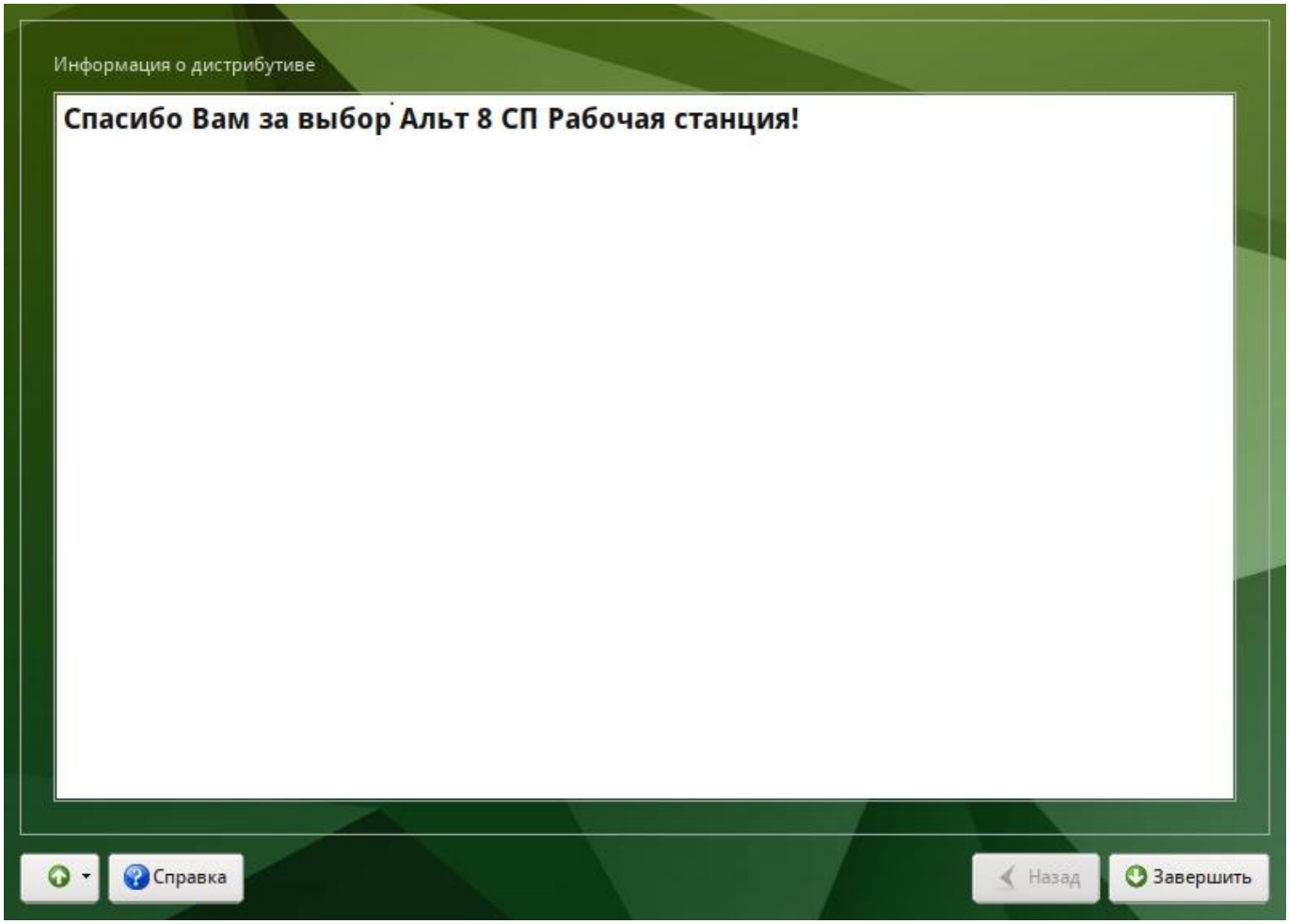

Рис. 21 – Установка. Завершение установки

После нажатия кнопки «Завершить» и перезагрузки компьютера выполняется штатная загрузка установленной ОС.

6. ПРОВЕРКА ОС АЛЬТ 8 СП

6.1. Запуск ОС

После включения вычислительного комплекса «Эльбрус» происходит инициализация программы начального старта.

Загрузка операционной системы начинается автоматически после небольшого времени ожидания (обычно несколько секунд).

```
Autoboot in xx  sec, PRESS SPACE TO DISABLE IT
```

Загрузка операционной системы может занять некоторое время, в зависимости от производительности компьютера. Основные этапы загрузки операционной системы – загрузка ядра, подключение (монтирование) файловых систем, запуск системных служб – периодически могут дополняться проверкой файловых систем на наличие ошибок. В этом случае время ожидания может занять больше времени, чем обычно.

Основной задачей программы начальной загрузки является загрузка ОС. Загрузку можно произвести по одной из четырех схем:

1) дождаться конца таймера обратного отсчета. В этом случае будет произведена загрузка заранее выбранной программы, с параметрами, хранящимися в энергонезависимой памяти либо в файле `/boot/boot.conf` (при его наличии) (метка, указанная как `default;`). Приоритетом обладает загрузка по параметрам, указанным в файле `/boot/boot.conf`. В этом случае из энергонезависимой памяти берется только значение номера устройства загрузки;
2) прервать таймер обратного отсчета и нажать клавишу `s`. В этом случае загрузка произойдет по параметрам, взятым из энергонезависимой памяти. Содержимое файла `/boot/boot.conf` приниматься в расчет не будет;
3) прервать таймер обратного отсчета и, нажав клавишу `c`, изменить параметры, взятые из энергонезависимой памяти. Потом, нажав клавишу `s`, загрузить программу;

4) прервать таймер обратного отсчета и, войдя в диалог загрузки с использованием конфигурационного файла `/boot/boot.conf` (`b` основного режима или `#boot` расширенного), загрузить одну из меток файла `/boot/boot.conf`.

П р и м е ч а н и е . Подробнее о работе с программой начального старта можно узнать из в штатной документации вычислительного комплекса «Эльбрус».

6.2. Идентификация и аутентификация в консольном режиме

При загрузке в консольном режиме работа загрузчика завершается запросом на ввод логина и пароля учетной записи. В случае необходимости на другую консоль можно перейти, нажав клавиш <Ctrl>+<Alt>+<F2>.

Для продолжения работы в консольном режиме необходимо ввести логин учетной записи пользователя и подтвердить его нажатием клавиши <Enter>. Затем ввести пароль и подтвердить его аналогичным образом.

В случае успешного прохождения процедуры аутентификации и входа в систему ОС перейдет к штатному режиму работы и предоставит дальнейший доступ к консоли (рис. 22).

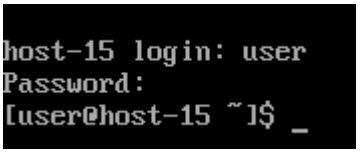

Рис. 22 – Аутентификация пользователя

6.3. Идентификация и аутентификация в графической оболочке MATE

В случае если графическая оболочка MATE была включена в состав ОС при установке, однако не стартовала автоматически, ее допускается вызвать вручную из консоли с помощью следующих команд:

```
~/.xinitrc
exec mate-session
```

Далее необходимо использовать команду `startx` для запуска MATE.

После загрузки ОС откроется окно входа в ОС Альт 8 СП по логину и паролю учетной записи (рис. 23). Необходимо ввести логин и пароль учетной записи, и нажать кнопку «Войти».

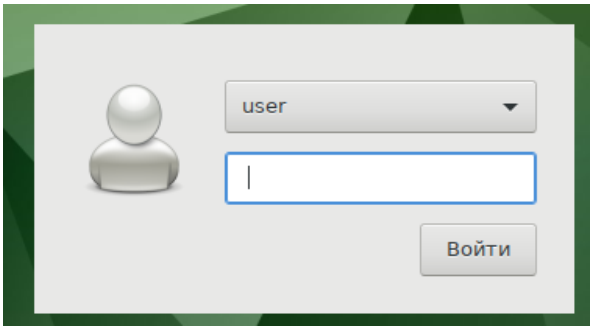

Рис. 23 – Ввод аутентификационных данных

Для продолжения работы и входа в ОС Альт 8 СП в графическом режиме необходимо выбрать одну из учетных записей, предлагаемых в окне аутентификации. Далее ввести пароль текущей учетной записи и нажать кнопку «Войти».

Для выбора учетной записи, не показанной в списке выбора, нужно раскрыть выпадающий список со значением логина текущей учетной записи и выбрать пункт «Другие…» (рис. 24).

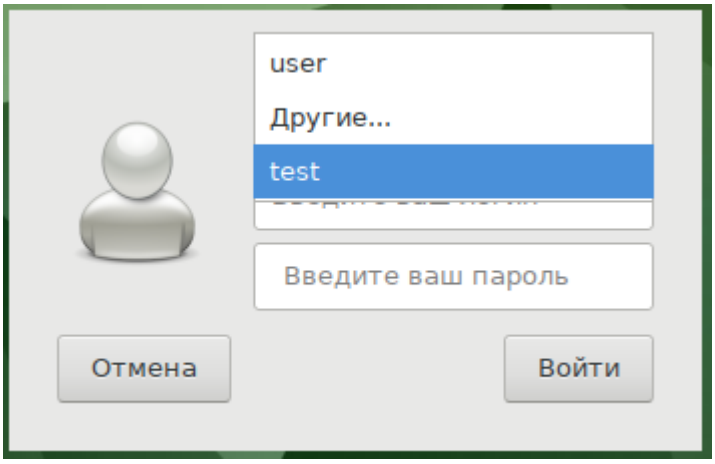

Рис. 24 – Выбор пользователя

После этого откроется окно ввода логина учетной записи (рис. 25), в котором нужно ввести логин и пароль учетной записи, и нажать кнопку «Войти».

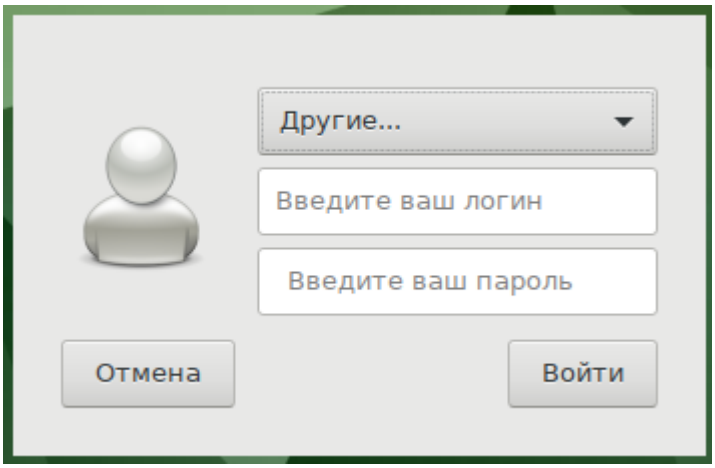

Рис. 25 – Аутентификация пользователя

В результате успешного прохождения процедуры аутентификации и входа в систему на экране появится графический интерфейс ОС Альт 8 СП (рис. 26).

Примечание. Работа в системе с использованием учетной записи администратора небезопасна, вследствие этого вход в систему в графическом режиме для администратора (root) запрещен. Попытка зарегистрироваться в системе будет прервана сообщением об ошибке.

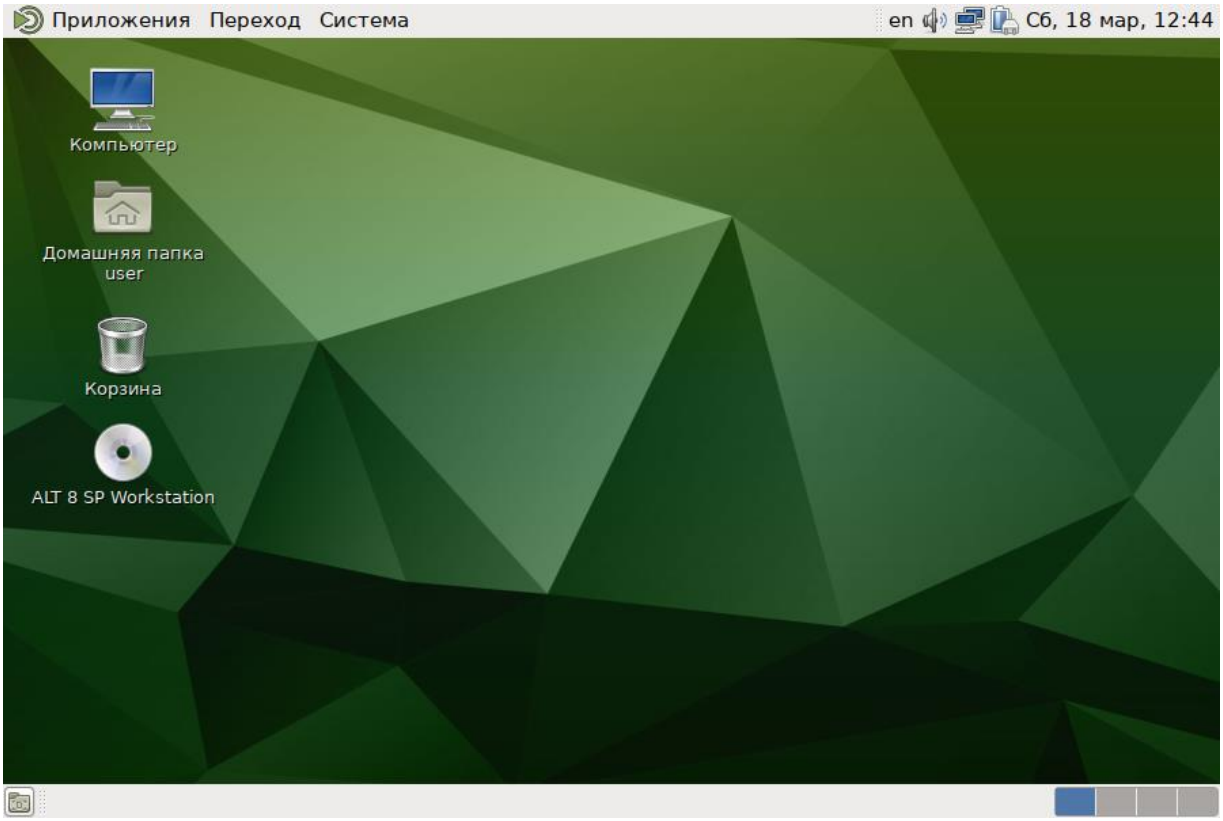

Рис. 26 – Графический интерфейс

6.4. Получение доступа к кодированным разделам

В случае если был создан кодированный раздел, потребуется вводить пароль при обращении к этому разделу (рис. 27).

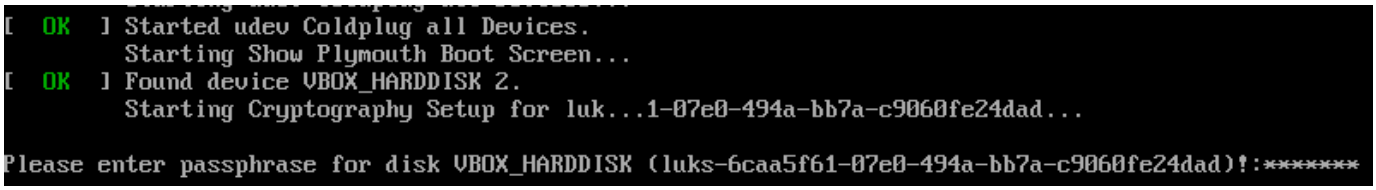

Рис. 27 – Запрос пароля для доступа к кодированным разделам

Если не ввести пароль за отведенный промежуток времени, то загрузка системы завершится ошибкой. В этом случае следует перезагрузить систему, нажав для этого два раза <Enter>, а затем клавиши <Ctrl>+<Alt>+<Delete>.

6.5. Определение параметров уничтожения данных

Для пользователей необходимо запретить использование команды `rm`. Для этого необходимо выполнить команду:

```
# chmod o-x /bin/rm
```

Команда `srm` предназначена для удаления данных без возможности их восстановления. `srm` выполняет безопасную перезапись/переименование/удаление целевого файла(ов). Использование команды `srm` аналогично использованию `rm`.

Команда `shred` переписывает несколько раз файл, скрывая его содержимое для того, чтобы сделать более трудоемким процесс восстановления данных даже в случае использования специального оборудования для восстановления:

```
shred [ОПЦИЯ] ФАЙЛ [...]
```

Стандартные опции для запуска команды:

1) `-f, --force` – изменить права для разрешения записи, если необходимо;
2) `-n, --iterations=N` – переписать N раз вместо указанных (25) по умолчанию;
3) `-s, --size=N` – очистить N байт (возможны суффиксы вида K, M, G);
4) `-u, --remove` – обрезать и удалить файл после перезаписи;
5) `-v, --verbose` – показывать индикатор прогресса

6) `-x, --exact` – не округлять размеры файлов до следующего целого блока;

7) `-z, --zero` – перезаписать в конце с нулями, чтобы скрыть перемешивание.

Если файл задан как -, перемешивать стандартный вывод.

Удаляет ФАЙЛЫ если указан `--remove (-u)`. По умолчанию файлы не удаляются, так как часто обрабатываются файлы-устройства вроде `/dev/hda`, а такие файлы нельзя удалять.

Команда `sfill` выполняет безопасную перезапись свободного пространства на разделе, в котором находится указанная директория и всех свободных индексных дескрипторов (inode) указанного каталога. Процесс удаления данных выглядит следующим образом:

- 1 проход с 0xff (все данные затираются значением 0xff);

- 5 случайных проходов с `/dev/urandom` используя RNG;

- 27 проходов со значениями Питера Гутмана;

- обрезает файл.

Стандартные опции для запуска команды:

1) `-d` – игнорировать специальные файлы "." и "..";

2) `-f` – быстрый (и небезопасный режим);

3) `-l` – выполнить только два прохода, с 0xff и случайное заполнение;

4) `-l -l` – выполнить только случайное заполнение (один проход);

5) `-r` – выполнить в рекурсивном режиме, удалить все подкаталоги;

6) `-v` – подробный режим;

7) `-z` – последний проход заполняет нулями, а не случайными данными.

Пользователю запрещено определять параметры уничтожения данных. Эти параметры определяет администратор.

Для определения параметров уничтожения данных в системе созданы скрипты с предопределенными настройками уничтожения данных, для их переопределения администратор должен внести правки в файл `/etc/sysconfig/s_rm`.

Пользователи для удаления данных должны использовать команды `s_rm` и `s_fill`.

6.6. dm-linear с безопасным удалением

Операции удаления обычно ограничиваются пометкой блоков данных как «неиспользуемых» в файловой системе. Dm-secdel, так же как dm-linear помечает блоки как не используемые, но заменяет очищение, записью случайных данных в освобождаемые блоки. Таким образом, данные удаляются надежно.

В силу своего абстрактного характера dm-linear поддерживает множество файловых систем, которые поддерживают опцию discard (например, ext3, ext4, xfs, btrfs).

Примечания к эксплуатации: следует создать сопоставленное устройство с помощью инструмента `secdelsetup`. Необходимо убедиться, что файловая система смонтирована на это, а не основное устройство. Необходимо убедиться, что файловая система (ФС) установлена с опцией `-o discard`.

Проверить смонтирована ли ФС в данный момент с этой опцией можно посмотрев вывод `mount`:

```
/dev/sdd1 on / type ext4 (rw,discard,errors=remount-ro)
```

Не следует включать ведение журнала данных. Необходимо обратить внимание, что при удалении файлов командой `rm` удаление будет выполняться асинхронно, поэтому чтобы убедиться, что данные уже удалены следует использовать команду `sync` или опцию монтирования файловой системы `-o sync` до использования команды `rm`.

Если необходимо, чтобы имена файлов также были уничтожены – удалите каталог самостоятельно, тогда его блоки освобождаются и стираются. При использовании команды `fstrim` все свободные блоки файловой системы будут отброшены (`discarded`) и, таким образом, стерты (файловая система должна быть примонтирована с опцией `-o discard`.)

Применение:

1) отображать sda5 в secdel5 (файловая система на secdel5 должна быть смонтирована с параметром `-o discard`):

```
secdelsetup /dev/sda5 [/dev/mapper/secdel5]
```

2) показать текущие (существующие) карты:

```
secdeltab –all
```

или

```
secdeltab –list
```

3) сохранить текущие карты в файл `/etc/secdeltab`, который был автоматически активирован после перезагрузки (системной службой `secdeltab.service`):

```
secdelsetup --save
```

4) отсоединить все активные карты:

```
secdeltab --detach-all
```

6.7. Центр управления системой

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

Примечание. При создании пользователя через ЦУС (альтератор) необходимо снимать отметку с пункта «Входит в группу администраторов».

ЦУС состоит из нескольких независимых диалогов-модулей. Каждый модуль отвечает за настройку определенной функции или свойства системы. Модули настройки сгруппированы по задачам.

Список установленных модулей можно просмотреть, выполнив команду:

```
$ alterator-standalone
```

ЦУС можно использовать для разных целей, например, (в скобках указаны имена соответствующих модулей):

- установка даты и времени (datetime);

- управление системными службами (services);

- просмотр системных журналов (logs);
- управление выключением удаленного компьютера (ahttpd-power, доступно только в веб-интерфейсе);
- настройка ограничений на использование внешних носителей (port-access, доступно только в веб-интерфейсе);
- конфигурирование сетевых интерфейсов (net-eth);
- настройка межсетевого экрана (net-iptables);
- управление пользователями (root и users).

Чтобы исключить возможность несанкционированного доступа к ЦУС по окончанию работы, необходимо завершить сеанс, нажав кнопку «Выйти».

Все модули ЦУС содержат встроенную справку, поясняющую назначение конкретного модуля. Справка вызывается кнопкой «Справка».

6.7.1. Графический интерфейс

Графический интерфейс ЦУС можно запустить следующими способами:

- комбинацией клавиш <ALT>+<F2> открыть окно быстрого запуска приложений и ввести в поле название программы – acc;
- выбрать на панели инструментов меню «Система» → «Администрирование» → «Центр управления системой» (рис. 28);
- при помощи консоли (приложение «Терминал среды MATE»), в которой необходимо ввести команду `acc`;

зная имя модуля, запустить графический интерфейс для него, можно также выполнив команду:

```
$ alterator-standalone <имя-модуля>
```

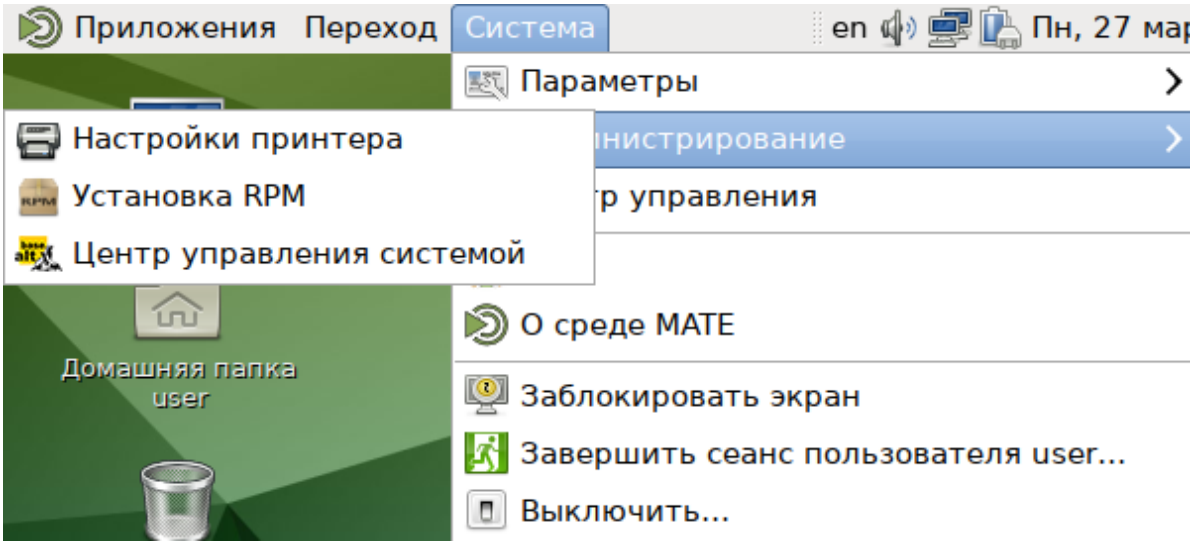

Рис. 28 – Запуск «Центра управления системой»

Запуск ЦУС (альтератора) требует прав администратора – ввести пароль root (рис. 29).

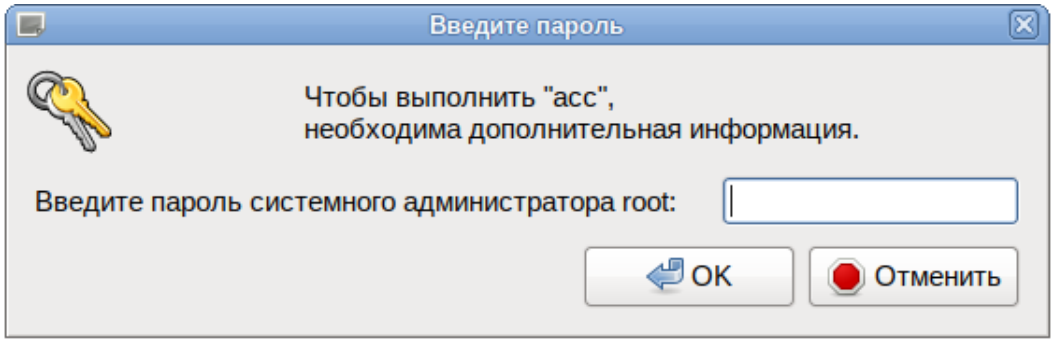

Рис. 29 – Запрос пароля для запуска «Центра управления системой»

После успешного входа откроется окно «Центра управления системой» (рис. 30).

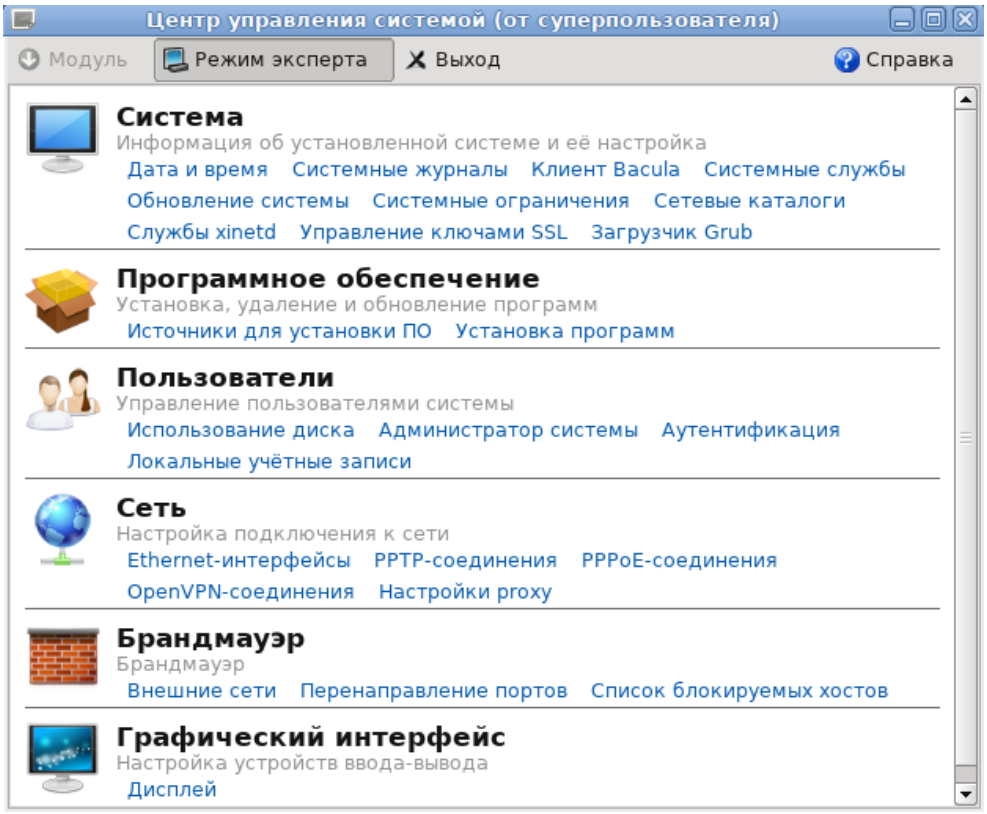

Рис. 30 – Окно «Центр управления системой»

Кнопка «Режим эксперта» позволяет выбрать один из режимов:

- основной режим (кнопка отжата);

- режим эксперта (кнопка нажата).

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

6.7.2. Веб-интерфейс ЦУС

ЦУС имеет веб-ориентированный интерфейс, позволяющий управлять сервером с любого компьютера сети.

Для запуска веб-ориентированного интерфейса, должен быть установлен пакет alterator-fbi:

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

Запущен сервис ahttpd:

```
# systemctl start ahttpd
```

Далее необходимо открыть в браузере адрес `https://localhost:8080/` или `https://ip-адрес:8080/`.

Для начала работы с ЦУС необходимо зарегистрироваться. Запуск ЦУС требует прав администратора, поэтому сначала необходимо предъявить пароль – ввести пароль root (рис. 31). Дополнительно на этапе регистрации можно выбрать язык интерфейса. По умолчанию предлагается язык, определенный настройками браузера.

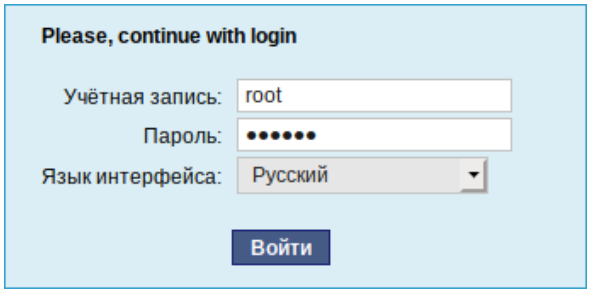

Рис. 31 – Запрос пароля администратора для запуска веб-интерфейса ЦУС

После успешного входа откроется окно «Центра управления системой» (рис. 32).

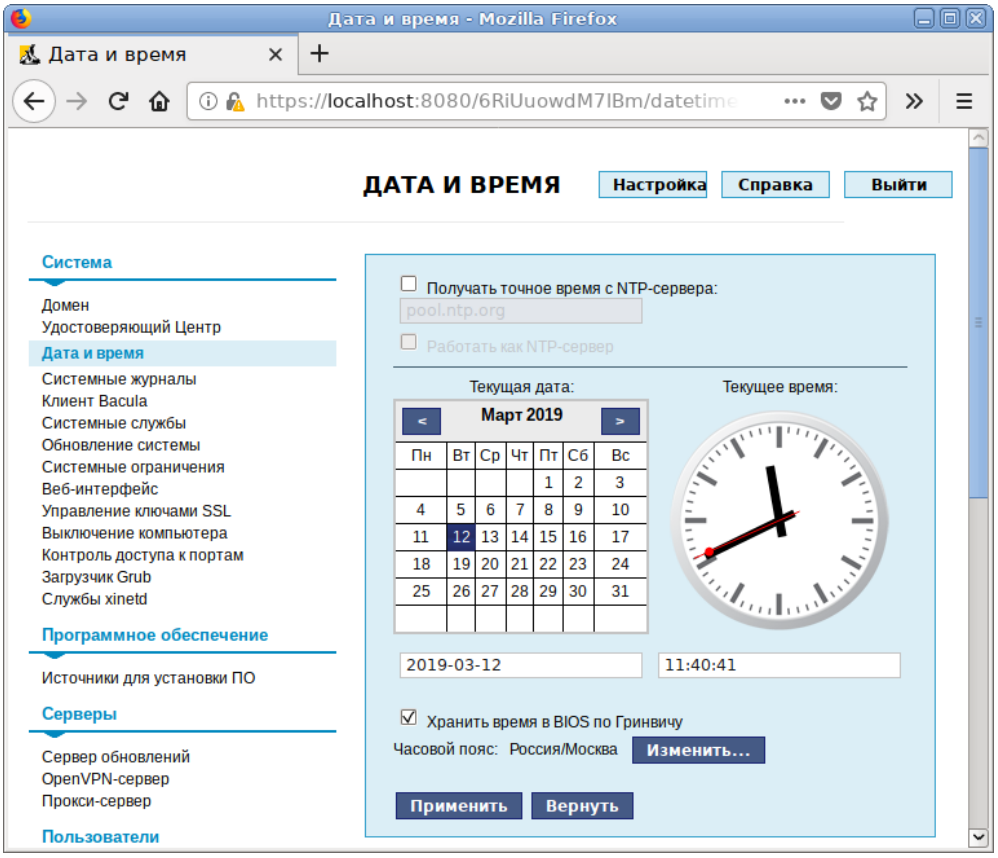

Рис. 32 – Окно веб-интерфейса «Центр управления системой»

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.

6.7.1. Установка и удаление модулей ЦУС

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

```
$ rpm –qa | grep alterator
```

Прочие пакеты для ЦУС можно найти, выполнив команду:

```
$ apt-cache search alterator*
# apt-get install
```

Модули можно дополнительно загружать и удалять как обычные программы:

```
# apt-get install alterator-net-openvpn
# apt-get remove alterator-net-openvpn
```

6.7.2. Права доступа к модулям ЦУС

Администратор 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 (рис. 37).

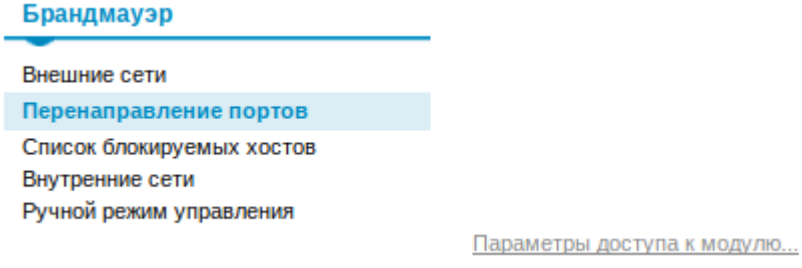

Рис. 33 – Ссылка «Параметры доступа к модулю»

В открывшемся окне, в списке «Новый пользователь» необходимо выбрать пользователя, который получит доступ к данному модулю и нажать кнопку «Добавить». Для сохранения настроек необходимо перезапустить HTTP-сервер, для этого достаточно нажать кнопку «Перезапустить HTTP-сервер» (рис. 34).

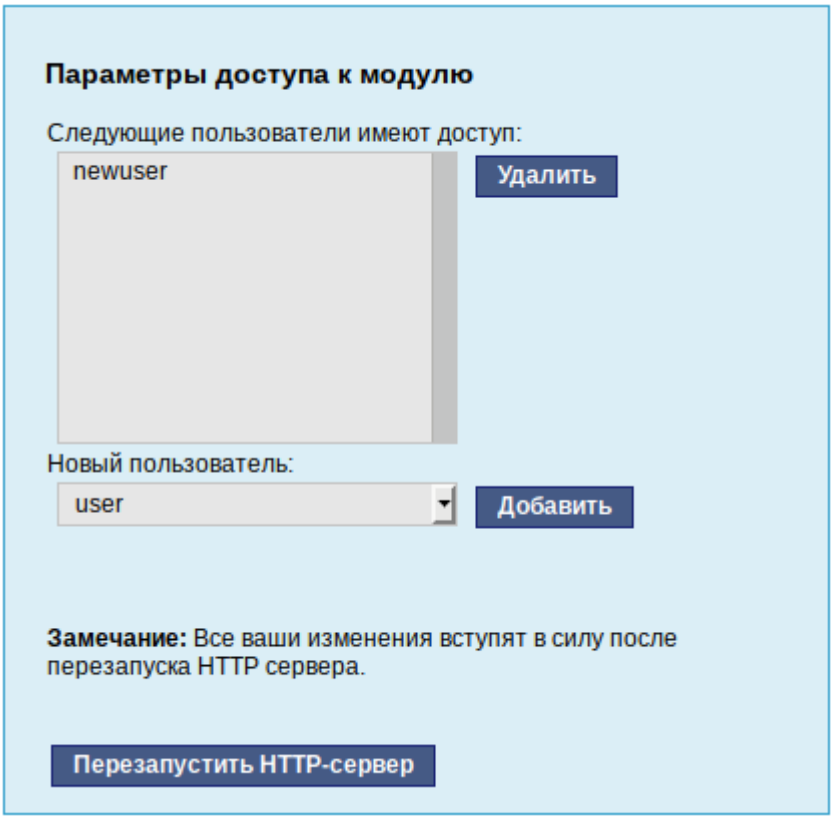

Рис. 34 – Параметры доступа к модулю

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, выбрать пользователя, нажать кнопку «Удалить» (рис. 34) и перезапустить HTTP-сервер.

Системный пользователь, пройдя процедуру аутентификации (рис. 35), может просматривать и вызывать модули, к которым он имеет доступ (рис. 36).

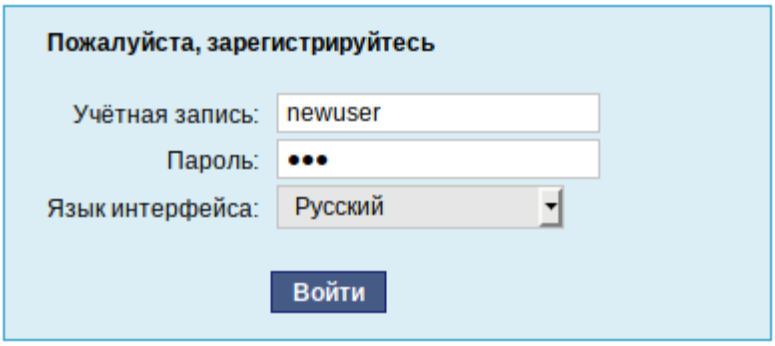

Рис. 35 – Запрос пароля для запуска веб-интерфейса ЦУС

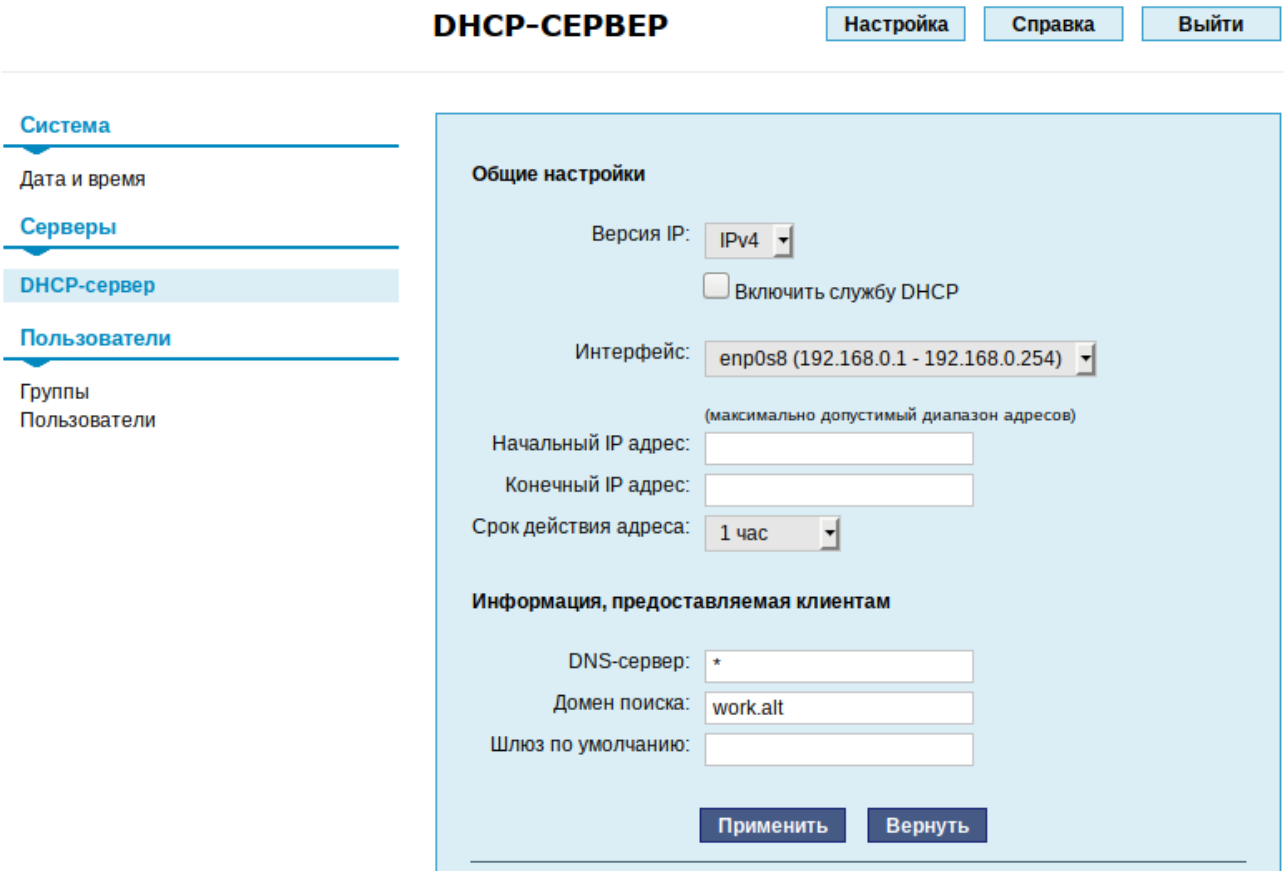

Рис. 36 – Веб-интерфейс ЦУС, запущенный от системного пользователя

6.7.3. Получение справочной информации

Все модули ЦУС содержат встроенную справку, поясняющую назначение конкретного модуля. Справка вызывается кнопкой «Справка» (рис. 37).

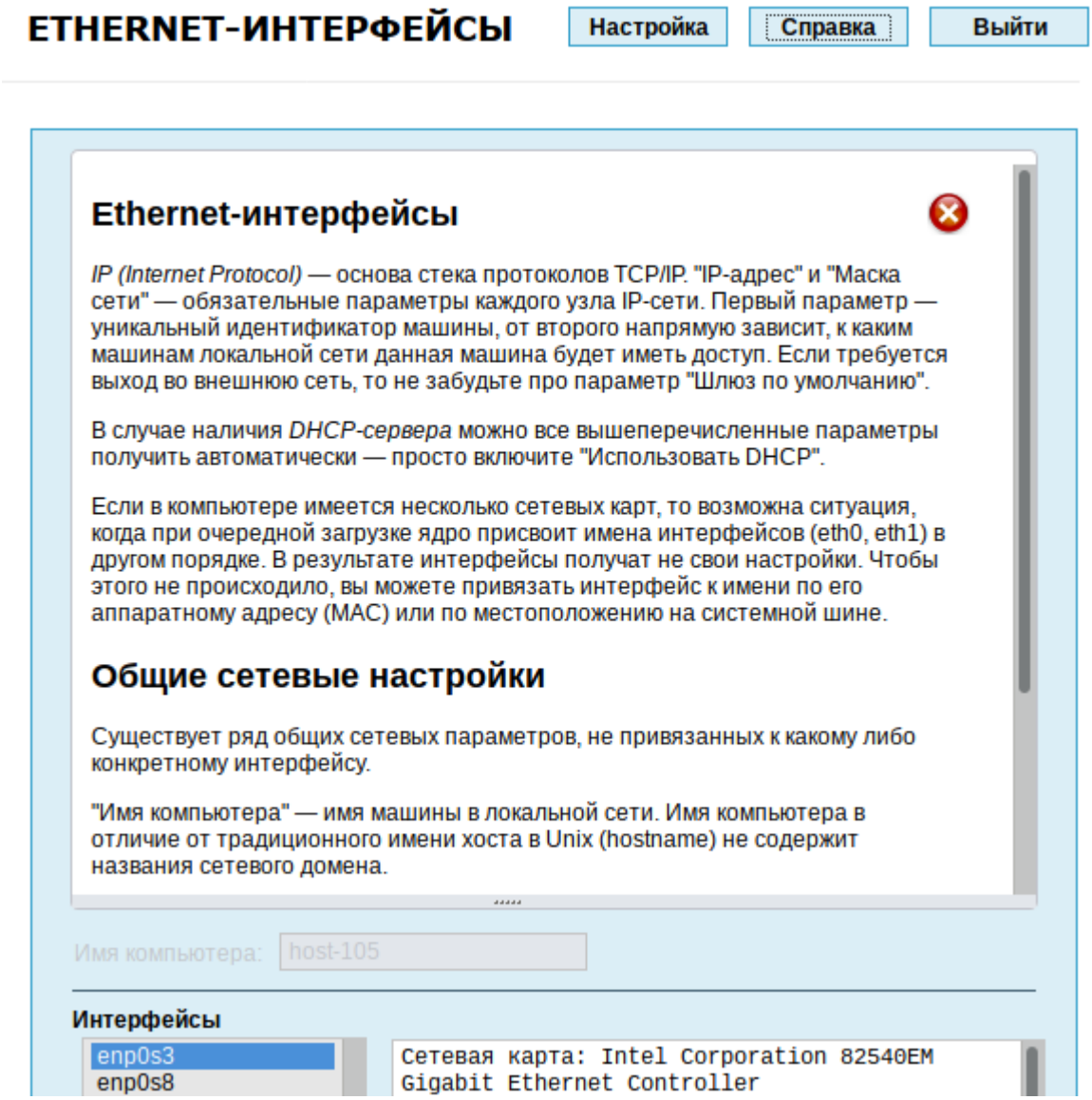

Рис. 37 – Получение справочной информации о модуле ЦУС в веб-интерфейсе

6.8. Завершение работы ОС

Для корректного завершения работы ОС (перезагрузки) во время ее работы запрещается выключать питание компьютера или перезагружать компьютер нажатием на кнопку «Reset», так как для корректного завершения работы требуется размонтирование файловой системы.

Перед окончанием работы с ОС необходимо завершить все работающие программы.

Для завершения работы ОС можно воспользоваться несколькими различными способами остановки системы:

- нажать комбинацию клавиш <Ctrl>+<Alt>+<Del>, что на рабочей станции приведет к вызову диалога завершения работы системы, а на сервере – к перезагрузке системы, при этом необходимо дождаться появления на экране сообщения «Reboot» (перезагрузка) и выключить питание системы;
- воспользоваться специальной командой `shutdown`, доступной пользователю с правами root (суперпользователь);
- при наличии графической оболочки следует воспользоваться диалогом доступным в меню «Система».

6.9. Настройки завершения сеанса пользователя

Для каждого пользователя можно настроить автоматическое завершение сеанса, после установленного времени бездействия (неактивности) пользователя. Для этого необходимо создать файл `/etc/logout`, в который поместить допустимое время простоя для каждого пользователя, например:

```
user1 300
user2 200
```

Формат файла `/etc/logout`:

```
<user> <время в секундах от момента последнего действия>
```

6.10. Блокирование сеанса доступа в информационную систему после установленного времени бездействия (неактивности) пользователя или по его запросу

После авторизации и загрузки графической рабочей среды MATE, пользователю предоставляется рабочий стол для работы с графическими приложениями.

При работе в графическом режиме блокирование сеанса доступа после установленного времени бездействия (по умолчанию 10 минут) происходит посредством срабатывания программы – хранителя экрана (screensaver).

Время бездействия системы устанавливается в меню: «Система» → «Параметры» → «Хранитель экрана». Также заблокировать сеанс доступа можно по запросу пользователя: «Система» → «Заблокировать экран» (<Ctrl>+<Alt>+<L>).

Программа vlock позволяет заблокировать сеанс при работе в консоли.

Для использования программы vlock, требуется ее предварительно установить:

```
# apt-get install vlock
```

6.11. Настройка блокировки возможности пользователя изменить настройки блокировки системы

Для блокировки возможности пользователя изменить настройки блокировки системы необходимо выполнить следующие действия:

1) создать файл `/etc/dconf/profile/user` со следующим содержимым:

```
user-db:user
system-db:local
```

2) создать каталоги `/etc/dconf/db/local.d/` и `/etc/dconf/db/local.d/locks`:

```
# mkdir /etc/dconf/db/local.d/
# mkdir /etc/dconf/db/local.d/locks
```

3) создать файл `/etc/dconf/db/local.d/screensaver`, в который поместить текст:

```
[org/mate/screensaver]
idle-activation-enabled=true
lock-enabled=true
```

4) в файле `/etc/dconf/db/local.d/session` установить время бездействия в минутах:

```
[org/mate/session]
idle-delay=2
```

5) запретить пользователям изменять заставку, для этого создать файл `/etc/dconf/db/local.d/locks/00-screensaver` со следующим содержимым:

```
#prevent users from changing screensaver
```

```
/org/mate/screensaver/idle-activation-enabled
/org/mate/screensaver/lock-enabled
/org/mate/desktop/session/idle-delay
```

6) выполнить обновление:

```
# dconf update
```

6.12. Идентификация и аутентификация средствами openvpn

6.12.1. Общие сведения

OpenVPN – свободная реализация технологии Виртуальной Частной Сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами. Она позволяет устанавливать соединения между компьютерами, находящимися за NAT-firewall, без необходимости изменения их настроек.

Для обеспечения безопасности управляющего канала и потока данных OpenVPN использует библиотеку OpenSSL. Это позволяет задействовать весь набор алгоритмов шифрования, доступных в данной библиотеке. Также может использоваться пакетная авторизация HMAC, для обеспечения большей безопасности, и аппаратное ускорение для улучшения производительности шифрования. Эта библиотека использует OpenSSL, а точнее протоколы SSLv3/TLSv1.2.

Аутентификация в OpenVPN возможна несколькими способами:

- статическим ключом, распространяемым на каждого клиента;

парой логин/пароль (как через самописный скрипт, так и с помощью плагинов: PAM, RADIUS и других);

- с использованием SSL-сертификатов;

- двухфакторная аутентификация (с использованием смарт-карт).

Размещение файлов OpenVPN:

- `/var/lib/openvpn/` – корневой каталог после инициализации демона (chroot);

- /var/lib/openvpn/etc/openvpn/ccd – каталог, в котором размещаются файлы особых параметров для подключаемых клиентов (Client Config Directory);
- /var/lib/openvpn/cache – рабочий каталог, является текущим для работы демона после инициализации (в него демон записывает файлы, у которых не указан путь, обычно это ipp и status);
- /etc/openvpn/ – каталог с файлами настройки;
- /etc/openvpn/ccd – символическая ссылка на /var/lib/openvpn/etc/openvpn/ccd (файлы доступны и до, и после chroot). Требуется для отладки, когда openvpn запускается без chroot;
- /etc/openvpn/keys/ – каталог для хранения ключей (информации ограниченного доступа).

6.12.2. Конфигурирование

Каждый файл конфигурации по маске /etc/openvpn/*.conf является конфигурацией отдельного экземпляра демона openvpn. Для имени экземпляра берется имя файла без суффикса .conf.

Настройки стартового скрипта располагаются в файле /etc/sysconfig/openvpn, по умолчанию он устанавливает следующие переменные окружения:

```
CHROOT=yes
OPENVPNUSER=openvpn
OPENVPNGROUP=openvpn
MANUAL=""
```

Стартовый скрипт /etc/init.d/openvpn может запускать и останавливать как все экземпляры демона, так и каждый по отдельности. Значение переменной MANUAL в /etc/sysconfig/openvpn указывает экземпляры, которые нужно запустить при старте системы (и при запуске стартового скрипта без параметра).

Для ручного запуска (остановки, проверки) одного экземпляра в конце командной строки указываем имя экземпляра. Например, для экземпляра openvpn с конфигурационным файлом /etc/openvpn/server.conf:

```
# service openvpn start server
Adjusting environment for openvpn:
[ DONE ]
Starting openvpn service:
[ DONE ]
# service openvpn status client-one
openvpn is stopped
```

При запуске сервиса, демон openvpn запускается, читает файл конфигурации из /etc/openvpn/, читает оттуда же файлы dh, ca и ключи. Этот каталог доступен демону только при его запуске.

Далее демон выполняет chroot в /var/lib/openvpn/ и cd в /var/lib/openvpn/cache, понижает привилегии до пользователя openvpn, затем инициализирует работу с сетью.

Таким образом, файл конфигурации должен быть размещен в /etc/openvpn, все ключи – в /etc/openvpn/keys, файлы настроек клиентов – в /etc/openvpn/ccd/ или /var/lib/openvpn/etc/openvpn/ccd/.

Важно правильно указать права доступа. Ключи должны быть доступны только администратору, конфигурации клиентов должны быть доступны на чтение пользователю openvpn:

```
# chown root:root /etc/openvpn/keys/* ; chmod 600 /etc/openvpn/keys/*
# chown root:openvpn /var/lib/openvpn/etc/openvpn/ccd/* ; chmod 640 /var/lib/openvpn/etc/openvpn/ccd/*
```

В файле конфигурации должны быть указаны:

- ifconfig-pool-persist и status – без полного пути либо с путем /cache/;
- ca, dh, cert, key – с путем /etc/openvpn/keys/;
- client-config-dir /etc/openvpn/ccd.

Далее приводится пример конфигурации в файле server.conf:

```
$ cat /etc/openvpn/server.conf
```

```
port 1194
proto udp
dev tun
ca /etc/openvpn/keys/admin.ca
dh /etc/openvpn/keys/dh4096.pem
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
comp-lzo
server 192.168.254.0 255.255.255.0
tls-server
cipher AES-256-CBC
verb 3
mute 10
keepalive 10 60
user nobody
group nogroup
persist-key
persist-tun
status server_status.log
ifconfig-pool-persist server_ipp.txt
verb 3
management localhost 1194
push "route 192.168.1.0 255.255.255.0"
client-config-dir /etc/openvpn/ccd
route 192.168.2.0 255.255.255.0
route 192.168.3.0 255.255.255.0
```

6.12.3. Создание ключей

6.12.3.1. Создание ключей для OpenVPN туннеля средствами утилиты openssl

Для создания туннеля средствами утилиты openssl необходимо выполнить следующие действия:

1) проверить наличие в системе установленного пакета openssl с помощью следующей команды:

```
# rpm -qa openssl
```

2) для возможности подписывать любые сертификаты, необходимо открыть файл `/var/lib/ssl/openssl.cnf` и изменить значение параметра `policy` на следующее:

```
policy = policy_anything
```

3) создать папку:

```
# mkdir -p /root/CA/demoCA
```

4) перейти в каталог:

```
# cd /root/CA
```

5) создать в каталоге `/root/CA` следующие папки и файлы:

```
# mkdir -p ./demoCA/newcerts
# touch ./demoCA/index.txt
# echo '01' > ./demoCA/serial
# echo '01' > ./demoCA/crlnumber
```

где:

- `demoCA/newcerts` – каталог сертификатов;
- `demoCA/index.txt` – текстовый файл, база с действующими и отозванными сертификатами;
- `demoCA/serial` – файл индекса для базы ключей и сертификатов;
- `demoCA/crlnumber` – файл индекса для базы отозванных сертификатов;

6) создать «самоподписанный» сертификат `my-ca.crt` и закрытый ключ `my-ca.pem`, которыми будут заверяться/подписываться ключи и сертификаты клиентов, желающих подключиться к серверу, с помощью следующей команды:

```
# openssl req -new -x509 -keyout my-ca.pem -out my-ca.crt
```

где:

- `-req` – запрос на создание сертификата;
- `-x509` – создать самоподписанный сертификат стандарта X.509;
- `-keyout` – записать закрытый ключ в файл;
- `-out` – записать сертификат в файл;

7) ввести пароль для закрытого ключа и ответить на запросы о владельце ключа;
8) создать пару «ключ-сертификат» для сервера и каждого клиента, желающего подключиться к серверу, с помощью следующей команды:

```
# openssl req -new -nodes -keyout server.pem -out server.crs
```

где `-nodes` – означает, что шифровать закрытый ключ не нужно;

9) подписать запрос на сертификат своим «самоподписанным» `my-ca.crt` сертификатом и ключом `my-ca.pem` с помощью следующей команды:

```
# openssl ca -cert my-ca.crt -keyfile my-ca.pem -days 3650 -in server.crs -out server.crt
```

где:

- `-cert`, корневой сертификат удостоверяющего центра;
- `-keyfile`, секретный ключ удостоверяющего центра;

10) после получения связки «ключ-сертификат» для сервера server сгенерировать запрос на сертификат для пользователя:

```
# openssl req -new -nodes -keyout user_1.pem -out user_1.crs
```

11) подписать запрос на сертификат своим `my-ca.crt` сертификатом и ключом `my-ca.pem`:

```
# openssl ca -cert my-ca.crt -keyfile my-ca.pem -days 365 -in user_1.crs -out user_1.crt
```

12) задать параметры Диффи-Хеллмана для сервера:

```
# openssl gendh -out server.dh 2048
```

13) удалить файлы запросов на сертификаты:

```
# rm *.crs
```

14) проверить состав каталога /root/CA (состав файлов должен соответствовать приведенному ниже):

```
# ls -l
итого 40
drwxr-xr-x 3 root root 4096 авг 26 15:07 demoCA
-rw-r--r-- 1 root root 1123 авг 26 14:47 my-ca.crt
-rw-r--r-- 1 root root 1834 авг 26 14:47 my-ca.pem
-rw-r--r-- 1 root root 4202 авг 26 14:58 server.crt
-rw-r--r-- 1 root root 424 авг 26 15:14 server.dh
-rw-r--r-- 1 root root 1708 авг 26 14:52 server.pem
-rw-r--r-- 1 root root 4190 авг 26 15:07 user_1.crt
-rw-r--r-- 1 root root 1708 авг 26 15:05 user_1.pem
```

15) разместить ключи и сертификаты в каталогах сервера и клиента следующим образом:

- my-ca.crt – для сервера и клиентов;
- my-ca.pem – только для подписи сертификатов (лучше хранить на отдельном от OpenVPN сервера компьютере);
- my-ca.crt, server.crt, server.dh, server.pem – для сервера OpenVPN;
- my-ca.crt, user_1.crt, user_1.pem – для клиента OpenVPN;

16) для новых клиентов создать новые ключи и отдать комплектом my-ca.crt, новый_сертификат.crt, новый_ключ.pem;

17) в конфигурационном файле OpenVPN сервера поместить ссылки на эти ключи:

```
ca /root/CA/my-ca.crt
cert /root/CA/server.crt
key /root/CA/server.pem
dh /root/CA/server.dh
```

18) в конфигурационном файле OpenVPN клиента поместить ссылки на эти ключи:

```
ca /etc/net/ifaces/tun0/my-ca.crt
cert /var/lib/ssl/certs/user_1.crt
```

```
key /var/lib/ssl/private/user_1.pem
```

19) просмотреть базу ключей:

```
# cat /root/CA/demoCA/index.txt
V 250823115811Z 01 unknown /C=RU/CN=vpn-server
V 160825120737Z 02 unknown /C=RU/CN=user_1
```

где v – действующий (валидный) ключ.

6.12.3.2. Создание списка отзыва сертификатов

Для создания списка отзыва сертификатов необходимо выполнить следующие действия:

1) выполнить следующую команду:

```
# openssl ca -cert my-ca.crt -keyfile my-ca.pem -gencrl -out crl.pem
```

2) просмотреть содержимое файла `crl.pem` с помощью следующей команды:

```
# openssl crl -noout -text -in crl.pem
```

3) отозвать сертификат `user_1.crt`:

```
# openssl ca -cert my-ca.crt -keyfile my-ca.pem -revoke user_1.crt -out crl.pem
```

4) обновить список (обязательно после каждого отзыва сертификата):

```
# openssl ca -cert my-ca.crt -keyfile my-ca.pem -gencrl -out crl.pem
```

5) просмотреть `crl.pem`:

```
# openssl crl -noout -text -in crl.pem
```

6) поместить файл `crl.pem` в каталог `/var/lib/openvpn`.

6.12.3.3. Создание ключей для OpenVPN туннеля средствами Easy-Rsa скриптов

Для работы с утилитой Easy-Rsa необходимо установить пакет easy-rsa с помощью следующей команды:

```
# apt-get install easy-rsa
```

Далее нужно выполнить поиск по ключевому слову `easyrsa*`, чтобы посмотреть, куда выполнялась установка утилиты:

```
# find / -name "easyrsa*"
/usr/bin/easyrsa
```

```
/usr/share/easyrsa3
```

В OpenSSL есть пример файла `openssl.cnf`, который находится в соответствующей папке. По умолчанию утилита openssl обращается к файлу `/var/lib/ssl/openssl.cnf`. В файле конфигурации есть несколько полезных параметров – например, местонахождение серийных номеров и списка отозванных сертификатов (Certificate Revocation List).

Однако некоторые записи из раздела `CA_default` ссылаются на директории и файлы, которые, в случае их отсутствия, могут привести к проблемам при развертывании центра сертификации. В связи с этим необходимо создать все требуемые файлы и папки перед тем, как подписывать CSR. В составе OpenSSL включена утилита CA.pl, которая упрощает процесс подготовки директорий и файлов.

В каталоге `/usr/share/easyrsa3` находятся следующие файлы:

```
openssl-1.0.cnf vars.example x509-types/
```

Файл `openssl-1.0.cnf`, является конфигуратором для утилиты openssl, запущенной через скрипты easy-rsa. Программа упрощает процесс создания инфраструктуры каталогов PKI.

Нужно перейти в каталог, в котором будет создаваться инфраструктура каталогов для ключей и сертификатов, с помощью следующей команды:

```
# cd /root
```

Затем необходимо создать структуру каталогов с помощью следующей команды:

```
# easyrsa init-pki
```

В текущей директории будет создан каталог `pki` с вложенными каталогами для ключей и запросов.

Дальнейшие действия также необходимо выполнять в текущей директории, иначе утилита будет выводить ошибки из-за отсутствия `pki` каталога в текущей директории при запуске easyrsa команды.

6.12.3.3.1. Создание ключей центра сертификации с помощью Easy-Rsa скриптов

Для создания ключей центра сертификации необходимо создать корневой сертификат. Для этого необходимо запустить easyrsa с помощью следующей команды:

```
# easyrsa build-ca
```

Далее будет выведен процесс генерации, в ходе которого нужно указать сложный пароль и Common Name сервера, например CA-ORG:

```
Generating a 2048 bit RSA private key
.....................................................+++
.....................................................+++
writing new private key to '/root/pki/private/ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
...
-----
Common Name (eg: your user, host, or server name) [Easy-RSA
CA]:CA-ORG
CA creation complete and you may now import and sign cert
requests.
Your new CA certificate file for publishing is at:
/root/pki/ca.crt
```

Затем нужно создать ключи Диффи-Хелмана:

```
# easyrsa gen-dh
```

Создание ключа занимает некоторое продолжительное время. Далее необходимо проверить содержание каталога `pki` с помощью следующей команды:

```
# ls -l ./pki
```

Содержание каталога должно соответствовать приведенному ниже:

```
итого 28
-rw------- 1 root root 1151 авг 27 09:32 ca.crt
drwx------ 2 root root 4096 авг 27 09:32 certs_by_serial
-rw------- 1 root root 424 авг 27 09:38 dh.pem
```

```
-rw------- 1 root root 0 авг 27 09:32 index.txt
drwx------ 2 root root 4096 авг 27 09:32 issued
drwx------ 2 root root 4096 авг 27 09:32 private
drwx------ 2 root root 4096 авг 27 09:28 reqs
-rw------- 1 root root 3 авг 27 09:32 serial
```

где:

- `ca.crt` – сертификат корневого центра сертификации;

- `dh.pem` – ключ Диффи-Хелмана;

- `./private/ca.key` – секретный ключ центра сертификации.

6.12.3.3.2. Создание ключей сервера с помощью Easy-Rsa скриптов

Создать запрос на сертификат для сервера OVPN. Сертификат будет не зашифрован (запаролен), за это отвечает параметр nopass, иначе при каждом старте OpenVPN будет запрашивать этот пароль:

```
easyrsa gen-req vpn-server nopass
```

Скопировать полученные ключи в рабочий каталог openvpn и в конфигурации сервера указать полный путь к ключам:

```
cp ./pki/ca.crt /etc/openvpn/keys
cp ./pki/issued/vpn-server.crt /etc/openvpn/keys
cp ./pki/private/vpn-server.key /etc/openvpn/keys
cp ./pki/dh.pem /etc/openvpn/keys
```

Для создания пары ключ/сертификат минуя создания запросов и подписи необходимо выполнить команду:

```
easyrsa build-server-full vpn-server nopass - без пароля
easyrsa build-server-full vpn-server - с паролем
```

6.12.3.3.3. Создание клиентских ключей с помощью Easy-Rsa скриптов

Процесс создания ключей клиентам аналогичен созданию ключей для сервера. Создание запроса запароленного ключа для клиента (потребуется вводить при каждом подключении) с именем User выполняется с помощью следующей команды:

```
easyrsa gen-req User
```

Создание запроса без парольного ключа для клиента выполняется с помощью следующей команды:

```
easyrsa gen-req User nopass
```

Создание ключа пользователя выполняется с помощью следующей команды:

```
easyrsa sign-req client User
```

Создание ключа пользователя с ограничением действия сертификата в 90 дней (после истечения срока можно только перевыпустить) выполняется с помощью следующей команды:

```
./easyrsa sign-req client User -days 90
```

Передача файлов клиенту выполняется с помощью следующей команды:

```
./pki/issued/User.crt
./pki/private/User.key
./pki/ca.crt
```

Для создания пары ключ/сертификат минуя создания запросов и подписи необходимо выполнить команду:

`easyrsa build-client-full User nopass` – без пароля;

`easyrsa build-client-full User` – с паролем.

6.12.4. Отзыв сертификатов

Генерация файла отозванных ключей выполняется с помощью следующей команды:

```
# easyrsa gen-crl
```

Сделать символическую ссылку в каталог с ключами:

```
# ln -s /root/pki/crl.pem /var/lib/openvpn
```

В файл конфигурации openvpn сервера добавить строку:

```
# crl-verify crl.pem
```

Отзыв сертификата пользователя User выполняется с помощью следующей команды:

```
# easyrsa revoke User
```

Каждый раз при отзыве сертификата необходимо обновлять `crl.pem`, чтобы внести в него изменения:

```
# easyrsa gen-crl
```

Одноименный файл ключа не может быть создан, пока не отозван старый. Для исключения возможности mitm атаки служит параметр `remote-cert-tls server`.

Список валидных и отозванных сертификатов можно посмотреть в файле `./pki/index.txt`. Начало строки описания каждого сертификата начинается с букв `V` или `R`, что значит `Valid` и `Revoked` (действующий и отозванный).

6.13. Виртуальная консоль

В процессе работы ОС Альт 8 СП активно несколько виртуальных консолей. Каждая виртуальная консоль доступна по одновременному нажатию клавиш <Ctrl>, <Alt> и функциональной клавиши с номером этой консоли от <F1> до <F6>.

На первых шести виртуальных консолях (от <Ctrl>+<Alt>+<F1> до <Ctrl>+<Alt>+<F6>) пользователь может зарегистрироваться и работать в текстовом режиме. Если была установлена графическая оболочка MATE, она будет загружаться в первой виртуальной консоли. Двенадцатая виртуальная консоль (<Ctrl>+<Alt>+<F12>) выполняет функцию системной консоли – на нее выводятся сообщения о происходящих в системе событиях.

7. ОПИСАНИЕ ОПЕРАЦИЙ АДМИНИСТРИРОВАНИЯ

Администрирование ОС Альт 8 СП если не указано иное выполняется из командной строки ОС Альт 8 СП.

7.1. Управление системными сервисами и командами

7.1.1. Сервисы

Сервисы – это программы, которые запускаются и останавливаются через инициализированные скрипты, расположенные в каталоге `/etc/init.d`. Многие из этих сервисов запускаются на этапе старта ОС Альт 8 СП.

Каталог `/sbin/service` обеспечивает интерфейс (взаимодействие) пользователя с инициализированными скриптами. В свою очередь, скрипты обеспечивают интерфейс для управления сервисами, предоставляя пользователю опции для запуска, остановки, перезапуска, запроса состояния сервиса и выполнения других воздействий на сервис.

Инициализированный скрипт сервиса openssh имеет следующие опции:

```
/etc/init.d/sshd
Usage: sshd
{start|stop|reload|restart|condstop|condrestart|condreload|check|
status}
```

Текущее состояние всех системных служб в ОС Альт 8 СП можно просмотреть с помощью команды `systemctl`:

```
systemctl
...
sshd.service
loaded active running   OpenSSH server daemon
  systemd-binfmt.service
loaded active exited    Set Up Additional Binary F
  systemd-fsck-root.service
loaded active exited    File System Check on Roo
...
```

Информация о запущенности и включенности сервисов может быть получена или изменена с помощью команды `systemctl`. Например, для службы удаленного доступа ssh установки по умолчанию выглядят следующим образом:

```
/sbin/systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: ena
   Active: active (running) since Mon 2019-04-01 09:48:34 MSK; 4h 0min ago
  Process: 921 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Process: 904 ExecStartPre=/usr/bin/ssh-keygen -A (code=exited, status=0/SUCCES
 Main PID: 942 (sshd)
   CGroup: /system.slice/sshd.service
           └─942 /usr/sbin/sshd -D
```

Сервис sshd запускается автоматически. Для того чтобы отключить его автоматический запуск сервиса, можно воспользоваться следующей опцией команды `systemctl`:

```
/sbin/systemctl disable sshd
```

Запуск, остановка, перезапуск и перезагрузка настроек служб выполняются соответственно командами:

```
/sbin/systemctl start <служба>
/sbin/systemctl stop <служба>
/sbin/systemctl restart <служба>
/sbin/systemctl reload <служба>
```

7.1.2. Команды

Далее приведены основные команды, использующиеся в ОС Альт 8 СП:

- `ar` – создание и работа с библиотечными архивами;
- `at` – формирование или удаление отложенного задания;
- `awk` – язык обработки строковых шаблонов;
- `batch` – планирование команд в очереди загрузки;

- bc – строковый калькулятор;
- chfn – управление информацией учетной записи (имя, описание);
- chsh – управление выбором командного интерпретатора (по умолчанию – для учетной записи);
- cut – разбивка файла на секции, задаваемые контекстными разделителями;
- df – вывод отчета об использовании дискового пространства;
- dmesg – вывод содержимого системного буфера сообщений;
- du – вычисление количества использованного пространства элементов ФС;
- echo – вывод содержимого аргументов на стандартный вывод;
- egrep – поиск в файлах содержимого согласно регулярным выражениям;
- fgrep – поиск в файлах содержимого согласно фиксированным шаблонам;
- file – определение типа файла;
- find – поиск файла по различным признакам в иерархии каталогов;
- gettext – получение строки интернационализации из каталогов перевода;
- grep – вывод строки, содержащей шаблон поиска;
- groupadd – создание новой учетной записи группы;
- groupdel – удаление учетной записи группы;
- groupmod – изменение учетной записи группы;
- groups – вывод списка групп;
- gunzip – распаковка файла;
- gzip – упаковка файла;
- hostname – вывод и задание имени хоста;
- install – копирование файла с установкой атрибутов;
- ipcrm – удаление ресурса IPC;
- ipcs – вывод характеристик ресурса IPC;
- kill – прекращение выполнения процесса;
- killall – удаление процессов по имени;
- lpr – система печати;
- ls – вывод содержимого каталога;

- `lsb_release` – вывод информации о дистрибутиве;
- `m4` – запуск макропроцессора;
- `md5sum` – генерация и проверка MD5-сообщения;
- `mknod` – создание файла специального типа;
- `mktemp` – генерация уникального имени файла;
- `more` – постраничный вывод содержимого файла;
- `mount` – монтирование ФС;
- `msgfmt` – создание объектного файла сообщений из файла сообщений;
- `newgrp` – смена идентификатора группы;
- `nice` – изменение приоритета процесса перед его запуском;
- `nohup` – работа процесса после выхода из системы;
- `od` – вывод содержимого файла в восьмеричном и других видах;
- `passwd` – смена пароля учетной записи;
- `patch` – применение файла описания изменений к оригинальному файлу;
- `pidof` – вывод идентификатора процесса по его имени;
- `ps` – вывод информации о процессах;
- `renice` – изменение уровня приоритета процесса;
- `rm` – удаление файлов или каталогов;
- `sed` – строковый редактор;
- `sendmail` – транспорт системы электронных сообщений;
- `sh` – командный интерпретатор;
- `shutdown` – команда останова системы;
- `srm` – безопасная перезапись/переименование/удаление целевого файла;
- `su` – изменение идентификатора запускаемого процесса;
- `sync` – сброс системных буферов на носители;
- `tar` – файловый архиватор;
- `umount` – размонтирование файловой системы;
- `useradd` – создание новой учетной записи или обновление существующей;
- `userdel` – удаление учетной записи и соответствующих файлов окружения;

- `usermod` – модификация информации об учетной записи;
- `w` – список пользователей, кто в настоящий момент работает в системе и с какими файлами;
- `who` – вывод списка пользователей системы.

Узнать об опциях команд можно с помощью команды `man`.

7.1.3. Средства архивирования файлов

Команды `tar`, `cpio`, `gzip` представляют собой инструменты создания резервных копий и архивирования ФС.

При создании архива командами `tar` и `gzip` передается список файлов и каталогов, указываемых как параметры командной строки. Любой указанный каталог просматривается рекурсивно.

При создании архива с помощью команды `cpio` ей предоставляется список объектов (имена файлов и каталогов, символические имена любых устройств, гнезда доменов UNIX, именованные каналы).

7.1.3.1. Команда `tar`

Команда `tar` предназначена для создания архивов.

Синтаксис:

```
tar [Опции] [АРГ]
```

Опции:

1) `-c` – создает архив;
2) `-x` – восстанавливает файлы из архива на устройстве, заданном по умолчанию или определенном опцией `f`;
3) `-f` name – создает (или читает) архив с `name`, где `name` – имя файла или устройства, определенного в `/dev`, например `/dev/rmt0`;
4) `-Z` – сжимает или распаковывает архив с помощью `compress`;
5) `-z` – сжимает или распаковывает архив с помощью `gzip`;
6) `-M` – создает многотомный архив;
7) `-t` – создает список сохраненных в архиве файлов и выводит его на консоль;

8) `-v` – выводит подробную информацию о процессе.

7.1.3.2. Команда `cpio`

Команда `cpio` предназначена для копирования файлов. Ее можно использовать с опцией `-o` для создания резервных архивов и с опцией `-i` – для восстановления файлов. Команда получает информацию от стандартного устройства ввода и посылает выводимую информацию на стандартное устройство вывода.

Команда `cpio` может архивировать любой набор файлов и специальные файлы, хранит информацию более эффективно, чем `tar`, пропускает сбойные сектора или блоки при восстановлении данных, и ее архивы могут быть восстановлены в ОС.

Недостатком команды `cpio` является то, что для обновления архива следует использовать язык программирования оболочки, чтобы создать соответствующий сценарий.

Синтаксис:

```
cpio [Опции] < список-имен [> архив]
```

Опции:

1) `-o` – создание архива в стандартное устройство вывода;
2) `-l` – восстановление файлов из архива, передаваемого на стандартное устройство ввода;
3) `-t` – создание списка содержимого стандартного устройства ввода;

Ниже приводятся примеры использования команды `cpio` для решения различных задач.

Копирование файлов из каталога `/home` в архив `home.cpio` выполняется следующим образом:

```
find /home/* | cpio -o > /tmp/home.cpio
```

Восстановление файлов из архива `home.cpio` с сохранением дерева каталогов и создание списка в файле `bkup.index` выполняется следующим образом:

```
cpio -id < /tmp/home.cpio > bkup.index
```

Использование команды `find` для поиска измененных за последние сутки файлов и сохранение их в архив `home.new.cpio` выполняется следующим образом:

```
find /home -mtime 1 -type f | cpio -o > /tmp/home.new.cpio
```

Восстановление файла `/home/dave/notes.txt` из архива `home.cpio` выполняется следующим образом:

```
cpio -id /home/dave/notes.txt < home.cpio
```

Для восстановления файла с помощью `cpio` следует указывать его полное имя.

Все эти команды могут выполняться автоматически путем их размещения в файле crontab пользователя с идентификатором root.

Пример записи, выполняющей резервное копирование каталога `/home` ежедневно в 01:30:

```
30 02 *** ls /home : cpio -o > /tmp/home.cpio
```

При необходимости выполнения резервного копирования более сложного уровня можно создать соответствующий сценарий оболочки. Запуск подобных сценариев также может быть осуществлен посредством cron.

Создание резервных копий означает определение политики создания резервных копий для снижения потерь и восстановления информации после возможной аварии системы.

7.1.4. Средства редактирования файлов

7.1.4.1. Текстовый редактор Vi

Текстовый редактор Vi – системный редактор, назначаемый ОС по умолчанию для работы с текстовыми файлами.

Текстовый редактор Vi имеет модальный интерфейс – одни и те же клавиши в разных режимах работы выполняют разные действия.

В редакторе Vi есть несколько режимов работы:

1) командный режим – перемещение по файлу, удаление текста и другие редактирующие функции. По умолчанию, работа начинается в командном режиме. Перейти в него из любого другого режима `<ESC>`, иногда два раза;

2) режим ввода – ввод текста (удаление и ввод текста происходит в двух разных режимах). Переход в режим ввода из командного режима осуществляется командой <i>;
3) режим строчного редактора ED – это специальный режим, в котором редактору даются сложные команды. При вводе этих команд они отображаются в последней строке экрана. Например, команда <wq> позволяет записать файл и покинуть редактор Vi, а команда <q!> – выйти из редактора Vi без сохранения изменений. В этом режиме обычно вводятся команды, название которых состоит из нескольких символов. Переход в него из командного режима осуществляется командой <:>.

Далее описаны операции, которые можно произвести с файлом в командном режиме.

7.1.4.1.1. Открыть (создать) файл

Управляющая команда открытия файла выглядит следующим образом:

```
vi <имя_файла>
```

Создание файла происходит при помощи той же команды, поскольку создание файла происходит в момент сохранения.

Для открытия или создания нового файла в командном режиме необходимо набрать:

```
:e filename
```

Перед этим нужно сохранить предыдущий файл с помощью следующих команд:

- <:w> – сохраняет файл с существующим именем;
- <:sav filename> – или «Сохранить как».

7.1.4.1.2. Навигация по файлу

Навигация по файлу происходит с помощью управляющих клавиш на клавиатуре. Также допускается использовать клавиши быстрого перемещения:

- <^> или <0> – в начало текущей строки;
- <$> – в конец текущей строки;
- <w> – на слово вправо;

- <b> – на слово влево.

7.1.4.1.3. Редактирование файла

Для редактирования текста необходимо перейти в режим ввода (нажать <i>).

Основные команды редактирования:

- <R>, <i> – переход в режим ввода, замена текста под курсором;
- <I> – переход в режим ввода с начала текущей строки;
- <o> – переход в режим ввода с новой строки под курсором;
- <O> – переход в режим ввода с новой строки над курсором;
- <a> – переход в режим ввода после курсора;
- <x> – стирание символа под курсором;
- <X> – стирание символа перед курсором;
- <dd> – стирание текущей строки;
- <d<число>d> – стирание выбранного числа строк, начиная с текущей;
- <yy> – копирование текущей строки в неименованный буфер;
- <y<число>y> – копирование выбранного числа строк, начиная с текущей в неименованный буфер;
- <p> – вставка строки из неименованного буфера под курсор;
- <P> – вставка строки из неименованного буфера над курсором;
- <J> – слияние текущей строки со следующей;
- <u> – отмена последней команды;
- <.> – повтор последней команды.

Для перехода в режим строчного редактора ED необходимо нажать <Shift>+<:>.

7.1.4.1.4. Запись в файл и выход из редактора

Запись в файл выполняется следующей командой:

<Esc>:w<Enter>

В случае, если файл заблокирован другим пользователем либо отсутствуют права на запись, необходимо использовать следующую команду:

<Esc>:w!<Enter>

При попытке записи без «!» будет выдано соответствующее предупреждение.

Создать новый файл <имя_файла> и записать в него текущее содержимое:

<Esc>:w имя_файла <Enter>

В случае, если файл с таким именем уже существует, редактор выдаст предупреждение. После успешного создания файла и осуществления записи информации в него работа продолжится со старым файлом.

Для выхода из редактора необходимо использовать следующую команду:

<Esc>:q<Enter>

В случае, если в файл были внесены изменения, необходимо добавлять после команды «!».

Выйти из редактора не сохраняя изменения:

<Esc>:q!<Enter>

Сохранить изменения в файле и выйти:

<Esc>:wq<Enter> или <Esc>ZZ<Enter>.

7.1.4.1.5. Дополнительные возможности

Текстовый редактор Vi обладает рядом дополнительных возможностей, которые вызываются следующими командами:

- ^G – показать информацию о файле;
- G – перейти в конец файла;
- <number>G – перейти на конкретную строку <number>;
- :<number> – перейти на <number> строк вперед;
- :setnu[mber] – отобразить слева нумерацию строк (:setnonu[mber] – спрятать нумерацию);
- :setwrap – переносить длинные строки (:setnowrap – не переносить);
- :colorscheme<name> – задать цветовую тему (где <name> имя темы, TAB работает как авто-дополнение);
- /мама – поиск текста «мама» в файле;
- n – повторить поиск;
- :h или :help – список возможной помощи (:viusage, :exusage).

Привести концы строк в файле к виду dos или unix соответственно:

:set fileformat=dos

```
:setfileformat=unix
```

Задать размер табуляции в четыре пробела:

```
:setts=4
```

7.1.4.2. Редактор VIM

VIM – свободный режимный текстовый редактор, созданный на основе Vi.

7.1.4.2.1. Основной режим работы

Основной режим работы VIM предназначен для просмотра файлов, ввода команд и перехода из него в другие режимы. В командный режим можно попасть по нажатию клавиши <Esc>.

При нажатии клавиши «:» становится доступна командная строка ViM, в которой вводятся следующие команды:

- команда выхода – `quit` либо `q`;
- команда сохранения – `write` либо `w`, параметром которой может быть имя файла;
- вызов справки – `help` либо `h`.

Для остальных клавиш (и их последовательностей) допускается задавать любые команды либо использовать значения по умолчанию.

Перечисленные ниже команды вводятся в основном режиме (если нет специального уточнения). Все они имеют командностроочные аналоги и могут быть легко переопределены.

7.1.4.2.2. Визуальный режим работы

Визуальный режим работы предназначен, в первую очередь, для выделения блоков текста. Переход в визуальный режим выполняется с помощью следующих сочетаний клавиш:

- <v> для посимвольного выбора;
- <Shift>+<v> для построчного выбора;
- <Ctrl>+<v> для блочного выбора.

В режиме посимвольного выделения (при переходе по клавише «v») допускается оперировать следующими сущностями:

- слово («w»);

- предложение («s»);

- параграф («p»);

- блок («b»).

Выделение при этом необходимо начинать с позиции курсора («a»), или же с начала блока («i»). Например, выделение текущего блока (участка, ограниченного парными элементами) можно произвести следующим образом:

```
<Esc>vib
```

Копирование в буфер выделенного текста осуществляется по «y», вырезание по «d» а вставка, соответственно, «p».

7.1.4.2.3. Режим редактирования

Режим редактирования предназначен для ввода текста. Переключение на режим редактирования осуществляется нажатием клавиши <Insert>.

7.1.4.2.4. Переходы

Для перехода на строку с номером n используется команда `G`. Так для перехода к началу текста нужно набрать `1G`, для сотой строки `100G`, а для перехода в конец текста – `$G`.

Для перехода на n символов в нужную сторону используются клавиши навигации на клавиатуре. То есть для перехода на 1000 символов вниз нужно набрать «1000» и нажать клавишу «↓».

Для перемещения по тексту допускается использовать следующие команды:

- «(», «)» – для перемещения по предложениям;

- «{», «}» – для параграфов;

- «[[«, «]]» – для функций;

- «%» – переход к парной скобке;

- «’» – к предыдущему положению;

- <Ctrl>+<O>, <Ctrl>+<I> – соответственно, назад и вперед по истории переходов.

7.1.4.2.5. Метки

Используются для отметки позиции (<буква>-метка, где меткой является любая буква) и быстрого к ней перехода (<`>-метка). Метки нижнего регистра действительны в пределах данного файла, метки верхнего регистра действуют во всех открытых файлах.

Список всех меток можно получить командой `marks`.

7.1.4.2.6. Регистры

Регистр отмечается видом <"буква>. К нему применимы все стандартные действия: копирование в него ("<метка>y), вырезание ("<метка>d), и вставка из него ("<метка>p), можете вместо `p` использовать `[p,]p` для вставки соответственно перед, или после курсора).

В режиме редактирования вставка из регистра осуществляется по <Ctrl>+R<метка>. Для добавления данных в регистр используйте заглавную метку.

Также допускается писать в регистр, воспользовавшись командой «q<метка>» и завершив запись по `q`. Таким образом сохраняется макрос, выполнить который можно по «@<метка>».

Регистры с метками «*» и «+» совпадают с X-Window clipboards, «%» – соответствует редактируемому файлу. Для просмотра содержимого всех регистров нужно воспользоваться командой `:registers`, либо `:reg метка1метка2...` для просмотра только выбранных регистров.

7.1.4.2.7. Фолды

Фолды предназначены для сокрытия строк, ненужных в данный момент.

По умолчанию фолды активированы в режиме ручной расстановки. Все команды для работы с фолдами начинаются с `z`:

- создание фолд выполняется командой `zf`;
- открытие фолд производится командой `zo` либо нажатием навигационной стрелки «→»;
- закрытие кода в существующий фолд – по `zc`.

Для автоматического подключения фолд по отношению к табуляции необходимо добавить в файл настроек следующую строку:

```
set foldmethod=indent
```

7.1.4.2.8. Сессии

Сессии предназначены для сохранения текущего состояния и настройки редактора таким образом, что при следующем запуске работа продолжится с того же места.

Сессии создаются следующей командой:

```
:mksession /path/to/Session.vim
```

Чтение сессий выполняется командой:

```
:so /path/to/Session.vim
```

Для сохранения текущего контекста (текст, положение курсора в коде, текущая расстановка фолдов) нужно использовать команду `:mkview`, а для чтения – `:loadview`.

Автоматическое сохранение и чтение контекста при начале и окончании редактирования файла может быть реализовано следующим кодом (применяется для всех файлов, имеющих точку в имени):

```
au BufWinLeave *.* mkview
au BufWinEnter *.* silent loadview
```

7.1.4.2.9. Поиск и замена

Поиск по тексту осуществляется следующими командами:

- `/` – поиск по регулярному выражению вперед;
- `?` – поиск по регулярному выражению в обратном направлении;
- `n` – продолжение поиска далее по тексту;
- `N` – повторение предыдущего запроса;
- `# либо *` – поиск слова под установленным курсором.

Для поиска с заменой рекомендуется использовать следующую команду:

```
%s/что/на что/gic
```

где `%` означает работу со всем текстом (а не с текущей строкой), `g` – глобальная замена (а не первое совпадение), `i` – игнорирование регистра, а `c` – подтверждение каждого действия.

7.1.4.2.10. Автодополнение, Отмена, Смена регистра, Повтор

Автодополнение производится по содержимому данного файла, а также указанных в переменной dictionary по нажатию клавиш ``.

Для отмены предыдущих действий в режиме автодополнения используется `u`.

Для смены регистра выделенного участка (или буквы под курсором) используется `~`. При этом команда `U` – принудительно устанавливает верхний регистр, а `u` – нижний.

Для повтора прошлой команды используется символ «.».

7.1.4.2.11. Конфигурация

Файл конфигурации используется для настройки различных аспектов поведения и внешнего вида Vim. Комментарии в этом файле начинаются с символа «"» (двойная кавычка) и продолжаются до конца строки. Основным конфигурационным файлом является `~/.vimrc`. Активация русского шрифта в GUI-режиме, плюс выбор темы для обоих режимов осуществляется, например, следующим кодом:

```
if has("gui_running")
colorscheme candy
set  guifont=-cronyx-courier-medium-r-normal-*-*-120-*-*-m-*-koi8-
r
endif
if !has("gui_running")
colorscheme elflord
endif
```

В файл конфигурации можно добавить привычное поведение и привычные сочетания клавиш:

```
"Выход по F10
nmap <F10> :q<CR>
imap <F10> <ESC>:q<CR>
"Сохранение по F2
nmap <F2> :w<CR>
imap <F2> <ESC>:w<CR>i<Right>
"Компиляция по F9
```

```
nmap <F9> :make<CR>
imap <F9> <ESC>:make<CR>
```

В Vim присутствует подробная документация по настройкам – `:options`.

Основным конфигурационным файлом является `~/.vimrc`. Активация русского шрифта в GUI-режиме и выбор темы для обоих режимов осуществляется следующим образом:

```
if has("gui_running")
colorscheme candy
set  guifont=-cronyx-courier-medium-r-normal-*-*-120-*-*-m-*-koi8-
r
endif
if !has("gui_running")
colorscheme elflord
endif
Быстрые клавиши:
"Выход по F10
nmap <F10> :q<CR>
imap <F10> <ESC>:q<CR>
"Сохранение по F2
nmap <F2> :w<CR>
imap <F2> <ESC>:w<CR>i<Right>
"Компиляция по F9
nmap <F9> :make<CR>
imap <F9> <ESC>:make<CR>
```

7.1.5. Средства настройки отложенного исполнения команд

7.1.5.1. Служба crond

Для регулярного запуска команд в ОС Альт 8 СП используется служба crond.

Служба crond запускается при загрузке системы и проверяет очередь заданий at и заданий пользователей в файлах crontab. При запуске, служба crond сначала проверяет каталог `/var/spool/cron` на наличие файлов crontab, файлы crontab имеют имена пользователей, соответствующие именам пользователей из `/etc/passwd`. Каждый пользователь может иметь только один файл crontab, записей в файле может быть несколько.

В случае, если задание не было обнаружено, crond переходит в режим ожидания на одну минуту и затем вновь приступает к поискам команды, которую следует запустить в этот момент. Большую часть времени служба crond проводит в режиме ожидания, и для ее работы используется минимум системных ресурсов.

Чтобы определить список задач для cron, используется команда crontab.

7.1.5.1.1. Crontab

Утилита crontab управляет доступом пользователя к службе crond путем копирования, создания, выдачи содержимого и удаления файлов crontab, таблиц заданий. При вызове без опций, crontab копирует указанный файл или стандартный входной поток (если файл не указан) в каталог, в котором хранятся пользовательские таблицы заданий cron. Каждый пользователь может иметь свои собственные файлы crontab, и, хотя эти файлы доступны в `/var/spool/cron`, они не предназначены для редактирования напрямую.

Синтаксис:

```
crontab [имя_файла]
crontab [ -elr ] имя_пользователя
```

Опции:

1) `-e` – редактирует копию файла crontab текущего пользователя или создает пустой файл для редактирования, если соответствующего файла crontab не существует. Когда редактирование завершается, файл устанавливается в качестве пользовательского файла crontab. Переменная среды EDITOR задает редактор, вызываемый при указании опции `-e`. Все задания в файле crontab должны создаваться с помощью утилиты crontab;
2) `-l` – отображает текущий файл crontab на стандартный вывод;
3) `-r` – удаляет текущий файл crontab.

7.1.5.1.2. Контроль доступа к crontab

Доступ пользователя к crontab разрешен, если:

- имя пользователя указано в файле `/etc/cron.d/cron.allow`;
- файл `/etc/cron.d/cron.allow` не существует и имя пользователя не указано в файле `/etc/cron.d/cron.deny`.

Доступ пользователя к crontab не разрешен, если:

- файл `/etc/cron.d/cron.allow` существует и имя пользователя в нем не указано;
- файл `/etc/cron.d/cron.allow` не существует и имя пользователя указано в файле `/etc/cron.d/cron.deny`.

Правила разрешения и запрещения выполнения заданий применимы к пользователю root, только если существуют файлы `allow/deny`.

В файлах `allow/deny` надо задавать по одному имени пользователя в строке.

7.1.5.1.3. Формат записи файла crontab

Редактировать crontab пользователя можно используя команду:

```
crontab -e
```

Файл crontab состоит из строк, содержащие шесть полей. Поля разделяются пробелами или символами табуляции. Первые пять полей – целочисленные шаблоны, задающие:

- минуту (0 – 59);
- час (0 – 23);
- день месяца (1 – 31);
- месяц года (1 – 12);
- день недели (0 – 6, причем 0=воскресенье).

Каждый из этих шаблонов может представлять собой звездочку (которая обозначает все допустимые значения) или список элементов через запятые. Элемент – число или два числа через дефис (что обозначает закрытый интервал). Обратите внимание, что дни можно указывать в двух полях (день месяца и день недели). Оба поля учитываются, если заданы в виде списка элементов (запись: 30 4 1,15 * 5 приведет к выполнению команды в 4:30 пополуночи первого и пятнадцатого числа каждого месяца, плюс в каждую пятницу). При указании диапазона можно пропускать некоторые его значения, указав шаг в форме «/число». Например: «0-23/2» для поля час означает запуск команды через два часа. Шаг можно указывать также после звездочки: «каждые два часа» соответствует значению «*/2». Для задания полей месяц и день_недели можно использовать имена. Указывайте

первые три буквы нужного дня или месяца на английском, регистр букв не имеет значения. Диапазоны или списки имен не разрешены.

Служба crond запускает команды, когда значения полей минута, час, месяц и хотя бы одно из полей число и день_недели, совпадают с текущим временем. Служба crond сверяет директивы с текущим временем раз в минуту.

Вместо первых пяти полей допустимо указание одного из восьми специальных триггеров:

- @reboot – выполнить команду один раз, при запуске cron;
- @yearly – выполнять команду каждое 1 января, «0 0 1 1 *»;
- @annually – эквивалентно @yearly;
- @monthly – выполнять команду в начале каждого месяца, «0 0 1 * *»;
- @weekly – выполнять команду каждое воскресенье, «0 0 * * 0»;
- @daily – выполнять команду в полночь, «0 0 * * *»;
- @midnight – эквивалентно @daily;
- @hourly – выполнять команду раз в час, «0 * * * *».

Шестое поле в строке файла crontab – строка, выполняемая командным интерпретатором в указанные моменты времени. Символ % (процент) в этом поле, если он не замаскирован «\» (обратной косой), преобразуется в символ новой строки.

Только первая строка (до символа % или до конца строки) поля команды выполняется командным интерпретатором. Другие строки передаются команде как стандартный входной поток. Пустые строки, ведущие пробелы и символы табуляции игнорируются. Строки, начинающиеся с символа («#») считаются комментариями и игнорируются. Комментарии не допускаются в тех же строках, где расположены команды cron, так как они будут распознаны как части команды. По этой же причине комментарии не разрешены в строках, задающих переменные среды.

Строка-директива представляет собой либо задание переменной среды, либо команду cron.

Демон crond предоставляет каждому командному интерпретатору стандартную среду, задавая переменные `HOME`, `LOGNAME`, `SHELL(=/bin/sh)`, `TZ` и `PATH`. Стандартное значение переменной PATH для пользовательских заданий cron – `/usr/bin`, а для заданий cron пользователя root – `/usr/sbin:/usr/bin`.

Если стандартный выходной поток и стандартный поток ошибок команд не перенаправлены, любые сгенерированные результаты или сообщения об ошибках будут отправлены пользователю по электронной почте.

7.1.5.1.4. Примеры

Далее приведены примеры использования таблиц crontab в ходе администрирования ОС Альт 8 СП.

Пример 1

```
$ crontab -e
#minute (0-59),
#| hour (0-23),
#| | day of the month (1-31),
#| | | month of the year (1-12),
#| | | | day of the week (0-6 with 0=Sunday).
#| | | | | commands
# Каждые 5 минут записывать результат вывода
# команды date в файл date.txt в домашнем каталоге
*/5 * * * * date > ~/date.txt
# Выполнять задание в 18 часов 7 минут 13 числа
# каждого месяца и по пятницам
7 18 13 * 5 /home/www/myscript.pl
# Выполнять задание по воскресеньям в 10 час 30 минут
30 10 * * 0 /home/www/myscript.pl
crontab: installing new crontab
```

Вывод `crontab: installing new crontab` означает, что новый crontab успешно установлен.

Пример 2

```
# использовать для запуска команд /bin/sh
# не обращая внимание на то, что написано в /etc/passwd
SHELL=/bin/sh
```

```
# отправлять вывод выполнения команд по электронной
# почте пользователю 'paul'
# не обращая внимания на то, чей это crontab
MAILTO=paul
#
# запускать пять минут пополуночи, каждый день
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# запускать в 14:15 первого числа каждого месяца
15 14 1 * * $HOME/bin/monthly
# запускать в 22.00 каждый рабочий день
0 22 * * 1-5 mail -s "Уже 10 вечера"
23 0-23/2 * * * echo "запуск в 00:23, 2:23, 4:23 ..., каждый
день"
5 4 * * sun echo "запуск в 4:05 каждое воскресенье"
```

7.1.5.1.5. Дополнительные возможности таблиц

Талицы crontab обладают следующими дополнительными возможностями:

- при задании дня недели 0 и 7 соответствуют воскресенью;
- допускается указывать одновременно и списки, и диапазоны в одном и том же поле;
- допускается указывать диапазоны с пропусками – например, «1-9/2» соответствует «1,3,5,7,9»;
- допустимо указание месяцев или дней недели по имени;
- в crontab разрешено задавать переменные среды вручную;
- вывод команд отсылается почтой владельцу файла crontab, а также может отправляться кому-либо другому, либо отправка может быть отключена (функция не поддерживается в SysV);
- любая из команд с префиксом «@» может заменять первые пять полей файла.

7.1.5.2. Команда `at`

Для запуска одной или более команд в заранее определенное время используется команда `at`. В ней можно определить время и (или) дату запуска той или иной команды.

Команда `at` требует двух (или большего числа) параметров – как минимум, следует указать время запуска и какая команда должна быть запущена. Параметры запуска с помощью команды at указываются в виде списка строк, следующих за ней. Ввод каждой строки завершается нажатием клавиши <Enter>. По окончании ввода всей команды нажать клавиши <Ctrl>+<D> для ее завершения.

Например, если необходимо запустить команды в 1:23, следует ввести:

```
at 1:23
lpr /usr/sales/reports/.
echo "Files printed"
```

В указанном примере будут распечатаны все файлы каталога `/usr/sales/reports`, и пользователю будет выведено сообщение на экран монитора. После ввода всей команды отобразится следующая запись:

```
job 756603300.a at Tues Jan 21 01:23:00 2007
```

Это означает, что указанные команды будут запущены, как и было задано, в 1:23. В сообщении также приведен идентификатор задания (756603300.a), который понадобится, если необходимо отменить задание:

```
at -d 756603300.a
```

В случае, если список команд находится в файле, например, `getdone`, и необходимо запустить все перечисленные в нем команды в 10:00, следует воспользоваться одной из двух форм команды `at`:

`at 10:00 < getdone` либо `at 10:00 -f getdone`

Обе приведенные команды эквивалентны. Разница заключается в том, что в первой команде используется механизм перенаправления потоков ввода-вывода, во второй команде – дисковый файл.

Кроме времени, в команде `at` может быть также определена дата:

```
at 17:00 Jan 24
lp /usr/sales/reports/
echo "Files printed"
```

Задания, определяемые администратором, помещаются в очередь, которую ОС периодически просматривает. Администратору необязательно находиться в

системе для того, чтобы at отработала задания, команда будет работать в фоновом режиме.

Для того чтобы просмотреть очередь заданий, нужно ввести следующую команду:

```
at -l
```

В случае если предыдущие примеры были запущены, то будет выведено:

```
job 756603300.a at Sat Dec 20 01:23:00 2007 job 756604200.a at
Sat Jan 24
17:00:00 2008
```

Администратор видит только свои задания по команде at.

Для удаления задания из очереди следует запустить at с опцией -d и номером удаляемого задания следующим образом:

```
at -d 756604200.a
```

Далее представлены варианты использования команды at.

Выполнить задание во время hh:mm в 24-часовом формате:

```
at hh:mm
```

Выполнить задание во время hh:mm в 24-часовом формате в соответствующий день:

```
at hh:mm месяц день год
```

Вывести список заданий в очереди (псевдоним команды – atq):

```
at -l
```

Выполнить задание через определенное время, которое задано параметром count в соответствующих единицах – неделях, днях, часах или минутах:

```
at now+count time-units
```

Удалить задание с идентификатором job_ID из очереди (псевдоним команды – atrm):

```
at -d job_ID
```

Администратор может применять все эти команды. Для других пользователей права доступа к команде at определяются файлами /etc/at.allow и /etc/at.deny. В случае если, существует файл /etc/at.allow, то применять команду at могут только перечисленные в нем пользователи. В случае, если же такого файла нет,

проверяется наличие файла `/etc/at.deny`, в котором отражено, кому запрещено пользоваться командой `at`. Также если ни одного из файлов, описывающих доступ к «alt», нет, то команда `at` доступна только пользователю с идентификатором root.

7.1.5.3. Команда `batch`

Команда `batch` позволяет ОС самой решить, когда наступает подходящий момент для запуска задачи – например, когда система находится в состоянии наименьшей загрузки, и процессы запускаются в фоновом режиме.

Формат команды `batch` представляет собой список заданий для выполнения, следующих в строках за ней, заканчивается список комбинацией клавиш <Ctrl>+<D>. Также допускается поместить список команд в файл и перенаправить его на стандартный ввод команды `batch`.

Например, для сортировки набора файлов, печати результатов и вывода сообщения нужно ввести следующие команды:

```
batch
sort /usr/sales/reports ; lp
echo "Files printed"
```

В ответ на это система выдаст:

```
job 7789001234.b at Fri Feb 21 11:43:09 1999
```

Примечание. Дата и время, приведенные в сообщении, соответствуют нажатию клавиш <Ctrl>+<D>.

7.1.6. Средство управления процессами xinetd

Средство управления процессами xinetd (далее – сервер xinetd) выполняет функции управления процессами, которые обеспечивают работу сервисов подключения к локальным и глобальным сетям.

Сервер xinetd представляет собой единственный процесс, который выполняет прослушивание всех портов на наличие запросов от других сервисов, перечисленных в файле конфигурации `xinetd.conf` (расположен в директории `/etc`): когда на порт поступает запрос, сервер xinetd запускает соответствующий сервер.

Сервисы, перечисленные в конфигурационном файле сервера xinetd, можно разделить на две группы.

Сервисы из первой группы («multi-threaded») на каждый новый запрос запускают новый серверный процесс. Для таких сервисов сервер xinetd продолжает прослушивать сеть на соответствующем порту, ожидая новых запросов на порождение нового процесса.

В другую группу («single-threaded») включаются сервисы службы, которые в состоянии обрабатывать новые соединения. В ходе работы с ними сервер xinetd прекращает обработку новых запросов до тех пор, пока серверный процесс не завершит свою работу. Сервисы в этой группе также обычно относят к группе «datagram-based», работающих с дейтаграммными протоколами передачи данных формата UDP.

Сервер xinetd позволяет сохранять системные ресурсы за счет контроля запуска серверных процессов. Полностью соответствуя назначению запускать требуемые сервисы, сервер xinetd осуществляет так же функции контроля доступа и регистрации событий. Кроме того, сервер xinetd не ограничен сервисами, перечисленными в файле `/etc/services`. Также допускается использовать сервер xinetd для запуска сервисов специального назначения.

Синтаксис:

```
xinetd [опции]
```

Опции:

- `-d` – активирует режим отладки. Указание этой опции приводит к большому количеству отладочных сообщений, которые делают возможным использование отладчика на xinetd;

- -syslog syslog_facility – разрешает протоколирование создаваемых xinetd сообщений через syslog с заданным syslogfacility. Поддерживаются следующие имена facility: daemon, auth, user, local[0-7] (назначение можно просмотреть в syslog.conf). Данная опция неэффективна в режиме отладки, так как все необходимые сообщения отправляются на терминал;
- -filelog файл_журнала – сообщения, создаваемые xinetd будут помещаться в указанный файл. Сообщения всегда добавляются к уже существующему файлу. Если файл не существует, то он будет создан. Данная опция неэффективна в режиме отладки, так как все необходимые сообщения отправляются на терминал;
- -f файл_настроек – задает файл, который xinetd использует для настройки. По умолчанию это /etc/xinetd.conf;
- -pidfile pid_файл – в этот файл записывается идентификатор процесса. Данная опция неэффективна в режиме отладки;
- -stayalive – xinetd будет оставаться запущенным, даже если не задано никаких служб;
- -loop rate – устанавливает верхнюю величину цикла, по которой определяется, что служба работает с ошибками и по которой она отключается. Величина цикла задается в терминах количества серверов в секунду, которое может быть запущено в обработку (fork). Для этой опции, корректное значение определяется скоростью вашей машины. По умолчанию равно 10;
- -reuse – xinetd будет устанавливать опцию сокета SO_REUSEADDR перед привязкой сокета службы к какому-либо интернет адресу. Это позволяет привязать адрес, даже если есть программа, которая уже использует его, например, в том случае, если некоторые серверы были запущены во время предыдущего запуска xinetd и еще не завершили свою работу. Данная опция не оказывает влияния на службу RPC;

- `--limit proc_limit` – устанавливает ограничение на количество одновременно запущенных процессов, которые может запустить xinetd. Ее назначение предотвращать переполнение таблицы процессов;
- `--logprocs limit` – устанавливает ограничение на количество одновременно запущенных серверов на один идентификатор удаленного пользователя;
- `--shutdownprocs limit` – устанавливает ограничение на количество одновременно запущенных серверов для завершения работы службы;
- `--version` – вывести информацию о версии xinetd;
- `--cc interval` – xinetd будет выполнять периодические проверки своего внутреннего состояния каждые `interval` секунд.

Опции `syslog` и `filelog` являются взаимно исключающими. Если ни одна из них не задана, то по умолчанию используется `syslog` с `daemonfacility`. Не путайте сообщения xinetd с сообщениями, которые создаются службами. Последние протоколируются только если это задано в файле с настройками.

Сервер xinetd выполняет определенные действия при получении определенных сигналов. Действия, ассоциированные с соответствующими сигналами, могут быть переопределены путем редактирования `config.h` и последующей компиляции.

Сигналы:

- SIGHUP – заставляет выполнить жесткую перенастройку, означающую, что xinetd перечитает файл с настройками и завершит работу серверов для тех служб, которые больше не доступны. Управление доступом выполняется снова на уже запущенных серверах через проверку удаленных подключений, времени доступа и копий серверов. Если количество копий серверов уменьшается, то некоторые произвольно выбранные сервера будут убиты, чтобы соблюсти ограничение; это случится после завершения работы тех серверов, которые попадают под ограничение доступа с удаленных адресов или ограничение времени доступа. Также, если флаг INTERCEPT был сброшен и происходит его установка, то будет завершена работа любых запущенных серверов для служб с этим флагом. Цель такого поведения –

убедиться, что после жесткой перенастройки не будет запущено серверов, которые могут принимать пакеты с тех адресов, которые не соответствуют критериями управления доступом;

- SIGQUIT – приводит к завершению работы;
- SIGTERM – завершает работу всех запущенных серверов перед завершением работы xinetd;
- SIGUSR1 – приводит к снятию дампа внутреннего состояния (по умолчанию файл дампа это `/var/run/xinetd.dump`; чтобы изменить данное имя файла нужна правка `config.h` и перекомпиляция);
- SIGIOT – производит внутреннюю проверку того, что структуры данных, используемые программой не повреждены. Когда проверка завершится, xinetd сгенерирует сообщение, которое скажет успешно прошла проверка или нет.

При реконфигурации файлы журналов закрываются и вновь открываются. Это позволяет удалять старые файлы журналов.

7.1.7. Администрирование многопользовательской и многозадачной среды

7.1.7.1. Команда `who`

Для получения списка пользователей, работающих в ОС, используется команда `who`, которая позволяет вывести в консоль идентификаторы активных пользователей, терминалы и время входа в систему.

Для получения списка пользователей, зарегистрировавшихся в системе, необходимо выполнить команду `who`. Задавая различные опции, с помощью команды `who` можно получить информацию о времени начала и конца сеансов работы пользователей, перезагрузок, корректировках системных часов, а также о других процессах, порожденных процессом init.

Синтаксис команды `who`:

```
who [-u] [-T] [-l] [-H] [-q] [-p] [-d] [-b] [-r] [-t] [-a] [-s] [имя файла]
```

Опции команды `who` приведены в таблице 1.

Т а б л и ц а 1 – Опции команды `who`

Опция	Описание
`-u`	Позволяет вывести информацию о пользователях, которые в настоящее время являются активными (работают в ОС).
`-H`	Опция, аналогичная опции `-u` (дополнительно в консоль выводится название столбцов).
`-s`	Позволяет вывести в консоль имена активных пользователей и терминальных линий, а также время и дату начала сессии пользователей.
`-t`	Позволяет вывести информацию о последней корректировке системных часов администратором.
`-r`	Позволяет вывести текущий уровень выполнения процесса init, кроме этого, будут выведены идентификатор процесса, системный код завершения и пользовательский код завершения процесса.
`-a`	Позволяет обработать файл `/etc/utmp` или файл, указанный в команде, считая, что все опции (кроме THqs) включены.
`-b`	Позволяет вывести время и дату последней загрузки системы.
`-d`	Позволяет вывести информацию обо всех процессах, которые прекратили существование и не были заново порождены процессом init.
`-p`	Позволяет вывести список всех других процессов, активных в настоящий момент, которые были порождены процессом init.
`-q`	Позволяет вывести имена и количество пользователей, работающих в настоящий момент в системе.
`-l`	Позволяет вывести список линий, на которых система ожидает входа в нее какого-либо пользователя.
`-T`	Аналогична опции -s с той разницей, что дополнительно в позиции STATE выводится информация о состоянии терминальной линии.

Сообщения, выводимые после выполнения команды `who`, имеют следующий формат:

```
NAME [STATE] LINE TIME [IDLE] [PID] [COMMENT] [EXIT]
```

Информация NAME, LINE и TIME выводится при использовании всех опций, кроме `-q`, `STATE` – только при использовании опции `-T`, `IDLE` и `PID` – только при использовании опции `-u` и `-l`, `COMMENT` и `EXIT` – только при использовании опции `-a`.

В сообщениях, выводимых после выполнения команды `who`, фигурируют следующие параметры:

- `NAME` – имя пользователя;
- `STATE` – состояние терминальной линии (состояние – возможность передавать сообщения на терминал от кого-либо другого терминала: состояние «+» – свидетельствует о том, что терминалу может передавать сообщения любой другой терминал, состояние «–» – терминалу сообщения передаваться не могут; пользователь root может передавать сообщения во все линии, которым отвечает состояние «+» или «–»; при обнаружении неисправной линии выводится «?»);
- `LINE` – имя терминальной линии;
- `TIME` – время и дата начала сеанса работы пользователя в системе;
- `IDLE` – время, прошедшее со времени последней активной работы пользователя;
- `PID` – идентификатор процесса входной оболочки пользователя;
- `COMMENT` – комментарий, характеризующий данную линию (если таковые имеются в файле `/etc/inittab` – этот файл может содержать, например, сведения о местоположении терминала, телефонном номере комнаты или о типе физического терминала).

Чтобы получить сведения о сеансе, учетной записи и PID запущенного процесса необходимо выполнить следующую команду:

```
who -uH
```

На экран монитора будет выведено сообщение следующего вида:

```
ИМЯ ЛИНИЯ ВРЕМЯ IDLE PID КОММЕНТАРИЙ
user-name line-name mm-dd hh:mm . 10340 (:0)
```

где:

- `user-name` – имя пользователя;
- `line-name` – имя терминальной линии;
- `mm-dd hh:mm` – дата (в формате мм- дд, мм – месяц, дд – день) и время (в формате чч:мм, чч – час, мм – минута) начала сеанса работы пользователя;

- `10340` – PID-идентификатор процесса;

- `(:0)` – отсутствующий комментарий.

Точка (.) в параметре IDLE свидетельствует о том, что данный терминал находился в активном состоянии не более минуты тому назад.

7.1.7.2. Команда `ps`

Для получения информации о состоянии запущенных процессов используется команда `ps`. Она выдает следующую информацию о процессах: какие из них выполнены, какие вызвали проблемы в системе, как долго выполняется тот или иной процесс, какие он затребовал системные ресурсы, идентификатор процесса (который будет необходим, например, для прекращения работы процесса с помощью команды `kill`).

Команда `ps`, запущенная без опций командной строки, выдает список процессов, которые порождены учетной записью администратора.

Наиболее распространенное применение `ps` – отслеживание работы фоновых и других процессов в системе. Поскольку в большинстве случаев фоновые процессы никак не взаимодействуют ни с экраном, ни с клавиатурой, команда `ps` остается основным средством наблюдения за ними.

Синтаксис команды `ps`:

```
ps [-e] [-d] [-a] [-f] [-l] [-n файл_с_системой] [-t
список_терминалов]
[-p список_идентификаторов_процессов]
[-u список_идентификаторов_пользователей]
[-g список_идентификаторов_лидеров_групп]
```

Опции команды `ps` приведены в таблице 2.

Т а б л и ц а 2 – Опции команды `ps`

Опция	Описание
`-e`	Позволяет вывести информацию обо всех процессах.
`-d`	Позволяет вывести информацию обо всех процессах, кроме лидеров групп.
`-a`	Позволяет вывести информацию обо всех наиболее часто запрашиваемых процессах, то есть обо всех процессах, кроме лидеров групп и процессов, не ассоциированных с терминалом.
`-f`	Позволяет сгенерировать полный листинг.
`-l`	Генерировать листинг в длинном формате.
`-n файл_с_системой`	Считать, что операционная система загружена из файла_с_системой, а не из файла `/unix`.
`-t список_терминалов`	Позволяет вывести информацию только о процессах, ассоциированных с терминалами из заданного списка_терминалов (терминал – это либо имя файла-устройства, например, tty, номер или console, либо просто номер, если имя файла начинается с tty).
`-p`	Список_идентификаторов_процессов – позволяет вывести информацию только об указанных процессах.
`-u`	Список_идентификаторов_пользователей – позволяет вывести информацию только о процессах с заданными идентификаторами или входными именами пользователей (идентификатор пользователя выводится в числовом виде, а при наличии опции `-f` – в символьном).
`-g`	Список_идентификаторов_лидеров_групп – позволяет вывести информацию только о процессах, для которых указаны идентификаторы лидеров групп (лидер группы – это процесс, номер которого идентичен его идентификатору группы).

`ps` выводит четыре основных поля информации для каждого процесса:

- PID – идентификатор процесса;

- TTY – терминал, с которого был запущен процесс;

- TIME – время работы процесса;

- COMMAND – имя выполненной команды.

При указании опции `-f` команда `ps` пытается определить имя команды и аргументы, с которыми был создан процесс, исследуя пользовательский блок процесса. В случае если это не удается, имя процесса выводится так же, как и при отсутствии опции `-f`, только заключается в квадратные скобки.

В таблице 3 приводятся заголовки колонок листинга, и поясняется смысл их содержимого. Буквы `l` или `f` в скобках означают, что эта колонка появляется соответственно при длинном или полном формате листинга, отсутствие букв означает, что данная колонка выводится всегда. При этом опции `-l` и `-f` влияют только на формат выдачи, но не на список процессов, информация о которых будет предоставлена.

Таблица 3 – Описание заголовков колонок листинга

Заголовок	Значение	Описание
F (l)	Флаги (шестнадцатеричные), логическая сумма которых характеризует процессы следующим образом:	
	00	Процесс терминирован, элемент таблицы процессов свободен.
	01	Системный процесс: всегда в основной памяти.
	02	Процесс трассируется родительским процессом.
	04	Родительский трассировочный сигнал остановил процесс, родительский процесс находится в состоянии ожидания.
	08	Процесс не может быть разбужен сигналом.
	10	Процесс в основной памяти.
	20	Процесс в основной памяти, блокирован до завершения события.
	40	Идет сигнал к удаленной системе.
	80	Процесс в очереди на ввод/вывод.

Окончание таблицы 3

Заголовок	Значение	Описание
S (l)	Статус процесса:	
	O	Процесс обрабатывается процессором.
	S	Процесс ожидает завершения события.
	R	Процесс стоит в очереди на выполнение.
	I	Процесс создается.
	Z	Процесс завершен, но родительский процесс не ждет этого.
	T	Процесс остановлен сигналом, так как родительский процесс трассирует его.
S (l)	X	Процесс ожидает получения большего объема основной памяти.
UID (f,l)		Идентификатор владельца процесса, при указании опции `-f` выдается входное имя пользователя.
PID		Идентификатор процесса (необходим для терминирования процесса).
PPID(f,l)		Идентификатор родительского процесса.
C (f,l)		Доля выделенного планировщиком времени центрального процессора.
STIME (f)		Время запуска процесса (часы:минуты:секунды). Если процесс запущен более чем 24 часа назад, выводится месяц и день запуска.
PRI (l)		Приоритет процесса: большее число означает меньший приоритет.
NI (l)		Поправка к приоритету.
ADDR (l)		Адрес процесса в памяти.
SZ (l)		Размер (в блоках по 512 байт) образа процесса в памяти.
WCHAN (l)		Адрес события, которого ожидает процесс (у активного процесса эта колонка пуста).
TTY		Управляющий терминал (обычно – терминал, с которого был запущен процесс). В случае если такового нет, выводится символ «?».
TIME		Истраченное процессом время на выполнение центральным процессором.
COMMAND		Имя программы: если указана опция `-f`, выводится полное имя команды и ее аргументы.

7.1.7.3. Команда `nohup`

Команда `nohup` применяется для того, чтобы процесс продолжал выполняться даже после выхода из системы, поскольку выполнение стандартного дочернего процесса завершается сразу после прекращения работы родительского, и если был запущен фоновый процесс, он также прекращает работу при выходе из системы.

При выполнении, команду `nohup` следует поместить в начало командной строки следующим образом:

```
nohup sort sales.dat &
```

В данном примере `nohup` заставляет ОС игнорировать выход из нее и продолжать выполнение до тех пор, пока процесс не закончится сам по себе. Будет запущен процесс, который продолжит свое выполнение, не требуя контроля администратора.

7.1.7.4. Команда `nice`

Команда `nice` позволяет запустить другую команду с предопределенным приоритетом выполнения, предоставляя администратору возможность определять приоритет при выполнении своих задач.

При обычном запуске все задачи имеют один и тот же приоритет, и ОС равномерно распределяет между ними процессорное время. С помощью команды nice можно понизить приоритет какой-либо задачи, предоставив другим задачам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только пользователь с идентификатором root.

Команда `nice` обладает следующим синтаксисом:

```
nice -number command
```

Уровень приоритета определяется параметром `number`, при этом большее его значение означает меньший приоритет команды. Значение по умолчанию равно «10», и `number` представляет собой число, на которое он должен быть уменьшен.

Например, если запущен процесс сортировки:

```
sort sales.dat > sales.srt &
```

Далее, чтобы дать ему преимущество над следующим процессом, нужно запустить следующий процесс с уменьшенным приоритетом:

```
nice -5 lp mail_list &
```

Для того чтобы назначить процессу самый низкий приоритет из возможных, необходимо выполнить следующую команду:

```
nice -10 lp mail_list &
```

П р и м е ч а н и е . В случае команды `nice` тире означает знак опции.

Только пользователь с идентификатором root может повысить приоритет того или иного процесса, применяя для этого отрицательное значение аргумента. Максимально возможный приоритет – «20», присвоить его процессу пользователь с идентификатором root может с помощью команды:

```
nice --10 job &
```

Наличие символа «&» в примере достаточно условно, можно изменять приоритеты, как фоновых процессов, так и процессов переднего плана.

7.1.7.5. Команда `renice`

Команда `renice` позволяет изменить приоритет работающего процесса. Формат этой команды подобен формату команды `nice`:

```
renice -number PID
```

Для изменения приоритета работающего процесса необходимо знать его идентификатор, получить который можно с помощью команды `ps`, например, вызвав:

```
ps -e : grep name
```

В данной команде необходимо заменить name именем интересующего процесса. Команда `grep` отфильтрует только те записи, в которых будет встречаться имя нужной команды. В случае если необходимо изменить приоритет всех процессов пользователя или группы пользователей, в `команде renice` используется идентификатор пользователя или группы.

Далее приводится пример использования команды `renice`, предположив, что имя пользователя – pav:

```
ps -ef : grep $LOGNAME
pav 11805 11804 0 Dec 22 ttysb 0:01 sort sales.dat > sales srt
```

```
pav 19955 19938 4 16:13:02 ttypo 0:00 grep pav
pav 19938 1 0 16:11:04 ttypo 0-00 bash
pav 19940 19938 42 16:13:02 ttypo 0:33 find . -name core -exec nn
{};
```

Теперь, чтобы понизить приоритет процесса `find` с идентификатором 19940, нужно ввести:

```
renice -5 19940
```

В случае команды `renice` работают те же правила, что и в случае команды `nice`, а именно:

- ее можно использовать только со своими процессами;
- пользователь с идентификатором root может применить ее к любому процессу;
- только пользователь с идентификатором root может повысить приоритет процесса.

7.1.7.6. Команда `kill`

В отдельных ситуациях необходимо прекратить выполнение процесса, не дожидаясь его нормального завершения. Это может произойти в следующих случаях:

- процесс использует слишком много времени процессора и ресурсов компьютера;
- процесс работает слишком долго, не давая ожидаемых результатов;
- процесс производит слишком большой вывод информации на экран или в файл;
- процесс привел к блокировке терминала или другой сессии;
- из-за ошибки пользователя или программы используются не те файлы или параметры командной строки;
- дальнейшее выполнение процесса бесполезно.

В случае если процесс работает не в фоновом режиме, нажатие клавиш <Ctrl>+<C> должно прервать его выполнение, но если процесс фоновый прервать его выполнение можно только с помощью команды `kill`, которая посылает

процессу сигнал, требующий от процесса завершения. Для этого используются две формы:

```
kill PID(s)
kill -signal PID(s)
```

Для завершения процесса с идентификатором 127 ввести:

```
kill 127
```

Для того чтобы завершить процессы 115, 225 и 325, ввести:

```
kill 115 225 325
```

С помощью опции `-signal` можно, например, заставить процесс перечитать конфигурационные файлы без прекращения работы. Список доступных сигналов можно получить с помощью команды:

```
kill -l
```

При успешном завершении процесса никакое сообщение не выводится. Сообщение появится при попытке завершения процесса без наличия соответствующих прав доступа или при попытке завершить несуществующий процесс.

Завершение родительского процесса иногда приводит к завершению дочерних, однако для полной уверенности в завершении всех процессов, связанных с данным, следует указывать их в команде `kill`.

В случае, если терминал оказался заблокированным, можно войти в систему с другого терминала:

```
ps -ef: grep $LOGNAME
```

и завершить работу оболочки на блокированном терминале.

При выполнении команда `kill` посылает процессу соответствующий сигнал. Программы ОС могут посылать и принимать более 20 сигналов, каждый из которых имеет свой номер. Например, при выходе администратора ОС посылает всем его процессам сигнал 1, который заставляет все процессы (кроме запущенных с помощью `nohup`) прекратить работу. Программы могут быть написаны и таким образом, что будут игнорировать посылаемые им сигналы, включая сигнал 15, который возникает при запуске команды `kill` без указания конкретного сигнала.

Однако сигнал 9 не может быть проигнорирован – процесс все равно будет завершен. Таким образом, если команда `kill PID` не смогла завершить процесс (он виден при использовании команды `ps`), необходимо воспользоваться следующей командой:

```
kill -9 PID
```

Команда `kill -9` прекращает процесс, не давая возможности, например, корректно закрыть файлы, что может привести к потере данных. Использовать эту возможность следует только в случае крайней необходимости.

Для завершения всех фоновых процессов необходимо ввести следующую команду:

```
kill 0
```

Преимущественное право контроля над процессом принадлежит владельцу. Права владельца могут отменяться только пользователем с идентификатором root.

Ядро назначает каждому процессу четыре идентификатора: реальный и эффективный UID, реальный и эффективный GID. Реальные ID используются для учета использования системных ресурсов, а эффективные – для определения прав доступа. Как правило, реальные и эффективные ID совпадают. Владелец процесса может посылать в процесс сигналы, а также понижать приоритет процесса.

Процесс, приступающий к выполнению другого программного файла, осуществляет один из системных вызовов семейства `exec`. Когда такое случается, эффективные UID и GID процесса могут быть установлены равными UID и GID файла, содержащего образ новой программы, если у этого файла установлены биты смены идентификатора пользователя и идентификатора группы.

Системный вызов `exec` – это механизм, с помощью которого такие команды, как `passwd`, временно получают права пользователя с идентификатором root (команде `passwd` они нужны для того, чтобы изменить `/etc/passwd`).

7.1.8. Верификация версии

Администратор имеет возможность верифицировать версию ОС Альт 8 СП выполнив команду: `# cat /root/.install-log/diskinfo`

7.2. Управление программными пакетами

Для установки, удаления и обновления программ и поддержания целостности системы в ОС семейства Linux используются менеджеры пакетов типа «rpm». Для автоматизации этого процесса и применяется Усовершенствованная система управления программными пакетами APT (Advanced Packaging Tool).

Автоматизация достигается созданием одного или нескольких внешних репозиториев, в которых хранятся пакеты программ и относительно которых производится сверка пакетов, установленных в системе. Репозитории могут содержать как официальную версию дистрибутива, обновляемую его разработчиками по мере выхода новых версий программ, так и локальные наработки, например, пакеты, разработанные внутри компании.

Таким образом, 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, руководствуется сведениями о внешнем репозитории для разрешения конфликтов и поиска корректного пути их устранения.

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

7.2.1. Источники программ (репозитории)

7.2.1.1. Репозитории

Репозитории, с которыми работает APT, отличаются от обычного набора пакетов наличием мета информации – индексов пакетов, содержащихся в репозитории, и сведений о них. Поэтому, чтобы получить всю информацию о репозитории, APT достаточно получить его индексы.

APT может работать с любым количеством репозиториев 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, а расположение в том или ином репозитории не имеет значения. Если потребуется, APT в рамках одной операции установки группы пакетов может пользоваться несколькими репозиториями.

Подключая одновременно несколько репозиториев, нужно следить за тем, чтобы они были совместимы друг с другом по пакетной базе – отражали один определенный этап разработки. Совместимыми являются основной репозиторий дистрибутива и репозиторий обновлений по безопасности к данному дистрибутиву. В то же время смешение среди источников APT репозиториев, относящихся к разным дистрибутивам, или смешение стабильного репозитория с нестабильной веткой разработки (Sisyphus) чревато различными неожиданными трудностями при обновлении пакетов.

APT позволяет взаимодействовать с репозиторием с помощью различных протоколов доступа. Наиболее популярные – HTTP и FTP, однако существуют и некоторые дополнительные методы.

Для того чтобы APT мог использовать тот или иной репозиторий, информацию о нем необходимо поместить в файл.

Файлы описания источников находятся в директории `/etc/apt/source.list.d/` и имеют расширение `.list`, например:

```
altsp.list
sources.list
```

Так же, есть файл с предопределенным именем: `/etc/apt/source.list`. Утилита apt-get, в момент работы, просматривает одновременно все эти файлы.

Описания репозиториев заносятся в этот файл в следующем виде:

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

где:

- `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`, `cdrom`, `copy`;
- `путь` – путь к репозиторию в терминах выбранного метода;
- `база` – относительный путь к базе данных репозитория;
- `название` – название репозитория.

Синтаксис, описывающий источники:

```
$ cat /etc/apt/sources.list.d/altsp.list
# update.altsp.su (IVK, Moscow)

# ALT Certified 8
#rpm    [cert8]    ftp://update.altsp.su/pub/distributions/ALTLinux
c8/branch/x86_64 classic
#rpm    [cert8]    ftp://update.altsp.su/pub/distributions/ALTLinux
c8/branch/x86_64-i586 classic
#rpm    [cert8]    ftp://update.altsp.su/pub/distributions/ALTLinux
c8/branch/noarch classic

#rpm    [cert8]    http://update.altsp.su/pub/distributions/ALTLinux
c8/branch/x86_64 classic
#rpm    [cert8]    http://update.altsp.su/pub/distributions/ALTLinux
c8/branch/x86_64-i586 classic
#rpm    [cert8]    http://update.altsp.su/pub/distributions/ALTLinux
c8/branch/noarch classic
```

Если первым символом идет символ комментария – строка считается простым текстом, а не описанием источника. У активной записи, в начале строки этот символ отсутствует.

Описание источника состоит из ключевых элементов:

- тип репозитория – применяется пакетная система rpm (все источники описывают rpm-репозитории);
- ключ подписи – пакеты в репозитории подписаны и могут быть проверены, если указать ключ. Списки доступных ключей хранятся в каталоге

/etc/apt/vendors.list.d в файлах с расширением .list. Так же, есть файл /etc/apt/vendors.list. В примере использован ключ [cert8];

- адрес – адрес расположения репозитория. Репозитории доступны несколькими способами (ftp://, http:// и rsync://). После описания способа доступа, прописан адрес;
- тип данных – репозиторий может содержать как исполняемые пакеты, так и пакеты для разработчиков или пакеты с данными общего характера;
- название – название репозитория.

Для добавления в sources.list репозитория на CD/DVD-носителе информации в APT предусмотрена специальная утилита – apt-cdrom. Чтобы добавить запись о репозитории на носителе, достаточно вставить его в привод для чтения (записи) CD (DVD)-носителей информации и выполнить следующую команду:

```
# apt-cdrom add
```

В случае, если при установке ОС Альт 8 СП был выбран профиль «Рабочая станция», то записи для cdrom в файле /etc/fstab не будет. В данном случае потребуется примонтировать носитель информации вручную:

```
# mount /dev/cdrom /media/ALTLinux
```

Затем использовать команду добавления носителя с дополнительным ключом:

```
# apt-cdrom add -m
```

После этого в sources.list появится запись о подключенном диске примерно такого вида:

```
rpm cdrom:[ ALT 8 SP Workstation]/ ALTLinux main
```

После того как список репозиториев в sources.list будет отредактирован, необходимо обновить локальную базу данных APT о доступных пакетах, выполнив команду:

```
# apt-get update
```

В случае если в sources.list присутствует репозиторий, содержимое которого может изменяться, как происходит с любым постоянно разрабатываемым репозиторием, в частности, обновлений по безопасности (updates), то прежде чем

работать с APT, необходимо синхронизировать локальную базу данных с удаленным сервером:

```
# apt-get update
```

Локальная база данных создается заново каждый раз, когда в репозитории происходит изменение: добавление, удаление или переименование пакета. Для репозиториев, находящихся на извлекаемых носителях информации и подключенных командой `apt-cdrom add`, синхронизация производится единожды в момент подключения.

При выборе пакетов для установки APT руководствуется всеми доступными репозиториями вне зависимости от способа доступа к ним. Таким образом, если в репозитории, доступном по сети Интернет, обнаружена более новая версия программы, чем на CD (DVD)-носителе информации, APT начнет загружать данный пакет по сети.

7.2.2. Обновление информации о репозиториях

Практически любое действие с системой apt начинается с обновления данных от активированных источников. Список источников необходимо обновлять при поиске новой версии пакета, установке пакетов или обновлении установленных пакетов новыми версиями.

Обновление данных осуществляется командой:

```
# apt-get update
```

Программа загрузит данные с активированных источников в свой кеш.

После выполнения этой команды, apt обновит свой кеш новой информацией.

7.2.3. Поиск пакетов

Утилита apt-cache предназначена для поиска программных пакетов, в репозитории, и позволяет искать не только по имени пакета, но и по его описанию.

Команда `apt-cache search <подстрока>` позволяет найти все пакеты, в именах или описании которых присутствует указанная подстрока. Пример поиска может выглядеть следующим образом:

```
$ apt-cache search ^gimp
gimp - The GNU Image Manipulation Program
```

```
libgimp - GIMP libraries
libgimp-devel - GIMP plugin and extension development kit
gimp-help-en - English help files for the GIMP
gimp-help-ru - Russian help files for the GIMP
gimp-plugin-separateplus - Improved version of the CMYK
Separation plug-in [...]
gimp-script-ISONoiseReduction - Gimp script for reducing sensor
noise [...]
gimp-plugin-gutenprint - GIMP plug-in for gutenprint
gimp-plugin-ufraw - GIMP plugin for opening and converting RAW
files [...]
```

Символ «^» в поисковом выражении, указывает на то, что необходимо найти совпадения только в начале строки (в данном случае – в начале имени пакета).

Для того чтобы подробнее узнать о каждом из найденных пакетов и прочитать его описание, можно воспользоваться командой `apt-cache show`, которая покажет информацию о пакете из репозитория:

```
$ apt-cache show gimp-help-ru
Package: gimp-help-ru
Section: Graphics
Installed Size: 37095561
Maintainer: Alexey Tourbin <at@altlinux.org>
Version: 2.6.1-alt2
Pre-Depends: rpmlib(PayloadIsLzma)
Provides: gimp-help-ru (= 2.6.1-alt2)
Obsoletes: gimp-help-common (< 2.6.1-alt2)
Architecture: noarch
Size: 28561160
MD5Sum: 0802d8f5ec1f78af6a4a19005af4e37d
Filename: gimp-help-ru-2.6.1-alt2.noarch.rpm
Description: Russian help files for the GIMP
Russian help files for the GIMP.
```

apt-cache позволяет осуществлять поиск и по русскому слову, однако в этом случае будут найдены только те пакеты, у которых есть описание на русском языке.

7.2.4. Управление установкой (инсталляцией) компонентов программного обеспечения

Установку пакетов может производить только администратор.

ВНИМАНИЕ!

Обновление пакетов выполняется при отсутствии нарушений целостности системы. Проверка целостности системы выполняется:

1) с помощью команды:
   ```
   # integalert
   ```
   При отсутствии изменений вывод команды: `integrity check OK`
2) или просмотр записей osec в системном журнале с помощью команды:
   ```
   # journalctl | grep osec
   ```
   При отсутствии изменений в записях журнала присутствует: `No changes[osec]`

ВНИМАНИЕ!

Если в системе инициализирована система контроля целостности ima-evm (должна быть инициализирована) то установка/обновление пакетов должно происходить посредством команды `updater-start` (см. п. 7.2.4.1) или штатным методом с использованием команды `integrity-applier` (см. п. 7.2.4.2).

Если система контроля целостности не используется, то обновление пакетов необходимо производить в следующем порядке:

1) если используется `control++` (черные/белые списки), необходимо выключить черные/белые списки, выполнив сброс текущего режима (просмотреть установленный режим можно, выполнив команду `control++ status`):
   ```
   # control++ reset
   ```
2) установить пакеты/обновить систему при помощи apt-get;
3) включить черные/белые списки, выполнив команду (в зависимости от вывода в пункте 1):
   ```
   # control++ blacklist
   ```
   или
   ```
   # control++ wl
   ```
4) выполнить команду:
   ```
   # integalert fix
   ```

7.2.4.1. Команда updater-start

Для того чтобы система сохранила все настройки безопасности для установки/обновления пакетов может использоваться команда `updater-start` (из пакета `updater`).

В результате запуска данной команды будет обновлена система и ядро системы, а также включена система контроля целостности ima-evm. Необходимо дождаться завершения работы команды (система будет несколько раз перезагружена).

Примечание. Выполнение команды может занять довольно продолжительное время (время зависит от количества установленных в системе файлов).

Примечание. Если после отработки команды updater-start не запускается сервис auditd, необходимо переименовать/удалить старый журнал аудита (`/var/log/audit/audit.log`) и потом выполнить команду `systemctl start auditd`:

```
# mv /var/log/audit/audit.log /var/log/audit/audit.log_old
# systemctl start auditd
```

Команда `updater-start` также запускает скрипты из `/etc/updater.d/*` с параметром `remove` перед установкой пакетов и их же с параметром `apply` после.

В частности, если используется `control++` со списками, то в `/etc/updater.d/` нужно положить скрипт, вызывающий `control++` и снимающий списки доустановки пакетов и устанавливающий их после установки. Последовательность действий:

1) в каталоге `/etc/updater.d` создать файл (с произвольным названием) с содержимым:

```
#!/bin/bash
if [ "$1" == "remove" ] ;
 then
  control++ reset
 fi
if [ "$1" == "apply" ] ;
 then
 control++ blacklist
 fi
```

2) сделать этот файл исполняемым:

```
# chmod +x /etc/updater.d/<имя_файла>
```

3) запустить обновление: `# updater-start`

4) переименовать файл записи аудита `/var/log/audit/audit.log`:

```
# mv /var/log/audit/audit.log /var/log/audit/audit_old.log
```

5) выполнить запуск аудита: `# service auditd start`

7.2.4.2. Команда integrity-applier

Для того чтобы система сохранила все настройки безопасности установку/обновление пакетов необходимо производить в следующем порядке:

1) установить пакеты/обновить систему при помощи apt-get;
2) выполнить команду для инициализации контроля целостности:

```
# /usr/bin/integrity-applier
```

3) дождаться завершения работы команды (система будет перезагружена четыре раза);
4) переименовать файл записи аудита `/var/log/audit/audit.log`:

```
# mv /var/log/audit/audit.log /var/log/audit/audit_old.log
```

5) выполнить запуск аудита:

```
# service auditd start
```

7.2.5. Установка или обновление пакета командой apt

Установка пакета с помощью APT выполняется командой:

```
# apt-get install имя_пакета
```

Если пакет уже установлен и в подключенном репозитории нет обновлений для данного пакета, система сообщит об уже установленном пакете последней версии. Если в репозитории присутствует более новая версия или новое обновление – программа начнет процесс установки.

`apt-get` позволяет устанавливать в систему другие, пока еще не установленные пакеты, требуемые для работы. Он определяет, какие пакеты необходимо установить, и устанавливает их, пользуясь всеми доступными репозиториями.

Установка пакета gimp командой `apt-get install gimp` приведет к следующему диалогу с APT:

```
# apt-get install gimp
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие дополнительные пакеты будут установлены:
icc-profiles libbabl libgegl libgimp libjavascriptcoregtk2
libopenraw libspiro libwebkitgtk2 libwmf
Следующие НОВЫЕ пакеты будут установлены:
gimp icc-profiles libbabl libgegl libgimp libjavascriptcoregtk2
libopenraw libspiro libweb-kitgtk2 libwmf
```

```
0 будет обновлено, 10 новых установлено, 0 пакетов будет удалено
и 0 не будет обновлено.
Необходимо получить 0B/24,6MB архивов.
После распаковки потребуется дополнительно 105MB дискового
пространства.
Продолжить? [Y/n] y
. . .
Получено 24,6MB за 0s (44,1MB/s).
Совершаем изменения...
Preparing...              ######################## [100%]
1: libbabl                ######################## [ 10%]
2: libwmf                 ######################## [ 20%]
3: libjavascriptcoregtk2  ######################## [ 30%]
4: libwebkitgtk2          ######################## [ 40%]
5: icc-profiles           ######################## [ 50%]
6: libspiro               ######################## [ 60%]
7: libopenraw             ######################## [ 70%]
8: libgegl                ######################## [ 80%]
9: libgimp                ######################## [ 90%]
10: gimp                  ######################## [100%]
Running /usr/lib/rpm/posttrans-filetriggers
Завершено.
```

Команда `apt-get install имя_пакета` используется и для обновления уже установленного пакета или группы пакетов. В этом случае `apt-get` дополнительно проверяет, не обновилась ли версия пакета в репозитории по сравнению с установленным в системе.

При помощи APT можно установить и отдельный бинарный rpm-пакет, не входящий ни в один из репозиториев. Для этого достаточно выполнить команду `apt-get install путь_к_файлу.rpm`. При этом APT проведет стандартную процедуру проверки зависимостей и конфликтов с уже установленными пакетами.

В результате операций с пакетами без использования APT может нарушиться целостность ОС Альт 8 СП, и `apt-get` в таком случае откажется выполнять операции установки, удаления или обновления.

Для восстановления целостности ОС Альт 8 СП необходимо повторить операцию, задав опцию `-f`, заставляющую `apt-get` исправить нарушенные зависимости, удалить или заменить конфликтующие пакеты. Любые действия в этом режиме обязательно требуют подтверждения со стороны пользователя.

При установке пакетов происходит запись в системный журнал вида:

```
apt-get: имя-пакета installed
```

7.2.6. Удаление установленного пакета командой apt

Для удаления пакета используется команда `apt-get remove <имя_пакета>`. Удаление пакета с сохранением его файлов настройки производится при помощи следующей команды:

```
# apt-get remove <значимая_часть_имени_пакета>
```

В случае, если при этом необходимо полностью очистить систему от всех компонент удаляемого пакета, то применяется команда:

```
# apt-get remove --purge <значимая_часть_имени_пакета>
```

Для того чтобы не нарушать целостность системы, будут удалены и все пакеты, зависящие от удаляемого.

В случае удаления с помощью `apt-get` базового компонента системы появится запрос на подтверждение операции:

```
# apt-get remove filesystem
Обработка файловых зависимостей... Завершено
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
basesystem filesystem ppp sudo
Внимание: следующие базовые пакеты будут удалены:
В обычных условиях этого не должно было произойти, надеемся, вы
точно представляете, чего требуете!
basesystem filesystem (по причине basesystem)
0 пакетов будет обновлено, 0 будет добавлено новых, 4 будет
удалено(заменено) и 0 не будет обновлено.
Необходимо получить 0B архивов. После распаковки 588kБ будет
освобождено.
Вы делаете нечто потенциально опасное!
Введите фразу 'Yes, do as I say!' чтобы продолжить.
```

Каждую ситуацию, в которой APT выдает такое сообщение, необходимо рассматривать отдельно. Однако, вероятность того, что после выполнения этой команды система окажется неработоспособной, очень велика.

При удалении пакетов происходит запись в системный журнал вида:

```
apt-get: имя-пакета removed
```

7.2.7. Обновление всех установленных пакетов

Полное обновление всех установленных в системе пакетов производится при помощи команды:

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

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

7.2.8. Обновление ядра и модулей 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 командой: `$ uname -r`

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

Новое ядро загрузится только после перезагрузки системы.

Установка/обновление модулей ядра выполняется командой:

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

Например, для установки модуля VirtualBox, если текущий тип ядра std-def, следует выполнить команду:

```
# apt-get install kernel-modules-virtualbox-std-def
```

7.2.9. Удаление старых версий ядра

После успешной загрузки на обновленном ядре можно удалить старое, выполнив команду: `# remove-old-kernels`

7.2.10. Обновление изолированного окружения (chrooted environment)

Команда `update_chrooted –list` выводит список всех типов модулей для `update_chrooted`, которые установлены в системе:

```
# update_chrooted --list
List of registered types: all conf lib
```

С помощью команды `update_chrooted <имя_типа>` можно выполнить все модули указанного типа.

После изменения общесистемных конфигурационных файлов типа `/etc/resolv.conf`, для того чтобы синхронизировать эти изменения во всех многочисленных `chrooted environments` следует выполнить команду:

```
# update_chrooted conf
```

После изменения системных библиотек следует выполнить команду:

```
# update_chrooted lib
```

Для синхронизации изменений конфигурационных файлов и системных библиотек следует используется команда:

```
# update_chrooted all
```

7.2.11. Проверка подлинности пакетов

Подлинность пакетов при обновлении обеспечивается средствами кодирования, подтверждающих как целостность самих пакетов, так и целостность индексов, описывающих репозитории.

Ключевая информация для проверки подлинности распространяется вместе с дистрибутивом на сертифицированном носителе и защищена от потенциальной подмены при передаче по каналам связи.

Проверить подлинность и целостность пакета можно командой:

```
# rpm –vK имя_пакета
```

7.2.12. Получение уведомлений о выходе обновлений

Системы, имеющие прямой выход в Интернет, могут получать обновления при помощи модуля «Сервер обновлений» Центра управления по сети Интернет из следующего репозитория в соответствии с выбранной веткой для нужного дистрибутива: `http://update.altsp.su/pub/distributions/ALTLinux/c8/`

7.2.13. Обновление систем, не имеющих выхода в Интернет

Для систем, не имеющих прямого выхода в Интернет, рекомендуется установка отдельного сервера обновлений на базе ОС Альт 8 СП, находящегося вне защищенного контура и организация ограниченного доступа к этому серверу. Настройка сервера обновлений осуществляется при помощи модуля «Сервер обновлений» Центра управления.

В случае, когда это невозможно, следует получать ISO образы обновлений по адресу: `http://ftp.altsp.su/pub/distributions/ALTLinux/c8/images`, и доставлять их к обновляемым компьютерам на дисках.

Для добавления диска в качестве источника установки следует воспользоваться командой `apt-cdrom add`.

Модуль alterator-mirror предназначен для зеркалирования репозиториев и публикации их для обновлений рабочих станций и серверов.

Настройка сервера обновлений осуществляется при помощи модуля «Сервер обновлений» ЦУС (рис. 38).

Рис. 38 – Меню «Сервер обновлений»

На этой странице можно выбрать, как часто выполнять закачку пакетов. Также можно выставить время, когда начинать зеркалирование (рис. 39).

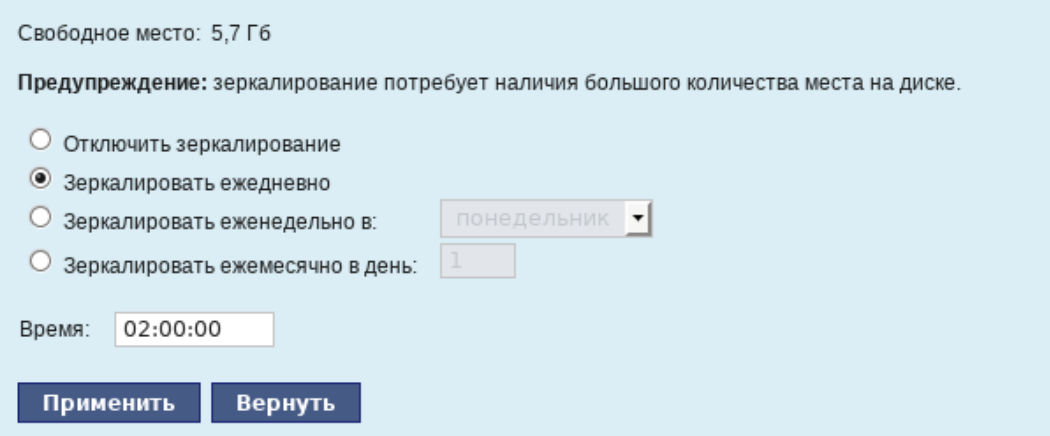

Рис. 39 – Настройка расписания

Так же можно выбрать репозитории, локальные срезы которых необходимы. Далее при нажатии на название репозитория, появляются настройки этого репозитория (рис. 40). Необходимо выбрать источник (сайт, откуда будет скачиваться репозиторий), архитектуру процессора (если их несколько, то стоит выбрать соответствующие).

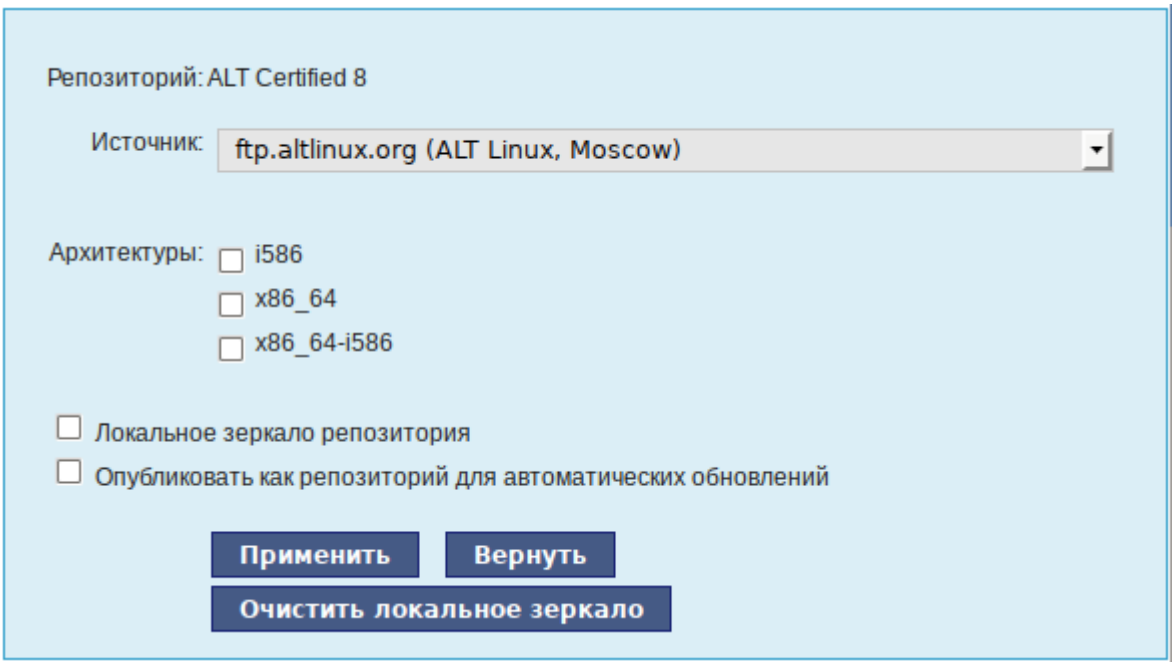

Рис. 40 – Настройки репозитория

Настройка локального репозитория заканчивается нажатием на кнопку «Применить».

Далее необходимо отредактировать файл `/etc/httpd2/conf/sites-available/default.conf`, изменив следующие строки:

```
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
```

Этим серверу apache будет разрешено обрабатывать символические ссылки. Перезапустить apache:

```
# /etc/init.d/httpd2 restart
```

Перейти в папку веб-сервера:

```
cd /var/www/html
```

Создать здесь символическую ссылку на репозиторий:

```
ln -s /srv/public/mirror mirror
```

На клиентских машинах настроить репозитории. Для этого необходимо запустить Synaptic, в параметрах выбрать репозитории. И далее настроить URL доступных репозиториев:

```
http://ваш ip/mirror/
```

Так же со стороны клиентских машин на них необходимо настроить модуль «Обновление системы» отметив в нем «Обновление системы, управляемое сервером» (рис. 41). Источник обновлений вычисляется автоматически (при выбранном режиме «Обновление системы, управляемое сервером» и наличии в локальной сети настроенного сервера). Процесс обновления будет запускаться автоматически согласно заданному расписанию.

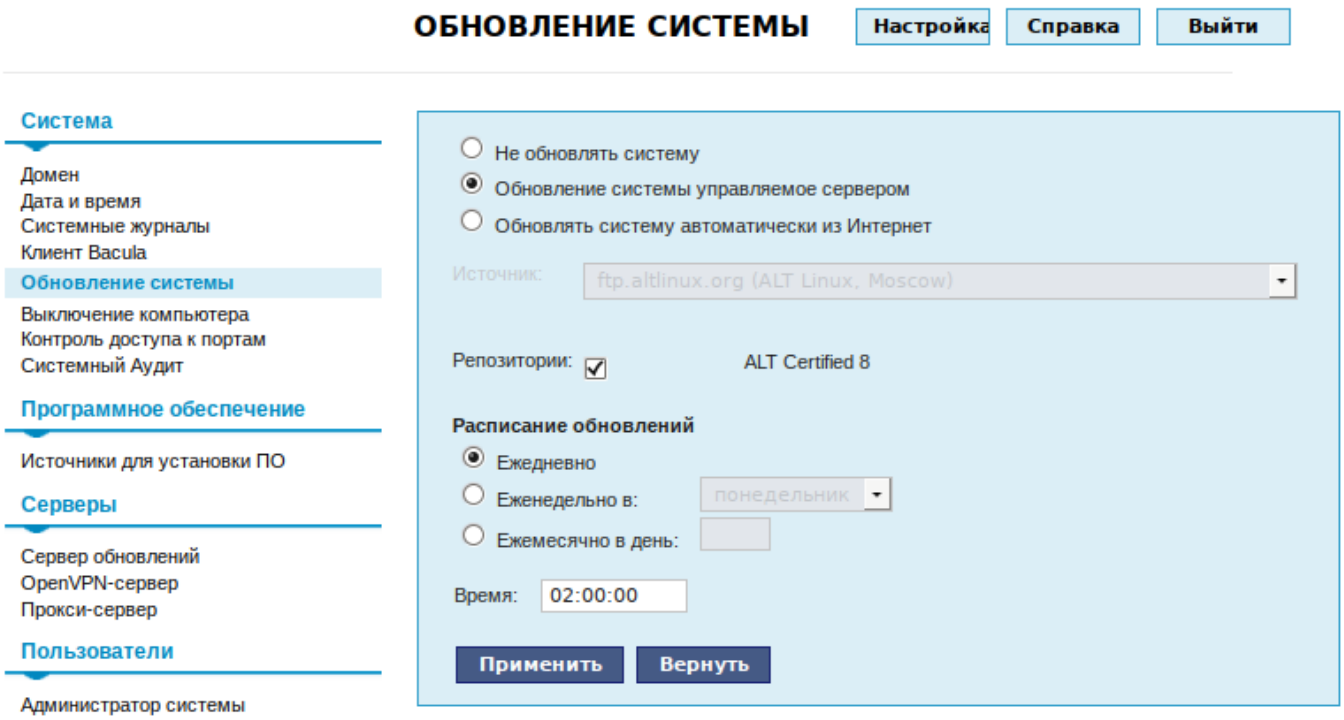

Рис. 41 – Модуль «Обновление системы»

7.3. Работа со смарт-картами

Для настройки работы со смарт-картами необходимо установить дополнительные пакеты:

1) синхронизировать файлы описаний пакетов с источником пакетов, выполнив команду:

```
# apt-get update
```

2) установить пакеты для поддержки программно-аппаратного комплекса электронно-цифровой подписи и хранения ключевой информации «RUTOKEN», выполнив команду:

```
# apt-get install opensc pcsc-lite pam_pkcs11 pcsc-lite-ccid librtpkcs11ecp libopensc lightdm-gtk-greeter-pd openssl-engine_pkcs11
```

7.3.1. Двухфакторная аутентификация

На токене должны присутствовать ключевая пара и сертификат.

Для генерирования ключевой пары на токене и создания самоподписанного сертификата, используя openssl, необходимо выполнить следующие действия (путь зависит от архитектуры):

1) запустить сервис поддержки смарт-карт, выполнив команду:

```
# systemctl start pcscd
```

2) сгенерировать ключевую пару, выполнив команду:

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so --keypairgen
--key-type rsa:2048 -l --id 45
```

3) сгенерировать сертификат в формате PEM:

```
# openssl
OpenSSL> engine dynamic -pre
SO_PATH:/usr/lib64/openssl/engines/libpkcs11.so -pre ID:pkcs11
-pre LIST_ADD:1 -pre LOAD -pre
MODULE_PATH:/usr/lib64/librtpkcs11ecp.so
OpenSSL> req -engine pkcs11 -new -key 45 -keyform engine -x509
-out CA.pem  -text
```

4) конвертировать сертификат из формата PEM в формат CRT:

```
OpenSSL> x509 -in CA.pem -out cert.crt -outform DER
```

5) сохранить сертификат на аутентифицирующий носитель:

```
# pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so -l -y cert
-w cert.crt --id 45
```

Для настройки двухфакторной аутентификации необходимо выполнить следующие действия:

1) отредактировать файл `/etc/security/pam_pkcs11/pam_pkcs11.conf` для установки аутентификации по «RUTOKEN» следующим образом:

   - закомментировать строку `# use_pkcs11_module = opensc;` и добавить строку `use_pkcs11_module = rutoken;`:

     ```
     # use_pkcs11_module = opensc;
     use_pkcs11_module = rutoken;
     ```

   - после строки `use_pkcs11_module = rutoken;` добавить модуль rutoken:

     ```
     use_pkcs11_module = rutoken;
     ```

```
pkcs11_module rutoken {
ca_dir = /etc/security/pam_pkcs11/cacerts;
crl_dir = /etc/security/pam_pkcs11/crls;
module = /usr/lib64/librtpkcs11ecp.so;
cert_policy = subject;
description = "Rutoken ECP";
slot_description = "none";
}
```

2) включить сервисы поддержки смарт-карт, выполнив команды:

```
# systemctl enable pcscd
# systemctl start pcscd
```

3) включить системную аутентификацию по смарт-картам в графическом интерфейсе, выполнив команду:

```
# control system-auth pkcs11
```

4) добавить информацию об удостоверяющем центре на машину (файл о сертификате создан в начальных условиях):

```
cp CA.pem /etc/security/pam_pkcs11/cacerts/
certutil -A -n 'Root CA' -t 'CT,C,C' -a -d /etc/pki/nssdb/ -i
./CA.pem
```

5) добавить информацию о сертификате в домашний каталог пользователя:

```
mkdir /home/user/.eid/
cat CA.pem > /home/user/.eid/authorized_certificates
```

6) для возможности аутентификации по сертификату в консоли необходимо в файл `/etc/pam.d/login` вначале добавить строку:

```
auth       [success=done authinfo_unavail=ignore ignore=ignore
default=die] pam_pkcs11.so
```

7.4. Блокировка удаления открытых файлов

Блокировка возможности удаления открытого файла реализована в модуле ядра AltHA. Использовать этом механизм не рекомендуется.

Для включения механизма блокировки необходимо:

1) передать ядру параметр `altha=1`. Для этого в файле `/boot/boot.conf` к значению переменной `CMDLINE_LINUX_DEFAULT` добавить `altha=1`. Перезагрузить ОС;
2) установить значение переменной `kernel.altha.olock.enabled` равным 1, выполнив команду:

```
# sysctl -w kernel.altha.olock.enabled=1
```

3) переменная `kernel.altha.olock.dirs` должна содержать разделенный двоеточиями список каталогов, например:

```
/var/lib/something:/tmp/something
```

Для изменения значения переменной `kernel.altha.olock.dirs` выполнить команду:

```
# sysctl -w kernel.altha.olock.dirs=список:каталогов
```

При необходимости устанавливать эти переменные автоматически при каждой загрузке ОС, необходимо добавить их в файл `/etc/sysctl.conf`. После редактирования `sysctl.conf` применить изменения, без перезагрузки ОС, можно выполнив команду:

```
# sysctl -p
```

7.5. Настройка разграничения доступа к подключаемым устройствам

7.5.1. Общие сведения

В ОС Альт 8 СП осуществляется разграничение доступа к символьным и блочным устройствам, для которых в каталоге `/dev` создаются файлы устройств. Разграничение выполняется с использованием генерации правил менеджера устройств udev.

П р и м е ч а н и е . При разграничении доступа к устройствам типа видеокарт, либо сетевых карт, названный метод не используется.

Для решения задачи разграничения доступа к устройствам на основе генерации правил менеджера устройств udev в ОС реализованы:

- средства разграничения доступа к устройствам на основе правил udev;
- средства регистрации устройств.

Средства разграничения доступа к устройствам на основе генерации правил udev обеспечивают дискреционное разграничение доступа пользователей к подключаемым, в первую очередь, через интерфейс USB, устройствам (сканерам, съемным накопителям, видеокамерам).

Средства регистрации устройств обеспечивают учет подключаемых устройств и съемных носителей в системе, установку дискреционных атрибутов доступа пользователей к устройствам и создание дополнительных правил доступа к устройству (например, ограничение на подключение устройства только к определенному USB-порту).

7.5.2. Ограничения при помощи правил Udev

Udev – сервис, который подхватывает и конфигурирует внешние устройства, получая уведомления от ядра ОС. Udev гибко настраивается под оборудование и задачи с помощью специальных правил.

Разграничение доступа к устройству осуществляется на основе соответствующего правила для менеджера устройств udev, которое хранится в файле в каталоге `/etc/udev/rules.d`. Файл правил обязательно должен иметь расширение `.rules`.

Далее приведен пример правила для съемного USB-носителя:

```
ENV{ID_SERIAL}=="JetFlash_TS256MJF120_OYLIXNA6", OWNER="user",
GROUP="users"
```

В приведенном примере для съемного USB-носителя с серийным номером `JetFlash_TS256MJF120_OYLIXNA6` разрешено его использование владельцу устройства: пользователю user и пользователям, входящим в группу users.

Типовое правило udev состоит из нескольких пар «ключ – значение» разделенных запятой.

Одни ключи используются для проверки соответствия устройства определенному правилу, в таких ключах используется знак «==» для разделения пары. Следующий пример отражает применение правила только для случая, если значения ключа SUBSYSTEM для этого устройства равно «block»:

```
SUBSYSTEM=="block"
```

Другие ключи используются для указания действия, если все условия соответствия выполняются. Для разделения пар в таких ключах используется знак равно «=».

Например, в случае с `NAME="mydisk"` правило будет выглядеть следующим образом:

```
SUBSYSTEM=="block", ATTR{size}=="1343153213", NAME="mydisk"
```

Это правило выполнится только для устройства подсистемы block и с размером 1343153213 байт.

Для правил udev существуют следующие ключи соответствия:

- SUBSYSTEM – подсистема устройства;
- KERNEL – имя выдаваемое устройству ядром;
- DRIVER – драйвер обслуживающий устройство;
- ATTR – sysfs атрибут устройства;
- SUBSYSTEMS – подсистема родительского устройства.

Для действий используются ключи:

- NAME – установить имя файла устройства;
- SYMLINK – альтернативное имя устройства;
- RUN – выполнить скрипт при подключении устройства;
- GROUP – группа, у которой есть доступ к файлу;
- OWNER – владелец файла устройства;
- MODE – маска прав доступа.

Ключ ATTR позволяет получить информацию об устройстве. Посмотреть все возможные Udev параметры для устройства можно с помощью команды `udevadm`.

Например, для диска `/dev/sda` команда просмотра параметров будет выглядеть следующим образом:

```
$ udevadm info -a -n sda1
```

Для создания файла с правилами нужно выполнить следующую команду:

```
touch /etc/udev/rules.d/usb.rules
```

Правило отключения ручного монтирования, для всех пользователей не из группы «plugdev», которое необходимо добавить в файл `usb.rules`, будет выглядеть следующим образом:

```
BUS=="usb", SUBSYSTEM=="block", KERNEL=="sd*", ACTION=="add", GROUP="plugdev", MODE="660"
```

Правило, которое при подключении USB-устройства запускает скрипт `/etc/udev/usb_on.sh`, и сделает необходимые действия (например, запишет в log-файл необходимую информацию), будет выглядеть следующим образом:

```
ACTION=="add", SUBSYSTEM=="block", ENV{ID_BUS}=="usb|mmc|memstick|ieee1394", RUN+="/bin/bash /etc/udev/usb_on.sh %E{ID_SERIAL_SHORT} %E{ID_MODEL} %E{ID_VENDOR}"
```

где:

- ACTION – отслеживаемое действие, add – подключение устройств, remove – отключение;
- ENV – перечень отслеживаемых устройств по типу;
- RUN – исполняемое действие.

Скрипту `usb_on.sh` udev передает следующие данные:

- %E{ID_SERIAL_SHORT} – серийный номер USB устройства;
- %E{ID_MODEL} – модель USB устройства;
- %E{ID_VENDOR} – производитель USB устройства.

Использование скрипта позволяет выполнять более гибкую настройку правил: можно не только монтировать устройства, но и выполнять другие действия (копировать, менять владельца и так далее). Также допускается задавать тип доступа к информации на носителе, например, «только для чтения».

Далее приводятся примеры оформления других возможных правил для udev:

- отключить все USB-порты:

```
BUS=="usb", OPTIONS+="ignore_device"
```

- отключить все блочные устройства, присоединенные к USB-портам:

```
BUS=="usb", SUBSYSTEM=="block", OPTIONS+="ignore_device"
```

- назначить постоянное имя файлу устройства второго IDE-диска:

```
KERNEL=="sdb", NAME="my_spare"
```

- игнорировать второй USB SCSI/IDE-диск, подключенный по USB:

```
BUS=="usb", KERNEL=="hdb", OPTIONS+="ignore_device"
```

7.5.3. Управление монтированием блочных устройств

При монтировании блочных устройств используется утилита mount, модифицированная для монтирования устройства владельцем или пользователем. В процессе монтирования от имени пользователя ожидается два параметра: конкретное наименование файла устройства и конкретное наименование точки монтирования.

Для предоставления локальным пользователям возможности монтирования ФС съемных накопителей необходимо наличие в файле `/etc/fstab` следующей записи:

```
/dev/sdb1 /media/usb     vfat rw,noauto,user 0 0
```

7.5.4. Настройка ограничений в веб-интерфейсе alterator-ports-access

Настроить ограничения на использование внешних носителей можно в интерфейсе alterator-fbi.

Должны быть установлены пакеты alterator-fbi и alterator-ports-access:

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

Далее необходимо запустить службу ahttpd:

```
# systemctl start ahttpd
```

Открыть в браузере адрес `https://localhost:8080` или `https://ip_address:8080` и ввести пароль администратора. Далее в меню «Система» необходимо выбрать пункт «Контроль доступа к портам» (рис. 42).

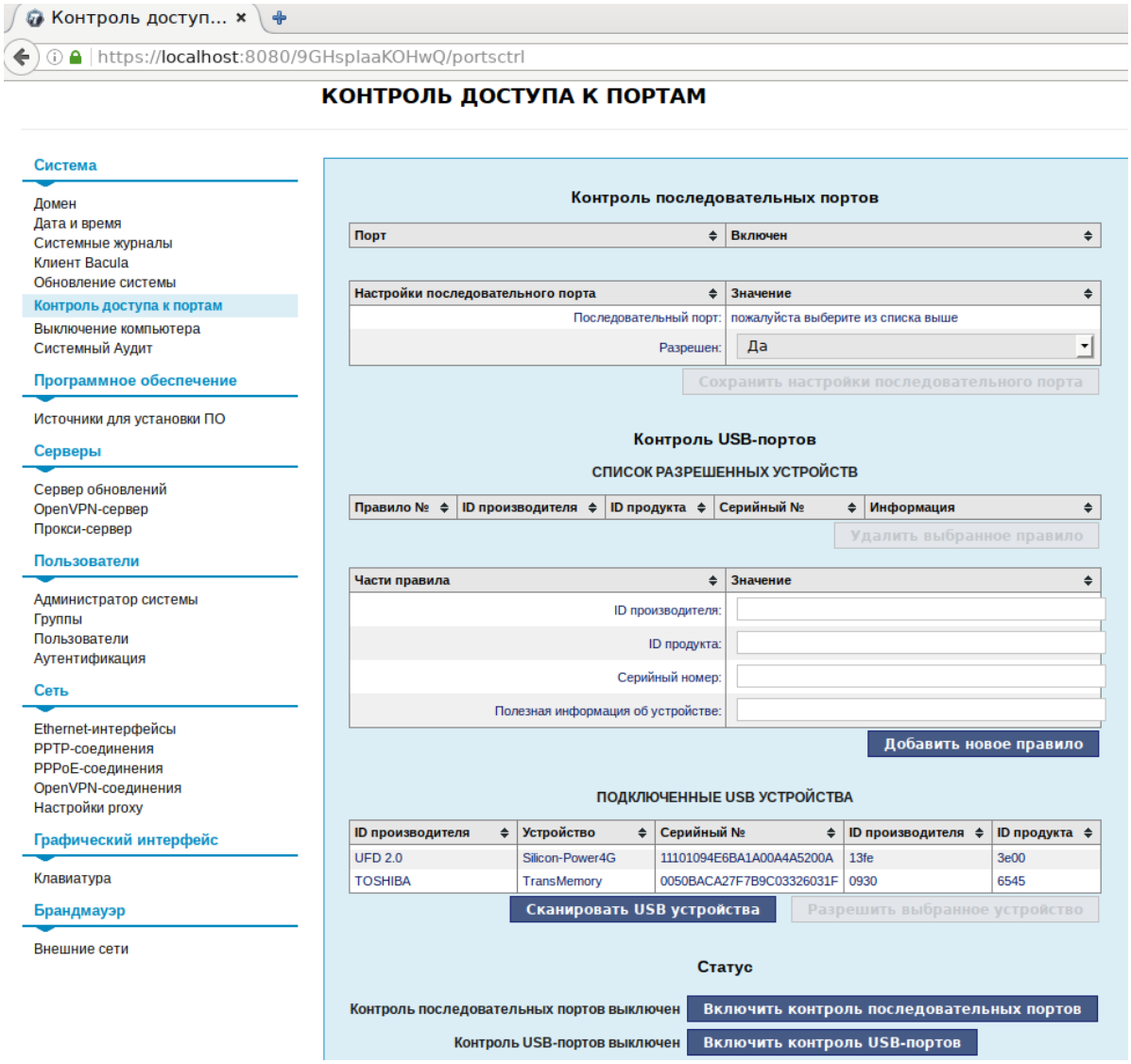

Рис. 42 – Контроль доступа к портам

Для того чтобы отключить поддержку всех USB устройств кроме заданных, необходимо нажать на кнопку «Включить контроль USB-портов».

Для того чтобы добавить USB-устройство в список разрешенных можно ввести HID устройства в поле ID продукта и нажать кнопку «Добавить правило».

Для определения подключенных USB устройств нужно нажать кнопку «Сканировать USB устройства», выделить устройство, которое необходимо разрешить и нажать кнопку «Разрешить выбранное устройство» (рис. 43).

Для исключения устройства из списка разрешенных, необходимо выделить правило, разрешающее данное устройство и нажать кнопку «Удалить выбранное правило».

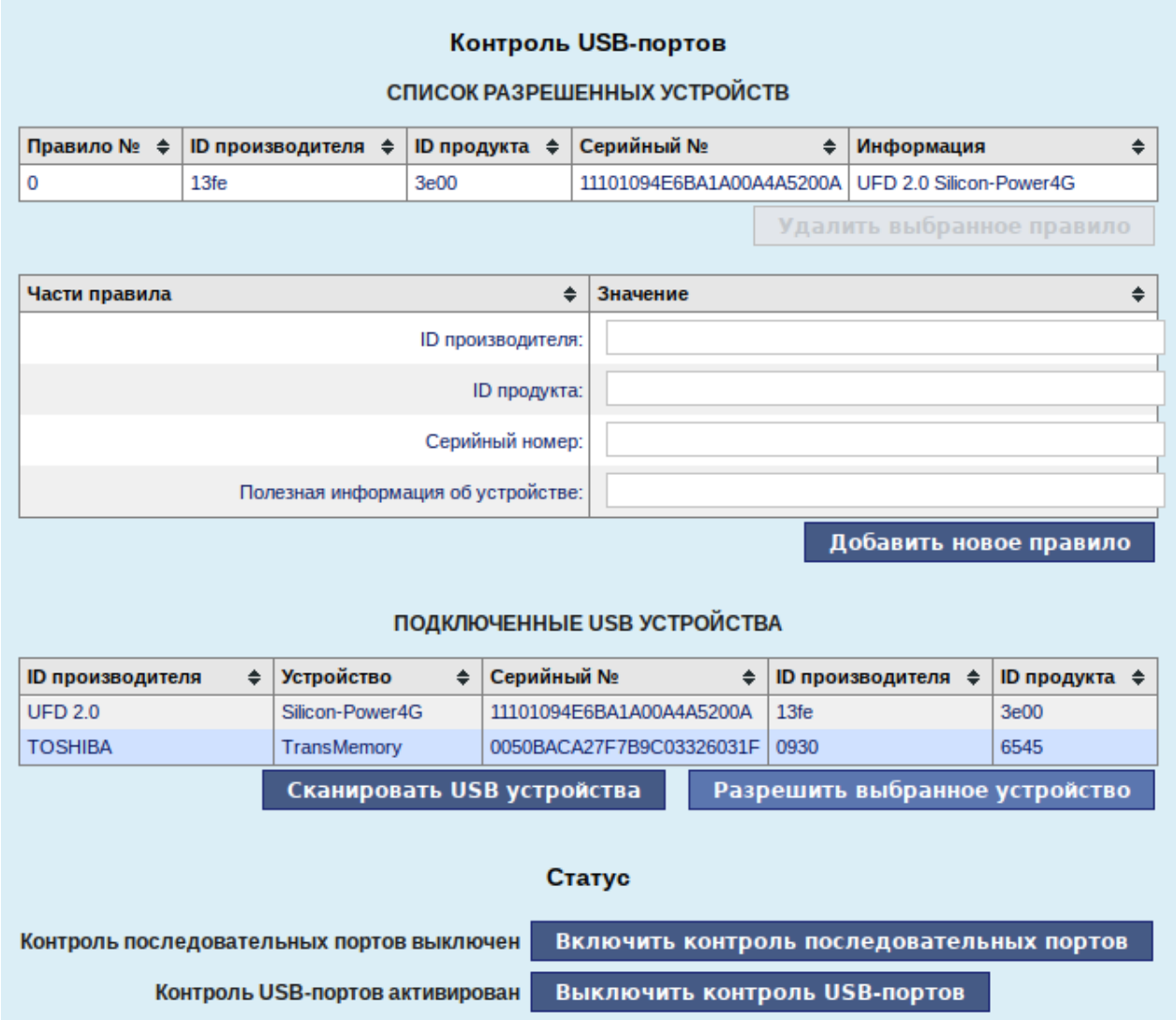

Рис. 43 – Добавление USB устройства в список разрешенных устройств

7.6. Настройка сети с помощью набора пакетов `/etc/net`

Набор пакетов `/etc/net` – это система конфигурации сети в ОС семейства Linux, которая позволяет администратору произвести настройки сети.

7.6.1. Устройство `/etc/net`

`/etc/net` интегрирован в ОС Альт 8 СП в виде пакетов:

- etcnet – базовые сценарии;
- etcnet-full – виртуальный пакет с зависимостями на все пакеты, которые могут использоваться сценариями `/etc/net`, с указанием их точных версий;
- etcnet-defaults-desktop – умолчания для рабочей станции;
- etcnet-defaults-server – умолчания для сервера.

Переменные sysctl в ОС Альт 8 СП конфигурируются в следующих местах: `/etc/sysctl.conf` (глобальные системные), `/etc/sysconfig/network-scripts/sysctl.conf` (общие сетевые в net-scripts), `/etc/net/sysctl.conf` (общие сетевые в `/etc/net`), `/etc/net/ifaces/*/sysctl.conf*` (частные для конкретных интерфейсов или их типов в `/etc/net`).

7.6.1.1. Организация опций `/etc/net` по умолчанию

Методология работы `/etc/net` предусматривает несколько шагов наследования опций, первый из которых – загрузка опций по умолчанию. Для их хранения предназначен каталог `/etc/net/options.d`, из которого будут последовательно прочитаны все файлы. В этом каталоге содержится файл `/etc/net/options.d/00-default`, содержащий значения по умолчанию, а также файл `/etc/net/options.d/50-ALTLinux-server` со специфичными для дистрибутива значениями.

Для изменения набора функций по умолчанию допускается создать файл с еще более высоким номером и определить настройки по умолчанию для своей системы. В результате такого подхода:

- не изменяются файлы с опциями, принадлежащие пакету. Это делает обновление пакета намного более корректным;
- можно легко увидеть, какие опции переопределяются на каждом этапе.

7.6.1.2. Интерфейсы lo, default и unknown

Сразу после установки пакета etcnet в каталоге `/etc/net/ifaces` (в котором хранятся конфигурации интерфейсов) создаются три каталога:

- `lo`;
- `default`;
- `unknown`.

Интерфейс `lo` – стандартная «петля» (loopback), которая должна быть во всякой Linux-системе, поэтому конфигурация для него включена по умолчанию. В остальном он ничем не отличается от любого другого интерфейса и конфигурируется точно так же файлами `options` и `ipv4address`.

Интерфейс `default` – специальный каталог, файлы в котором обрабатываются следующим образом:

- `resolv.conf` – если присутствует, то копируется в `/etc/resolv.conf`;
- `options` – файл опций, читается после опций по умолчанию;
- `options-<вид интерфейса>` – файл содержит опции, специфичные для данного вида интерфейсов. Некоторые из них не обязательны и позволяют использовать особенности данного вида интерфейсов, например, LINKDETECT в options-eth; другие обязательны;
- `sysctl.conf-<вид интерфейса>` – файл с переменными `sysctl`, которые необходимо изменить. Файл `sysctl.conf-dvb` отключает `return path filter`, что нужно в случае асимметричной маршрутизации;
- `iplink-<вид интерфейса>` – файл с командами `iplink`, специфичными для данного вида;
- `selectprofile` – если этот файл исполняемый, то он будет вызван из сценариев `ifup/ifdown, setup/shutdown` для того, чтобы вернуть на стандартном выводе имя профиля, которое необходимо использовать. Это позволяет автоматически переключать профили в зависимости от каких-либо условий. В поставку включен пример сценария:
  `/etc/net/scripts/contrib/selectprofile`
- `fw` – каталог с настройками сетевого экрана по умолчанию.

Интерфейс `unknown` – специальная конфигурация, которая будет использована в том случае, когда `/etc/net` выполняет настройку hotplug-интерфейса, для которого не существует каталога конфигурации. Она будет работать только в том случае, если включена опция `ALLOW_UNKNOWN`.

7.6.1.3. Сценарии конфигурации сети и hotplug-интерфейсы

7.6.1.3.1. Сценарии конфигурации сети

Существует несколько сценариев конфигурации сети.

Первый сценарий – выполнение `service network start` при старте системы или вручную. При этом требуется только сформировать погруппные (потиповые) списки интерфейсов, подлежащих обработке, и последовательно выполнить

требуемые действия. Модули ядра при этом загружаются сценариями `/etc/net`, при этом имена модулей берутся из опции MODULE (в этой опции можно в кавычках перечислить несколько имен, и они будут последовательно загружены). Этот метод часто используется на практике и лучше всего подходит для маршрутизаторов. Преимущество метода в том, что вся необходимая информация сконцентрирована в одном месте – каталоге `/etc/net`. Если опция MODULE не определена, то будет предпринята попытка загрузки по имени интерфейса, подразумевая, что файл `/etc/modules.conf` заполнен правильно.

Второй сценарий – реакция на событие `ifplugd`. В части загрузки модуля этот сценарий не отличается от первого.

Третий сценарий – реакция на появление или исчезновение сменного устройства. Для обработки таких событий предназначены сценарии `/etc/net/scripts/{ifup,ifdown}-removable`, которые вызываются из сценариев пакетов hotplug и pcmcia-cs. Сложность заключается в том, что для сменных PCMCIA-карт вызовы могут дублироваться: для одного и того же события первый раз `ifup-removable` будет вызван из hotplug, второй – из pcmcia-cs. Кроме того, hotplug также реагирует на загрузку модулей ядра для обычных карт PCI и, более того, включает сценарии, которые пытаются загружать модули самостоятельно. В этом контексте `/etc/net` получает слишком много вызовов от hotplug и по умолчанию их игнорирует (`USE_HOTPLUG=no`).

7.6.1.3.2. hotplug-интерфейсы

Для настройки сменной карты в файле options необходимо задать следующий параметр:

```
USE_HOTPLUG=yes
```

После этого `/etc/net` при получении события от hotplug будет автоматически вызывать управляющий модуль устройства при его подключении и выгружать из памяти в случае отсоединения устройства.

П р и м е ч а н и е . Съемные интерфейсы будут пропущены при обычном старте сети, так как их присутствие ОС определяет только после получения вызова от hotplug.

В случае, если необходимо вручную расконфигурировать hotplug-интерфейс до его извлечения, допускается использовать команду `ifdown`. Для повторной конфигурации интерфейса нужно подключить его к ПЭВМ еще раз.

Также существует опция `USE_PCMCIA`. В случае, если события для интерфейса и карты генерирует демон pcmcia-cs, то нужно ее включить. Также, если события генерируются только hotplug, то рекомендуется использовать опцию `USE_HOTPLUG`.

7.6.2. Быстрая настройка сетевого интерфейса стандарта Ethernet

Для настройки сетевого интерфейса стандарта Ethernet следует выполнить следующие шаги:

1) узнать имя сетевого интерфейса:

   ```
   $ ip link show
   ```

   Примечание. Если система не загрузила модуль ядра для сетевой карты автоматически, то его следует загрузить вручную. Для определения модуля можно использовать команду `lspci`. Чтобы загрузить модуль вручную можно использовать команду `modprobe`, например: `modprobe e1000`;

2) создать каталог конфигурации интерфейса `/etc/net/ifaces/<название интерфейса>`, в котором будут храниться файлы с настройками;

3) в каталоге конфигурации сетевого интерфейса создать файл `options` и записать в этот файл строку:

   ```
   MODULE=<имя модуля>
   ```

   На данном этапе работу с файлом `options` можно завершить;

4) выяснить, какой IP-адрес должен быть назначен интерфейсу. Если сетевой интерфейс конфигурируется по DHCP), то в файл `/etc/net/ifaces/enp0s3/options` следует записать строку:

   ```
   BOOTPROTO=dhcp
   ```

   и перейти к шагу 7.

   Примечания:

   1. В ряде случаев в файле options может понадобиться запись:

   ```
   DHCP_HOSTNAME=<имя машины без домена>
   ```

2. В конце файла options необходимо наличие пустой строки.

У сетевого интерфейса существуют два взаимосвязанных атрибута:

- IP-адрес;
- сетевая маска (mask);

5) текущие значение адреса можно просмотреть командой:

```
$ ip address show
```

Вероятнее всего интерфейс-петля lo (loopback) уже сконфигурирован с адресом 127.0.0.1/8 (что эквивалентно IP-адресу 127.0.0.1 и маске подсети 255.0.0.0). /8 означает длину префикса CIDR (Classless InterDomain Routing).

Для задания IP-адреса и маски подсети интерфейса enp0s3 необходимо создать файл `/etc/net/ifaces/enp0s3/ipv4address`, в который следует записать IP-адрес с длиной маски, например:

```
10.0.0.20/24
```

6) выяснить адрес шлюза (маршрут по умолчанию). Создать файл `/etc/net/ifaces/<название интерфейса>/ipv4route`, в который записать строку:

```
default via <ip-шлюза>
```

7) убедиться, что все выполнено правильно, выполнив команду:

```
# systemctl restart network
```

На данном этапе сетевой интерфейс должен быть успешно сконфигурирован.

В случае, если интерфейс был сконфигурирован с помощью DHCP-сервера, но адрес не был назначен, то следует искать сообщение от DHCP-сервера в файле `/var/log/messages`.

7.6.3. Настройка ifplugd

Для корректного использования ifplugd необходимо выполнить команду:

```
# systemctl disable ifplugd
```

Затем назначить переменную `USE_IFPLUGD` в файлах `options` соответствующих интерфейсов (`/etc/net/ifaces/<имя_интерфейса>/options`).

7.6.4. Настройка PPtP-интерфейса и PPPoE-интерфейса

В `/etc/net` введена опция PPPTYPE для единообразной настройки интерфейсов PPP, PPPoE и PPtP.

PPPTYPE может принимать следующие значения:

- dialup – обычный PPP-интерфейс;
- pppoe – интерфейс PPPoE;
- pptp – интерфейс PPtP.

Для `PPPTYPE=pppoe` необходимо в опции HOST указывать имя Ethernet-интерфейса, через который будет производиться работа PPPoE. В дальнейшем, при необходимости, этот интерфейс будет настраиваться автоматически.

Для `PPPTYPE=pptp` необходимо в опции PPTP_SERVER указывать имя хоста или IP-адрес PPtP-сервера, к которому будет производиться подключение. Кроме того, в большинстве случаев необходимо указать в опции REQUIRES интерфейс, через который будет достижим хост, указанный в PPTP_SERVER.

Для настройки PPPoE-соединения необходимо выполнить следующие действия:

1) создать каталог конфигурации PPP-интерфейса, например, `/etc/net/ifaces/ppp5` (допускается задавать имена PPP-интерфейса вида pppN, pppNN, pppNNN, где N – любая цифра от 0 до 9);
2) создать файл с опциями `/etc/net /etc/net/ifaces/ppp5/options` следующего содержания:

```
PPPTYPE=dialup
PPPPERSIST=on
PPPMAXFAIL=0
HOST=enp0s3
```

3) создать файл с опциями `pppd /etc/net/ifaces/ppp5/pppoptions` следующего содержания:

```
defaultroute
mtu 1476
usepeerdns
user ppp_username
```

```
password ppp_password
nomppe
```

7.6.5. Команды сервиса network

У сервиса network имеется ряд команд:

- `start` – запустить все стационарные интерфейсы. hotplug-интерфейсы будут сконфигурированы при поступлении соответствующего вызова от hotplug;
- `startwith <имя профиля>` – старт с указанным именем профиля, а не определенным автоматически;
- `stop` – остановить все стационарные интерфейсы. hotplug-интерфейсы будут расконфигурированы при поступлении соответствующего вызова от hotplug;
- `stopwith <имя профиля>` – стоп с указанным именем профиля, а не определенным автоматически;
- `restart` – эквивалентно «stop» с последующим «start», как и в большинстве других сервисов;
- `restartwith <имя профиля>` – рестарт в контексте указанного профиля, эквивалентно `stopwith <имя профиля>` и `startwith <имя профиля>`;
- `switchto <имя профиля>` – переключение в указанный профиль, эквивалентно `stop` и `startwith <имя профиля>`;
- `reload` – семантически обозначает «актуализировать текущую конфигурацию». Для всех сконфигурированных в настоящий момент интерфейсов будет выполнена реконфигурация при наличии конфигурации;
- `check` – автоматическая проверка конфигурационной базы.

7.6.6. Протоколы конфигурации адресов

Опция `BOOTPROTO` позволяет управлять назначением адресов и маршрутов сетевого интерфейса. Управление выполняется с помощью следующих команд:

- `static` – адреса и маршруты будут взяты из ipv4address/ipv6address и ipv4route/ipv6route;
- `dhcp` – интерфейс будет сконфигурирован по DHCP;
- `dhcp6` – интерфейс будет сконфигурирован по DHCPv6;

- `ipv4ll` – интерфейс будет сконфигурирован с помощью IPv4LL (link-local), ранее известному как ZCIP (zeroconf IP), это значит, что из сети 169.254.0.0/16 будет подобран еще не использованный адрес и назначен на интерфейс.

Существует несколько комбинированных способов:

- `dhcp-static` – если конфигурация по DHCP не удалась, конфигурировать методом static;
- `dhcp6-static` – если конфигурация по DHCPv6 не удалась, конфигурировать методом static;
- `dhcp-ipv4ll` – если конфигурация по DHCP не удалась, конфигурировать методом ipv4ll;
- `dhcp-ipv4ll-static` – если конфигурация по DHCP не удалась, конфигурировать методом ipv4ll.

7.6.7. Расширенные возможности `/etc/net`

7.6.7.1. Несколько IP-адресов или маршрутов на одном интерфейсе

В файл `ipv4address` можно помещать произвольное количество IP-адресов по одному адресу на каждой строке. То же самое относится к статическим маршрутам и файлу `ipv4route`.

`/etc/net` не анализирует содержимое этих файлов, а формирует на основе каждой строки командную строку для утилиты ip. Это означает, что можно помещать в этих файлах произвольные поддерживаемые ip опции и они будут обработаны. Например, в файле `ipv4route` можно поместить строку:

```
10.0.1.0/24 via 10.0.0.253 metric 50 weight 5 table 100
```

7.6.7.2. Зависимости между интерфейсами

У интерфейсов `vlan`, `bond`, `bri`, `teql` входящих в группу зависимых физических, должна быть определена опция HOST со списком интерфейсов, необходимых для инициализации текущего интерфейса. Если хост-интерфейс не сконфигурирован при поднятии зависимого интерфейса, то это будет исправлено.

Кроме обязательной для определенных интерфейсов опции HOST, может быть задана и необязательная для всех остальных интерфейсов опция REQUIRES.

Интерфейсы, перечисленные в этой опции, будут считаться зависимостями текущего интерфейса. Например, по умолчанию попытка сконфигурировать интерфейс А, который зависит от Б и В, приведет сначала к конфигурации Б и В. Аналогично, при расконфигурации Б или В сначала будет расконфигурирован А.

Зависимость одного интерфейса от другого не всегда формальна. Например, в сценарии ifup-pre одного интерфейса может использоваться команда, которая потребует разрешения DNS-имени, которое может быть разрешено только с помощью `resolv.conf`, инсталлируемого другим интерфейсом. Или это может быть PPPoE/PPtP-интерфейс, требующий Ethernet-интерфейс для работы.

7.6.7.3. Пользовательские сценарии post и pre

Существует возможность поместить в каталог конфигурации интерфейса файлы, которые будут выполнены в определенные моменты. Для этого они должны быть исполняемыми и называться следующим образом:

- `ifup-pre` – для выполнения перед конфигурированием интерфейса;
- `ifup-post` – для выполнения после конфигурирования интерфейса. Например, можно запустить почтовую систему;
- `ifdown-pre` – для выполнения перед расконфигурированием интерфейса. Например, можно остановить почтовую систему;
- `ifdown-post` – для выполнения после расконфигурирования интерфейса.

7.6.7.4. Управление канальными параметрами интерфейсов

Если поместить в конфигурационный каталог интерфейса файл `iplink`, в котором в каждой строке будут записаны команды режима `link` утилиты ip, то они будут выполнены при конфигурации интерфейса.

Например, если необходимо, чтобы интерфейс enp3s0 имел MAC-адрес `aa:bb:cc:dd:ee:ff` и MTU 200 байт, то в файл `/etc/net/ifaces/enp3s0/iplink` нужно поместить следующее:

```
address aa:bb:cc:dd:ee:ff
mtu 200
```

7.6.7.5. Управление физическими параметрами интерфейсов

Если поместить в конфигурационный каталог интерфейса файл `ethtool`, в котором будет строка с параметрами программы ethtool, то она будет выполнена при конфигурации интерфейса.

Например, если есть необходимость, чтобы интерфейс enp3s0 имел скорость 10 Мбит/с и авто-согласование скорости было отключено, то в файл `/etc/net/ifaces/enp3s0/ethtool` нужно поместить следующую строку:

```
speed 10 autoneg off
```

7.6.7.6. Настройка Ethernet-моста

Etcnet использует утилиту brctl для настройки Ethernet-моста (далее – моста). В случае если интерфейсы, входящие в состав моста, единственные физически подключенные, и настройка моста происходит с удаленного узла через эти интерфейсы, то требуется соблюдать осторожность, поскольку эти интерфейсы перестанут быть доступны.

В случае ошибки в конфигурации, потребуется физический доступ к серверу. Для страховки перед перезапуском сервиса network можно открыть еще одну консоль и запустить там, например, команду:

```
sleep 500 &&reboot
```

Для настройки моста необходимо завести каталог `/etc/net/ifaces/<имя_моста>` и создать там файлы со следующими данными:

```
- brctl:
 stp AUTO on
- ipv4address:
 192.168.100.200/24
- options:
 TYPE=bri
 HOST='enp0s3 tap0'
 BOOTPROTO=static
```

Содержимое файла `brctl` передается утилите brctl. AUTO означает, что скрипт `setup-bri` самостоятельно определит имя bridge-интерфейса.

IP-адрес для интерфейса, будет взят из `ipv4address`.

В опции HOST файла `options` нужно указать те интерфейсы, которые будут входить в мост. Если в него будут входить интерфейсы, которые до этого имели ip-адрес (например, enp0s3), то этот адрес должен быть удален (например, можно закомментировать содержимое файла `ifaces/enp0s3/ipv4address`).

При старте сети сначала поднимаются интерфейсы, входящие в мост, затем сам мост (автоматически). Для назначения адреса мосту можно так же использовать DHCP (`BOOTPROTO=dhcp`).

7.6.7.7. Настройка VLAN

Для настройки 802.1q VLAN (например, id 4094 на enp0s3) следует, создав каталог `ifaces/enp0s3.4094`, поместить в него файлы со следующим содержимым:

```
- ipv4address:
 192.168.100.200/24
- options:
 TYPE=vlan
 HOST=enp0s3
 VID=4094
 BOOTPROTO=static
```

Содержимое переменных HOST и VID будет передано утилите vconfig; использование файла `vlantab` необязательно (и не рекомендуется по причине невозможности использовать ifup для отдельного интерфейса).

Следует обратить внимание, что 4094 является верхней допустимой границей идентификатора валидного VLAN, а 4095 используется технически в процессе отбрасывания трафика по неверным VLAN. (следует отметить, что это не ограничение Linux: в стандарте под VID отведено 12 бит).

Для настройки Q-in-Q интерфейса, например, `enp0s3.123.513` (дважды тегированный трафик: внешняя метка – 123, внутренняя – 513) следует файл `options` в каталоге `ifaces/enp0s3.123.513` заполнить следующим образом:

```
TYPE=vlan
HOST=enp0s3.123     # «родительский» интерфейс;
VID=513
VLAN_REORDER_HDR=0
```

```
BOOTPROTO=static
```

Родительский интерфейс должен быть сконфигурирован (можно с или без BOOTPROTO, с или без ipv4address).

Таким образом, можно каскадировать интерфейсы как «угодно глубоко» (Q-in-Q-in-Q-in-Q...). Необходимо только учитывать, что длина имени интерфейса ограничена (16-ю символами).

7.6.7.8. Настройка tun/tap интерфейса

Etcnet поддерживает простое создание интерфейсов типа tun/tap. Это виртуальный тип интерфейсов для передачи пакетов между ядром и программами, который не передает данных через физические устройства. tun – это интерфейс типа point-to-point, работающий с кадрами IP. tap – интерфейс типа ethernet, работающий с кадрами ethernet.

Потребуется использование утилиты tunctl, находящейся в одноименном пакете. Пусть требуется создать и настроить tun/tap интерфейс, например, с именем tap0. Для этого необходимо:

1) создать каталог интерфейса `/etc/net/ifaces/tap0`;
2) создать в каталоге интерфейса `/etc/net/ifaces/tap0` файл настройки options со следующим содержанием:

```
TYPE=tuntap
TUNTAP_USER=combr
```

TUNTAP_USER – аккаунт или цифровой id пользователя, которому будут даны права на использование интерфейса tap0 (устройство `/dev/net/tun`). Этот параметр будет передан утилите tunctl как аргумент опции `-u`.

Для создания интерфейса через `/dev/net/tun` требуется привилегия CAP_NET_ADMIN. В общем случае, данная привилегия назначена только для учетной записи root, и обычный пользователь, имеющий доступ к `/dev/net/tun`, может использовать только уже созданные интерфейсы, к которым разрешен доступ для его UID.

7.6.7.8.1. Настройка и использование IP-туннелей

IP-туннели – средство, позволяющее улучшить IP-сети. Поддерживаются IP-туннели трех видов: IPIP, GRE и SIT.

Каждый вид туннеля по степени сложности организации предназначен для решения задач разных уровней:

- туннели IPIP – самые простые;
- туннели SIT предназначены для транспортировки пакетов IPv6 через сети IPv4;
- туннели GRE (general incapsulation) обычно используются в маршрутизаторах Cisco.

По туннелям типа GRE могут передаваться «broadcast» и «multicast» пакеты. Кроме того, эти туннели поддерживают контрольные суммы и контроль упорядоченности пакетов. Также GRE-туннели обладают опциональным атрибутом key в виде произвольного 4-байтового числа, который позволяет конфигурировать несколько GRE туннелей между одной парой IP-адресов несущей сети (в отличие от IPIP-туннелей, с которыми это невозможно).

Тип туннеля определяется опцией TUNTYPE (ipip, gre, sit). По умолчанию `TUNTYPE=ipip`. Кроме типа туннеля для конфигурации всегда требуется адрес удаленного хоста и почти всегда – локальный адрес. Эти адреса определяются опциями TUNREMOTE и TUNLOCAL соответственно. В некоторых случаях локальный адрес можно не указывать. В этом случае опция TUNLOCAL все равно обязательна, но принимает значение any. Не забудьте назначить туннельному интерфейсу адреса и маршруты в соответствующих файлах.

Далее, в качестве примера, выполняется конфигурация GRE-туннеля между 10.0.1.2 и 10.0.2.3 с двумя ключами для исходящих и входящих пакетов, проверкой очередности пакетов, TTL-8 и вычислением контрольных сумм. Туннель должен использовать только определенный интерфейс. Пусть имя создаваемого туннеля будет mytunnel.

Необходимо сделать следующие операции:

1) создать каталог туннеля `/etc/net/ifaces/mytunnel`;

2) создать в каталоге туннеля файл настроек `options` `/etc/net/ifaces/mytunnel/options`;

3) отредактировать файл настроек `options`:

```
TYPE=iptun
TUNTYPE=gre
TUNLOCAL=10.0.1.2
TUNREMOTE=10.0.2.3
TUNTTL=8
HOST=enp0s3
TUNOPTIONS='seq ikey 2020 okey 2030 csum'
```

При настройке VPN-подключения часто не учитывают, что при использовании опции `pppd 'defaultroute'` маршрут по-умолчанию после подключения будет изменен. При этом, если VPN-сервер находится в другой, отличной от клиента, сети, то после подключения (и изменения маршрута по-умолчанию) VPN-сервер становится недоступным, следовательно, недоступными становятся все внешние адреса, и подключение, как правило, прекращается по тайм-ауту.

Решением служит указание отдельного маршрута на VPN-сервер (или его сеть). Для этого необходимо прописать (в примере – для маршрута через enp0s3) в `/etc/net/ifaces/enp0s3/ipv4route` строку вида:

```
10.0.1.0/24 via 10.0.0.1
```

В данном примере подразумевается, что VPN-сервер находится в сети 10.0.1.0/24 (например, имеет адрес 10.0.1.1), клиент – в сети 10.0.0.0/24 (и имеет адрес, например, 10.0.0.10), а маршрутизатор имеет адрес 10.0.0.1.

Теперь, при использовании опции `'defaulroute'` для pppd (которая указывает, что необходимо изменить на вновь созданное подключение маршрут по умолчанию), даже после замены маршрута по умолчанию новым, сеть 10.0.1.0, в которой в нашем примере и находится VPN-сервер, останется доступной.

Как более точечный вариант можно использовать скрипты `ifup-pre` и `ifdown-post` в каталоге конфигурируемого PPP-интерфейса.

Например:

```
#!/bin/sh
```

```
# sample /etc/net/ifaces/ppp0/ifup-pre; replace variables
yourself
ip route add VPN_SERVER via DEF_GW
#!/bin/sh
# sample /etc/net/ifaces/ppp0/ifdown-post; replace variables
yourself
ip route del VPN_SERVER via DEF_GW
```

Далее необходимо подставить нужные IP-адреса вместо VPN_SERVER и DEF_GW (не сеть, где VPN-сервер, а ее /32 префикс CIDR) и выполнить команду:

```
chmod +x ifup-pre ifdown-post
```

7.6.7.9. Сложная маршрутизация

Под «сложной маршрутизацией» подразумевается наличие нескольких таблиц маршрутизации. Для их использования необходимо сконфигурировать правила ядра. В правилах по умолчанию можно увидеть следующее:

```
# ip rule show
0: from all lookup local
32766: from all lookup main
32767: from all lookup default
```

Для настройки «сложной маршрутизации» необходимо выполнить следующие операции:

1) сами таблицы определены в файле `/etc/iproute2/rt_tables`. Для создания конфигурации «сложной маршрутизации» необходимо вначале «создать» нужные таблицы в этом файле (если хотите использовать имена таблиц, а не числа);
2) необходимо заполнить таблицы. В конфигурационном каталоге интерфейса в файле ipv4route необходимо добавить маршрутные записи, не забывая указать tableXX. Важно учитывать, что если строка начинается с режима iproute (add, del, replace, append, change), то по умолчанию будет использован режим DEFAULT_IPV4ROUTE_CMD (append);
3) определить правила в файле `ipv4rule`. Если строка не начинается с операции `del` или `add`, то нужный режим будет подставлен автоматически. Это подходит для тех случаев, когда при «поднятии» интерфейса

необходимо добавить правила, а при «опускании» – удалить. Возможность указывать `del` или `add` реализована для обратных случаев: если при «поднятии» интерфейса необходимо удалить правила, а при «опускании» – добавить. В этом случае `add` и `del` будут в нужный момент автоматически заменены на `del` и `add`.

7.6.7.10. Простое переключение маршрутов

При необходимости настроить второй маршрут по умолчанию через беспроводной интерфейс, в обход работы основного проводного сетевого интерфейса, но с меньшей метрикой, чем у проводного интерфейса используется простое переключение маршрутов.

В этом случае при настройке Wi-Fi маршрут настроится по умолчанию для ethernet-интерфейса файл настроек `/etc/net/ifaces/enp0s3/ipv4route` будет следующим:

```
default via 192.168.3.254 metric 10
```

Для Wi-Fi-интерфейса файл настроек `/etc/net/ifaces/wlp2s0/ipv4route` таким:

```
 default via 192.168.123.1 metric 5
```

7.6.7.11. Настройка Wi-Fi

Большинство беспроводных интерфейсов сейчас представлено системе как интерфейсы Ethernet. Соответственно беспроводный интерфейс будет иметь `TYPE=eth`. Чтобы интерфейс нормально функционировал, необходимо кроме загрузки модуля с параметрами, воспользоваться утилитами iwconfig из пакета wireless-tools или wpa_supplicant из такого же пакета.

Для автоматического запуска поместите в конфигурационный каталог интерфейса файл `iwconfig` с командами `iwconfig` или файл `wpa_supplicant.conf` с конфигурацией `wpa_supplicant`.

Пример конфигурации:

Файл `/etc/net/ifaces/wlp2s0/options`:

```
TYPE=eth
MODULE=ndiswrapper
```

```
NEVER_RMMOD=yes
BOOTPROTO=dhcp
USE_HOTPLUG=no
ONBOOT=no
```

Файл `/etc/net/ifaces/wlp2s0/iwconfig`:

```
essid default
#key bababababa
```

Пример использования etcnet для настройки беспроводной сети:

- файл `/etc/net/ifaces/wlp2s0/options`:

```
TYPE=eth
USE_HOTPLUG=NO
BOOTPROTO=static
module=ipw2200
WPA_DRIVER=wext
```

- файл `/etc/net/ifaces/enp0s3/iwconfig`:

```
essid homenet
mode 1
ap 00:11:D8:22:AD:0D
channel 3
rate 11M
```

- файл `/etc/net/ifaces/enp0s3/wpa_supplicant.conf`:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=1
ap_scan=1
fast_reauth=1
network={
   ssid="homenet"
   bssid=00:11:D8:22:AD:0D
   proto=WPA
   key_mgmt=WPA-PSK
   pairwise=CCMP TKIP
   group=TKIP
   psk="this is my mega secret password string to wpa
```

```
supplicant"
   priority=2
}
```

7.6.7.12. Использование автодополнения в `sysctl.conf`

В конфигурационном каталоге интерфейса может находиться файл `sysctl.conf`, в котором можно перечислить переменные sysctl. Переменные могут быть как общесистемными, так и относящимися к интерфейсу. Естественно, запись в `sysctl.conf` настроек вида `net.ipv4.conf.enp0s3.log_martians = 1` достаточно неудобна, а при переименовании интерфейса велик риск не отредактировать файл `sysctl.conf` соответствующим образом.

Эта проблема решается следующим способом: производится запись в файл только имени переменной и значение, а система `/etc/net` сама найдет путь к этой переменной и вызовет sysctl с полным именем.

Пример содержания файла `sysctl.conf`:

```
log_martians=1
rp_filter=1
```

7.6.7.13. Подключение к Wi-Fi с сертификатом на аппаратном токене

Для беспроводного подключения в корпоративных сетях могут использоваться сертификаты, записанные на аппаратном токене, например, Aladdin eToken. Для настройки такого подключения необходимо использовать `/etc/net/ifaces/wlp2s0/wpa_supplicant.conf`:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
#eapol_version=1
#ap_scan=2
#fast_reauth=1
pkcs11_engine_path=/usr/lib/openssl/engines/engine_pkcs11.so
pkcs11_module_path=/usr/lib/libeTPkcs11.so
update_config=0
network={
     ssid="test"
     key_mgmt=WPA-EAP
```

```
        pairwise=CCMP TKIP
        group=CCMP TKIP
        eap=TLS
        identity="email@address.ru"
        engine_id="pkcs11"
        key_id="xxxxxxxxx"
        cert_id="xxxxxxxxx"
        engine=1
}
```

где `key_id` и `cerd_id` взяты из вывода команды:

```
# pkcs11-tool --module /usr/lib/libeTPkcs11.so -O -l
```

Используются оригинальные драйвера Aladdin – pkiclient-5.00.28-0, и пакет openssl-engine_pkcs11-0.1.5-alt1.

7.6.7.14. Профили конфигурации

7.6.7.14.1. Определение профилей

Профиль – именованный вариант конфигурации, в той или иной степени изменяющий базовую конфигурацию системы. Профили могут быть применены, например, для конфигурации ноутбука в разных сетевых окружениях, или при подготовке новой или тестовой конфигурации с возможностью быстрого возврата к старой. Практически профили реализуются следующим образом: для какого-либо из файлов, составляющих общесистемную конфигурацию или конфигурацию интерфейса, создается альтернативный вариант, который отличается добавлением в конце названия файла знака «#» и имени профиля.

Например, пусть единственное отличие между профилями заключается в том, какой модуль ядра будет загружен для интерфейса enp0s3. В этом случае файл `/etc/net/ifaces/enp0s3/options` необходимо скопировать в `/etc/net/ifaces/enp0s3/options#profile1` и изменить значение переменной MODULE в одном из них. Далее при использовании конфигурации по умолчанию будет использован файл `options`, а при использовании профиля profile1 – файл `options#profile1`.

Профили могут использоваться также и для отключения каких-то параметров конфигурации. Например, если используется файл `ipv4route` для установки маршрутов для интерфейса, то можно создать файл нулевого размера `ipv4route#profile2`, чтобы при использовании профиля profile2 никаких маршрутов не конфигурировалось.

7.6.7.14.2. Выбор профиля по умолчанию

Если требуется, чтобы определенный профиль конфигурации использовался по умолчанию, то необходимо записать его название в файл `/etc/net/profile`. Этот метод имеет приоритет над параметром ядра `netprofile`. Использование такого способа выбора профиля целесообразно, когда переключение между конфигурациями происходит реже, чем перезагрузка системы.

7.6.7.14.3. Смена профиля во время работы

Если требуется переконфигурировать сеть без перезагрузки или редактирования файла `/etc/net/profile`, то следует использовать параметры сервиса network. Этот метод имеет приоритет над профилем по умолчанию и профилем, выбранным при загрузке. Целесообразно его использовать, если смена сетевого окружения происходит чаще, чем перезагрузка системы.

7.6.7.14.4. Определение профиля во время конфигурации интерфейса

Если в каталоге конфигурации интерфейса существует исполняемый файл ненулевого размера с именем `selectprofile`, то этот файл будет выполнен и первое слово первой строки его стандартного вывода использовано как имя профиля, которое должно быть использовано для конфигурации данного интерфейса. Этот метод имеет приоритет над всеми остальными методами. Исходной задачей, требующей такого решения, являлось конфигурирование беспроводного интерфейса в зависимости от доступных точек доступа.

Следует учитывать, что число вызовов файла `selectprofile` может меняться в зависимости от контекста и время его выполнения может быть различным, поэтому при написании такого файла следует учитывать, что первым параметром будет являться имя текущего сценария. В настоящее время это могут быть `ifup*`,

`ifdown*`, `setup*` и `shutdown*`. Для приведенного выше примера имеет смысл реагировать только на вызовы из ifup или ifup-common.

7.6.8. Настройка сетевого экрана в `/etc/net`

`/etc/net` содержит поддержку управления сетевым экраном (firewall). В данный момент поддерживается iptables, ip6tables, ipset и ebtables. Реализация основана на группировке таблиц и цепочек в таблицах. Таблицы могут быть только системные, цепочки же, кроме системных, могут быть заданы пользователем.

Ниже приведены файлы и каталоги, используемые для настройки сетевого экрана.

`/etc/net/ifaces/default/fw/options` – файл с настройками сетевого экрана по-умолчанию:

1) FW_TYPE – тип сетевого экрана. Здесь можно указать только iptables, другие типы пока не поддерживаются. Обратите внимание на этот параметр, т. к. по умолчанию он не указан в файле настроек;
2) IPTABLES_HUMAN_SYNTAX – включает или отключает использование поддержки удобочитаемого синтаксиса правил для iptables. Значение: `yes` или `no`;
3) IPTABLES_SYSTEM_CHAINS – список системных цепочек в таблицах. Все цепочки, не указанные здесь, будут автоматически создаваться и удаляться. Значение: названия цепочек (все названия чувствительны к регистру!), разделенные пробелом;
4) IPTABLES_INPUT_POLICY – действие по-умолчанию для пакетов, попадающих в системную цепочку INPUT таблицы filter. Значение: одно из ACCEPT, DROP, QUEUE или RETURN;
5) IPTABLES_FORWARD_POLICY – действие по-умолчанию для пакетов, попадающих в системную цепочку FORWARD таблицы filter. Значение: одно из ACCEPT, DROP, QUEUE или RETURN;
6) IPTABLES_OUTPUT_POLICY – действие по-умолчанию для пакетов, попадающих в системную цепочку OUTPUT таблицы filter. Значение: одно из ACCEPT, DROP, QUEUE или RETURN;

7) IPTABLES_RULE_EMBEDDING – способ добавления нового правила в цепочку. Значение: APPEND или INSERT, что означает добавление в конец списка правил или, соответственно, в начало.

`/etc/net/ifaces/default/fw/iptables/filter`, `/etc/net/ifaces/default/fw/iptables/nat`, `/etc/net/ifaces/default/fw/iptables/mangle` – каталоги, соответствующие таблицам iptables. В каталогах создаются файлы, соответствующие необходимым системным или пользовательским цепочкам, в которых уже и прописываются сами правила iptables.

`/etc/net/ifaces/default/fw/iptables/loadorder`, `/etc/net/ifaces/default/fw/tablename/loadorder` – если такой файл существует и не пуст, то обработка таблиц и (или) цепочек в таблице происходит в том порядке, который указан в файле (по одному значению на строку). Все таблицы и цепочки, которые не указаны, обрабатываться не будут.

`/etc/net/ifaces/default/fw/iptables/modules` – список модулей ядра, которые необходимо загрузить перед запуском сетевого экрана. При остановке эти модули выгружаются.

`/etc/net/ifaces/default/fw/iptables/syntax` – описание замен при использовании удобочитаемого синтаксиса правил iptables.

7.6.8.1. Алгоритм работы сетевого экрана

Алгоритм работы сетевого экрана:

1) при запуске службы network, виртуальный интерфейс default:
   - если опция CONFIG_FW (в файле `/etc/net/ifaces/default/options`) не установлена в yes, то ничего не делает и происходит выход из процедуры запуска сетевого экрана, иначе переходим к следующему пункту;
   - считывается файл настроек `/etc/net/ifaces/default/fw/iptables/options`;

- до настройки любого интерфейса и обработки значений sysctl устанавливаются действия по умолчанию (policy) для системных цепочек таблицы filter;
- считывается файл со списком модулей ядра `/etc/net/ifaces/default/fw/iptables/modules`, и все указанные в нем модули (по одному на строку) загружаются. При отсутствии файла никакие модули не загружаются;
- создаются все пользовательские цепочки во всех таблицах (пользовательскими считаются все цепочки, не указанные в переменной IPTABLES_SYSTEM_CHAINS);
- считывается файл `/etc/net/ifaces/default/fw/iptables/loadorder`, и в указанном в нем порядке происходит обработка таблиц iptables. При отсутствии файла обработка происходит в соответствии с сортировкой названий таблиц по имени;
- считывается файл `/etc/net/ifaces/default/fw/iptables/tablename/loadorder` в каждой обрабатываемой таблице, и происходит обработка и загрузка правил для каждой цепочки в порядке, указанном в файле. При отсутствии файла обработка опять же происходит в соответствии с сортировкой по имени;
- если опция IPTABLES_HUMAN_SYNTAX установлена в yes, то считывается и обрабатывается файл с «синтаксисом» `/etc/net/ifaces/default/fw/iptables/syntax`;
- файл с правилами обрабатывает построчно (одно правило на строку); если указана опция `IPTABLES_HUMAN_SYNTAX`, то правило обрабатывается интерпретатором в соответствии с синтаксисом и превращается в реальные опции для команды iptables, после чего запускается iptables с этими параметрами; иначе правило без обработки передается iptables;

2) при «поднятии» любого интерфейса, кроме default – выполняются все подпункты пункта 1, только все файлы и каталоги ищутся в каталоге текущего интерфейса;
3) при «опускании» любого интерфейса, кроме default – все подпункты пункта 1 выполняются в обратном порядке, все правила удаляются из цепочек в обратном порядке, все модули ядра выгружаются в обратном порядке. Все файлы и каталоги ищутся в каталоге текущего интерфейса;
4) при остановке службы network виртуальный интерфейс default – все подпункты пункта 1 выполняются в обратном порядке, все правила из всех цепочек удаляются командой `iptables -F`, все модули выгружаются в обратном порядке, все пользовательские цепочки удаляются.

Действия по умолчанию (policy) для системных цепочек устанавливается в ACCEPT.

7.6.8.2. Правила для iptables

Правила для iptables можно писать с помощью синтаксиса, подобного синтаксису ipfw и других. Сделано это с помощью простой замены слов на опции iptables. Сами замены описаны в файле `/etc/net/ifaces/default/fw/iptables/syntax` (там также описано некоторое количество вспомогательных слов, так что правила можно писать практически на английском литературном). Синтаксис правила можно совмещать (то есть использовать и заданный в «etcnet» синтаксис, и реальные опции команды iptables).

Во всех правилах нельзя использовать названия цепочки и (или) таблицы. Они будут добавляться автоматически.

В правилах можно использовать любые переменные окружения, выполнять любые команды `shell` (они должны быть указаны в одну строку). Переменная $NAME содержит имя текущего интерфейса. Переменные $IPV4ADDRESS и $IPV6ADDRESS содержат массив IPV4/IPV6 адресов текущего интерфейса (это обычные «bash arrays», можно обращаться к ним по индексу: ${IPV4ADDRESS[2]} или просто $IPV4ADDRESS для первого значения). Для удобства можно

использовать файлы `options`, в которых прописывать какие-либо переменные, к примеру, адреса gateway, ISP, сетей и т. д.

Во всех файлах можно использовать комментарии (строка должна начинаться с символа #).

Нет необходимости копировать все файлы настроек в каталог каждого интерфейса. Сначала будут считаны настройки виртуального интерфейса default, а уже потом у текущего интерфейса, соответственно, можно переопределять только требуемые для настройки параметры.

Описания всех правил в настройках виртуального интерфейса default достаточно для поднятия простого сетевого экрана. При наличии же большого количества правил и интерфейсов есть смысл разделить логически все правила по каждому интерфейсу (опять же, не будет нагружаться процессор без необходимости, если интерфейс, к которому относится много правил, сейчас не «поднят»).

В начале каждого правила можно указать, что с этим правилом делать. Может быть одно из трех значений:

1) `-A` – добавление в конец списка правил (при включенном удобочитаемом таксисе соответствует команде `append`);
2) `-I [num]` – добавление в начало списка правил; если указан необязательный параметр num, то правило будет вставлено в строку правил с таким номером (iptables считает несуществующий номер строки ошибкой и добавляет правило). При включенном удобочитаемом синтаксисе соответствует команде `insert [num]`);
3) `-D` – удаление правила из списка правил (соответственно, при «остановке» интерфейса правило наоборот будет добавлено). При включенном удобочитаемом синтаксисе соответствует команде `delete`.

Если никакое действие не указано, то правила добавляются в цепочку в соответствии со значением переменной IPTABLES_RULE_EMBEDDING.

Если изменяется какое-то правило в конфигурационных файлах при уже загруженных правилах iptables, то для того, чтобы в памяти не остались старые правила, необходимо или выгрузить все правила для текущего интерфейса (если

настраивается для конкретного интерфейса, а не для default) перед изменением файлов или после изменения использовать команду `/etc/net/scripts/contrib/efw default restart` (она полностью удалит все правила, однако, пользовательские цепочки других интерфейсов не будут затронуты), и далее загрузить заново правила для нужного или всех интерфейсов.

7.6.8.3. Примеры

Пример настройки сетевого экрана в etcnet (файл – содержание):

Файл `/etc/net/ifaces/enp0s3/fw/options`:

```
# Our WAN IP address
WAN_IP=5.6.7.8/24
# First net
NET1=1.2.3.0/24
# Second net
NET2=4.3.2.0/24
# Friend net
FRIEND_NET=5.6.7.0/24
```

Файл `/etc/net/ifaces/enp0s3/fw/iptables/filter/INPUT`:

```
accept all from any to $IPV4ADDRESS
jump-to COUNT-CHAIN if marked as 0x11
```

Файл `/etc/net/ifaces/enp0s3/fw/iptables/filter/FORWARD`:

```
jump-to FRIEND-NET if from $FRIEND-NET
append drop tcp from net $NET1 to net NET2
delete drop udp from $NET1 to $NET2
insert reject udp to $WAN_IP
drop icmp to $(somescript.sh)
```

Файл `/etc/net/ifaces/enp0s3/fw/iptables/filter/FRIEND-NET`:

```
policy reject
```

Файл `/etc/net/ifaces/enp0s3/fw/iptables/mangle/PREROUTING`:

```
insert 15 mark tcp as 0x10 if from-iface $NAME and dport is 22
mark tcp as 0x11 if from net $NET1 and from-iface $NAME
```

Файл `/etc/net/ifaces/enp0s3/fw/iptables/nat/POSTROUTING`:

```
snat-to $WAN_IP if marked as 0x10
```

7.6.8.4. Утилиты

В `scripts/contrib` находятся вспомогательные утилиты.

Скрипт `efw` предназначен для ручного управления сетевым экраном.

Синтаксис:

```
efw –ips[et]|[--ipt[ables]|--ip6t[ables]|--ebt[ables]|--no-
ips[et]|--no-ipt[ables]|
--no-ip6t[ables]|--no-ebt[ables]] [iface] [table|settype]
[chain|set] <action> [правило или опции для action]
```

Параметры:

1) `--ipset` – обработать только ipset;
2) `--iptables` – обработать только iptables;
3) `--ip6tables` – обработать только ip6tables;
4) `--ebtables` – обработать только ebtables;
5) `--no-iptables` – обработать все типы за исключением iptables;
6) `--no-ip6tables` – обработать все типы за исключением ip6tables;
7) `--no-ebtables` – обработать все типы за исключением ebtables;
8) `iface` – 'default' (по умолчанию), имя интерфейса или 'all' для всех интерфейсов;
9) `table` – 'mangle' (только для iptables и ip6tables), 'broute' (только для ebtables), 'filter' (по умолчанию), 'nat' или 'all' для всех таблиц;
10) `chain` – системная либо пользовательская цепочка (чувствительно к регистру!) или 'all' для всех цепочек;
11) `action` – 'start', 'stop', 'restart', 'load', 'unload', 'reload', 'flush', 'show|list', 'count|counters', 'rule', 'new|create', 'remove|delete', 'zero', 'policy', 'rename'.

Действия (`action`):

1) `start` – обработать все таблицы и цепочки для заданного интерфейса (даже если задано конкретное имя цепочки либо таблицы);
2) `stop` – обработать все таблицы и цепочки для заданного интерфейса (даже если задано конкретное имя цепочки либо таблицы);
3) `restart` – равносильно сначала 'stop' затем 'start';

4) `load` – загрузить правила для заданного интерфейса, таблицы и цепочки;
5) `unload` – выгрузить правила для заданного интерфейса, таблицы и цепочки;
6) `reload` – равносильно сначала 'unload' затем 'load';
7) `flush` – очистить правила для заданного интерфейса, таблицы и цепочки;
8) `show` – показать правила для заданного интерфейса, таблицы и цепочки;
9) `list` – тоже что и 'show';
10) `count` – показать значения счетчиков для заданной таблицы и цепочки;
11) `counters` – тоже что и 'count';
12) `rule` – разобрать правило и передать его в iptables и (или) ip6tables и (или) ebtables;
13) `new` – создать новую цепочку;
14) `create` – тоже что и 'new';
15) `remove` – удалить цепочку;
16) `delete` – тоже что и 'remove';
17) `zero` – очистить счетчики пакетов и байтов в цепочке;
18) `policy` – задать политику для цепочки;
19) `rename` – переименовать цепочку.

Опции для действий `show` и `list`:

1) `-n` или `numeric` – цифровой вывод IP-адресов, портов и сервисов;
2) `-v` или `verbose` – детальный вывод правил;
3) `-x` или `exact` – не округлять числа;
4) `--line-numbers` или `lines` – показать номера каждой строки.

На данный момент скрипт `efw` «умеет» частично «угадывать» интерфейс, таблицу и цепочку (если их не передали в командной строке) и все действия, кроме `counters`. Так же поддерживается маска «all» для интерфейсов, таблиц и цепочек.

Примеры команд

Выгрузить (flush) все правила из всех цепочек всех таблиц, удалить цепочки, пользователем, выгрузить все загруженные модули:

```
/etc/net/scripts/contrib/efw default stop
```

Выгрузить (путем удаления каждого правила в обратном порядке) все правила из цепочки FORWARD таблицы filter для интерфейса enp0s3:

```
/etc/net/scripts/contrib/efw enp0s3 unload
```

Загрузить все правила для всех цепочек во всех таблицах всех интерфейсов:

```
/etc/net/scripts/contrib/efw all all all load
```

Обработать правило и добавить его во все цепочки таблицы filter:

```
/etc/net/scripts/contrib/efw default filter all rule accept all
from any
```

Если изменяется какое-либо правило в конфигурационных файлах при уже загруженных правилах iptables, то для того, чтобы в памяти не остались старые правила, необходимо:

- вариант 1: выгрузить все правила для текущего интерфейса (если настраивается для конкретного интерфейса, а не `default`) перед изменением файлов;
- вариант 2: после изменения использовать команду `efw default restart` (она полностью удалит все правила, однако, пользовательские цепочки других интерфейсов не будут затронуты), и далее загрузить заново правила для требуемого или всех интерфейсов.

Таким образом, наиболее используемой командой при изменении конфигурации сетевого экрана является:

```
/etc/net/scripts/contrib/efw default stop;
/etc/net/scripts/contrib/efw all start
```

7.7. Настройка удаленного подключения

Для получения удаленного доступа к другим ПЭВМ и предоставления такого доступа в ОС Альт 8 СП используется протокол SSH (Secure Shell).

SSH реализует соединение с удаленным компьютером, защищающее от следующих угроз:

- прослушивание данных, передаваемых по этому соединению;
- манипулирование данными на пути от клиента к серверу;
- подмена клиента либо сервера путем манипулирования IP-адресами, DNS либо маршрутизацией.

SSH обладает следующими возможностями:

- сжатие передаваемых данных;
- туннелирование каналов внутри установленного соединения – в том числе соединений с X-сервером;
- широкая распространенность: существуют реализации SSH для самых различных аппаратных платформ и операционных систем.

OpenSSH – реализация SSH, входящая в состав дистрибутива. Эта реализация включает в себя следующие программы и утилиты:

- клиентские программы ssh, scp и sftp (используются для запуска программ на удаленных серверах и копирования файлов по сети);
- серверные программы sshd, sftp-server (используются для предоставления доступа по протоколу SSH);
- вспомогательные программы scp, rescp, ssh-keygen, ssh-add, ssh-agent, ssh-copy-id, ssh-keyscan.

7.7.1. OpenSSH, сервер протокола SSH (sshd)

OpenSSH Daemon (sshd) – программа-сервер, обслуживающая запросы программы-клиента ssh. Вместе эти программы заменяют rlogin и rsh и обеспечивают защищенную и кодированную связь между двумя непроверенными компьютерами через незащищенную сеть.

sshd – это служба, принимающая запросы на соединения от клиентов. Для каждого нового соединения создается (с помощью вызова «fork») новый экземпляр службы. Ответвленный экземпляр обрабатывает обмен ключами, кодирование, аутентификацию, выполнение команд и обмен данными.

Параметры определяются при помощи ключей командной строки или файла конфигурации (по умолчанию – `sshd_config`). Ключи командной строки имеют больший приоритет, чем значения, указанные в файле конфигурации. При получении сигнала отбоя SIGHUP перечитывает свой файл конфигурации путем запуска собственной копии с тем же самым именем, с которым был запущен, например, `/usr/sbin/sshd`.

Синтаксис команды:

```
sshd [-46Ddeiqt] [-b длина ключа_1] [-f файл конфигурации] [-g
время_задержки_регистрации] [-h файл_ключа_хоста] [-k
частота_генерации_ключа] [-o директива] [-p порт] [-u длина]
```

Доступны ключи, приведенные в таблице 4.

Т а б л и ц а 4 – Ключи команды sshd

Ключ	Описание
`-4`	Использовать только адреса IPv4.
`-6`	Использовать только адреса IPv6.
`-b длина_ключа_1`	Определяет число битов в ключе сервера протокола версии 1 (по умолчанию 1024).
`-D`	Не переходить в фоновый режим и не становиться службой. Это упрощает слежение за экземпляром sshd.
`-d`	Режим отладки. Сервер посылает расширенную отладочную информацию в файл журнала событий системы и не переходит в фоновый режим работы. Сервер не создает дочерних процессов и обрабатывает только одно соединение. Параметр предназначен только для отладки работы сервера. Несколько параметров -d указанных один за другим, повышают уровень отладки. Максимум – это 3.
`-e`	Направлять вывод в консоль (stderr) вместо механизма журналирования событий системы.
`-f файл_конфигурации`	Определяет имя файла конфигурации (по умолчанию – `/etc/openssh/sshd_config`). Не работает, если нет файла конфигурации.

Продолжение таблицы 4

Ключ	Описание
`-g время_задержки_регистрации`	Определяет период, в течение которого клиент должен себя идентифицировать (по умолчанию – 120 секунд). Если клиент не смог идентифицировать себя в течение этого времени, экземпляр сервера прекращает свою работу. Значение равное нулю отменяет ограничение на время ожидания.
`-h файл_ключа_хоста`	Определяет файл, из которого будет считан ключ хоста. Этот параметр должен быть указан, если запущен не от имени пользователя с идентификатором root (так как обычно стандартные файлы хоста доступны для чтения только пользователю с идентификатором root). Стандартное расположение файла – `/etc/openssh/ssh_host_key` для протокола версии 1, и `/etc/openssh/ssh_host_dsa_key`, `/etc/openssh/ssh_host_ecdsa_key` и `/etc/openssh/ssh_host_rsa_key` для протокола версии 2. Можно иметь несколько ключей хоста для разных версий протокола и алгоритмов генерации ключей.
`-i`	Позволяет уведомить программу о том, что она запускается службой inetd. Обычно sshd не запускается из inetd, так как требуется генерировать ключ сервера до ответа клиенту, а это может отнять десятки секунд. Клиент будет вынужден ожидать слишком долго, если ключ будет повторно генерироваться каждый раз. Однако, при малых размерах ключа (например, 512), использование из inetd может быть оправдано.
`-k частота_генерации_ключа`	Определяет, как часто будет регенерироваться ключ сервера протокола версии 1 (по умолчанию 3600 секунд – один раз в час). Значение ноль означает, что ключ никогда не будет регенерирован.
`-o директива`	Позволяет указывать директивы в формате файла конфигурации, например, такие, для которых нет соответствующего ключа командной строки. Директивы файла конфигурации описаны в sshd_config.

Окончание таблицы 4

Ключ	Описание
`-p порт`	Порт, на котором сервер будет ожидать соединения (по умолчанию – 22). Возможно указание нескольких ключей с разными портами. Если данный ключ указан, параметр Port файла конфигурации игнорируется, однако порты, указанные в ListenAddress имеют больший приоритет, чем указанные в командной строке.
`-q`	Не заносить в системный журнал регистрации событий никакой информации. В обычном режиме в нем фиксируется подключение, аутентификация и разрыв каждого соединения.
`-t`	Режим тестирования. Выполняется только проверка соответствия файла конфигурации и готовность ключей. Полезно для проверки состояния службы после обновления, при котором были изменены файлы конфигурации.
`-u длина`	Размер поля в структуре utmp хранящей имя удаленного хоста. Если разрешенное имя хоста превышает указанное значение, то взамен будет использован десятичное представление IP-адреса через точку. Это позволяет уникально идентифицировать машины со слишком длинными именами. Указание `-u0` включает использование в файле utmp IP-адресов во всех случаях. При этом будет производить DNS-запросы только если это явно требуется конфигурацией (`from="pattern-list"`) или механизмом аутентификации (либо RhostsRSAAuthentication либо HostbasedAuthentication). Использование DNS также обязательно в случае задания параметрам AllowUsers и DenyUsers значения в формате USER@HOST.

7.7.1.1. Аутентификация

Служба OpenSSH SSH поддерживает версии протокола SSH 1 и 2. При этом использование протокола версии 1 крайне не рекомендуется. Запретить использование одного протокола версии 1 можно, указав в параметре `Protocol` файла `sshd_config`:

```
Protocol 2
```

Протокол 2 поддерживает ключи DSA, ECDSA и RSA; протокол 1 поддерживает только ключи RSA. Независимо от протокола, каждый подключающийся хост имеет собственный, обычно 2048-битный идентифицирующий его ключ.

Для протокола версии 1 подтверждение субъекта сервера обеспечивается 768-битным ключом, который генерируется при запуске сервера. Ключ генерируется заново каждый час, при условии его использования, и не хранится на диске. При получении запроса на подключение со стороны клиента служба посылает в ответ свой открытый ключ и свои ключи. Клиент сравнивает ключ хоста RSA со своими данными, чтобы убедиться в том, что это тот же сервер. Затем клиент генерирует 256-битное произвольное число, шифрует его при помощи обоих ключей (своего и сервера) и отправляет результат серверу. Это число становится ключом сеанса, и с его помощью выполняется кодирование всех последующих данных, по согласованному методу – Blowfish или 3DES (клиент выбирает метод из предложенных сервером). В настоящее время по умолчанию используется 3DES.

Для протокола версии 2 подтверждение субъекта сервера обеспечивается по схеме Диффи-Хеллмана, в результате которой также получается общий ключ сеанса. Дальнейший обмен данными шифруется симметричным кодом, 128-битным AES, Blowfish, 3DES, CAST128, Arcfour, 192-битным AES или 256-битным AES, который выбирает клиент из предложенных сервером. Кроме того, целостность передаваемых данных обеспечивается кодом подтверждения подлинности сообщения (hmac-md5, hmac-sha1, umac-64, hmac-ripemd160, hmac-sha2-256 или hmac-sha2-512).

Далее, сервер и клиент переходят в режим аутентификации. Клиент пытается аутентифицировать себя по своему хосту, открытому ключу, паролю или с помощью беспарольного механизма («вызов-ответ»).

Независимо от типа аутентификации служба проверяет доступность соответствующей учетной записи в системе. Так, она может быть заблокирована посредством добавления ее в параметр DenyUsers или ее группы в DenyGroups. Для

запрета только аутентификации по паролю укажите в файле `passwd 'NP'` или `'*NP*'`.

После успешной аутентификации себя клиентом связь переходит в режим подготовки сеанса. В этот момент клиент может запросить такие вещи, как выделение псевдо-терминала, перенаправление соединения X11, перенаправление соединения TCP/IP или перенаправление соединения агента аутентификации через защищенный канал.

Наконец, клиент запрашивает оболочку или выполнение команды, после чего стороны входят в режим сеанса. В этом режиме, каждая из сторон в любой момент может пересылать данные и эти данные будут переданы оболочке или команде на стороне сервера и на пользовательский терминал соответственно.

По завершении работы пользовательской программы и закрытии всех перенаправленных X11 и других соединений сервер посылает клиенту команду со статусом выхода и сеанс завершается.

7.7.1.2. Вход в систему

После успешной аутентификации пользователя выполняются следующие действия:

- если регистрация в системе произведена на терминале (tty) и не указана никакая команда, то отображается время последнего входа в систему и содержимое файла `/etc/motd` (если только это не отключено в файле конфигурации или `~/.hushlogin`);
- если регистрация в системе произведена на терминале, записывается время регистрации;
- проверяется `/etc/nologin` если он присутствует, выводится его содержимое и завершается работа (исключение – root);
- осуществляется переход к выполнению с правами обычного пользователя;
- устанавливаются значения основных переменных среды;
- интерпретируется файл `~/.ssh/environment`, если таковой имеется, и пользователям разрешено изменять среду;
- происходит переход в домашний каталог пользователя;

- если имеется файл `~/.ssh/rc`, то производится его выполнение, а если нет и имеется `/etc/openssh/sshrc`, то выполняется он, в противном случае выполняется xauth. Файлам `rc` на стандартный ввод передается протокол аутентификации X11 и cookie;
- запускается оболочка пользователя или выполняется указанная команда.

7.7.1.3. SSHRC

Если файл `~/.ssh/rc` существует, он будет выполняться после файлов определения переменных среды, но перед запуском оболочки пользователя или команды. Если используется подмена X11, то на его стандартный ввод будет передана пара «proto cookie», также ему будет доступна переменная среды DISPLAY. Сценарий должен вызывать xauth самостоятельно для добавления cookie X11.

Основная цель этого файла состоит в выполнении процедур инициализации, необходимые прежде, чем станет доступным основной каталог пользователя. AFS – пример такой среды.

Этот файл будет, содержать блок аналогичный следующему:

```
if read proto cookie && [ -n "$DISPLAY" ]; then
if [ `echo $DISPLAY | cut -c1-10` = 'localhost:' ]; then
# X11UseLocalhost=yes
echo add unix:`echo $DISPLAY |
cut -c11-` $proto $cookie
else
# X11UseLocalhost=no
echo add $DISPLAY $proto $cookie
fi | xauth -q -
fi
```

Если этот файл отсутствует, то выполняется `/etc/openssh/sshrc`, а если отсутствует и он, то для добавления cookie используется xauth.

7.7.1.4. Формат файла `authorized_keys`

Параметр `AuthorizedKeysFile` файла конфигурации определяет путь к файлу с открытыми ключами. Значение по умолчанию – `~/.ssh/authorized_keys` и

`~/.ssh/authorized_keys2`. Каждая строка файла содержит один ключ (пустые строки или строки, начинающиеся с символа «#» считаются комментариями и игнорируются). Открытые ключи протокола 1 (RSA) состоят из следующих полей, разделенных пробелами: параметры, битность, порядок, модуль, комментарий. Открытые ключи протокола версии 2 состоят из полей: параметр, тип ключа, ключ в виде base64, комментарий. Поля параметров необязательны; их отсутствие определяется наличием в начале строки цифры (поле параметра никогда не начинается с цифры). Поля битности, порядка, модуля и комментарии определяют ключ RSA; поле комментария не используется (но может быть удобно пользователю для отметки ключа). Для протокола версии 2 типом ключа является ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, ssh-dss или ssh-rsa.

Строки в этих файлах, обычно имеют длину в несколько сотен байт (из-за размера открытого ключа RSA) и могут достигать длины в 8 килобайт (таким образом, максимальный размер ключа DSA – 8 килобит, а RSA – 16 килобит). Очевидно, не стоит вводить их вручную. Вместо этого следует скопировать файл `identity.pub`, `id_dsa.pub` или `id_rsa.pub` и отредактировать.

Минимальная длина модуля RSA независимо от протокола составляет 768 бит.

Параметры (если таковые имеются) состоят из разделенных запятой определений. Для указания пробелов следует воспользоваться двойными кавычками. Поддерживаются следующие определения параметров (регистра названий параметров не учитывается):

1) `command="команда"` – выполнять команду при каждом использовании данного ключа для аутентификации. Команда, передаваемая пользователем, будет игнорироваться. Команда выполняется на псевдо-терминале, если последний запрашивается клиентом; в противном случае она выполняется без терминала. Если требуется «чистый» 8-битный канал, запрашивать псевдо-терминал или указывать no-pty нельзя. В команду может быть включена кавычка, предваренная обратной косой чертой. Данный параметр полезен для ограничения использования определенных RSA-ключей. Примером может служить ключ, по которому можно

выполнять удаленные операции резервного копирования и ничего более. Учтите, что клиент по-прежнему может запросить перенаправление TCP и (или) X11, если только это не запрещено явно. Команда, запрашиваемая клиентом, заносится в переменную SSH_ORIGINAL_COMMAND. Заметьте, что данный параметр относится к выполнению оболочки, команды или подсистемы;

2) `environment="ПЕРЕМЕННАЯ=значение"` – добавить переменную в среду (или переопределить ее значение) при регистрации в системе с использованием данного ключа. Допускается указание нескольких таких директив. По умолчанию изменение переменных среды таким образом отключено. За его включение отвечает параметр PermitUserEnvironment. Этот параметр отключается автоматически при включении UseLogin;
3) `From="список-шаблонов"` – если параметр определен, то в дополнение к прохождению аутентификации по открытому ключу каноническое имя удаленного хоста должно соответствовать одному из шаблонов в списке (шаблоны указываются через запятую). Цель этого параметра – увеличение степени защиты: если частный ключ хоста каким-либо образом удастся похитить, то он позволит злоумышленнику войти в систему из любой точки мира. Этот дополнительный параметр делает использование ворованных ключей более затруднительным (кроме перехвата ключа, требуется взлом серверов имен и (или) маршрутизаторов). Смотреть секцию ШАБЛОНЫ в `ssh_config`;
4) `no-agent-forwarding` – запретить перенаправление агента аутентификации при аутентификации данным ключом;
5) `no-port-forwarding` – запретить перенаправление TCP/IP при аутентификации данным ключом. Любой запрос на перенаправление порта приведет к получению клиентом сообщения об ошибке. Это может быть использовано, например, вместе с параметром `command`;
6) `no-pty` – запретить назначение терминала (запросы на назначение псевдо-терминала не будут удовлетворены);

7) `no-X11-forwarding` – запретить перенаправление X11 при аутентификации данным ключом. Любой запрос на перенаправление порта возвратит клиенту сообщение об ошибке;
8) `permitopen="хост:порт"` – для функции перенаправления данных с локального клиентского порта на порт удаленной системы (выполняемого при указании `ssh -L`) ограничить набор возможных целей для перенаправления указанной машиной и портом. Для указания адресов IPv6 можно использовать альтернативный синтаксис: хост/порт. Допускается указание нескольких целей через запятую. Значение параметра не интерпретируется как шаблон (т. е. является литеральным);
9) `tunnel="n"` – принудительно использовать устройство tun на сервере. Без этого параметра при запросе клиентом туннеля используется ближайшее доступное для этого устройство.

Пример файла `authorized_keys`:

```
# допустимы комментарии только на всю строку
ssh-rsa AAAAB3Nza...LiPk== user@example.test
from="*.sales.example.test,!pc.sales.example.test" ssh-rsa
AAAAB2...19Q== test@example.test
command="dump /home",no-pty,no-port-forwarding ssh-dss
AAAAC3...51R== example.test
permitopen="192.0.2.1:80",permitopen="192.0.2.2:25" ssh-dss
AAAAB5...21S==
tunnel="0",command="sh /etc/netstart tun0" ssh-rsa AAAA...==
user@example.test
```

7.7.1.5. Формат файла `ssh_known_hosts`

В файлах `/etc/openssh/ssh_known_hosts` и `~/.ssh/known_hosts` хранятся открытые ключи всех машин, с которыми когда-либо устанавливалась связь. Глобальный файл должен быть подготовлен администратором (это необязательно), пользовательский файл поддерживается автоматически: каждый раз, когда поступает запрос на соединение от неизвестной машины, ее ключ автоматически заносится в пользовательский файл.

Каждая строка в этом файле содержит следующие поля: имена хостов, битность, порядок, модуль, комментарий. Поля разделены пробелами.

Имена хостов – это разделенный запятыми список шаблонов (символы подстановки – ('*' и '?'); каждый шаблон сопоставляется с каноническим именем машины (при аутентификации клиента) или с именем, которое указано пользователем (при аутентификации сервера). Этот шаблон может также быть предварен знаком '!' для обозначения отрицания: если имя машины соответствует отрицаемому шаблону, оно будет отвергнуто (этой строкой) даже если оно соответствует другому шаблону в этой же строке. Также можно, заключив имя хоста или IP-адрес в квадратные скобки – '[' и '],' – через ':' указать нестандартный порт.

Вместо имен хостов можно записывать их хэши. Это позволит скрыть их от злоумышленника в случае попадания файла в его руки. Для различия хэшей от имен хостов первые предваряются символом '|'. На одной строке может быть не больше одного хэша, операция отрицания в этом случае не доступна.

Разрядность, порядок и модуль копируются из ключа хоста RSA, например, `/etc/openssh/ssh_host_key.pub` Необязательное поле комментария занимает всю оставшуюся часть строки и игнорируется.

Комментариями также считаются пустые и строки, начинающиеся с «#».

Идентификация машины принимается, если любая совпавшая строка содержит правильный ключ. Таким образом, можно (хотя это не рекомендуется) иметь несколько строк или различных ключей для одного и того же хоста. Это неизбежно случается при помещении в файл кратких форм имен хостов из

различных доменов. В файлах может содержаться противоречивая информация. Идентификация принимается, если адекватная информация имеется в любом из них.

Заметьте, что строки в этих файлах, обычно имеют длину в несколько сотен символов и, очевидно, не стоит вводить имена хостов вручную. Вместо этого их можно сгенерировать при помощи сценария оболочки или взять из файла `/etc/ssh/ssh_host_key.pub`, добавив вначале имя хоста.

Пример файла `ssh_known_hosts`:

```
# допустимы явные комментарии только на всю строку
closenet,...,192.0.2.53 1024 37 159...93 closenet.example.test
cvs.example.test,192.0.2.10 ssh-rsa AAAA1234.....=
# хэш имени хоста
|1|JfKTdBh7rNbXkVAQCRp4OQoPfmI=|USECr3SWf1JUPsms5AqfD5QfxkM= ssh-
rsa
AAAA1234.....=
```

7.7.1.6. Файлы

`~/.hushlogin` – позволяет отключить вывод времени последнего входа в систему и содержимого файла `/etc/motd`, если в файле конфигурации включены соответственно PrintLastLog и PrintMotd. Файл не влияет на вывод содержимого Banner.

`~/.rhosts` – используется для аутентификации по хосту. На некоторых машинах, если каталог пользователя находится на разделе NFS, для того чтобы он был доступен пользователю root, он должен быть доступен для чтения всем. Файл должен принадлежать пользователю и не должен быть доступен для записи другим. Рекомендуемый набор прав доступа в общем случае – чтение/запись для пользователя и недоступность для других.

`~/.shosts` – аналогичен файлу `.rhosts`, но позволяет проводить аутентификацию на основе хоста, не разрешая вход в систему с помощью `rlogin/rsh`.

`~/.ssh/authorized_keys` – содержит список открытых ключей (DSA/ECDSA/RSA), которые могут быть использованы для регистрации данного пользователя. Формат файла описан выше. Этот файл не очень важен для

злоумышленника, но мы рекомендуем сделать его доступным только пользователю (чтение/запись).

Если этот файл, каталог `~/.ssh` или домашний каталог пользователя доступны для записи другим пользователям, этот файл может быть изменен или заменен любым пользователем системы, имеющим сколько угодно мало прав. В этом случае sshd не будет использовать этот файл, если только параметр StrictModes не имеет значение «no». Установить рекомендуемый набор прав доступа можно командой `chmodgo-w ~/ ~/.ssh ~/.ssh/authorized_keys`.

`~/.ssh/environment` – этот файл (при его наличии) считывается в среду при регистрации в системе. Он может содержать только пустые строки, строки комментария (начинающиеся с «#»), и определения значений переменных в виде: переменная=значение. Правом на запись этого файла должен обладать только пользователь; он не должен быть доступен остальным. Задание переменных среды отключено по умолчанию, за что отвечает параметр `PermitUserEnvironment`.

`~/.ssh/known_hosts` – список адресов, к которым когда-либо подключался пользователь, и которые отсутствуют в общесистемном файле, и соответствующих им открытых ключей. Формат файла описан выше. Файл должен быть доступен для записи только владельцу и администратору. Он может также быть доступен для чтения всем остальным, но это не обязательно.

`~/.ssh/rc` – сценарий инициализации, запускаемый перед запуском оболочки пользователя или команды. Этот файл должен быть доступен для записи только пользователю и не должен быть вообще доступен другим.

`/etc/hosts.allow` и `/etc/hosts.deny` – данные о разрешении и запрете соединений с хостами для надстроек TCP.

`/etc/hosts.equiv` – используется для аутентификации на основе хоста. Должен быть доступен для записи только root.

`/etc/openssh/moduli` – модули для схемы Диффи-Хеллмана.

`/etc/motd` – содержимое файла отображается программой login после того как осуществлен успешный вход в систему, перед запуском команды интерпретатора.

/etc/nologin – если существует, подключение будет разрешено только пользователю с идентификатором root. Любому, кто пытается войти в систему, будет показано содержимое этого файла, и запросы на регистрацию в качестве не пользователя с идентификатором root будут отвергнуты. Этот файл должен быть доступен для чтения всем.

/etc/shosts.equiv – аналогичен hosts.equiv, но позволяет проводить аутентификацию на основе хоста, не разрешая вход в систему с помощью rlogin/rsh.

/etc/openssh/ssh_known_hosts – общесистемный список известных хостов и их ключей. Этот файл должен составляться администратором. В него следует включать открытые ключи всех компьютеров организации. Формат файла описан выше. Файл должен быть доступен всем для чтения и владельцу/администратору для записи.

/etc/openssh/ssh_host_key, /etc/openssh/ssh_host_dsa_key, /etc/openssh/ssh_host_ecdsa_key, /etc/openssh/ssh_host_rsa_key – содержат частные ключи хостов. Файлы должны принадлежать root, и быть доступными только для него. Не запустится если эти файлы доступны для чтения кому-либо кроме пользователя с идентификатором root.

/etc/openssh/ssh_host_key.pub, /etc/openssh/ssh_host_dsa_key.pub, /etc/openssh/ssh_host_ecdsa_key.pub, /etc/openssh/ssh_host_rsa_key.pub – содержат открытые ключи хостов. Должны быть доступны всем для чтения и только пользователю с идентификатором root для записи. Содержимое файлов должно соответствовать содержимому соответствующих файлов с частными ключами. Эти файлы не используются программой и предназначены для копирования пользователем в файлы known_hosts. Эти файлы создаются командой ssh-keygen.

/etc/openssh/sshd_config – конфигурация службы sshd.

/etc/openssh/sshrc – аналогичен ~/.ssh/rc, позволяет задавать инициализационный сценарий глобально для всех пользователей. Должен быть доступен всем для чтения и только root для записи.

/var/empty – каталог chroot используемый при отделении полномочий на предаутентификационном этапе. В папке не должно быть никаких файлов, она должна принадлежать только root и не должна быть доступна другим для записи.

/var/run/sshd.pid – идентификатор процесса, ожидающего запросов на подключение (если одновременно работает несколько экземпляров служб для нескольких портов, в него записывается идентификатор экземпляра, запущенного последним). Содержимое этого файла может не быть защищено и может быть доступно всем.

7.7.2. SSHD_CONFIG

7.7.2.1. Описание файла конфигурации

Служба sshd считывает данные о конфигурации из файла /etc/openssh/sshd_config (или из файла, указанного в командной строке при помощи параметра -f). Файл содержит пары «параметр-значение», по одной на строку. Пустые строки и строки, начинающиеся с «#» интерпретируются как комментарии. В случае, если аргументы содержат пробелы, они должны быть заключены в двойные кавычки (”).

Файл /etc/openssh/sshd_config должен быть доступен для записи только пользователю root, и рекомендуется делать его доступным для чтения всем.

В таблице 5 приведены описания возможных параметров (регистр имен аргументов учитывается, регистр имен параметров – нет).

Т а б л и ц а 5 – Описание параметров

Параметр	Описание
AcceptEnv	Список переменных среды, которые, будучи заданы клиентом, будут копироваться в environ сеанса. Соответствующая настройка на стороне клиента выполняется параметром SendEnv и описана в `ssh_config`. Переменные указываются по имени, допускаются символы подстановки «*» и «?» Несколько переменных среды можно указывать через пробелы или в нескольких параметрах AcceptEnv. Данный параметр введен для предотвращения обхода ограничений среды пользователя посредством изменения значений переменных среды. По умолчанию не принимаются никакие переменные среды.
AddressFamily	Семейство адресов, которое должна использовать служба sshd. Допустимые значения: «any» «inet» (только IPv4) и «inet6» (только IPv6). Значение по умолчанию – «any».
AllowGroups	Список шаблонов имен групп через пробел. Если параметр определен, регистрация в системе разрешается только тем пользователям, чья главная или вспомогательная группы соответствуют какому-либо из шаблонов. Допустимы только имена групп. По умолчанию разрешена регистрация в системе для членов всех групп. Разрешающие/запрещающие (`allow/deny`) директивы обрабатываются в следующем порядке: `DenyUsers AllowUsers DenyGroups AllowGroups`.
AllowTcpForwarding	Определяет, будет ли разрешено перенаправление TCP. Значение по умолчанию – «yes». Отключение пересылки TCP не увеличит уровень защищенности системы, пока пользователям не запрещен доступ к командной оболочке, так как они всегда могут установить свои собственные перенаправления.
AllowUsers	Список имен пользователей через пробел. Если параметр определен, регистрация в системе будет разрешена только пользователям, чьи имена соответствуют одному из шаблонов. Допустимы только имена пользователей; числовой идентификатор пользователя не распознается. По умолчанию разрешена регистрация в системе для всех пользователей.

Продолжение таблицы 5

Параметр	Описание
	Если шаблон указывается в форме ПОЛЬЗОВАТЕЛЬ@ХОСТ, его две части проверяются отдельно, таким образом, разрешая доступ только пользователям с указанными именами, подключающимся с указанных хостов. Разрешающие/запрещающие (`allow/deny`) директивы обрабатываются в следующем порядке: `DenyUsers AllowUsers DenyGroups AllowGroups`.
AuthorizedKeysFile	Файл с открытыми ключами, которые могут быть использованы для аутентификации пользователей. Допустимо указание шаблонов, они преобразуются при настройке соединения: %% заменяется на символ '%', %h заменяется на домашний каталог идентифицируемого пользователя, %u – на имя пользователя. После преобразования AuthorizedKeysFile интерпретируется либо как абсолютный путь, либо как путь относительно домашнего каталога пользователя. Значение по умолчанию – `/etc/openssh/authorized_keys/%u /etc/openssh/authorized_keys2/%u .ssh/authorized_keys .ssh/authorized_keys2`.
Banner	Содержимое указанного файла будет отправлено удаленному пользователю прежде, чем будет разрешена аутентификация. Этот параметр доступен только с протоколом версии 2. По умолчанию не выводится никакой информации.
ChallengeResponseAuthentication	Определяет, разрешается ли беспарольная аутентификация «вызов-ответ». Поддерживаются все схемы аутентификации `login.conf`. Значение по умолчанию – «no».
Ciphers	Допустимые для протокола версии 2 шифры. Несколько кодов указываются через запятую. Поддерживаются следующие шифры: «3des-cbc», «aes128-cbc», «aes192-cbc», «aes256-cbc», «aes128-ctr», «aes192-ctr», «aes256-ctr», «arcfour128», «arcfour256», «arcfour», «blowfish-cbc» и «cast128-cbc». Значение по умолчанию: - aes256-ctr,aes192-ctr,aes128-ctr,arcfour256,arcfour128; - blowfish-cbc,aes256-cbc,aes192-cbc,aes128-cbc,3des-cbc; - cast128-cbc,arcfour.

Продолжение таблицы 5

Параметр	Описание
ClientAliveCountMax	Количество запросов, проверяющих доступность клиента, которые могут оставаться без ответа. Если предел достигнут, sshd отключит клиента и завершит сеанс. Запросы client alive отличаются от TCPKeepAlive. Данные запросы отправляются через защищенный канал и поэтому не могут быть подменены. Параметр TCPKeepAlive допускает возможность подмены данных. Механизм client alive полезен, если поведение клиента или сервера зависит от активности соединения. Значение по умолчанию – 3. Если ClientAliveInterval равно 15, а для ClientAliveCountMax оставлено значение по умолчанию, не отвечающие клиенты SSH будут отключаться приблизительно через 45. Данный параметр относится только к протоколу версии 2.
ClientAliveInterval	Время бездействия со стороны клиента в секундах, после которого sshd отправляет через защищенный канал запрос отклика клиенту. Значение по умолчанию – 0, что означает, что клиенту не будут направляться такие запросы. Этот параметр применим только с протоколом версии 2.
Compression	Разрешить сжатие сразу, после аутентификации или вообще запретить его. Допустимые значения – «yes», «delayed» и «no». Значение по умолчанию – «delayed».
DenyGroups	Список шаблонов имен групп через пробел. Если параметр определен, регистрация в системе пользователям, чья главная или вспомогательная группа соответствуют содержащимся в списке шаблонам, не разрешается. Допустимы только имена групп. По умолчанию регистрация в системе разрешена для всех групп. Разрешающие/запрещающие (`allow/deny`) директивы обрабатываются в следующем порядке: `DenyUsers AllowUsers DenyGroups AllowGroups`.
DenyUsers	Список имен пользователей через пробел. Если параметр определен, регистрация в системе пользователей, чьи имена соответствуют одному из шаблонов, будет запрещена. Допустимы только имена пользователей; числовой идентификатор пользователя не распознается. По умолчанию разрешена регистрация в системе для всех пользователей. Если шаблон указывается в форме ПОЛЬЗОВАТЕЛЬ@ХОСТ, его две части проверяются отдельно, таким образом, запрещается доступ только пользователям с указанными именами, подключающимся с указанных хостов.

Продолжение таблицы 5

Параметр	Описание
	Разрешающие/запрещающие (`allow`/`de`ny) директивы обрабатываются в следующем порядке: `DenyUsers AllowUsers DenyGroups AllowGroups`.
ForceCommand	Выполнять указанную команду после регистрации пользователя в системе, игнорируя команду, запрашиваемую им. Команда запускается оболочкой пользователя с ключом -c. Это относится к выполнению оболочки, команды или подсистемы, обычно применяется внутри блока Match. Команда, запрошенная пользователем, помещается в переменную среды SSH_ORIGINAL_COMMAND.
GatewayPorts	Определяет, разрешено ли удаленным машинам подключение к портам, выделенным для туннелирования трафика клиентов. По умолчанию sshd делает доступными порты, используемые для туннелирования инициируемого сервером, только для кольцевого (loopback) адреса, то есть удаленные машины подключаться к перенаправляемым портам не могут. С помощью данного параметра можно исправить такое положение дел. Значение «no» разрешает туннелирование только в рамках данной системы, «yes» разрешает туннелирование для хостов соответствующих шаблону, а «clientspecified» позволяет клиенту самостоятельно выбирать адрес для туннелирования. Значение по умолчанию – «no».
GSSAPIAuthentication	Допускать аутентификацию по GSSAPI. Значение по умолчанию – «no» Данный параметр относится только к протоколу версии 2.
GSSAPICleanupCredentials	Очищать ли кэш аутентификационных данных клиента при завершении сеанса. Значение по умолчанию – «yes» Данный параметр относится только к протоколу версии 2.
HostbasedAuthentication	Допускать аутентификацию по хостам, т. е. аутентификацию по rhosts или `/etc/hosts.equiv` в сочетании с открытым ключом клиента. Этот параметр схож с RhostsRSAAuthentication и применим только к протоколу версии 2. Значение по умолчанию – «no».

Продолжение таблицы 5

Параметр	Описание
HostbasedUsesNameFromPacketOnly	Отключить выполнение запросов имени хоста при обработке файлов `~/.shosts`, `~/.rhosts` и `/etc/hosts.equiv` в рамках аутентификации по хосту (HostbasedAuthentication). При значении «yes» для сравнения будет использоваться имя указанное клиентом, а не имя которое может быть получено стандартными средствами соединения TCP. Значение по умолчанию – «no».
HostKey	Файл с частными ключами хоста. Значение по умолчанию – `/etc/ssh/ssh_host_key` для протокола 1, и `/etc/ssh/ssh_host_dsa_key`, `/etc/ssh/ssh_host_ecdsa_key` и `/etc/ssh/ssh_host_rsa_key` для протокола 2. sshd не будет принимать файлы частных ключей доступные для чтения всей группе или вообще всем пользователям. Можно указывать несколько файлов с ключами хоста. Ключи «rsa1» используются для протокола версии 1, ключи «dsa», «ecdsa» и «rsa» – для версии 2 протокола SSH.
IgnoreRhosts	Не учитывать содержимое файлов `.rhosts` и `.shosts` при аутентификации RhostsRSAAuthentication и HostbasedAuthentication. При этом будут учитываться только `/etc/hosts.equiv` и `/etc/openssh/shosts.equiv`. Значение по умолчанию – «yes».
IgnoreUserKnownHosts	Не учитывать содержимое файла `~/.ssh/known_hosts` при RhostsRSAAuthentication или HostbasedAuthentication Значение по умолчанию – «no».
KerberosAuthentication	Определяет, дозволена ли аутентификация Kerberos: Проверять ли пароль, указанный пользователем для аутентификации PasswordAuthentication в Kerberos KDC. Это может быть либо в форме тикетов Kerberos или, если PasswordAuthentication установлена в «yes», пароль, предоставленный пользователем, будет утвержден через Kerberos KDC. Для использования этого параметра серверу необходима Kerberos servtab, которая разрешит проверку субъекта KDC. Значение по умолчанию – «no».

Продолжение таблицы 5

Параметр	Описание
KerberosGetAFSToken	Если AFS активна и у пользователя имеется Kerberos 5 TGT, получать талон AFS перед обращением к домашнему каталогу пользователя. Значение по умолчанию – «no».
KerberosOrLocalPasswd	В случае непринятия аутентификации посредством Kerberos, проверять пароль другими механизмами, такими как `/etc/passwd`. Значение по умолчанию – «yes».
KerberosTicketCleanup	Очищать ли кэш талонов пользователя при завершении сеанса. Значение по умолчанию – «yes».
KeyRegenerationInterval	В протоколе версии 1 эфемерный ключ сервера будет автоматически регенерироваться по истечении этого количества секунд. Цель регенерации состоит в том, чтобы предохранить кодированные установленные сеансы от более поздних вторжений на машину и захвата ключей. Ключ нигде не сохраняется. Если установлено значение 0, то ключ не будет регенерироваться. Значение по умолчанию – 3600 (секунд).

ListenAddress	Локальные адреса, по которым sshd должен ожидать соединения. Может быт использован следующие форматы записей:
	`ListenAddress`
	`хост|адрес-IPv4|адрес-IPv6`
	`ListenAddress`
	`хост|адрес-IPv4:порт`
	`ListenAddress`
	`[хост|адрес-IPv6]:порт`
	Если порт не указан, sshd будет ожидать соединения на указанном адресе и на всех указанных ранее (но не после) в параметре Port портах. По умолчанию ожидается соединение на всех локальных адресах. Допустимо указание нескольких параметров.

LoginGraceTime	Сервер отключается по истечении этого времени, если пользователю не удалась регистрация в системе. Если стоит значение 0, то время ожидания не ограничено. Значение по умолчанию – 120 секунд.
LogLevel	Задает степень подробности сообщений для протоколов sshd. Допустимыми являются значения: QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2, и DEBUG3. Значение по умолчанию – INFO. Значения DEBUG и DEBUG1 эквивалентны. Использование значения DEBUG* нарушает конфиденциальность пользователей и потому не рекомендуется.

Продолжение таблицы 5

Параметр	Описание
MACs	Допустимые алгоритмы MAC (Message Authentication Code – код установления подлинности сообщения). Они используются в протоколе версии 2 для гарантирования целостности данных. Несколько алгоритмов следует указывать через запятую. Значение по умолчанию: `hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160,hmac-sha1-96,hmac-md5-96,hmac-sha2-256,hmac-sha256-96,hmac-sha2-512,hmac-sha2-512-96`.
Match	Начинает условный блок. Если все критерии на строке Match удовлетворены, указанные в блоке директивы будут иметь больший приоритет чем указанные в глобальном разделе файла конфигурации. Концом блока считается либо следующая директива Match, либо конец файла. В качестве аргументов Match принимаются пары критерий-шаблон. Допустимые критерии: User Group Host и Address В самом блоке Match допустимо указание следующих параметров: AllowAgentForwarding, AllowTcpForwarding, AuthorizedKeysFile, AuthorizedPrincipalsFile, Banner, ChrootDirectory, ForceCommand, GatewayPorts, GSSAPIAuthentication, HostbasedAuthentication, HostbasedUsesNameFromPacketOnly, KbdInteractiveAuthentication, Kerber-osAuthentication, Match, MaxAuthTries, MaxSessions, PasswordAuthentication, PermitEmptyPasswords, PermitOpen, PermitRoot-Login, PermitTunnel, PubkeyAuthentication, RhostsRSAAuthentication, RSAAuthentication, X11DisplayOffset, X11Forwarding и X11UseLocalHost.
MaxAuthTries	Ограничение на число попыток идентифицировать себя в течение одного соединения. При достижении количества неудачных попыток аутентификации записи о последующих неудачах будут вноситься в протокол. Значение по умолчанию: 6.
MaxSesstions	Ограничение на число одновременно открытых сессий в каждом сетевом соединении. Значение по умолчанию – 10.

Продолжение таблицы 5

Параметр	Описание
MaxStartups	Ограничение на число одновременных соединений, в которых не был пройден этап аутентификации. Все последующие соединения не будут приниматься, пока на уже существующем соединении не будет произведена аутентификация или не истечет время, указанное в параметре LoginGraceTime. Значение по умолчанию – «10:30:20». Как альтернатива может быть задействован ранний случайный отказ в подключении путем указания трех разделенных через двоеточие значений «старт:норма:предел» (например, «10:30:60»). Соединение будет сбрасываться с вероятностью «норма/100» (30%) если имеется «старт» (10) (10) соединений с не пройденным этапом аутентификации. Вероятность возрастает линейно и постоянно, попытки будут отвергаться при достижении числа «предел» (60).
PasswordAuthentication	Допускать аутентификацию по паролю. Значение по умолчанию – «yes».
PermitEmptyPasswords	Допускать использование пустых паролей при аутентификации по паролю. Значение по умолчанию – «no».

PermitOpen	Ограничить возможные конечные точки для туннелирования TCP. Допустимые формы указания точек:
	`PermitOpen хост:порт`
	`PermitOpen адрес-IPv4:порт`
	`PermitOpen [адрес-IPv6]:порт`
	Возможно указание нескольких конечных точек через пробел. Значение «any» снимает ограничение и является значением по умолчанию.

PermitRootLogin	Допускать вход в систему через ssh в качестве пользователя с идентификатором root. Допустимые значения: «yes», «without-password», «forced-commands-only», «no». Значение по умолчанию – «yes». Если этот параметр установлен в значение «without-password» войти в систему в качестве пользователя с идентификатором root, указав для аутентификации пароль, будет невозможно.

Продолжение таблицы 5

Параметр	Описание
	Если этот параметр установлен в значение «forced-commands-only» будет разрешена регистрация пользователя с идентификатором root в системе по открытому ключу, но только если определен параметр command команда (может быть полезно для удаленного создания резервных копий, даже если регистрация пользователя с идентификатором root в системе не разрешена). Все другие методы аутентификации для пользователя с идентификатором root будут отключены. При значении «no» вход в систему в качестве root будет полностью запрещен.
PermitTunnel	Допускать использование перенаправления для устройств tun. Допустимые значения: «yes» «point-to-point» (уровень 3), «ethernet» (уровень 2), «no». Значение «yes» эквивалентно «point-to-point» и «ethernet» одновременно. Значение по умолчанию – «no».
PermitUserEnvironment	Учитывать ли файл `~/.ssh/environment` и параметры `environment=` в файле `~/.ssh/authorized_keys`. Значение по умолчанию – «no» Посредством изменения переменных среды пользователи могут обойти ограничения своих полномочий. Например, с помощью механизма LD_PRELOAD.
PidFile	Файл в который следует записывать идентификатор процесса службы SSH. Значение по умолчанию – `/var/run/sshd.pid`.
Port	Порт, на котором следует ожидать запросы на соединение. Значение по умолчанию – 22. Допустимо указание параметра несколько раз. См. также ListenAddress.
PrintLastLog	Выводить ли время и дату предыдущего входа в систему при интерактивной регистрации пользователя в ней. Значение по умолчанию – «yes».
PrintMotd	Выводить ли содержимое файла `/etc/motd` при интерактивной регистрации пользователя в системе (в некоторых системах это выполняется оболочкой, сценарием `/etc/profile` или аналогичным). Значение по умолчанию – «yes».
Protocol	Версии протокола, которые следует принимать. Допустимые значения – «1» и «2» Несколько значений указываются через запятую. Значение по умолчанию – «2». Порядок указания протоколов не имеет значения, т. к. протокол выбирается клиентом из списка доступных.

Продолжение таблицы 5

Параметр	Описание
PubkeyAuthentication	Допускать аутентификацию по открытому ключу. Значение по умолчанию – «yes». Данный параметр относится только к протоколу версии 2.
RhostsRSAAuthentication	Допускать аутентификацию по rhosts или `/etc/hosts.equiv` совместно с аутентификацией по хосту RSA. Значение по умолчанию – «no» Данный параметр относится только к протоколу версии 1.
RSAAuthentication	Допускать аутентификацию только по ключу RSA. Значение по умолчанию – «yes». Данный параметр относится только к протоколу версии 1.
ServerKeyBits	Длина ключа сервера для эфемерного протокола 1. Минимальное значение – 512 (по умолчанию – 1024).
StrictModes	Проверять наборы прав доступа и принадлежность конфигурационных файлов и домашнего каталога пользователя перед разрешением регистрации в системе. Это рекомендуется выполнять потому, что новички иногда оставляют свои каталоги или файлы доступными для записи всем. Значение по умолчанию – «yes».
Subsystem	Позволяет настроить внешнюю подсистему (например, службу FTP). В качестве параметров должны выступать имя подсистемы и команда, которая будет выполняться при запросе подсистемы. Команда sftp-server реализует подсистему пере-дачи файлов sftp. По умолчанию подсистемы не определены. Данный параметр относится только к протоколу версии 2.
SyslogFacility	Код источника сообщений для протокола syslog. Допустимые значения: `DAEMON`, `USER`, `AUTHPRIV`, `LOCAL0`, `LOCAL1`, `LOCAL2`, `LOCAL3`, `LOCAL4`, `LOCAL5`, `LOCAL6`, `LOCAL7`. Значение по умолчанию – `AUTHPRIV`.
TCPKeepAlive	Указывает, будет ли система посылать другой стороне контрольные сообщения для удержания соединения активным. Если они посылаются, то разрыв соединения или аварийный отказ одной из машин будут должным образом замечены. При этом временная потеря маршрута также повлечет за собой разрыв соединения. С другой стороны, если контрольные сообщения не посылаются, сеанс на сервере может зависнуть, оставив после себя «пользователей-привидений» и отнимая ресурсы сервера. Значение по умолчанию – «yes». Это позволяет избежать бесконечно долгих сеансов. Для отключения отправки сообщений TCP keepalive установите значение «no».

Продолжение таблицы 5

Параметр	Описание
UseDNS	Выполнять ли запросы DNS для получения имени удаленного хоста для того чтобы убеждаться в том, что обратное преобразование выдает тот же самый IP-адрес. Значение по умолчанию – «yes».
UseLogin	Использовать login для интерактивных сеансов регистрации в системе. Значение по умолчанию – «no». login никогда не используется для удаленного выполнения команд. Если этот параметр включен, функция X11Forwarding будет отключена, потому что login не может обрабатывать cookie xauth. В случае использования разделения полномочий (UsePrivilegeSeparation) данный параметр будет отключен после прохождения аутентификации.
UsePAM	Включить интерфейс модулей аутентификации Pluggable Authentication Module. При значении «yes» аутентификация PAM будет доступна через ChallengeResponseAuthentication и PasswordAuthentication в дополнение к учетной записи PAM и обработке модулей сеансов для всех типов аутентификации. Поскольку безпарольная аутентификация PAM «вызов-ответ» служит заменой аутентификации по паролю, необходимо отключить либо PasswordAuthentication, либо ChallengeResponseAuthentication. При включенном UsePAM службу sshd можно будет выполнять только с правами root. Значение по умолчанию – «yes».
UsePrivilegeSeparation	Разделять полномочия посредством создания дочернего процесса с меньшими правами для обработки входящего трафика. После прохождения аутентификации для работы с клиентом будет создан специальный процесс, соответствующий его правам. Если значение параметра равно «sandbox», то на непривилегированный процесс до прохождения аутентификации будут наложены дополнительные ограничения. Значение по умолчанию – «sandbox».
X11DisplayOffset	Номер первого дисплея доступного для туннелирования трафика X11 sshd (по умолчанию – 10). Позволяет избежать вмешательства sshd в работу настоящих серверов X11.

Окончание таблицы 5

Параметр	Описание
X11Forwarding	Допускать туннелирование X11. Допустимые значения – «yes» и «no». Значение по умолчанию – «yes». Если дисплей-посредник ожидает соединений от любых адресов (или по шаблону) sshd включение туннелирования X11 подвергает сервер и логические дисплеи клиентов дополнительной опасности. Поэтому такое поведение не является поведением по умолчанию. Проверка и подмена аутентификационных данных при атаке выполняются на стороне клиента. При туннелировании X11 графический сервер клиента может подвергаться атаке при запросе клиентом SSH туннелирования. Для большей защиты пользователей администратор может запретить туннелирование, установив значение «no». Туннелирование X11 отключается автоматически при включении UseLogin.
X11UseLocalhost	К какому адресу следует привязывать сервер туннелирования X11: к кольцевому (loopback) или адресу, указанному по шаблону. По умолчанию сервер туннелирования привязывается к кольцевому адресу, а в качестве хоста в переменную среды DISPLAY заносится «localhost». Это не позволяет удаленным хостам подключаться к дисплею-посреднику. Однако, в случае старых клиентов X11, такая конфигурация может не сработать. Установите тогда X11UseLocalhost в «no». Допустимые значения – «yes» и «no». Значение по умолчанию – «yes».
XAuthLocation	Путь к команде xauth. Значение по умолчанию – `/usr/bin/xauth`.

7.7.2.2. Указание времени

Ключи командной строки sshd и параметры файлы конфигурации могут требовать указания времени. Оно должно указываться в виде последовательности:

```
время [единицы]
```

где время – положительное целое, единицы могут принимать следующие значения:

- ничего – секунды;

- s | S – секунды;

- m | M – минуты;
- h | H – часы;
- d | D – дни;
- w | W – недели.

Итоговое время получается в результате сложения всех выражений. Примеры:

- 600 – 600 секунд (10 минут);
- 10m – 10 минут;
- 1h30m – 1 час 30 минут (90 минут).

7.8. Настройка FTP-сервера

В состав дистрибутива ОС Альт 8 СП входит vsftpd (Very Secure FTP Daemon) – полнофункциональный FTP-сервер, позволяющий обслуживать как анонимные запросы, так и запросы от пользователей, зарегистрированных на сервере и имеющих полноценный доступ к его ресурсам.

Для установки vsftpd необходимо выполнить следующую команду:

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

7.8.1. Организация анонимного доступа на основе vsftpd

В конфигурационном файле сервера `/etc/vsftpd.conf` за разрешение анонимного доступа к серверу vsftpd отвечает параметр `anonymous_enable`, который по умолчанию имеет значение YES, т. е. анонимный доступ к серверу разрешен.

При установке vsftpd в системе автоматически создается учетная запись псевдопользователя «novsftpd». Это регистрационное имя не должно использоваться кем-либо для входа в систему, поэтому реальный пароль для него не задается. Вместо командного интерпретатора указывается `/dev/null`.

При установке пакета anonftp автоматически создается каталог, который будет корневым при анонимном подключении, – `/var/ftp` с необходимыми правами доступа. Владельцем этого каталога является пользователь root. Группой-владельцем каталога является специальная группа ftpadmin, предназначенная для администраторов FTP-сервера.

Если требуется создать в области для анонимного доступа дерево каталогов, следует в каталоге `/var/ftp/pub` установить права доступа 2775. При этом анонимным пользователям FTP-сервера будет предоставлен доступ на чтение к файлам, находящимся в каталоге. Владельцем каталога следует назначить пользователя root. В качестве группы, которой принадлежит `/var/ftp/pub`, следует назначить группу ftpadmin, включив в нее пользователей, которым необходимо изменять содержимое каталогов FTP-сервера.

Примечание. Не рекомендуется работать с содержимым от имени пользователя с идентификатором root.

Чтобы разрешить анонимным пользователям сервера доступ на запись, необходимо создать каталог `/var/ftp/incoming` с правами доступа 3773 (владелец – «ftpadmin», группа-владелец – «ftpadmin»), тем самым предоставив анонимным пользователям право записи в этот каталог, но лишив их возможности просмотра его содержимого.

7.8.2. Доступ к серверу зарегистрированных пользователей

Чтобы предоставить доступ к FTP-серверу для локально зарегистрированных пользователей, необходимо внести изменения в конфигурационный файл `/etc/vsftpd.conf`. Для этого достаточно удалить знак комментария перед директивой `local_enable=YES`. В такой конфигурации клиенты FTP-сервера получат доступ к любым каталогам файловой системы, для которых такой доступ разрешен исходя из прав соответствующих локальных пользователей. Это могут быть как домашние каталоги пользователей, так и системные каталоги. Если в настройках vsftpd разрешена запись, клиенты получат и все права на запись, которыми располагают эти пользователи.

Сервер vsftpd позволяет ограничить возможность пользователей, зарегистрированных локально, перемещаться по дереву каталогов. При этом процесс, работающий с клиентом, будет выполняться в изолированной среде (chrooted environment), и пользователь будет иметь доступ только к своему домашнему каталогу и его подкаталогам. Чтобы ограничить таким образом доступ к

каталогам для отдельных пользователей, необходимо удалить знаки комментариев у следующих строк в конфигурационном файле:

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
```

В файле `/etc/vsftpd/chroot_list` следует перечислить регистрационные имена пользователей, для которых должна использоваться изолированная среда выполнения. Можно использовать для этого и другой файл, указав его имя в строке `chroot_list_file` конфигурационного файла.

Чтобы ограничить доступ к дереву каталогов для всех пользователей, зарегистрированных локально, следует добавить в конфигурационный файл директиву `chroot_local_user=YES`.

В этом случае имена пользователей, перечисленные в файле `/etc/vsftpd/chroot_list` (при условии, что у строк, указанных выше, удалены знаки комментария), имеют противоположное действие. Для них не используется изолированная среда выполнения, и перемещение по файловой иерархии не ограничивается домашним каталогом.

Чтобы запретить анонимный доступ к FTP-серверу, необходимо поставить знак комментария в начале строки `anonymous_enable=YES` в конфигурационном файле.

7.8.3. Дополнительные сведения о настройке сервера

Сервер vsftpd способен осуществлять всю передачу данных в пассивном режиме, что сопряжено со значительно меньшим риском.

Чтобы разрешить использование только пассивного режима, достаточно удалить символ комментария у директивы `port_enable=NO` в конфигурационном файле.

Чтобы разрешить запись файлов на сервер, следует удалить знак комментария у директивы `write_enable=YES`. Этого достаточно для того, чтобы пользователи, зарегистрированные локально, получили возможность загружать файлы в те каталоги, для которых они располагают правами на запись.

Чтобы разрешить запись файлов анонимным пользователям, необходимо, кроме этого, удалить знак комментария у строки `anon_upload_enable=YES`. Специальный непривилегированный пользователь, используемый для работы с анонимными клиентами, должен иметь права на запись в один или несколько каталогов, доступных таким клиентам.

Параметры использования vsftpd (в том числе относящиеся к безопасности) могут быть заданы при помощи xinetd. Этот сервер позволяет ограничить количество одновременно выполняемых процессов как по системе в целом, так и для каждого отдельного пользователя, указать пользователя, от имени которого будет выполняться служба, задать приоритет процесса (nice), указать адреса, с которых разрешено подключение к данной службе, а также время доступа и множество других параметров.

Пример файла конфигурации xinetd для 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 = 16M # лимит адресного пространства
server = /usr/sbin/vsftpd # путь к исполняемому файлу
only_from = 192.168.0.0 # доступ из всей подсети
# доступ с указанных адресов
only_from = 207.46.197.100 207.46.197.101
# only_from = 0.0.0.0 # неограниченный по адресам доступ
access_times = 2:00-9:00 12:00-24:00 # время, доступа
}
```

7.9. Настройка служб DNS (Bind)

7.9.1. Общие сведения

Службы DNS (Bind) в ОС Альт 8 СП отвечают за преобразование доменного имени в IP-адрес и за обратную операцию.

Если локальная сеть не подключена к сети Интернет, вполне возможно, что внутренний DNS-сервер в ней не нужен. За преобразование доменного имени в IP-адрес и обратно в различные механизмы, лишь один из которых базируется на службе доменных имен. В самом простом случае имена всех компьютеров вместе с их адресами можно записать в файл `/etc/hosts`. Порядок просмотра различных пространств имен указывается в файле `/etc/nsswitch.conf`. Строка `hosts: files dns` этого файла предписывает приложениям, пользующимся стандартной функцией `gethostbyname()` сначала обратиться в `/etc/hosts`, а затем отправить запрос к DNS-серверу.

Если задачу преобразования имен в адреса взял на себя провайдер, собственный DNS-сервер также не требуется. В этом случае на всех компьютерах в качестве сервера имен указывается сервер провайдера (поле «nameserver» в файле `/etc/resolv.conf`), к которому и идут все запросы. Даже если внутренняя сеть организована согласно RFC1918 (т. н. интранет) и адреса компьютеров в ней недоступны из внешней сети, DNS-запросы во внешнюю сеть будут выполняться. Между собой компьютерам предлагается общаться с помощью `/etc/hosts` или IP-адресов.

Некоторые службы и системные утилиты, работающие с доменными именами, запускаются в ОС Альт 8 СП с использованием chroot (в каталоге `/var/resolv`), поэтому после изменения упомянутых файлов рекомендуется выполнить команду:

```
update_chrootedconf
```

Собственную службу доменных имен рекомендуется настраивать для решения задач, описанных ниже.

7.9.2. Уменьшение времени ответа на DNS-запрос абонентов внутренней сети

Если канал подключения к сети Интернет обладает большим временем задержки, то работа с данными, включающими в себя много доменных имен (например, с www-страницами) может замедлиться. Общий объем трафика при этом не вырастет, поскольку система доменных имен – распределенная база данных, поддерживающая механизм кеширования запросов. Первое обращение к кеширующему DNS-серверу приводит к выполнению рекурсивного запроса: опрашивается сервер более высокого уровня, который, если не знает ответа, передаст запрос дальше. Результат запроса сохраняется в кэше, и таким образом все последующие обращения именно к этой записи дальше кеширующего сервера не уйдут. Время жизни (Time To Live, TTL) записи в кэше определяется хозяином запрошенного доменного имени. По истечении TTL запись из кэша удаляется.

7.9.3. Именование компьютеров в интранет-сети

Решение этой задачи может потребоваться, если среди компьютеров внутренней сети есть свои серверы (например, корпоративный www-сервер), к которым другие компьютеры обращаются по доменному имени.

Поскольку адреса такой сети не пойдут дальше межсетевого экрана, допускается использовать имя какого угодно – в том числе несуществующего – домена и сделать соответствующие записи `/etc/hosts`. Поддержание в актуальном состоянии файла `/etc/hosts` на всех компьютерах – нелегкая задача, и лучше все-таки воспользоваться DNS-сервером.

7.9.4. Примеры использования DNS-сервера Bind

Решение обеих поставленных задач предоставляется настройкой DNS-сервера Bind.

В ОС Альт 8 СП сервер Bind запускается с использованием `chroot`. В `/etc` от Bind остается символьная ссылка на главный файл настроек `named.conf`. Корневым каталогом является `/var/lib/bind`, где у Bind есть собственный каталог `/etc` содержащий набор включаемых друг в друга конфигурационных файлов, каталоги `/var` и `/dev`.

Примечание. Все пути к файлам и каталогам в настройках Bind начинаются именно из этого каталога, и `/zone` соответствует `/var/lib/bind/zone`.

Чтобы запустить named в кеширующем режиме, достаточно раскомментировать и заполнить раздел настройки forwarders (вышестоящие серверы) в файле `/var/lib/bind/etc/options.conf`.

В связи с возможными ограничениями на право обращаться к серверу с обычными и рекурсивными запросами (настройки `allow-query` и `allow-recursion`), допускается раскомментировать установки по умолчанию. Эти настройки открывают доступ только абонентам локальных сетей, к которым компьютер подключен непосредственно:

```
# grep allow- /var/lib/bind/etc/options.conf
// allow-query { localnets; };
// allow-recursion { localnets; };
```

Использование Bind для полноценного именования компьютеров в локальной сети требует создания двух зон (прямой и обратной), содержащих в виде записей определенного формата информацию о доменных именах компьютеров и об их роли в этих доменах.

Каждая зона должна включать запись типа SOA (StateOfAuthority, сведения об ответственности). В этой записи определяются основные временные и административные параметры домена, в том числе электронный адрес лица, ответственного за домен (администратора) и серийный номер зоны.

Серийный номер – число в диапазоне от 0 до 4294967295 (232); каждое изменение, вносимое в зону, должно сопровождаться увеличением этого номера. Обнаружив увеличение серийного номера, кеширующие и вторичные серверы признают все закешированные записи из этой зоны устаревшими. Удобно использовать формат «годмесяцчисловерсия», где все числа, кроме года, двузначные, а версия может обнуляться раз в день, соответствовать времени (например, по формуле 100*(часы*60+минуты)/(60*24) или иметь сквозную нумерацию (в этом случае появляется сложность с переходом от версии 99 к версии 100, то есть 0). Даже если серийный номер генерируется автоматически,

рекомендуется пользоваться этим форматом, наглядно отражающим время создания зоны.

Пример зоны, не содержащей ничего, кроме записи SOA и обязательной записи типа NS (NameServer), находится в файле `/var/lib/bind/zone/empty`.

Кроме записи типа SOA, в каждой зоне должна быть хотя бы одна запись типа NS, указывающая адрес DNS-сервера, авторитативного в этом домене (как минимум – адрес сервера, на котором запущен named).

Несколько зон включаются в настройку Bind автоматически (файл `/var/lib/bind/etc/rfc1912.conf`). Они нужны для обслуживания сети, привязанной к сетевой заглушке (127.0.0.1/8). Имя домена, который обслуживается зоной, задается в файле настроек, а в самом файле зоны можно использовать относительную адресацию (без «.» в конце имени), благодаря чему операция переименования домена выполняется редактированием одной строки.

В ОС Альт 8 СП рекомендуется добавлять описания зон в конфигурационный файл `/var/lib/bind/etc/local.conf`.

Прямая зона нужна для преобразования доменного имени в IP-адрес – операции, необходимой многим программам постоянно. Большинство записей в прямой зоне – типа A (Address) – предназначены именно для этого. Другие часто встречающиеся типы записей – это CNAME (CanonicalName, настоящее имя), позволяющий привязать несколько дополнительных имен к одному, и MX (MaileXchange, обмен почтой), указывающий, куда пересылать почтовые сообщения, в поле адресат которых встречается определенное доменное имя.

Пример прямой зоны для домена `internal.domain.net` (незначащие поля соответствующих файлов заменены на «. . .»):

```
# cat /var/lib/bind/etc/local.conf
. . .
zone "internal.domain.net" {
type master;
file "internal.domain.net";
};
. . .
```

```
# cat /var/lib/bind/zone/internal.domain.net
$TTL 1D
@ IN SOA server root.server (
2013082202 ; serial
12H ; refresh
1H ; retry
1W ; expire
1H ; ncache
)
IN NS server
MX 10 server
server A 10.10.10.1
www CNAME server
mail CNAME server
jack A 10.10.10.100
jill A 10.10.10.101
```

В этом примере используются правила по умолчанию: если в записи некоторое поле опущено, оно наследуется от предыдущей. Так, вместо A допускается написать INA, а вместо MX – @ IN MX, где @ означает имя домена, указанное в конфигурационном файле.

Как видно из примера, всю работу в сети делает компьютер с адресом 10.10.10.1, он же `server.internal.domain.net`, он же `www.internal.domain.net` и `mail.internal.domain.net`. Несмотря на наличие среди CNAME этого сервера имени «mail», MX-запись указывает на действительный адрес – так рекомендовано RFC (Request for Comments, документ из серии пронумерованных информационных документов Интернета, содержащих технические спецификации и стандарты, широко применяемые во всемирной сети).

Для того чтобы преобразовывать IP-адреса в доменные имена, у каждой сети должна быть обратная зона. Если такой зоны нет, и в файле `/etc/hosts` тоже ничего не написано, операция не выполнится. Такое преобразование нужно гораздо реже и в основном по соображениям административным: для того, чтобы выяснить принадлежность компьютера (с которого, допустим, пытаются атаковать сервер) по

его IP-адресу. Некоторые почтовые серверы проверяют, содержится ли IP-адрес машины, передающей сообщение, в обратной зоне и похоже ли полученное доменное имя на то, что указано в сообщении, и при несовпадении отказываются принимать письмо.

Обратная зона состоит почти целиком из записей типа PTR (Pointer, указатель). Чтобы не умножать сущностей, решено было не вводить новый способ работы сервера имен и представить обратное преобразование IP-адреса как прямое преобразование доменного имени специального вида. Например, чтобы выяснить доменное имя компьютера с адресом «1.2.3.4», необходимо запросить информацию о доменном имени `4.3.2.1.in-addr.arpa`. Таким образом, каждой подсети класса C (или выше) соответствует определенный домен, в котором можно найти ответ.

Обратная зона для домена, приведенного выше:

```
# cat /var/lib/bind/etc/local.conf
. . .
zone "12.11.10.in-addr.arpa" {
type master;
file "12.11.10.in-addr.arpa";
};
. . .
# cat /var/lib/bind/zone/12.11.10.in-addr.arpa
$TTL 1D
@           IN           SOA           server.internal.domain.net.
root.server.internal.domain.net (
2013082201 ; serial
12H ; refresh
1H ; retry
1W ; expire
1H ; ncache
)
IN NS server.internal.domain.net.
0 PTR internal.domain.net.
1 PTR server.internal.domain.net.
100 PTR jack.internal.domain.net.
```

```
101 PTR jill.internal.domain.net.
```

Относительные адреса, использованные в левой части записей PTR, раскрываются в полные вида – `адрес.12.11.10.in-addr.arpa`, а в правой части используются полные, которые могут указывать на имена в разных доменах.

Проверить синтаксическую правильность конфигурационного файла и файла зоны можно с помощью утилит named-checkconf и named-checkzone, входящих в пакет bind. Они же используются при запуске службы командой `service bind start`.

Стоит иметь ввиду, что, в отличие от прямых зон, обратные описывают административную принадлежность компьютеров, но сами принадлежат хозяину сети (как правило, провайдеру).

Существует особого рода затруднение, связанное с работой DNS-сервера уже не во внутренней сети, а в сети Интернет. Связано это с тем, что подсети класса C (сети /24, в которых сетевая маска занимает 24 бита, а адрес компьютера – 8) выдаются только организациям, способным такую подсеть освоить (в сети класса C 254 абонентских IP-адреса, один адрес сети и один широковещательный адрес). Чаще всего выдаются совсем маленькие подсети – от /30 (на два абонентских адреса) до /27 (на 30 адресов) – или другие диапазоны, сетевая маска которых не выровнена по границе байта. Таких подсетей в обратной зоне получится несколько, а возможности просто разделить ее, отдав часть адресов в администрирование хостам, нет. Провайдер в таких случаях пользуется RFC2317, предписывающем в обратной зоне заводить не записи вида PTR, а ссылки CNAME на адреса в «классифицированных» обратных зонах специального вида. Обратное преобразование становится двухступенчатым, зато администрирование каждой классифицированной зоны можно отдать хосту.

DNS-сервер, отвечающий на запросы из глобальной сети, должен быть зарегистрирован в родительском домене. Правила требуют, чтобы при регистрации домена было указано не менее двух DNS-серверов, которые будут его обслуживать.

Из всех зарегистрированных серверов (записей типа NS в родительской зоне) только одна соответствует первичному (master) серверу, а остальные – вторичным

(slave). Для внешнего пользователя вторичный сервер не отличается от первичного, отличия состоят только в способе администрирования: все изменения вносятся в зоны первичного сервера, а вторичный только кеширует эти зоны, целиком получая их по специальному межсерверному протоколу. Полученная зона складывается в файл, редактировать который бессмысленно: первичный сервер при изменении зоны рассылает всем своим вторичным указание скачать ее заново. Право на скачивание зоны можно ограничить настройкой allow-transfer (как правило, в ней перечисляются адреса вторичных серверов).

Пример задания вторичного сервера в файле настроек:

```
// We are a slave server for eng.example.com
zone "eng.example.com" {
type slave;
file "slave/eng.example.com";
// IP address of eng.example.com master server
masters { 192.168.4.12; };
};
```

Вторичный сервер рекомендуется размещать в сети, отличной от той, в которой помещается первичный, – так повышается надежность обработки запроса (если один сервер недоступен, возможно, ответит второй) и возрастает скорость распространения записей по кэшам промежуточных серверов.

Проверку работоспособности, доступности и вообще самочувствия DNS-сервера рекомендуется выполнять утилитой dig из пакета bind-utils, которая выдает максимум информации о том, что происходило с запросом (для информации об обратном преобразовании необходимо добавить ключ `-x`):

```
dig basealt.ru

; <<>> DiG 9.10.4-P5 <<>> basealt.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32751
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;basealt.ru. IN A
;; ANSWER SECTION:
basealt.ru.  86400 IN A 194.107.17.41
;; Query time: 1177 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Mar 01 10:07:17 MSK 2017
;; MSG SIZE rcvd: 55
```

Можно также использовать утилиту host из того же пакета:

```
host basealt.ru
basealt.ru has address 194.107.17.41
```

Для выяснения административной принадлежности тех или иных доменов и сетей можно воспользоваться утилитой whois из одноименного пакета, которая обращается к специальной сетевой базе данных (не имеющей отношения к DNS).

7.10. Настройка сервера электронной почты postfix

Postfix представляет собой агент передачи электронной почты и позволяет организовать обмен почтой внутри локальной сети, а также с внешней сетью.

Для расширения возможностей Postfix используется ряд дополнений, выделенных в отдельные пакеты, полный список которых можно получить с помощью следующей команды:

```
$ apt-cache search ^postfix-
```

Настройка сервера электронной почты Postfix осуществляется с помощью конфигурационных файлов, хранящихся в каталоге `/etc/postfix`. Основные параметры определяются в файле конфигурации `main.cf`. В файле `main.cf` указываются только параметры, выставленные администратором, и некоторые из значений по умолчанию, которые администратору с большой вероятностью нужно будет изменить. Значения по умолчанию для всех остальных параметров перечислены в файле `main.cf.default` (этот файл не следует редактировать, он служит только для справок).

Если конфигурация была изменена при запущенной службе postfix, новые настройки нужно активизировать командой:

```
# service postfix reload
```

Postfix сохраняет все сообщения в журнале `mail.log`, расположенном в каталоге `/var/log/`. Сообщения об ошибках и предупреждения сохраняются отдельно в журналы `mail.err` и `mail.warn` соответственно.

Запуск Postfix осуществляется с помощью следующей команды:

```
# postfix start
```

7.10.1. Утилиты командной строки

Postfix поставляется с набором утилит командной строки, которые помогают решать административные задачи. Они выполняют разнообразные функции (обращение к картам, просмотр файлов очередей, постановка сообщений в очередь и извлечение из очереди, изменение конфигурации).

Команда `postfix` останавливает, запускает и перезагружает конфигурацию с помощью параметров `stop`, `start` и `reload`.

Команда `postalias` создает индексированную карту псевдонимов из файла псевдонимов и работает аналогично команде `postmap`, при этом уделяя особое внимание нотации в файле псевдонимов (ключ и значение разделяются двоеточием).

Команда `postcat` выводит содержимое сообщения, находящегося в почтовой очереди. Для того чтобы прочитать сообщение, находящееся в очереди, необходимо знать идентификатор очереди. Для получения списка идентификаторов очередей следует выполнить следующую команду:

```
# mailq
```

После получения идентификатора очереди необходимо указать его в качестве параметра команды postcat для просмотра содержимого файла следующим образом:

```
# postcat -q <идентификатор очереди>
```

Основная задача команды `postmap` заключается в построении индексированных карт на основе обычных текстовых файлов.

Для того чтобы создать карту `/etc/postfix/virtual.db` на основе `/etc/postfix/virtual`, необходимо выполнить следующую команду:

```
# postmap hash:/etc/postfix/virtual
```

Также команда `postmap` обеспечивает возможность тестирования карт любого вида, поддерживаемых конфигурацией Postfix.

Команда `postdrop` считывает почту из стандартного ввода и записывает результат в каталог `maildrop` (программа работает в связке с утилитой sendmail).

Команда `postkick` отправляет запрос демону Postfix по локальному транспортному каналу, делая межпроцессное взаимодействие Postfix доступным для сценариев оболочки и других программ.

Команда `postlock` предоставляет монопольный доступ к файлам `mbox`, в которые выполняет запись Postfix, а затем исполняет команду, удерживая блокировку.

Команда `postlog` позволяет внешним программам, таким как сценарии командного интерпретатора, писать сообщения в журнал электронной почты (представляет собой Postfix-совместимый интерфейс регистрации).

Команда `postqueue` представляет собой пользовательский интерфейс для очередей Postfix, предоставляющий возможности, обычно доступные в рамках выполнения команды `sendmail`.

Команда `postqueue` с параметром `-f` просит диспетчер очередей доставить всю стоящую в очереди почту вне зависимости от места назначения:

```
# postqueue -f
```

Команда `postqueue` с параметром `-p` выводит содержимое очереди:

```
# postqueue -p
```

Команда `postqueue` с параметром `-s domain` пытается доставить всю стоящую в очереди почту для домена `domain`:

```
# postqueue -s example.com
```

Команда `postsuper` обслуживает задания внутри очередей Postfix (в отличие от `postqueue`, эта команда доступна только пользователю с идентификатором root, и она может быть выполнена, когда сервер не запущен).

7.10.2. Первичная настройка

В первую очередь после установки Postfix необходимо настроить параметры, отвечающие за домен и имя сервера. Чтобы установить значение параметра `myhostname`, необходимо отредактировать конфигурационный файл `main.cf`. (для параметра `myhostname` необходимо ввести полностью определенное доменное имя хоста):

```
myhostname = mail.example.com
```

Postfix может автоматически получить значение `mydomain` после того как параметр `myhostname` настроен, для этого Postfix отбрасывает первую часть значения `myhostname` до первой точки включительно:

```
mydomain = example.com
```

Далее необходимо указать домен, с которого отправляется локальная почта. Postfix будет добавлять значение из mydomain к любому адресу, если он задан не полностью. Для этого необходимо в конфигурационном файле `main.cf` для параметра `myorigin` установить следующее значение:

```
myorigin = $mydomain
```

П р и м е ч а н и е . Сообщение от процесса cron пользователю root получит адрес `root@$mydomain`, которое будет преобразовано в `root@example.com`.

Далее необходимо указать домены, для которых данный сервер является конечной точкой доставки электронной почты. Для того чтобы Postfix принимал любую почту, адресованную в домен `example.com` необходимо в файл конфигурации внести следующие изменения:

```
mydestination = $mydomain
```

Домены, для которых сервер получает почту, отличные от значения mydomain и не сконфигурированные как виртуальные домены Postfix, необходимо перечислить с помощью параметра `mydestination`, либо в дополнительном файле, на который ссылается этот параметр.

Адресаты указываются через запятую следующим образом:

```
mydestination =
$mydomain,
$myhostname
```

Аналогичным образом параметр `mynetworks` описывает блоки IP-адресов, которые считаются внутренними и с которых разрешен прием исходящих сообщений.

После внесения изменений в конфигурацию Postfix для применения новых настроек необходимо перезапустить службу Postfix:

```
# service postfix reload
```

7.10.3. Работа в режиме SMTP-сервера

После установки служба Postfix функционирует в режиме local, в котором сервер электронной почты Postfix не принимает соединения из внешней сети, ограничиваясь приемом локальных соединений посредством сокетов семейства UNIX (UNIX-domain socket).

Для настройки возможности приема сообщений по протоколу SMTP или ESMTP, как из внешней сети, так из внутренней, необходимо переключить службу Postfix в режим работы server с помощью следующей команды:

```
control postfix server
```

Рабочие станции в локальной сети или машины в сети провайдера, отделенной от внешней сети, должны перенаправлять исходящую почту на почтовый сервер, обслуживающий данную сеть.

Для того чтобы Postfix отправлял почту из локальной сети на SMTP-сервер провайдера, необходимо для параметра relayhost установить следующее значение:

```
relayhost = [smtp.provider.net]
```

7.10.4. SMTP-аутентификация

SMTP-аутентификация обеспечивает идентификацию клиентов независимо от их IP-адресов и позволяет серверу пересылать сообщения от почтовых клиентов, чьи IP-адреса не входят в список доверенных. Postfix реализует SMTP-аутентификацию при помощи протокола SASL (Simple Authentication and Security Layer) и использует библиотеки Cyrus-SASL.

Для защиты соединений используется протокол SSL/TLS (для включения поддержки необходимо установить пакет postfix-tls).

Для проверки поддержки SMTP-аутентификации Postfix необходимо от имени от имени администратора (root) выполнить следующую команду:

```
ldd ‘postconf h daemon_directory’/smtpd
```

Если в выводе команды присутствует строка `libsasl.so.2`, значит, пакет Postfix был собран с поддержкой SASL.

7.10.4.1. Настройки SMTP-аутентификации на сервере

Настройка SMTP-аутентификации на сервере осуществляется в несколько этапов:

1) включение SMTP-аутентификации на серверной части;
2) настройка механизмов SASL, которые будут предоставляться клиентам;
3) настройка поддержки SMTP-аутентификации для нестандартных почтовых клиентов;
4) настройка области (realm), которую Postfix будет передавать библиотеке SASL;
5) определение разрешения на пересылку в Postfix.

Чтобы включить SMTP-аутентификацию, необходимо в конфигурационный файл `main.cf` добавить следующую запись:

```
smtpd_sasl_auth_enable = yes
```

7.10.4.1.1. Настройка механизмов SASL

Управление предоставляемыми механизмами осуществляется с помощью параметра `smtpd_sasl_security_options`, в котором через запятые следует указать список из одного или более значений:

1) noanonymous – значение параметра, позволяющее включить проверку сервером верительных данных клиента (список значений параметра `smtpd_sasl_security_options` всегда должен включать в себя значение noanonymous);
2) noplaintext – значение параметра, позволяющее исключить использование всех механизмов открытого текста, таких как PLAIN и LOGIN (значение, рекомендуемое для использования, так как отправляемые открытым текстом верительные данные могут быть легко перехвачены в сети);

3) noactive – значение параметра, исключающее использование механизмов SASL, которые восприимчивы к активным атакам);
4) nodictionary – значение параметра, исключающее все механизмы, не устойчивые к атакам по словарю (атаки, осуществляемые методом полного перебора паролей);
5) mutual_auth – значение параметра, позволяющее включить поддержку только механизмов, обеспечивающих взаимную аутентификацию (сервер аутентифицирует себя для клиента).

7.10.4.1.2. Настройка SMTP-аутентификации для нестандартных почтовых клиентов

Для настройки альтернативной нотации для устаревших клиентов, не распознающих SMTP-аутентификацию по стандарту RFC 2222, но распознающих более раннюю нотацию, использованную в черновом варианте этого стандарта (где между командой `AUTH` и названиями механизмов стоял не пробел, а знак равенства), необходимо в конфигурационном файле `main.cf` установить параметр `broken_sasl_auth_clients`:

```
broken_sasl_auth_clients = yes
```

7.10.4.1.3. Настройка области SASL

Для аутентификации клиента сервер Postfix отправляет службе паролей Cyrus SASL область аутентификации (realm) вместе с верительными данными клиента. Такая необходимость определяется версией Cyrus SASL и выбором службы. Для указания области аутентификации в файле `main.cf` используется параметр `smtpd_sasl_local_domain`. По умолчанию этот параметр пуст и должен оставаться пустым, если только не используется вспомогательный плагин, которому действительно требуется область аутентификации.

7.10.4.1.4. Настройка разрешений на пересылку

Для разрешения пересылки для клиентов, прошедших аутентификацию SASL, необходимо добавить параметр `permit_sasl_authenticated` в список ограничений `smtpd_recipient_restrictions` своей конфигурации следующим образом:

```
smtpd_recipient_restrictions =
```

```
[...]
permit_sasl_authenticated,
permit_mynetworks,
reject_unauth_destination
[...]
```

Необходимо поместить ключевое слово `permit_sasl_authenticated` достаточно близко к началу списка ограничений, чтобы аутентифицированный клиент не был случайно отвергнут из-за несоответствия какому-то другому правилу (например, `reject_unauth_destination`).

7.10.4.2. Настройка SMTP-аутентификации на стороне клиента

Для настройки SMTP-аутентификации для клиента необходимо выполнить следующее:

1) запросить у удаленного сервера список поддерживаемых механизмов аутентификации;
2) включить SMTP-аутентификацию на клиентской части;
3) предоставить файл для хранения верительных данных;
4) настроить Postfix на работу с файлом верительных данных;
5) отключить ненадежные механизмы аутентификации.

Клиентская ПЭВМ должна поддерживать механизмы аутентификации, поддерживаемые сервером. Для получения списка механизмов аутентификации необходимо подключиться к почтовому серверу и отправить приветствие `EHLO` с помощью следующих команд:

```
$ telnet mail.remoteexample.com 25
EHLO mail.example.com
```

По умолчанию SMTP-аутентификация на стороне клиента выключена. Для того чтобы включить SMTP-аутентификацию необходимо в конфигурационный файл `main.cf` добавить следующую запись:

```
smtp_sasl_auth_enable = yes
```

После включения аутентификации на клиентской ПЭВМ необходимо сообщить серверу Postfix, где следует искать секретные данные, необходимые для

аутентификации, и какой из механизмов (из предлагаемых удаленным сервером) Postfix может использовать.

7.10.4.2.1. Хранение верительных данных

Необходимо подготовить данные, которые клиент Postfix будет использовать для того, чтобы аутентифицировать себя на сервере, для этого следует создать от имени root файл карты `/etc/postfix/sasl_passwd` (если он еще не существует) с помощью следующей команды:

```
# touch /etc/postfix/sasl_passwd
```

Далее необходимо отредактировать этот файл, поместив полностью определенное доменное имя почтового сервера, который требует аутентификации, с левой стороны, а разделенную двоеточием пару «имя пользователя – пароль» – с правой. Для имен пользователей `mail.example.com` и `relay.another.example.com`, а также соответствующих паролей файл `sasl_passwd` будет выглядеть следующим образом:

```
mail.example.com test:testpass
relay.another.example.com username:password
```

После редактирования файла `sasl_passwd` необходимо изменить права на него так, чтобы читать его мог только пользователь root (в файле хранится конфиденциальная информация, которая не должна быть доступна локальным пользователям), для этого необходимо использовать команды `chown` и `chmod`:

```
# chown root:root /etc/postfix/sasl_passwd && chmod 600
/etc/postfix/sasl_passwd
```

Затем необходимо преобразовать файл карты в индексированную карту для быстрого доступа Postfix (необходимо выполнять при каждом изменении файла `sasl_passwd`) с помощью следующей команды:

```
# postmap hash:/etc/postfix/sasl_passwd
```

7.10.4.2.2. Настройка Postfix для использования верительных данных

Необходимо сообщить клиенту Postfix, где хранится созданная карта верительных данных аутентификации; для этого в параметре `smtp_sasl_password_maps` в файле `main.cf` указать полный путь к файлу

sasl_passwd, указывая при этом (с помощью спецификатора hash:), что значения карты хранятся в хеш-файле, например:

```
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
```

7.10.4.2.3. Отключение некоторых механизмов аутентификации

Для отключения использования ненадежных механизмов, следует указать в параметре smtp_sasl_security_options список (через запятую) типов механизмов, которые клиент не может использовать. По умолчанию параметр smtp_sasl_security_options установлен в значение «noanonymous», но по возможности (если сервер поддерживает механизм с шифрованием, такой как DIGEST-MD5 или CRAM-MD5) следует также отключить использование механизмов открытого текста. Для этого необходимо добавить в файл main.cf следующую строку:

```
smtp_sasl_security_options = noanonymous, noplaintext
```

7.10.5. Триггеры ограничений

Ограничения позволяют почтовому серверу принять или отвергнуть сообщения на основе данных SMTP-соединения между клиентом и сервером. Информация, полученная из этого диалога, позволяет Postfix наложить или отменить ограничения на клиента (отправителя и получателя).

Postfix поддерживает следующие триггеры:

1) smtpd_client_restrictions – триггер применяется к IP-адресу или имени хоста клиента либо к ним обоим (по умолчанию Postfix разрешает подключение любому клиенту);
2) smtpd_helo_restrictions – триггер применяется к аргументу HELO/EHLO клиента и к IP-адресу и (или) имени хоста клиента (по умолчанию допускается любой аргумент HELO/EHLO);
3) smtpd_sender_restrictions – набор триггеров, который относится к частям конверта (Postfix применяет его к отправителю конверта, аргументу HELO/EHLO и клиенту, по умолчанию любому отправителю конверта разрешено отправлять сообщения);

4) `smtpd_recipient_restrictions` – триггер применяется к получателям конверта, отправителю конверта, аргументу HELO/EHLO и к IP-адресу и (или) имени хоста клиента (по умолчанию Postfix допускает любых получателей для клиентов, которые определены в параметре конфигурации `mynet-works`, для остальных же разрешены получатели в доменах из `relay_domains` и `mydomains`);
5) `smtpd_data_restrictions` – триггер выявляет клиенты, которые отправляют содержимое письма прежде, чем Postfix ответит на команду DATA (Postfix выполняет это посредством трассировки DATA, когда клиент отправляет команду на сервер, по умолчанию ограничения нет);
6) `smtpd_etrn_restrictions` – специальный триггер может ограничить клиенты, которые могут запрашивать у Postfix очистку очереди сообщений (по умолчанию всем клиентам разрешено выдавать команду ETRN).

В Postfix существуют несколько видов ограничений, которые можно разбить на четыре группы:

1) общие ограничения;
2) переключаемые ограничения;
3) настраиваемые ограничения;
4) дополнительные параметры контроля спама.

Общие ограничения выполняют следующие команды:

1) `permit` – разрешает запрос;
2) `defer` – откладывает запрос;
3) `reject` – отвергает запрос;
4) `warn_if_reject` – содействует последующим ограничениям (если ограничение после `warn_if_reject` решает отвергнуть запрос, то Postfix записывает в журнал сообщение reject_warning);
5) `reject_unauth_pipelining` – отвергает запрос, когда клиент отправляет команды SMTP раньше времени, еще не зная о том, действительно ли Postfix поддерживает конвейерную обработку команд ESMTP (таким

образом, достигается противодействие программам массовой рассылки, которые некорректно используют конвейерную обработку команд ESMTP для ускорения доставки).

Переключаемые ограничения работают как переключатели, при активации которых они проверяют выполнение некоторого условия. К переключаемым ограничениям относятся следующие:

1) `smtpd_helo_required` – ограничение, требующее от клиентов отправки команды HELO (или EHLO) в начале сеанса SMTP (наличия команды HELO/EHLO требуют RFC 821 и RFC 2821);
2) `strict_rfc821_envelopes` – ограничение, регулирующее степень терпимости Postfix к ошибкам в адресах, указанных в команде MAIL FROM (отправитель конверта) или RCPT TO;
3) `disable_vrfy_command` – SMTP-команда VRFY позволяет клиентам проверять существование получателя (ограничение позволяет отменить команды VRFY);
4) `allow_percent_hack` – ограничение, регулирующее преобразование из формы «user%domain» в «user@domain»;
5) `swap_bangpath` – ограничение, контролирующее преобразование из формы «site!user» в «user@site» (необходимо, если ПЭВМ подключена к сети UUCP).

Настраиваемые ограничения представляют собой карты, которые работают как фильтры. В каждой записи карты ключ является фильтром, а значение – тем действием, которое необходимо выполнить при совпадении:

1) HELO (EHLO) имя хоста – ограничения, относящиеся к именам хостов, которые клиенты могут отправлять с командой HELO или EHLO;
2) имя хоста/адрес клиента – ограничения, определяющие клиенты, которые могут устанавливать SMTP-соединения с почтовым сервером;
3) адрес отправителя – ограничения, определяющие адреса отправителей (конвертов), которые Postfix разрешает для использования в командах MAIL FROM;

4) адрес получателя – ограничения, определяющие адреса получателей (конвертов), которые Postfix разрешает для использования в командах RCPT TO;
5) ETRN!команды – ограничение, накладываемое на клиенты, которые могут выдавать команды ETRN;
6) проверка заголовка – ограничение, регулирующее заголовки сообщений;
7) проверка тела – ограничения, накладываемые на текст, который может появляться в строках тела сообщения;
8) черные списки DNSBL – черные списки, ограничивающие соединения от IP-адресов (клиентов), которые включены в черные списки DNSBL;
9) черные списки RHSBL – черные списки, запрещающие те домены отправителей (конверта), которые присутствуют в черных списках RHSBL.

Дополнительные параметры контроля спама поддерживают другие ограничения или возможности, не входящие в функциональность Postfix по умолчанию:

1) `default_rbl_reply` – создает шаблон ответа по умолчанию, который будет использоваться при блокировании запроса SMTP-клиента ограничением `reject_rbl_client` или `reject_rhsbl_sender`;
2) `permit_mx_backup_networks` – ограничивает использование функции контроля за пересылкой `permit_mx_backup` теми адресатами, у которых основные хосты MX входят в указанный список сетей;
3) `rbl_reply_maps` – определяет таблицы поиска и шаблоны ответов DNSBL, индексированные по имени домена DNSBL;
4) `relay_domains` – указывает Postfix на необходимость приема почты для этих доменов несмотря на то, что данный сервер не является местом их конечного назначения;
5) `smtpd_sender_login_maps` – определяет пользователя, которому разрешено использовать определенный адрес MAIL FROM.

В Postfix по умолчанию встроен набор ограничений. Для того чтобы посмотреть список ограничений необходимо выполнить следующую команду:

```
# postconf -d smtpd_recipient_restrictions
```

Для включения режима фильтрации почты в Postfix в зависимости от наличия в них нежелательной информации (спам) необходимо выполнить следующую команду:

```
control postfix filter
```

7.10.6. Алиасы и преобразование адресов

В Postfix для передачи сообщений электронной почты используются алиасы, которые позволяют создавать псевдонимы для длинных или плохо запоминаемых адресов электронной почты. Настройка алиасов в Postfix осуществляется с помощью таблиц aliases.

При установке Postfix в таблице создается алиас на имя пользователя root: вся корреспонденция, предназначенная администратору и поступающая на другие системные адреса, будет отправляться на имя реального пользователя, который осуществляет функции администратора.

Рабочий образ таблицы строится с помощью следующей команды:

```
newaliases
```

а также при актуализации всех изменений посредством следующей команды:

```
service postfix reload
```

При отправке сообщения Postfix формирует адрес отправителя автоматически из имени учетной записи пользователя и значения собственного домена (или значения «myorigin»). Преобразование адресов отправителей в глобальные адреса задаются в таблице типа canonical:

```
sender_canonical_maps = cdb:/etc/postfix/sender_canonical
```

Аналогичная таблица `recipient_canonical` и соответствующий параметр `recipient_canonical_maps` могут быть использованы для преобразования адресов назначения.

7.10.7. Настройка ограничений размера почтового ящика и отправляемого сообщения

По умолчанию размер файла почтового ящика при локальной доставке ограничен 51 200 000 байтами. Это ограничение можно изменить с помощью параметра `mailbox_size_limit`.

Например, снять ограничение можно установив этот параметр в 0:

```
mailbox_size_limit = 0
```

Также можно установить требуемый размер, указав в значении параметра необходимую величину:

```
mailbox_size_limit = <размер почтового ящика в байтах>
```

Для настройки размера отправляемого сообщения используется параметр `message_size_limit`:

```
message_size_limit = <размер сообщения в байтах>
```

Для настройки виртуальных аккаунтов используется параметр `virtual_mailbox_limit`:

```
virtual_mailbox_limit= <размер почтового ящика виртуального
аккаунта в байтах>
```

7.11. Настройка кэширующего прокси-сервера (Squid)

Для обеспечения контролируемого доступа ПЭВМ локальной сети к сети Интернет в составе ОС Альт 8 СП используется прокси-сервер Squid.

Для обеспечения возможности использования ПЭВМ, на которую установлен Squid, в качестве прокси-сервера необходимо настроить таблицы управления доступом (Access Control Lists, далее – ACL), которые хранятся в конфигурационном файле `squid.conf` в директории `/etc/squid/`.

Для того чтобы сервер Squid принимал соединения из всей внутренней сети, необходимо в раздел `# TAG: acl` включить следующую запись:

```
acl our_networks src <адреса внутренней сети>
http_access allow our_networks
```

При настройке таблиц управления доступом следует учитывать, что при обработке запроса на доступ к серверу Squid все строки `http_access` файла

squid.conf просматриваются последовательно сверху вниз до первой строки, соответствующей параметрам запроса.

7.11.1. Настройка прозрачного доступа через прокси-сервер

Для настройки прозрачного доступа пользователей локальной сети к сети Интернет через прокси-сервер необходимо выполнить настройку фильтра адресов, для этого необходимо выполнить команду iptables, перенаправляющую HTTP-запросы к внешним серверам на порт Squid:

```
# iptables -t NAT -A PREROUTING -d ! <прокси-сервер> \
-i <внутренний _интерфейс> -p tcp -m tcp --dport 80 \
-j REDIRECT --to-ports 3128
```

Также можно выполнить альтернативную команду:

```
# iptables -t nat -A PREROUTING -p tcp -d 0/0 --dport www \
-i <внутренний_сетевой_интерфейс> -j DNAT \
-to <локальный_адрес_на_котором_слушает_прокси>:3128
```

Настройка squid.conf при этом использует обратное проксирование. Далее необходимо добавить в конфигурационный файл squid.conf следующую строку:

```
http_port 80 transparent
```

7.11.2. Фильтрация доступа

В Squid существует гибкая схема фильтрации внешних ссылок, с помощью которой предоставляется возможность ограничить (запретить) доступ к определенным сетевым ресурсам. Содержимое фильтруется с помощью таблиц управления доступом ACL и настроек http_access deny, примеры которых приведены в конфигурационном файле squid.conf. При задании фильтруемого URL или доменного имени сервера можно использовать регулярные выражения, определяя в одной строке фильтр для целого класса адресов или доменных имен. Запрет доступа к домену baddomain.com, например, можно оформить следующим образом:

```
acl Bad dstdomain baddomain.com
http_access deny Bad
```

7.11.3. Авторизация доступа

Squid позволяет настраивать таблицы доступа ACL индивидуально для пользователей и (или) категорий пользователей. Если для определения того, какой именно пользователь подключается к серверу, недостаточно IP-адреса его компьютера, следует использовать схемы авторизации, принятые в Squid. Авторизация конфигурируется с помощью тега `TAG: auth_param`. Схемы (программы) авторизации, поддерживаемые Squid, хранятся в каталоге `/usr/lib/squid`.

Для настройки аутентификации в LDAP можно использовать следующую конфигурацию:

```
auth_param basic program /usr/lib/squid/squid_ldap_auth -b
ou=People,dc=office,dc=lan -f (uid=%s) -h ldap.office.lan
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
```

7.11.4. Кэширование данных

Squid обеспечивает возможность кэширования данных, полученных по запросам из сети Интернет (при повторных запросах данные извлекаются из сохраненной копии).

Настройка правил кэширования данных осуществляется с помощью таблиц доступа ACL, а также с помощью настройки конфигурационного файла `squid.conf`. Для отключения функции кэширования данных необходимо использовать параметр `always_direct`, для включения принудительного кэширования – `never_direct`.

Например, чтобы запретить кэширование данных, получаемых по протоколу FTP, необходимо в конфигурационный файл `squid.conf` добавить следующие строки:

```
acl FTP proto FTP
always_direct allow FTP
```

Squid поддерживает возможность обмена данными с кэшем авторизованного сервера (`parent peer` (родительский прокси-сервер) `/sibling peer` (братский

прокси-сервер)), например, если запрашиваемый ресурс в локальном кэше Squid не найден.

7.11.5. Настройка режима работы в качестве обратного прокси-сервера

Squid поддерживает режим работы в качестве обратного прокси-сервера. Работа в таком режиме обеспечивает ретрансляцию запросов из внешней сети на один или несколько серверов, логически расположенных во внутренней сети, и позволяет скрыть реальное расположение и структуру серверов, а также уменьшить нагрузку на них.

Для настройки сервера Squid для работы в качестве единственного обратного прокси-сервера, принимающего HTTP-запросы из внешней сети, необходимо в конфигурационный файл `squid.conf` добавить следующие строки:

```
http_port 80 defaultsite=internal.www.com
cache_peer <имя сервера> parent 80 <порт ICP> no-query originserver
```

Примечания:

1. В примере в качестве порта, принимающего запросы из внешней сети по протоколу HTTP, используется порт 80.

2. Так как сервер Squid играет роль единственного обратного прокси-сервера, необходимо выключить ICP, указав в качестве порта ICP значение 0.

3. parent (родительский прокси-сервер) – тип прокси-сервера в соответствии с иерархией серверов.

Для обратного проксирования нескольких внутренних серверов необходимо, чтобы внешние запросы к ресурсам сети Интернет с разными доменными именами попадали на вход Squid, который бы ставил в соответствие каждому имени действительный адрес сервера во внутренней сети и в соответствии с этим перенаправлял запрос. Делается это с помощью механизма виртуальных хостов.

Для организации прокси для двух серверов (`www1.foo.bar` и `www2.foo.bar`), адреса которых в DNS указывают на машину со Squid-сервером необходимо в конфигурационный файл `squid.conf` добавить следующую запись:

```
http_port 80 defaultsite=www1.foo.bar vhost
```

```
hosts_file /etc/hosts
```

Настройка defaultsite используется сервером для заполнения HTTP-заголовков. Для преобразования доменных имен в адреса серверов во внутренней сети следует использовать файл `/etc/hosts`:

```
10.0.0.1 www1.foo.bar
10.0.0.2 www2.foo.bar
```

7.11.6. Сбор статистики и ограничение полосы доступа

В состав Squid входит утилита кэш-менеджер, предназначенная для отображения статистики и загрузки сервера. Кэш-менеджер представляет собой cgi-приложение и должен выполняться под управлением сконфигурированного HTTP-сервера. Все настройки кэш-менеджера выполняются с помощью конфигурирования файла `squid.conf` (строки, которые относятся к кэш-менеджеру, обычно включают cachemgr).

Squid также обеспечивает возможность ограничения полосы пропускания для пользователей (для этого используются параметры `delay_pools` и `delay_class`).

7.11.7. Кеширование DNS-запросов

Squid содержит встроенный минисервер запросов DNS. Он выступает как посредник между Squid и внешними DNS-серверами. При запуске Squid производит начальное тестирование доступности DNS (можно отключить, используя опцию `–D`). Время кеширования удачного DNS-запроса по умолчанию составляет шесть часов.

7.12. Настройка фильтрации пакетов с помощью утилиты iptables

Утилита iptables – стандартный интерфейс командной строки для управления фильтрацией сетевых пакетов и сбора статистики сетевого взаимодействия.

Утилита iptables позволяет фильтровать сетевые пакеты по следующим параметрам:

- на основе сетевых адресов отправителя и получателя (IP-адреса, MAC-адреса);
- по протоколам tcp, udp, icmp;

- с учетом входного и выходного сетевого интерфейса;

- на основе используемого порта;

- с учетом даты и времени.

Фильтры состоят из правил. Каждое правило – это строка, содержащая в себе критерии, определяющие, подпадает ли пакет под заданное правило, и действие, которое необходимо выполнить в случае удовлетворения критерия.

7.12.1. Устройство фильтра iptables

Для iptables в общем виде правила выглядят так:

```
iptables [-t table] command [match] [target/jump]
```

Если в правило не включается спецификатор `[-t table]`, то по умолчанию предполагается использование таблицы filter, если же предполагается использование другой таблицы, то это требуется указать явно. Спецификатор таблицы так же можно указывать в любом месте строки правила, однако более или менее стандартом считается указание таблицы в начале правила.

Непосредственно за именем таблицы должна стоять команда управления фильтром. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие iptables (вставить правило, добавить правило в конец цепочки, или удалить правило). Тело команды в общем виде выглядит так:

```
[команда] [цепочка]
```

Ключ команда указывает на то, что нужно сделать с правилом, например, команда `-A` указывает на то, что правило нужно добавить в конец указанной цепочки.

Цепочка указывает, в какую цепочку нужно добавить правило. Стандартные цепочки – INPUT, OUTPUT, FORWARD, PREROUTING и POSTROUTING. Они находятся в таблицах фильтра. Не все таблицы содержат все стандартные цепочки. Подробнее таблицы и цепочки описаны ниже.

Раздел `[match]` задает критерии проверки, по которым определяется, подпадает ли пакет под действие этого правила или нет. Здесь можно указать самые разные критерии – IP-адрес источника пакета или сети, сетевой интерфейс.

Раздел `[target]` указывает, какое действие должно быть выполнено при условии выполнения критериев в правиле. Здесь можно передать пакет в другую цепочку правил, «сбросить» пакет и забыть про него, выдать на источник сообщение об ошибке и т. д.

Когда пакет приходит на сетевое устройство, он обрабатывается соответствующим драйвером и далее передается в фильтр в ядре ОС. Далее пакет проходит ряд таблиц и затем передается либо локальному приложению, либо переправляется на другую машину (рис. 44).

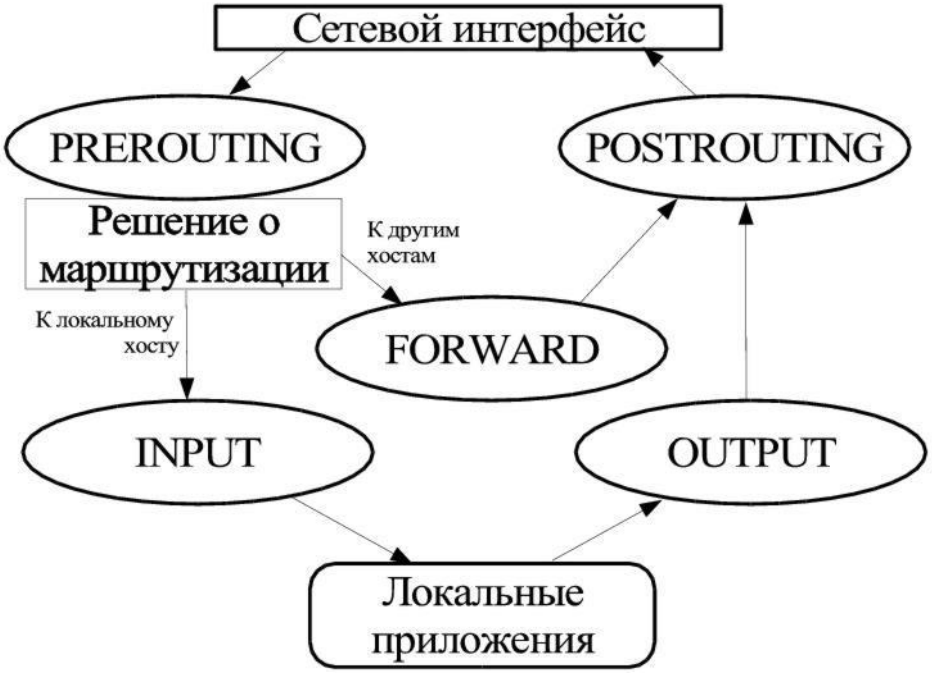

Рис. 44 – Схема движения пакетов в iptables

7.12.2. Встроенные таблицы фильтра iptables

По умолчанию используется таблица filter. Опция `-t` в правиле указывает на используемую таблицу. С ключом `-t` можно указывать следующие таблицы: nat, mangle, filter.

7.12.2.1. Таблица nat

Таблица nat используется главным образом для преобразования сетевых адресов Network Address Translation. Через эту таблицу проходит только первый пакет из потока. Преобразования адресов автоматически применяется ко всем

последующим пакетам. Таблица имеет три цепочки PREROUTING, OUTPUT и POSTROUTING:

- цепочка PREROUTING используется для внесения изменений в пакеты на входе в фильтр;
- цепочка OUTPUT используется для преобразования пакетов, созданных приложениями внутри компьютера, на котором установлен фильтр, перед принятием решения о маршрутизации;
- цепочка POSTROUTING используется для преобразования пакетов перед выдачей их в сеть.

7.12.2.2. Таблица mangle

Таблица mangle используется для внесения изменений в заголовки пакетов. Примером может служить изменение поля TTL, TOS или MARK. Таблица имеет две цепочки PREROUTING и OUTPUT:

- цепочка PREROUTING используется для внесения изменений на входе в фильтр перед принятием решения о маршрутизации;
- цепочка OUTPUT – для внесения изменений в пакеты, поступающие от внутренних приложений. Таблица mangle не должна использоваться для преобразования сетевых адресов (Network Address Translation) или маскарадинга (masquerading), для этих целей имеется таблица nat.

7.12.2.3. Таблица filter

Таблица filter используется, главным образом, для фильтрации пакетов.

Таблица имеет три цепочки – FORWARD, INPUT, OUTPUT:

- цепочка FORWARD используется для фильтрации пакетов, идущих транзитом через фильтрующий компьютер;
- цепочка INPUT предназначена для обработки входящих пакетов, направляемых локальным приложениям фильтрующего компьютера;
- цепочка OUTPUT используется для фильтрации исходящих пакетов, сгенерированных локальными приложениями фильтрующего компьютера.

7.12.3. Команды утилиты iptables

В таблице 6 приведены команды, которые используются в iptables.

Т а б л и ц а 6 – Команды утилиты iptables

Команда	Пример	Пояснения
`-A, --append`	`iptables -A INPUT`	Добавляет новое правило в конец заданной цепочки.
`-D, --delete`	`iptables -D INPUT --dport 80 -j DROP` `iptables -D INPUT 1`	Удаление правила из цепочки. Команда имеет два формата записи, первый – когда задается критерий сравнения с опцией `-D` (см. первый пример), второй – порядковый номер правила. Если задается критерий сравнения, то удаляется правило, которое имеет в себе этот критерий, если задается номер правила, то будет удалено правило с заданным номером. Счет правил в цепочках начинается с 1.
`-R, --replace`	`iptables -R INPUT 1 -s 192.168.0.1 -j DROP`	Данная команда заменяет одно правило другим. Используется в основном во время отладки новых правил.
`-I, --insert`	`iptables -I INPUT 1 -dport 80 -j ACCEPT`	Вставляет новое правило в цепочку. Число, следующее за именем цепочки, указывает номер правила, перед которым нужно вставить новое правило, другими словами число (задает номер для вставляемого правила. В примере, указывается, что данное правило должно быть 1-м в цепочке INPUT).

Продолжение таблицы 6

Команда	Пример	Пояснения
`-L, --list`	`iptables -L INPUT`	Вывод списка правил в заданной цепочке, в данном примере предполагается вывод правил из цепочки INPUT. Если имя цепочки не указывается, то выводится список правил для всех цепочек. Формат вывода зависит от наличия дополнительных ключей в команде, например `-n`, `-v`, и пр.
`-F, --flush`	`iptables -F INPUT`	Удаление всех правил из заданной цепочки (таблицы). Если имя цепочки и таблицы не указывается, то удаляются все правила, во всех цепочках.
`-Z, --zero`	`iptables -Z INPUT`	Обнуление всех счетчиков в заданной цепочке. Если имя цепочки не указывается, то подразумеваются все цепочки. При использовании ключа `-v` совместно с командой `-L`, на вывод будут поданы и состояния счетчиков пакетов, попавших под действие каждого правила. Допускается совместное использование команд `-L` и `-Z`. В этом случае будет выдан сначала список правил со счетчиками, а затем произойдет обнуление счетчиков.
`-N, --new-chain`	`iptables -N allowed`	Создается новая цепочка с заданным именем в заданной таблице. В приведенном выше примере создается новая цепочка с именем allowed. Имя цепочки должно быть уникальным и не должно совпадать с зарезервированными именами цепочек и действий (DROP, REJECT и т. п.).
`-X, --delete-chain`	`iptables -X allowed`	Удаление заданной цепочки из заданной таблицы. Удаляемая цепочка не должна иметь правил и не должно быть ссылок из других цепочек на удаляемую цепочку. Если имя цепочки не указано, то будут удалены все цепочки, определенные командой `-N` в заданной таблице.

Окончание таблицы 6

Команда	Пример	Пояснения
`-P, --policy`	`iptables -P INPUT DROP`	Определяет политику по умолчанию для заданной цепочки. Политика по умолчанию определяет действие, применяемое к пакетам, не попавшим под действие ни одного из правил в цепочке. В качестве политики по умолчанию допускается использовать DROP, ACCEPT и REJECT.
`-E, --rename-chain`	`iptables -E allowed disallowed`	Команда `-E` выполняет переименование пользовательской цепочки. В примере цепочка allowed будет переименована в цепочку disallowed. Эти переименования не изменяют порядок работы, а носят только косметический характер.

Команда должна быть указана всегда. Список доступных команд можно просмотреть с помощью команды `iptables -h` или, что то же самое, `iptables --help`. Некоторые команды могут использоваться совместно с дополнительными ключами.

7.12.4. Ключи утилиты iptables

В таблице 7 приводится список дополнительных ключей и описывается результат их действия.

Т а б л и ц а 7 – Ключи утилиты iptables

Ключ	Пример	Пояснения
`-x, --exact`	`--list`	Для всех чисел в выходных данных выводятся их точные значения без округления и без применения множителей K, M, G.
`-n, --numeric`	`--list`	Iptables выводит IP-адреса и номера портов в числовом виде, предотвращая попытки преобразовать их в символические имена.
`--line-numbers`	`--list`	Включает режим вывода номеров строк при отображении списка правил.

Окончание таблицы 7

Ключ	Пример	Пояснения
`-v, --verbose`	`--list, --append, --insert, --delete, --replace`	Используется для повышения информативности вывода и, как правило, используется совместно с командой `--list`. В случае использования с командой `--list`, в вывод этой команды включаются: имя интерфейса, счетчики пакетов и байт для каждого правила. Формат вывода счетчиков предполагает вывод кроме цифр числа еще и символьные множители K (x1000), M (x1,000,000) и G (x1,000,000,000). Для того чтобы заставить команду `--list` выводить полное число (без употребления множителей) требуется применять ключ `-x`. Если ключ `-v`, `--verbose` используется с командами `--append`, `--insert`, `--delete` или `--replace`, то на вывод будет выдан подробный отчет о произведенной операции.
`-c, --set-counters`	`--insert, --append, --replace`	Используется при создании нового правила для установки счетчиков пакетов и байт в заданное значение. Например, ключ `--set-counters 20 4000` установит счетчик пакетов = 20, а счетчик байт = 4000.
`--modprobe`	Любая команда	Определяет команду загрузки модуля ядра.

7.12.5. Основные действия над пакетами в фильтре iptables

В таблице 8 приведены доступные над пакетами действия.

Т а б л и ц а 8 – Действия над пакетами iptables

Действие	Пояснения
ACCEPT	Пакет прекращает движение по цепочке (и всем вызвавшим цепочкам, если текущая цепочка была вложенной) и считается принятым, тем не менее, пакет продолжит движение по цепочкам в других таблицах и может быть отвергнут там.
DROP	Отбрасывает пакет и iptables «забывает» о его существовании. Отброшенные пакеты прекращают свое движение полностью.
RETURN	Прекращает движение пакета по текущей цепочке правил и производит возврат в вызывающую цепочку, если текущая цепочка была вложенной, или, если текущая цепочка лежит на самом верхнем уровне (например, INPUT), то к пакету будет применена политика по умолчанию.

Окончание таблицы 8

Действие	Пояснения
LOG	Служит для журналирования отдельных пакетов и событий. В системный журнал могут заноситься заголовки IP пакетов, и другая интересующая вас информация.
REJECT	Используется, как правило, в тех же самых ситуациях, что и DROP, но в отличие от DROP, команда `REJECT` выдает сообщение об ошибке на хост, передавший пакет.
SNAT	Используется для преобразования сетевых адресов (Source Network Address Translation), т. е. изменение исходящего IP адреса в IP заголовке пакета.
DNAT	Destination Network Address Translation используется для преобразования адреса места назначения в IP заголовке пакета.
MASQUERADE	В основе своей представляет то же самое, что и SNAT только не имеет ключа `--to-source`. Причиной тому то, что маскарадинг может работать, например, с dialup подключением или DHCP, т. е. в тех случаях, когда IP адрес присваивается устройству динамически. Если используется динамическое подключение, то нужно использовать маскарадинг, если же используется статическое IP подключение, то лучшим выходом будет использование действия SNAT.
REDIRECT	Выполняет перенаправление пакетов и потоков на другой порт той же самой машины. К примеру, можно пакеты, поступающие на HTTP порт перенаправить на порт HTTP proxy. Действие REDIRECT очень удобно для выполнения «прозрачного» проксирования (transparent proxy), когда компьютеры в локальной сети даже не подозревают о существовании прокси.
TTL	Используется для изменения содержимого поля «время жизни» (Time To Live) в IP заголовке. Один из вариантов применения этого действия – это устанавливать значение поля «Time To Live» во всех исходящих пакетах в одно и то же значение. Если установить на все пакеты одно и то же значение TTL, то тем самым можно лишить провайдера одного из критериев определения того, что подключение к Интернету разделяется между несколькими компьютерами. Для примера можно привести число «TTL = 64», которое является стандартным для ядра Linux.

7.12.6. Основные критерии пакетов в фильтре iptables

В таблице 9 приведены возможные критерии для фильтрации пакетов в фильтре iptables.

Т а б л и ц а 9 – Критерии пакетов в фильтре iptables

Критерий	Пояснения
`-p, --protocol`	Используется для указания типа протокола. Примерами протоколов могут быть TCP, UDP и ICMP. Список протоколов можно посмотреть в файле `/etc/protocols`. Прежде всего, в качестве имени протокола в данный критерий можно передавать три вышеупомянутых протокола, а также ключевое слово ALL. В качестве протокола допускается передавать число – номер протокола.
`-s, --src, --source`	IP-адрес(а) источника пакета. Адрес источника может указываться без маски или префикса (например, 192.168.1.1), тогда подразумевается единственный IP-адрес. Можно указать адрес в виде `address/mask`, например, как 192.168.0.0/255.255.255.0, или более современным способом 192.168.0.0/24, т. е. фактически определяя диапазон адресов. Символ «!», установленный перед адресом, означает логическое отрицание, т. е. `--source ! 192.168.0.0/24` означает любой адрес кроме адресов 192.168.0.x.
`-d, --dst, --destination`	IP-адрес(а) получателя. Имеет синтаксис схожий с критерием `--source`, за исключением того, что подразумевает адрес места назначения. Точно так же может определять, как единственный IP-адрес, так и диапазон адресов. Символ «!» используется для логической инверсии критерия.
`-i, --in-interface`	Интерфейс, с которого был получен пакет. Использование этого критерия допускается только в цепочках INPUT, FORWARD и PREROUTING, в любых других случаях будет вызывать сообщение об ошибке.
`-o, --out-interface`	Задает имя выходного интерфейса. Этот критерий допускается использовать только в цепочках OUTPUT, FORWARD и POSTROUTING, в противном случае будет генерироваться сообщение об ошибке.
`-f, --fragment`	Правило распространяется на все фрагменты фрагментированного пакета, кроме первого, сделано это потому, что нет возможности определить исходящий/входящий порт для фрагмента пакета, а для ICMP-пакетов определить их тип. С помощью фрагментированных пакетов могут производиться атаки на межсетевой экран, так как фрагменты пакетов могут не отлавливаться другими правилами.

Продолжение таблицы 9

Критерий	Пояснения
`-sport`, `--source-port`	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Соответствие имен сервисов и номеров портов можно найти в файле `/etc/services`. При указании номеров портов правила отрабатывают несколько быстрее.
`--dport`, `--destination-port`	Порт, на который адресован пакет. Аргументы задаются в том же формате, что и для `--source-port`.
`--tcp-flags`	SYN, ACK, FIN SYN определяет маску и флаги tcp-пакета. Пакет считается удовлетворяющим критерию, если из перечисленных флагов в первом списке в единичное состояние установлены флаги из второго списка. В качестве аргументов критерия могут выступать флаги SYN, ACK, FIN, RST, URG, PSH, а также зарезервированные идентификаторы ALL и NONE. ALL означает ВСЕ флаги, а NONE – НИ ОДИН флаг. Так, критерий `--tcp-flags ALL NONE` означает, что все флаги в пакете должны быть сброшены. Символ «!» означает инверсию критерия. Имена флагов в каждом списке должны разделяться запятыми, пробелы служат для разделения списков.
`--icmp-type`	Тип сообщения ICMP определяется номером или именем. Числовые значения определяются в RFC 792. Чтобы получить список имен ICMP значений выполните команду `iptables --protocol icmp --help`. Символ «!» инвертирует критерий, например `--icmp-type ! 8`.
`--state`	Для использования данного критерия в правиле перед `--state` нужно явно указать `-m state`. Проверяется признак состояния соединения. Можно указывать 4 состояния: INVALID, ESTABLISHED, NEW и RELATED. INVALID подразумевает, что пакет связан с неизвестным потоком или соединением и, возможно содержит ошибку в данных или в заголовке. ESTABLISHED указывает на то, что пакет принадлежит уже установленному соединению, через которое пакеты идут в обоих направлениях. NEW подразумевает, что пакет открывает новое соединение или пакет принадлежит однонаправленному потоку. RELATED указывает на то, что пакет принадлежит уже существующему соединению, но при этом он открывает новое соединение. Примером может служить передача данных по FTP, или выдача сообщения ICMP об ошибке, которое связано с существующим TCP или UDP соединением.

Окончание таблицы 9

Критерий	Пояснения
	Признак NEW – это не то же самое, что установленный бит SYN в пакетах TCP, посредством которых открывается новое соединение, и, подобного рода пакеты могут быть потенциально опасны в случае, когда для защиты сети используется один сетевой экран.

7.12.7. Модули iptables

Возможности фильтрации пакетов расширяются через модули. Модули подключаются автоматически при выборе протокола (`-p/--protocol`) или вручную опцией `-m/--match`, после которой следует имя подключаемого фильтра и его опции.

Справку по опциям модуля можно получить с помощью ключа `-h/--help`. Допустимо указание нескольких модулей.

Результаты фильтрации, выдаваемые модулем, можно инвертировать указав ! перед его именем.

В таблице 10 приведены возможные критерии для фильтрации пакетов в фильтре iptables.

Т а б л и ц а 10 – Модули iptables

Модуль	Опции	Пояснение
connlimit	`[!] --connlimit-above n` – пакет подойдет под описание, если количество одновременных подключений на данный момент больше (меньше), чем `n` `--connlimit-mask bits` – позволяет задать маску блока адресов	Позволяет задавать возможное количество одновременных подключений к машине от заданного IP или блока адресов. Пример. Допускать не больше 20 соединений на порт 80 с одного хоста `iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 -j REJECT --reject-with tcp-reset`

Продолжение таблицы 10

Модуль	Опции	Пояснение
icmp	`--icmp-type [!] тип` – тип ICMP в виде числа или имени в соответствии с `iptables -p icmp -h`	Расширение загружается при указании `--protocol icmp`.
iprange	`[!]--src-range ip-ip` – диапазон IP-адресов отправителя `[!]--dst-range ip-ip` – диапазон IP-адресов получателя	Выделяет не один адрес, как `--src`, а все адреса от ip1 до ip2.

ipv4options	`--ssrr` – должен быть установлен флаг strict source routing (маршрутизация указывается источником);
	`--lsrr` – должен присутствовать флаг loose source routing (свободная маршрутизация);
	`--no-srr` – флаг, позволяющий источнику определить режим маршрутизации, должен отсутствовать;
	`[!] --rr` – должен присутствовать флаг RR;
	`[!] --ts` – должен присутствовать флаг TS;
	`[!] --ra` – должен присутствовать флаг оповещения маршрутизатора;
	`[!] --any-opt` – выдавать положительный результат если хотя бы один пункт из указанных выше был выполнен.
		Результат теста зависит от параметров заголовка IPv4, таких как параметры маршрутизации, запись маршрута, запрос времени, оповещение маршрутизатора.
		Примеры. Отбрасывать пакеты с флагом `record-route`:
		`iptables -A input -m ipv4options --rr -j DROP`
		Отбрасывать пакеты с флагом timestamp:
		`iptables -A input -m ipv4options --ts -j DROP`

length	`--length [!] размер[:размер]`	Позволяет проверять размеры пакетов (точно или по диапазону)

Продолжение таблицы 10

Модуль	Опции	Пояснение
limit	`--limit частота` – максимальная средняя частота положительных результатов. После числа можно указывать единицы: `` `/second' ``, `` `/minute' ``, `` `/hour' ``, `` `/day' ``; значение по умолчанию – `3/hour`. `--limit-burst number` – ограничение на исходное число пропускаемых пакетов (по умолчанию – 5).	Выдает положительный результат с фиксированной частотой. Правило использующее этот модуль будет выполняться до момента достижения лимита (и наоборот, если указан «!»). Может использоваться вместе с целью LOG для получения ограниченного протоколирования.
multiport	`[!]--source-ports port1,port2,port3:port4` – исходный порт равен одному из указанных; `[!]--destination-ports port1,port2,port3:port4` – порт назначения равен одному из указанных; `[!]--ports port1,port2,port3:port4` – исходный и порт назначения и равны одному из указанных.	Позволяет указывать в тексте правила несколько (до 15) портов и диапазонов портов (порт:порт). Используется только вместе с `-p tcp` или `-p udp`.
state	`--state состояния` – список фильтруемых состояний через запятую (см. таблицу 9).	Проверяется признак состояния соединения (state).
string	`--algo bm|kmp` – стратегия сравнения/поиска (`bm = Boyer-Moore, kmp = Knuth-Pratt-Morris`); `--from позиция` – позиция в данных с которой следует начинать поиск. Значение по умолчанию – 0;	Позволяет выполнять фильтрацию пакетов, основываясь на анализе содержимого области данных пакета.

Окончание таблицы 10

Модуль	Опции	Пояснение
	`--to позиция` – позиция в данных, при достижении которой следует прекращать поиск. Значение по умолчанию – размер пакета; `--string последовательность` – последовательность символов, которую следует искать в пакете; `--hex-string pattern` – последовательность символов, которую следует искать в пакете (в шестнадцатеричном представлении).	
tcp	см. таблицу 9	Это расширение загружается при указании `--protocol tcp`
u32	`--u32 "Start&Mask=Range"`	Позволяет извлекать из пакета данные размером до 4 байт, применять к ним операции логического И, сдвига, и проверять принадлежность получающихся данных определенным диапазонам. В простейшей форме, u32 вырезает блок из 4 байт начиная со Start, применяет к ним маску Mask и сравнивает результат с Range-m u32
udp	см. таблицу 9	Это расширение загружается при указании `--protocol udp`

Список доступных модулей можно просмотреть, выполнив команду:

```
# ls /lib/modules/$(uname -r)/kernel/net/netfilter/
```

Загруженные модули iptables можно найти в записи файловой системы proc `/proc/net/ip_tables_matches`:

```
# cat /proc/net/ip_tables_matches
```

Загрузка модуля:

```
# modprobe <модуль>
```

Например:

```
# modprobe xt_limit
# modprobe xt_length
# modprobe xt_u32
```

7.12.8. Использование фильтра iptables

ОС Альт 8 СП уже включает в себя предустановленный iptables. Для его настройки рекомендуется использовать возможности системы настройки сети `/etc/net`.

7.12.9. Примеры команд iptables

Список текущих правил:

```
iptables -nvL --line-numbers
```

Очистка всех правил:

```
iptables -F
```

Очистка правил в цепочке:

```
iptables -F INPUT
```

Удаления пятого правила в цепочке INPUT:

```
iptables -D INPUT 5
```

7.12.9.1. Фильтрация по источнику пакета

Для фильтрации по источнику используется опция `-s`.

Например, запретить все входящие пакеты с узла 192.168.1.95:

```
iptables -A INPUT -s 192.168.1.95 -j DROP
```

Можно использовать доменное имя для указания адреса хоста:

```
iptables -A INPUT -s test.host.net -j DROP
```

Также можно указать целую подсеть:

```
iptables -A INPUT -s 192.168.1.0/24 -j DROP
```

Можно использовать отрицание (знак «!»). Например, все пакеты с хостов отличных от 192.168.1.96 будут уничтожаться:

```
iptables -A INPUT ! -s 192.168.1.96 -j DROP
```

Разрешить трафик по localhost:

```
iptables -A INPUT 1 -i lo -j ACCEPT
```

Записывать в журнал попытки спуфинга с префиксом "IP_SPOOF A: " и запретить соединение:

```
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j LOG --log-prefix "IP_SPOOF A: "
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

7.12.9.2. Фильтрация по адресу назначения

Для фильтрации по адресу назначения используется опция `-d`.

Например, запретить все исходящие пакеты на хост 192.168.1.95:

```
iptables -A OUTPUT -d 192.168.156.156 -j DROP
```

Запретить доступ к ресурсу `vk.com`:

```
iptables -A OUTPUT -d vk.com -j REJECT
```

Как и в случае с источником пакета можно использовать адреса под сети и доменные имена. Отрицание также работает.

7.12.9.3. Фильтрация по протоколу

Опция `-p` указывает на протокол. Можно использовать all, icmp, tcp, udp или номер протокола (из `/etc/protocols`).

Разрешить входящие эхо-запросы:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

7.12.9.4. Фильтрация по порту источника

Разрешить все исходящие пакеты с порта 80:

```
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
```

Заблокировать все входящие запросы порта 80:

```
iptables -A INPUT -p tcp --dport 80 -j DROP
```

Для указания порта необходимо указать протокол (tcp или udp). Можно использовать отрицание.

Открыть диапазон портов:

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 7000:7010 -j ACCEPT
```

7.12.9.5. Фильтрация по порту назначения

Разрешить подключения по HTTP:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Разрешить подключения по SSH:

```
iptables -A INPUT -p tcp -i eth0 --dport 22 -j ACCEPT
```

Разрешить получать данные от DHCP-сервера:

```
iptables -A INPUT -p UDP --dport 68 --sport 67 -j ACCEPT
```

Разрешить rsync с определенной сети:

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.1.0/24 --dport 873 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 873 -m state --state ESTABLISHED -j ACCEPT
```

Разрешить IMAP/IMAP2 трафик:

```
iptables -A INPUT -i eth0 -p tcp --dport 143 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 143 -m state --state ESTABLISHED -j ACCEPT
```

Разрешить исходящие HTTP, FTP, DNS, SSH, SMTP:

```
iptables -A OUTPUT -p TCP -o eth0 --dport 443 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth0 --dport 80 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth0 --dport 53 -j ACCEPT
iptables -A OUTPUT -p UDP -o eth0 --dport 53 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth0 --dport 25 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth0 --dport 22 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth0 --dport 21 -j ACCEPT
```

Разрешить mysql для локальных пользователей:

```
iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
```

Разрешить CUPS (сервер печати, порт 631) для пользователей внутри локальной сети:

```
iptables -A INPUT -s 192.168.1.0/24 -p udp -m udp --dport 631 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -p tcp -m tcp --dport 631 -j ACCEPT
```

Разрешить синхронизацию времени NTP для пользователей внутри локальной сети:

```
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 123 -j ACCEPT
```

7.12.9.6. Перенаправление портов

Направим трафик с порта 442 на 22, это значит, что входящие ssh-соединения могут быть принятыми с порта 422 и 22:

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.1.15 --dport 422
-j DNAT --to 192.168.1.15:22
```

Также надо разрешить входящие соединения с порта 422:

```
iptables -A INPUT -i eth0 -p tcp --dport 422 -m state --state
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 422 -m state --state
ESTABLISHED -j ACCEPT
```

Как и в случае с портом источника нужно указать протокол. Можно использовать отрицание.

7.12.9.7. Ограничение по локальным пользователям

Ограничение по локальным пользователям нельзя поручить внешнему межсетевому экрану, так как он не имеет этой информации.

Отбросить все пакеты, исходящие от процессов пользователя с `UID=500`:

```
# iptables -A OUTPUT -m owner --uid-owner 500 -j DROP
```

Попытка соединения с удаленным узлом, пользователя с `UID=500`:

```
# su – test
$ wget ya.ru
--2017-03-07 13:53:14--  http://ya.ru/
Распознается ya.ru (ya.ru)… ошибка: Имя или служба не известны.
wget: не удается разрешить адрес «ya.ru»
```

Попытка соединения с локальным узлом, пользователя с `UID=500`:

```
# su – test
$ wget localhost
--2017-03-07 13:55:20--  http://localhost/
Распознается localhost (localhost)… 127.0.0.1
Подключение к localhost (localhost)|127.0.0.1|:80... ^C
```

7.12.9.8. Фильтрация по содержимому пакета

Отбросить все пакеты, данные в которых содержат подстроку virus:

```
# iptables -A INPUT -m string --algo kmp  --string "secret"  -j
LOG --log-level info --log-prefix "SECRET "
```

Записывать в журнал пакеты со строкой secret внутри:

```
# iptables -I INPUT -j DROP -p tcp -s 0.0.0.0/0 -m string --algo
kmp --string "virus "
```

Просмотр журнала:

```
# journalctl |grep SECRET
апр 03 16:47:18 host-15.localdomain kernel: SECRET IN=enp0s3 OUT=
MAC=08:00:27:d5:f3:78:74:e5:0b:3e:2c:88:08:00    SRC=192.168.3.101
DST=192.168.3.104  LEN=47  TOS=0x00  PREC=0x00  TTL=64  ID=30811  DF
PROTO=TCP SPT=53878 DPT=8080 WINDOW=229 RES=0x00 ACK PSH URGP=0
апр 03 16:58:47 host-15.localdomain kernel: SECRET IN=enp0s3 OUT=
MAC=08:00:27:d5:f3:78:74:e5:0b:3e:2c:88:08:00    SRC=192.168.3.101
DST=192.168.3.104  LEN=47  TOS=0x00  PREC=0x00  TTL=64  ID=38640  DF
PROTO=TCP SPT=54510 DPT=8080 WINDOW=229 RES=0x00 ACK PSH URGP=0
```

Статистика правил iptables и счетчики обработанных пакетов в цепочке INPUT:

```
# iptables -nvL INPUT --line-numbers
Chain INPUT (policy ACCEPT 1711 packets, 1400K bytes)
num    pkts bytes target       prot opt in       out       source
destination
1        47 49550 DROP         tcp  --  *       *         0.0.0.0/0
0.0.0.0/0            STRING match  "virus" ALGO name kmp TO 65535
2         0     0 DROP         tcp  --  *       *         0.0.0.0/0
0.0.0.0/0               STRING match  "virus " ALGO name kmp TO
65535
3        17 66141 LOG          tcp  --  *       *         0.0.0.0/0
0.0.0.0/0               STRING match  "secret" ALGO name kmp TO
65535 LOG flags 0 level 6 prefix "SECRET "
```

7.13. Настройка ПО для связи UNIX-машин с сетями Microsoft и LanManager (Samba)

Samba представляет собой комплект серверного и клиентского программного обеспечения, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS.

Для работы в сетях SMB необходимы:

- клиент;
- сервер;
- средства администрирования.

Для этого должны быть установлены пакеты samba, samba-client, samba-common, samba-winbind, samba-winbind-clients, входящие в состав дистрибутива.

При использовании SMB доступны следующие ресурсы:

- сетевые диски;
- прямые пути к дискам;
- принтеры;
- доменная авторизация и управление.

7.13.1. Основные каталоги и файлы, используемые для работы с Samba

Все файлы конфигурации и авторизации Samba расположены в каталоге `/etc/samba` и его подкаталогах:

1) `/usr/share/samba/codepages` – каталог, содержащий файлы с таблицами перекодировки;
2) `/etc/samba/lmhosts` – каталог предназначен для преобразования IP-адреса в имя NetBIOS;
3) `/var/lib/samba/private/secrets.tdb` – ключевой файл для идентификации машины в домене сети Microsoft;
4) `/etc/samba/smb.conf` – основной конфигурационный файл Samba;
5) `/var/lib/samba/private/passdb.td` – аналог `/etc/passwd` и `/etc/tcb/*/shadow` – файл пользователей сервера Samba с паролями.

Соответствие пользователей Samba и системных пользователей производится на основе общего UID; данный файл используется Samba при отсутствии данных о пользователе на PDC (Primary Domain Controller) или при отсутствии самого PDC;

6) `/etc/samba/smbusers` – файл соответствий имен сетевых и локальных пользователей SMB;
7) `/var/log/samba/*` – файлы журналов серверной части Samba. `log.smbd`, `log.nmbd`, `log.winbind` – журналы соответствующих процессов, а все прочие – журналы взаимодействия сервера с отдельными клиентскими хостами в формате `log.<Client_NetBIOS_NAME>`;
8) `/var/spool/samba` – каталог динамического спулинга печати сервера Samba;
9) `/var/cache/samba/*` – файлы, формируемые в процессе работы различных компонентов Samba;
10) `/var/lib/samba/` – служебные каталоги для администратора.

Список выполняемых файлов Samba можно получить командой:

```
$ rpm -ql `rpm -qa | grep samba` | grep bin/
```

Основными серверными компонентами являются:

1) `/usr/sbin/nmbd` – сервер преобразования имен и адресов;
2) `/usr/sbin/smbd` – файловый сервер;
3) `/usr/sbin/winbindd` – сервер импорта пользователей и групп с PDC;
4) `/etc/init.d/smb` и `/etc/init.d/winbind` – управляющие скрипты инициализации сервисов.

Скрипт `smb` имеет два режима перезапуска:

1) restart – производит полный перезапуск процессов smbd и nmbd со сбросом текущих соединений;
2) reload – принуждает файловый сервер smbd и сервер преобразования имен nmbd перечитывать файлы конфигурации без перезапуска и сброса соединений. При этом старые соединения продолжают существовать по

старым правилам, а ко всем новым соединениям будут применены уже новые правила на основании файлов конфигурации.

Основные клиентские компоненты:

1) `/usr/bin/smbclient` – интерактивное приложение для просмотра сетевых ресурсов;
2) `/sbin/mount.smb`, `/sbin/mount.smbfs`, `/usr/bin/smbumount`, `/usr/sbin/smbmnt`, `/sbin/mount.cifs` – средства монтирования/размонтирования сетевых файловых систем;
3) `/usr/bin/smbpasswd` – утилита управления пользователями и подключением к домену;
4) `/usr/bin/wbinfo` – утилита отображения списка пользователей, импортированных winbindd;
5) `/usr/bin/testparm` – утилита проверки синтаксиса конфигурационных файлов;
6) `/usr/bin/smbstatus` – утилита отображения статуса процессов smbd и nmbd;
7) `/usr/bin/nmblookup` – программа разрешения имен WINS (аналог nslookup для DNS).

7.13.2. Управление учетными записями пользователей

Управление учетными записями выполняется помощью утилиты smbpasswd.

Для управления учетными записями пользователей используются следующие основные команды (<User_name> – имя пользователя):

- создание нового пользователя:

```
# smbpasswd -a <User_name>
```

- смена пароля у существующего пользователя:

```
# smbpasswd <User_name>
```

- удаление существующего пользователя:

```
# smbpasswd -x <User_name>
```

- активация учетной записи:

```
# smbpasswd -e <User_name>
```

- приостановление учетной записи без удаления:

```
# smbpasswd -d <User_name>
```

- подключение данного компьютера к существующему домену:

```
# smbpasswd -j <Domain_name> -U <Administrator_name>
```

7.13.3. Настройка конфигурационного файла `smb.conf`

Для настройки Samba необходимо отредактировать основной конфигурационный файл `smb.conf` (расположен в директории `/etc/samba/`).

Файл конфигурации `smb.conf` включает следующие основные параметры:

- `[global]` – начало секции `[global]`, которая определяет общие настройки серверной части Samba;
- `netbios name` – позволяет указать Netbios имя сервера, по умолчанию используется первая часть доменного имени компьютера;
- `invalid users` – список пользователей, которым запрещен доступ (рекомендуется включить в этот список пользователя root);
- `interfaces` – позволяет указать сетевой интерфейс, используемый Samba (если машина имеет несколько сетевых интерфейсов);
- `security` – выбор режима безопасности, при `security=user` каждый пользователь должен иметь учетную запись (account) на GNU/Linux сервере, для того, чтобы Samba-сервер управлял доступом и пользователями, используйте `security=share`;
- `workgroup` – рабочая группа;
- `server string` – описание компьютера;
- `socket options` – параметры сокета, которые будут использоваться для обслуживания клиентов;
- `encrypt passwords` – включить/выключить шифрование паролей между сервером и клиентом;
- `wins support` –включить/выключить роль WINS-сервера;

- `os level` – приоритет данного сервера среди других компьютеров рабочей группы: определяет, кто именно будет главной машиной, отвечающей за отображение ресурсов сети;
- `domain master` – включить/выключить параметр `domain master`;
- `local master` – включить/выключить параметр `local master`;
- `domain logons` – включить/выключить функцию первичного контролера домена (PDC) для сервера Samba;
- `logon script` – пакетный файл запуска или файл сценария NT;
- `logon path` – каталог, в котором будут храниться пользовательские профили;
- `logon home` – домашний каталог при авторизации клиента;
- `name resolve order` – порядок разрешения имен;
- `dns proxy` – позволяет указать будет ли демон nmbd (например, если WINS не смог разрешить NetBIOS имя) выполнять запрос к DNS;
- `preserve case` – позволяет указать будут ли имена файлов, создаваемых клиентом оставаться такими как они есть или же они будут преобразовываться к значению по умолчанию;
- `short preserve case` – позволяет указать регистр имени файла для сохранения;
- `unix password sync` – позволяет выполнить синхронизацию пароля UNIX с паролем SMB при изменении зашифрованного пароля SMB в файле `smbpasswd`;
- `passwd program` – позволяет указать программу, которая будет использована для смены паролей UNIX;
- `passwd chat` – позволяет указать chat-протокол для смены пароля;
- `max log size` – позволяет указать максимальный размер файла журнала;
- `[NaMe123]` – позволяет указать название новой секции, где `NaMe123` – имя, видимое клиентам;
- `comment` – комментарий, видимый в сети как комментарий к ресурсу;

- `path` – позволяет указать путь к каталогу (общему ресурсу), доступ к которому будет разрешен пользователю;
- `public` – позволяет включить/выключить возможность доступа авторизованных пользователей к общему ресурсу без ввода пароля;
- `writable` – включить/выключить запрет на запись всем пользователям;
- `write list`– разрешение работы на запись пользователям, входящим в группу;
- `browseable` – включить/выключить отображение общего ресурса в списке доступных общих ресурсов в сетевом окружении и в списке просмотра;
- `force user, force group` – привязка к определенному имени пользователя или группе, имена через пробел:

```
force user = user1 user2
force group = group1 group2
```

Параметры конфигурационного файла поддерживают переменные:

- `%U` – имя пользователя сессии;
- `%G` – первичная группа `%U`;
- `%h` – DNS имя;
- `%m` – NETBIOS имя клиента;
- `%L` – NETBIOS имя сервера;
- `%v` – версия Samba;
- `%M` – DNS имя клиента;
- `%a` – архитектура клиента;
- `%I` – IP адрес клиента;
- `%i` – локальный IP адрес, к которому подключен клиент;
- `%T` – текущая дата и время;
- `%D` – имя домена или рабочей группы текущего пользователя;
- `%S` – имя ресурса;
- `%P` – корневая папка ресурса.

Примечание. Комментарии, помогающие при первичной настройке файла `smb.conf`, содержатся в файле `smb.conf.orig`. Кроме того, для ознакомления с полным списком возможностей `smb.conf` можно воспользоваться следующей командой:

```
man smb.conf
```

После сохранения любого вида изменений, внесенных в конфигурационный файл, рекомендуется выполнить его проверку на наличие синтаксических ошибок, для этого необходимо выполнить следующую команду:

```
testparm <полный путь к файлу конфигурации>
```

Например, опция `path = /tmp/%u` может быть интерпретирована как `path = /tmp/John`, если пользователь связан с именем пользователя John.

В случае отсутствия синтаксических ошибок файловый сервер smbd выполнит корректную загрузку конфигурационного файла.

7.13.4. Примеры использования Samba

7.13.4.1. Применение Samba в качестве файлохранилища с локальной авторизацией

Для применения сервера в качестве файлового хранилища с файловой авторизацией необходимо сконфигурировать файл `smb.conf` следующим образом:

```
[global]
netbios name = MYSAMBASERVER
server string = Samba Server Version %v
workgroup = WORKGROUP
passdb backend = tdbsam
security = user
null passwords = true
username map = /etc/samba/smbusers
name resolve order = hosts wins bcast
wins support = no
printing = CUPS
printcap name = CUPS
log file = /var/log/samba/log.%m
syslog = 0
```

```
syslog only = no
[SDA5-HOME]
path = /home
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
```

После сохранения данной конфигурации необходимо выполнить следующие действия:

1) внести пользователя в базу данных SMB и установить пароль для доступа к общим ресурсам с помощью следующей команды (user – имя пользователя):

   ```
   smbpasswd -a user
   ```

2) ввести пароль и нажать клавишу <Enter>, после чего пользователь будет добавлен в базу данных SMB;
3) включить пользователя с помощью следующей команды:

   ```
   smbpasswd -e user
   ```

4) создать псевдоним для имени пользователя user, для этого необходимо отредактировать файл `/etc/samba/smbusers` следующим образом:

   ```
   # Unix_name = SMB_name1 SMB_name2 …
   root = administrator admin
   nobody = guest pcguest smbguest
   user = andrey
   ```

П р и м е ч а н и е . Здесь `andrey` – имя пользователя, для которого создается псевдоним.

Далее необходимо перезапустить сервер Samba с помощью следующих команд:

```
service smb restart
service nmb restart
```

7.13.4.2. Создание общей папки с правами на чтение и запись

Для создания общей папки с правами на чтение и запись для всех пользователей можно воспользоваться таким файлом конфигурации `smb.conf`:

```
[global]
     server string = SambaServer
     log file = /var/log/samba/log.%m
     max log size = 50
[upload]
     path = /home/public
     read only = No
     guest ok = Yes
```

Для того чтобы предоставить возможность записи в общую папку только пользователям определенной группы необходимо добавить в конфигурационный файл следующую строку:

```
write list = @staff
```

При этом права на запись будут предоставлены пользователям, которые входят в группу staff.

Настройка закончена, следует перезапустить Samba, для этого воспользуйтесь

```
service smb restart
service nmb restart
```

7.13.4.3. Подключение Samba к существующему NT домену

Для подключения созданной рабочей станции Samba с именем COMP, администратором которой является пользователь Administrator и PDC этого домена реализован на другом компьютере, к существующему домену DOM необходимо выполнить следующие действия:

1) убедиться в уникальности имени подключаемой рабочей станции в домене, в противном случае необходимо выбрать другое имя или удалить машину из состава домена средствами самого PDC;
2) сконфигурировать файл `smb.conf` следующим образом:

```
[global]
     workgroup = DOM
     netbios name = MYSAMBASERVER
```

```
   server string = Samba Server Version %v
   security = DOMAIN
   password server = *
   log file = /var/log/samba/log.%m
   max log size = 50
   local master = No
   wins server = 192.168.7.19
   idmap config * : backend = tdb
   cups options = raw
[homes]
   comment = Home Directories
   read only = No
   browseable = No
[printers]
   comment = All Printers
   path = /var/spool/samba
   printable = Yes
   print ok = Yes
   browseable = No
[public]
   comment = Public Stuff
   path = /home/samba
   read only = No
   guest ok = Yes
```

Далее необходимо послать запрос на PDC с целью авторизации нового члена домена с помощью следующей команды:

```
$ smbpasswd -j DOM -r DOMPDC -U user
```

Также можно воспользоваться командой:

```
# net join member -U user
```

В ответ на запрос о вводе пароля необходимо ввести пароль пользователя, с которым пользователь зарегистрирован в домене. После успешной регистрации пользователя на экран будет выведено сообщение `Joined domain DOM`, что будет свидетельствовать об успешном добавлении новой станции к домену.

Далее следует перезапустить Samba:

```
service smb restart
service nmb restart
```

7.13.4.4. Настройка функционирования сервера в качестве контроллера домена

Для создания Primary Domain Controller (PDC) необходимо в `smb.conf` внести/изменить следующие записи:

```
[global]
# Имя сервера; если данный параметр не определен,
# то он примет значение, соответствующее имени хоста
netbios name = COOLSERVER
# Имя домена
workgroup = COOLDOMAIN
# Режим работы системы авторизации сервера
security = user
# Разрешение на использование кодированных паролей
encrypt passwords = yes
# Путь к локальному файлу паролей
smb passwd file = /etc/samba/smbpasswd
# Стать мастер-браузером для домена
local master = yes
# Быть PDC
domain master = yes
# Сразу при старте постараться стать мастер-браузером домена
preferred master = yes
# Быть сервером паролей домена
domain logons = yes
# Расположение профайла пользователей домена
logon path = \\%L\Profiles\%U
# Административная группа домена, присутствие в списке
# пользователя “administrator” весьма желательно, без
# этого данный пользователь не получит административных
# прав на клиентских машинах Windows
domain admin group = root @wheel administrator
# WINS-сервер имеет смысл, когда в сети более 10 машин,
# работающих по протоколу SMB. В сложных
```

```
# сетях существенно снижает широковещательный трафик.
wins support = yes
# Порядок разрешения имен NetBIOS
# Значение wins
# имеет смысл только при наличии в сети wins-сервера,
# в противном случае оно замедлит работу.
name resolve order = wins lmhosts bcast
```

Также необходимо создать ресурсы для работы домена.

Ресурс netlogon необходим для работы PDC и домена в целом. Он просто должен существовать.

```
[netlogon]
     comment = Network Logon Service
     path = /var/lib/samba/netlogon
     guest ok = yes
     writable = no
     write list = admin, administrator
```

Данный ресурс необходим для создания и хранения профайлов пользователей домена:

```
[Profiles]
     path = /var/lib/samba/profiles
     browseable = no
     read only = no
     create mask = 0600
     directory mask = 0700
```

При создании пользователя домена в `/var/lib/samba/profiles` автоматически создается каталог с именем, идентичным имени создаваемого пользователя и принадлежащий ему (с правами 0700). В этом каталоге будут храниться личные настройки пользователя.

Для того чтобы включить клиентскую машину в домен, необходимо произвести следующие действия.

Первый метод – вручную.

Прежде всего, необходимо создать локального пользователя системы с именем, соответствующим NetBIOS-name подключаемой к домену машины. К имени на конце добавляется символ «$».

Для добавления машины с именем machine_name необходимо от имени пользователя root выполнить следующие команды:

```
# /usr/sbin/useradd -g machines -d /dev/null -c "machine nickname" -s /bin/false machine_name$
# passwd -l machine_named$
```

Теперь, когда создан пользователь (символ «S» в конце имени означает что это NetBIOS-имя компьютера, а не имя пользователя), можно добавить его в домен, выполнив от имени root команду:

```
# smbpasswd -a -m machine_name
```

Теперь компьютер подключен к домену.

Второй метод – автоматический.

Работу по созданию машинного аккаунта можно переложить на Samba, включив в `smb.conf` следующую запись:

```
[global]
add user script = /usr/sbin/useradd -d /dev/null -g machines -s /bin/false -M %u
```

После выполнения описанной процедуры сервер Samba будет функционировать в роли контроллера домена, будет принимать от клиентских машин запросы на включение в домен и автоматически регистрировать их аналогично NT Server. В свою очередь пользователи смогут входить под своими именами и паролями с любой станции из состава домена с сохранением настроек, а также самостоятельно изменять свои пользовательские пароли без участия администратора.

7.13.4.5. Использование winbind

Сервис winbind представляет собой средство, предназначенное для интеграции Samba в домены Windows. Данный сервис считывает свою конфигурацию из `/etc/samba/smb.conf` и динамически взаимодействует с контроллером домена, автоматически синхронизируя списки пользователей и групп

домена и ПЭВМ Samba. Winbind позволяет автоматического поддерживать актуальность базы пользователей домена на рабочих станциях Samba.

Для корректного функционирования winbind в файле `smb.conf` должны быть объявлены следующие директивы:

```
# Диапазон номеров локальных пользователей, который будет
# использован для динамического создания пользователей домена
winbind uid = 10000-20000
# Диапазон номеров локальных групп пользователей
winbind gid = 10000-20000
# Символ-разделитель, используемый для составления доменных
# имен пользователей и располагающийся между именем домена и
# именем пользователя
winbind separator = +
# Интервал времени (в секундах) между запросами winbind к
# контроллеру домена в целях синхронизации списков
# пользователей и групп
winbind cache time = 10
# Шаблон имени домашних каталогов доменных пользователей,
# автоматически присваиваемых каждому пользователю.
# Сами каталоги, динамически не создаются.
# Вместо переменой %D подставляется имя домена, а
# вместо %U подставляется имя пользователя
template homedir = /home/%D/%U
# Командный интерпретатор, назначаемый по умолчанию для
# пользователей, авторизованных через winbindd
template shell = /bin/bash
```

Далее необходимо внести изменения в файле `/etc/nsswich.conf` в разделы passwd и group, вписав директиву `winbind` – например, таким образом:

```
passwd: files winbind
group: files winbind
```

После выполнения описанных действий можно использовать имена доменных пользователей в `/etc/samba/smb.conf` с целью разграничения доступа в правах на файлы и каталоги для подключения к сетевым ресурсам данного хоста со стороны других хостов.

7.13.4.6. Обычный клиент

В графическом окружении MATE клиентские функции Samba представлены встроенной в файловый менеджер поддержкой просмотра сетевого окружения и доступа к файлам на удаленных Samba-узлах. В сервере клиентские функции Samba предоставлены средствами просмотра сетевого окружения и монтирования файловых систем smbclient и mount.cifs (из пакета cifs-utils) соответственно.

При запуске эти программы считывают текущую конфигурацию из файла `/etc/samba/smb.conf` и используют доменные функции в случае, если машина подключена к домену Windows.

Также файловые системы, возможно, монтировать системной командой mount, указав в качестве типа файловой системы smbfs, и использовать эти записи в `/etc/fstab` для автоматического монтирования при загрузке системы.

Например, для того что бы смонтировать в каталог `/mnt/disk` ресурс public с машины SMALLSERVER под именем cooluser, нужно выполнить команду:

```
# mount -t cifs //smallserver/public /mnt/disk -o sec=ntlm
```

Регистр написания имен компьютеров, ресурсов и пользователей роли не играет. Для того чтобы получить список Samba-ресурсов данной машины и список машин рабочей группы или домена достаточно выполнить команду:

```
# smbclient -L localhost -N
```

Более подробные сведения можно прочесть в man-страницах по smbclient и mount.cifs.

7.13.4.7. Клиент в составе существующего домена NT

Подключение происходит аналогично рассмотренному подключению сервера в составе существующего домена NT.

7.13.5. Принт-сервер на CUPS

По умолчанию Samba сконфигурирована на использование CUPS (сервер печати для UNIX-подобных ОС) в качестве спулера печати. Подразумевается, что CUPS уже настроен и запущен. В `/etc/samba/smb.conf` присутствуют следующие директивы:

```
[global]
     printcap name = lpstat
     load printers = yes
     printing = cups
```

Также необходимо создать ресурс `[printers]` – его создание и назначение директив подробно описано в подпункте Обычный сервер в части «Особые ресурсы».

7.13.6. Особенности локализации клиента и сервера

Для того чтобы все компоненты Samba правильно работали с русскими именами файловых объектов и ресурсов, в `/etc/samba/smb.conf` необходимо добавить следующие директивы:

```
[global]
     client code page =
     character set =
```

Далее приводятся наборы значений этих директив и системных кодировок, наиболее часто используемых в России, Белоруссии и на Украине:

```
$LANG = ru_RU.KOI8-R
client code page = 866
characte set = koi8-r

$LANG = ru_RU.CP2151
client code page = 866
characte set = 1251

$LANG = be_BY.CP1251
client code page = 866
```

```
character set = 1251

$LANG = uk_UA.KOI8-U
client code page = 1125
character set = koi8-u


$LANG = uk_UA.CP1251
client code page = 1125
character set = 1251U

$LANG = ru_UA.CP1251
client code page = 1125
character set = 1251U
```

Также необходимо проследить, чтобы на тех компьютерах (с установленной ОС Windows), с которыми предполагается взаимодействие через Samba, были установлены соответствующие системные настройки локализации. В противном случае велика вероятность, что вместо кириллических символов будут отображены знаки «?» либо другие непрошенные символы.

Указанные директивы `/etc/samba/smb.conf` воздействуют на работу всех компонентов Samba – и серверных, и клиентских. На данный момент поддерживаются кириллические написания имен – файлов, каталогов и ресурсов.

7.13.7. Некоторые вопросы безопасности

Данный раздел относится в основном к серверной части Samba.

Прежде всего, необходимо определить, какие интерфейсы должны прослушиваться Samba в ожидании запроса на соединение (по умолчанию прослушиваются все имеющиеся в системе).

Например, для того, чтобы ограничить прослушивание локальным хостом и первой сетевой картой, необходимо написать в `/etc/samba/smb.conf`:

```
[global]
     interfaces = 127.0.0.1 enp0s3
     bind interfaces only = Yes
```

Далее можно ограничить диапазоны адресов, с которых позволительно обращаться к данному серверу. Действие данных директив аналогично воздействию `/etc/hosts.allow` и `/etc/hosts.deny` на xinetd и ssh: если IP-адрес хоста не подпадает под разрешающее правило, то соединение не будет установлено вовсе. Для того чтобы ограничить доступ двумя подсетями и локальной системой, дополнительно исключив при этом один хост, можно написать:

```
[global]
    hosts allow = 192.168.1. 192.168.2. 127.
    hostsdeny = 192.168.1.12
```

Все вышеперечисленные директивы ограничивают соединения на уровне интерфейсов и IP-адресов до какой-либо авторизации. Следующие директивы управляют режимом авторизации пользователей.

Во избежание перехвата чувствительных данных при передаче их по сети открытым текстом принято шифровать пароли. Данная директива включает шифрование паролей в Samba:

```
[global]
    encrypt passwords = yes
```

Файл переопределений имен пользователей является весьма мощным средством управления пользовательскими акаунтами, однако при неразумном использовании это средство опасно и поэтому по умолчанию отключено. Внимательно ознакомьтесь с содержимым файла `/etc/samba/smbusers`, прежде чем использовать его.

```
[global]
; username map = /etc/samba/smbusers
```

7.13.8. Samba 4 в роли контроллера домена Active Directory

Использование Samba 4 в роли контроллера домена Active Directory позволяет вводить Windows 7/8 в домен без манипуляций с реестром.

Поддерживаются следующие базовые возможности Active Directory:

- аутентификация рабочих станций Windows и Linux и служб;

- авторизация и предоставление ресурсов;

- групповые политики (GPO);

- перемещаемые профили (Roaming Profiles);
- поддержка инструментов Microsoft для управления серверами (Remote Server Administration Tools) с компьютеров под управлением Windows;
- поддержка протоколов SMB2 и SMB3 (в том числе с поддержкой шифрования);
- репликация с другими серверами (в том числе с Windows 2012).

Samba AD DC несовместима с OpenLDAP и MIT Kerberos, поэтому службы, использующие MIT Kerberos, несовместимы с ним.

Samba AD DC функционирует на уровне контроллера доменов Windows 2008 R2. Можно ввести его в домен Windows 2012 как клиента, но не как контроллер домена.

7.13.8.1. Установка

Для установки Samba AD DC выполняются следующие шаги:

установить пакет task-samba-dc, который установит все необходимое:

```
# apt-get install task-samba-dc
```

так как Samba в режиме контроллера домена (Doman Controller, DC) использует как свой LDAP, так и свой сервер Kerberos, несовместимый с MIT Kerberos, перед установкой необходимо остановить конфликтующие службы krb5kdc и slapd, а также bind:

```
# for service in smb nmb krb5kdc slapd bind; do chkconfig $service off; service $service stop; done
```

7.13.8.2. Миграция существующего сервера

Для миграции существующего сервера необходимо:

- скопировать данные для миграции в один каталог:

```
mkdir /var/lib/samba/dbdir
cp -pv /var/lib/samba/private/* /var/lib/samba/dbdir
cp -pv
/var/lib/samba/{account_policy,gencache_notrans,group_mapping}.tdb /var/lib/samba/dbdir
```

При этом должно скопироваться пять файлов `.tdb`;

- запустить «classicupgrade» (с правами администратора):

```
# samba-tool domain classicupgrade --dbdir=/var/lib/samba/dbdir
--use-xattrs=yes --realm=test.alt /etc/samba/smb.conf
```

7.13.8.3. Создание нового домена

7.13.8.3.1. Восстановление к начальному состоянию samba

Если домен уже создавался, необходимо очистить базу и конфигурацию Samba:

```
rm -f /etc/samba/smb.conf
rm - rf /var/lib/samba
rm -rf /var/cache/samba
mkdir -p /var/lib/samba/sysvol
```

Перед созданием домена нужно обязательно удалить `/etc/samba/smb.conf`:

```
rm -f /etc/samba/smb.conf
```

7.13.8.3.2. Выбор имени домена

Имя домена для разворачиваемого DC должно состоять минимум из двух компонентов, разделенных точкой.

При этом должно быть установлено правильное имя узла и домена для сервера:

- `HOSTNAME=dc.test.alt` в `/etc/sysconfig/network`;
- `hostname dc.test.alt`;
- `domainname test.alt`.

При указании домена, имеющего суффикс `.local`, на сервере и подключаемых компьютерах под управлением Linux потребуется отключить службу avahi-daemon.

7.13.8.3.3. Создание домена

Создание контроллера домена `test.alt`, выполняется командой:

```
samba-tool domain provision --realm=test.alt --domain test --
adminpass='Pa$$word' --dns-backend=SAMBA_INTERNAL --server-
role=dc --use-rfc2307 --use-xattrs=yes
```

где:

1) `--realm` – задает область Kerberos (LDAP), и DNS имя домена;

2) `--domain` – задает имя домена (имя рабочей группы);

3) `--adminpass` – пароль основного администратора домена;

4) `--server-role` – тип серверной роли.

7.13.8.3.4. Интерактивное создание домена

Для интерактивного развертывания нужно выполнить команду `samba-tool domain provision`, это запустит утилиту развертывания, которая будет задавать различные вопросы о требованиях к установке. В примере показано создание домена `test.alt`:

```
# samba-tool domain provision
Realm [TEST.ALT]:
Domain [TEST]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE)
[SAMBA_INTERNAL]:
DNS forwarder IP address (write 'none' to disable forwarding)
[127.0.0.1]:
Administrator password:
Retype password:
Looking up IPv4 addresses
More than one IPv4 address found. Using 192.168.1.1
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=test,DC=alt
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
```

```
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=test,DC=alt
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba 4 has been generated
at /var/lib/samba/private/krb5.conf
Once the above files are installed, your Samba4 server will be
ready to use
Server Role:           active directory domain controller
Hostname:              c228
NetBIOS Domain:        TEST
DNS Domain:            test.alt
DOMAIN SID:            S-1-5-21-80639820-2350372464-3293631772
```

При запросе ввода следует нажимать клавишу <Enter>, за исключением запроса пароля администратора (`Administrator password:` и `Retype password:`).

Пароль администратора должен быть не менее 7 символов и содержать символы как минимум трех групп из четырех возможных: латинских букв в верхнем и нижнем регистрах, чисел и других небуквенно-цифровых символов. Пароль не полностью соответствующий требованиям это одна из причин завершения развертывания домена ошибкой.

Параметры `--use-rfc2307 --use-xattrs=yes` позволяют поддерживать расширенные атрибуты типа UID и GID в схеме LDAP и ACL на файловой системе Linux.

7.13.8.4. Запуск службы

Службу samba необходимо установить по умолчанию и запустить ее:

```
# chkconfig samba on
# service samba start
```

7.13.8.5. Проверка работоспособности

Просмотр общей информации о домене:

```
# samba-tool domain info 127.0.0.1
  Forest                :       test.alt
  Domain                :       test.alt
  Netbios domain        :       TEST
  DC name               :       c228.test.alt
  DC netbios name       :       C228
  Server site           :       Default-First-Site-Name
  Client site           :       Default-First-Site-Name
```

Просмотр предоставляемых служб:

```
# smbclient -L localhost -Uadministrator
Enter administrator's password:
Domain=[TEST] OS=[Unix] Server=[Samba 4.0.21]
     Sharename   Type      Comment
     ---------   ----      -------
     netlogon    Disk
     sysvol      Disk
     IPC$        IPC       IPC Service (Samba
                           4.0.21)
Domain=[TEST] OS=[Unix] Server=[Samba 4.0.21]
     Sharename         Comment
     ---------         -------
     Workgroup         Master
     ---------         -------
     TEST.ALT          C228
     WORKGROUP         HOST-15
```

Общие ресурсы netlogon и sysvol создаваемые по умолчанию нужны для функционирования сервера AD и создаются в `smb.conf` в процессе развертывания/модернизации.

Для проверки конфигурации DNS, необходимо выполнить шаги:

- убедиться в наличии nameserver 127.0.0.1 в `/etc/resolv.conf`:

```
host test.alt
```

- проверить имена хостов:

```
# host -t SRV _kerberos._udp.test.alt.
_kerberos._udp.test.alt has SRV record 0 100 88 c228.test.alt.
# host -t SRV _ldap._tcp.test.alt.
_ldap._tcp.test.alt has SRV record 0 100 389 c228.test.alt.
# host -t A c228.test.alt.
c228.test.alt has address 192.168.1.1
```

Если имена не находятся, необходимо проверить выключение службы named.

Для проверки настройки необходимо запросить билет Kerberos для администратора домена (имя домена должно быть указано в верхнем регистре):

```
# kinit administrator@TEST.ALT
Password for administrator@TEST.ALT:
Warning: Your password will expire in 41 days on Вт 14 фев 2017
08:58:30
```

Просмотр полученного билета:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@TEST.ALT
    Valid               Expires           Service principal
    starting
    31.01.2017          31.01.2017        krbtgt/TEST.ALT@TEST.ALT
    10:23:54            20:23:54
    renew until 01.02.2017 10:23:45
```

7.13.8.6. Управление пользователями

Создать пользователя с паролем:

```
samba-tool user create <имя пользователя>
```

```
samba-tool user setexpiry <имя пользователя>
```

Например:

```
samba-tool user create ivanov --given-name='Иван Иванов' --mail-address='ivanov@test.alt'
```

Просмотреть список доступных пользователей:

```
samba-tool user list
```

Разблокировать пользователя:

```
samba-tool user setexpiry <имя пользователя> --noexpiry
```

Не допускайте одинаковые имена для пользователей и компьютера, так как это может привести к коллизиям (например, такого пользователя нельзя добавить в группу).

Если компьютер с таким именем заведен, то удалить его можно командой:

```
pdbedit -x -m <имя>
```

7.13.8.7. Заведение вторичного DC

Присоединение дополнительного Samba DC к существующему AD отличается от инициализации первого DC в лесу AD. Необходимо выполнить следующие действия (в примере используется узел: `dc2.test.alt` с ip-адресом 192.168.1.106):

1) на Primary Domain Controller (PDC) необходимо выключить службу bind и, если она была включена, перезапустить службу samba:

```
# service bind stop
# service samba restart
```

2) завести адрес IP для dc2 (указание аутентифицирующей информации (имени пользователя и пароля) обязательно!):

```
# samba-tool dns add 192.168.1.1 test.alt DC2 A 192.168.1.106 -Uadministrator
```

3) установить следующие параметры в файле конфигурации клиента Kerberos (на `dc2.test.alt` файл `/etc/krb5.conf`):

```
[libdefaults]
default_realm = TEST.ALT
dns_lookup_realm = true
dns_lookup_kdc = true
```

В `resolvconf` обязательно должен быть добавлен PDC как nameserver.

4) для проверки настройки необходимо запросить билет Kerberos для администратора домена (имя домена должно быть указано в верхнем регистре):

```
# kinit administrator@TEST.ALT
Password for administrator@TEST.ALT:
Warning: Your password will expire in 37 days on Пт 17 фев 2017 14:31:40
```

5) убедиться, что билет получен:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@TEST.ALT
Valid              Expires            Service principal
 starting
07.01.2017         08.01.2017         krbtgt/TEST.ALT@TEST.ALT
 18:51:01           04:51:01
renew until 08.01.2017 18:50:51
```

6) ввести в домен test.alt в качестве контроллера домена (DC):

```
# samba-tool domain join test.alt DC -Uadministrator --realm=test.alt
```

В результате будет выведена информация о присоединении к домену:

```
Joined domain TEST (SID S-1-5-21-80639820-2350372464-3293631772) as a DC
```

Для получения дополнительной информации можно воспользоваться командой:

```
samba-tool domain join --help
```

7) поставить службу samba в автозагрузку:

```
# chkconfig samba on
```

Если подключение к DC выполнялось под управлением Windows, необходимо запустить службу samba:

```
# service samba start
```

7.13.8.8. Репликация

Без успешной двунаправленной репликации в течение 14 дней DC исключается из Active Directory. Указание аутентифицирующей информации (имени пользователя и пароля) обязательно.

Для выполнения двунаправленной репликации необходимо выполнить следующие действия:

- реплицировать на вторичном DC (с первичного):

```
# samba-tool drs replicate dc2.test.alt c228.test.alt dc=test,dc=test -Uadministrator
```

Сначала указывается приемник, затем источник, после этого реплицируемая ветка в LDAP;

- реплицировать на вторичном DC (на первичный):

```
# samba-tool drs replicate c228.test.alt dc2.test.alt dc=test,dc=alt -Uadministrator
```

Сначала указывается приемник, затем источник, после этого реплицируемая ветка в LDAP.

Имя домена в именах серверов можно опустить (если они одинаковые);

- для просмотра статуса репликации на PDC, необходимо запустить на Samba DC команду:

```
# samba-tool drs showrepl
```

Если репликация на Windows не работает, нужно добавить в Active Directory Sites and Services новое соединение Active Directory. После этого реплицировать на DC, подождать минут 5 и пробовать реплицировать с Samba на Windows.

7.14. Система мониторинга Zabbix

Zabbix – система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

Для управления системой мониторинга и чтения данных используется веб-интерфейс.

Должен быть установлен и запущен сервер PostgreSQL, с созданным пользователем zabbix и созданной базой zabbix.

Веб-интерфейс zabbix доступен по адресу:

```
http://ip-сервера/zabbix
```

При первом заходе на страницу запустится мастер, который шаг за шагом проверит возможности веб-сервера, интерпретатора PHP и сконфигурирует подключение к базе данных (параметры подключения нужно указывать такие же, как у сервера zabbix). На последней странице мастера будет доступен для скачивания конфигурационный файл `zabbix.conf.php`, который необходимо сохранить в `/var/www/webapps/zabbix/frontends/php/conf`.

После этого появится экран входа в интерфейс управления системой мониторинга. Пользователь для входа по умолчанию Admin, пароль zabbix. Войдя в систему, нужно сменить ему пароль, завести других пользователей и можно начать настраивать zabbix.

7.15. Настройка экспорта аудита на удаленный узел

Для настройки экспорта аудита на удаленный узел необходимо настроить openvpn соединение между принимающей и передающей стороной, настроить межсетевой экран и внести изменения в конфигурационные файлы аудита.

На принимающей стороне – сервер:

1) скопировать файл `/usr/share/doc/openvpn-*/server.conf` (* – версия openvpn) в директорию `/etc/openvpn/` для его редактирования и последующего запуска сервера VPN;
2) в скопированном на предыдущем этапе файле `server.conf`, проверьте имена и пути файлов сертификата сервера (`.crt`), его ключа (`.key`), а также сертификата CA (`.crt`) и DH (`dh*.pem`), а также закомментировать параметр `proto udp` и раскомментировать `proto tcp`;
3) установить утилиту easy-rsa:

   ```
   # apt-get install easy-rsa
   ```

4) сгенерировать все необходимые ключи и сертификаты. Ввести для них пароли:

   ```
   # easyrsa init-pki
   # easyrsa build-ca
   ```

```
# easyrsa build-server-full server
# easyrsa build-client-full client1
# easyrsa gen-dh
```

5) перенести полученные ключи и сертификаты в каталог `/etc/openvpn/keys/`.

Настройка openVPN клиента на передающей стороне:

1) скопировать из `/usr/share/doc/openvpn-*/client.conf` (* – версия openvpn) в директорию `/etc/openvpn/` для его редактирования и последующего запуска клиента VPN;
2) скопировать раннее сгенерированные ключи и сертификаты в директорию `/etc/openvpn/keys/` и указать их в `client.conf`;
3) открыть `client.conf` найти строку `remote` и изменить ее на:

   `remote 10.10.3.87 1194`

   где `10.10.3.87` – это IP-адрес сервера на внешнем интерфейсе принимающей стороны.

   Также, закомментировать параметр `proto udp` и раскомментировать `proto tcp`.

Отредактировать конфигурационные файлы аудита:

- на принимающей стороне в файле `/etc/audit/auditd.conf` исправить параметр `tcp_listen_port=1060`;
- на передающей стороне в файле `/etc/audisp/audisp-remote.conf` исправить параметры:

  ```
  remote-server 10.8.0.1
  port 1060
  #queue_error_action
  ```

  где `10.8.0.1` – IP-адрес сервера vpn на созданном интерфейсе-туннеле принимающей стороны;
- на передающей стороне изменить параметр: `active = yes` в файле `/etc/audisp/plugins.d/au-remote.conf`;
- перезапустить систему на принимающей и передающей сторонах.

Запустить сервер на принимающей стороне:

```
# openvpn /etc/openvpn/server.conf
```

Запустить клиент OpenVPN на передающей стороне:

```
# openvpn /etc/openvpn/client.conf
```

Команды установки правила пропуска tcp пакетов с портом назначения 1060 только через устройство vpn (например, tun0) на принимающей стороне:

```
# iptables –A INPUT –p tcp --dport 1060 –i tun0 –j ACCEPT
# iptables –A INPUT –p tcp --dport 1060 –j DROP
```

7.16. Настройка системы сигнализации на основе nagios

Главной задачей системы мониторинга является оповещение администратора, о том, что поведение наблюдаемых объектов изменилось. Также оповещения должны отсылаться, когда состояние объекта возвращается в норму. Nagios позволяет использовать в качестве инструмента оповещения программы, разработанные пользователями.

Система сигнализации состоит из сервера мониторинга (управляющей машины) и удаленных узлов с датчиками мониторинга (управляемые машины).

На управляющей машине должны работать:

- nagios – осуществляет мониторинг сервисов на удаленных компьютерах, их доступности и статуса;
- apache2 – служит интерфейсом nagios;
- nagstamon – забирает данные из apache2, позволяет администратору блокировать удаленные станции через ssh и vlock.

На управляемых машинах должны работать:

- nagwad – собирает события из journal для доступа из nagios;
- nagios-nrpe – позволяет nagios забирать данные о различных сервисах с машины, в том числе о событиях безопасности, которые собираются nagwad при помощи audit и journald.

7.16.1. Настройка сервера мониторинга

7.16.1.1. Установка пакета nagios

Установить пакеты nagios-full, nagios-www-apache2, nagios-addons-nrpe, nagwad-server, apache2-mod_ssl (если они еще не установлены):

```
# apt-get install nagios-full
# apt-get install nagios-www-apache2
# apt-get install apache2-mod_ssl
# apt-get install nagwad-server
# apt-get install nagios-addons-nrpe
```

Примечание. Пакеты будут установлены по умолчанию, если на этапе установки была выбрана группа пакетов «Рабочее место контролера событий безопасности».

7.16.1.2. Конфигурация nagios

Выполнить настройку apache2:

```
# a2enmod ssl
# a2enmod alias
# a2enextra httpd-addon.d
# a2enmod authn_core
```

Добавить службы в автозагрузку:

```
# systemctl enable nagios
# systemctl restart nagios
# systemctl restart httpd2
```

7.16.2. Настройка удаленных хостов

7.16.2.1. Установка пакетов

Расширение NRPE предназначено для выполнения плагинов Nagios на удаленных машинах. Основная задача – позволить Nagios контролировать «локальные» ресурсы (например, загрузку процессора, использование памяти) на удаленных машинах. Поскольку эти ресурсы обычно не подвергаются воздействию внешних машин, то на удаленных машинах должен быть установлен агент, такой как NRPE (рис. 45).

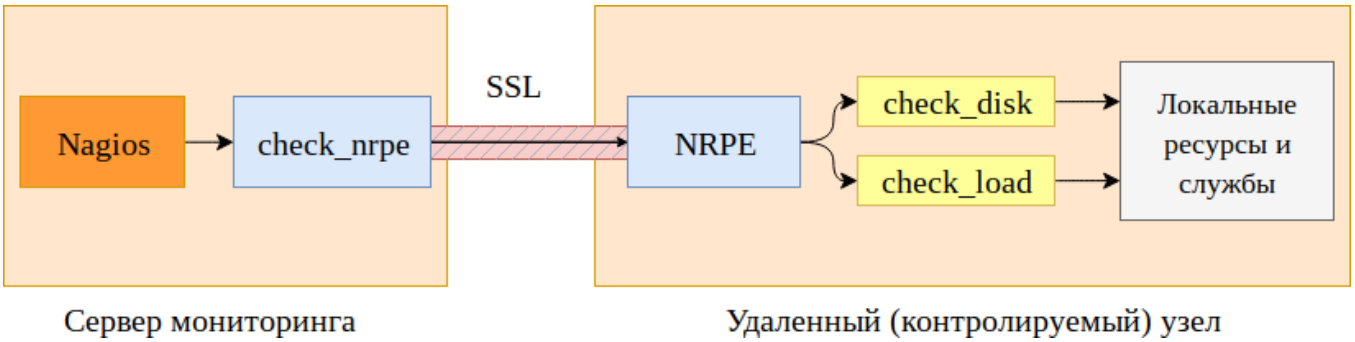

Рис. 45 – Взаимодействие сервера мониторинга с удаленным узлом

На удаленном хосте, за которым необходимо наблюдать установить пакеты nagwad и nagios-nrpe, и добавить их в автозагрузку:

```
# apt-get install nagwad
# systemctl enable nagwad
# apt-get install nagios-nrpe
# systemctl enable nrpe
```

Примечание. Пакеты будут установлены по умолчанию, если на этапе установки была выбрана группа пакетов «Датчики системы сигнализации».

7.16.2.2. Конфигурирование nrpe

Скопировать конфигурацию из `/usr/share/doc/examples/nrpe/nrpe.cfg` в `/etc/nagios`:

```
cp /usr/share/doc/examples/nrpe/nrpe.cfg /etc/nagios
```

IP-адрес сервера мониторинга Nagios необходимо добавить к `nrpe.cfg`. Для этого в файле `/etc/nagios/nrpe.cfg` измените следующую строку:

```
allowed_hosts=192.168.7.100 #сервер с nagios
```

Файл конфигурации NRPE, который был установлен, содержит несколько определений команд, которые можно использовать для мониторинга этой машины. Можно редактировать определения команд, добавлять новые команды и т. д. редактируя конфигурационный файл NRPE с помощью текстового редактора.

Для сигнализации администратору безопасности о нарушении целостности КСЗ и/или объектов контроля целостности необходимо, предварительно настроив подсистему контроля целостности osec, добавить в файл `/etc/nagios/nrpe.cfg` строку объявления команды:

```
command[check_osec]=/usr/lib/nagios/plugins/check_osec
```

Для сигнализации о попытках несанкционированного изменения полномочий пользователей в ОС, а также изменения, добавления и удаления учетных данных пользователей необходимо добавить строку объявления команды:

```
command[check_authdata]=/usr/lib/nagios/plugins/check_authdata
```

Для сигнализации администратору безопасности о попытках подключения к СВТ незарегистрированных устройств ввода-вывода информации или о попытках ввода/вывода информации с/на неучтенные устройства ввода-вывода, в том числе съемные носители информации необходимо добавить строку объявления команды:

```
command[check_devices]=/usr/lib/nagios/plugins/check_devices
```

Для сигнализации администратору безопасности о попытках НСД к защищаемой в ОС информации о попытках несанкционированного запуска программ пользователями ОС необходимо, предварительно настроив аудит, добавить строку объявления команды:

```
command[check_authdata]=/usr/lib/nagios/plugins/check_audit
```

Для настройки сигнализации администратору безопасности о событиях превышения пользователями заданного порогового значения предъявления незарегистрированного идентификатора или ввода неверной аутентифицирующей информации при входе в ОС необходимо, предварительно настроив ограничения неуспешных попыток входа, выполнить следующие действия.

На удаленном хосте, за которым необходимо наблюдать, создать каталоги:

```
# mkdir /var/lib/nagwad/logindata
# mkdir /var/lib/nagwad/logindata_archived
# mkdir /usr/lib/nagwad/logindata
```

Создать файл `/usr/lib/nagwad/logindata/logindata.regexp`:

```
# vim /usr/lib/nagwad/logindata/logindata.regexp
pam_tally2(.*:auth): user\|too many bad attempts
```

Задать права:

```
# chmod +x /usr/lib/nagwad/logindata/logindata.regexp
```

В файл `/usr/lib/nagwad/nagwad.sh` дописать следущие строки:

```
(/bin/journalctl -f | grep --line-buffered -f
/usr/lib/nagwad/logindata/logindata.regexp |
while read -r i
do
  grep -xs "$i" /var/lib/nagwad/logindata_archived
  # do not repeat archived events
  if [ $? -ne 0 ];
  then

    NAME=`mktemp -p /var/lib/nagwad/logindata/ logindataXXX`
    chown nagios $NAME
    echo $i > $NAME
  fi
done
) &
```

Создать файл `/usr/lib/nagios/plugins/check_logindata`:

```
# vim /usr/lib/nagios/plugins/check_logindata
#!/bin/bash
NAME=`ls -b /var/lib/nagwad/logindata | head -1 `
if [ "$NAME" == "" ]
 then
    echo "OK"
    exit 0
 else
    echo "Too many bad attempts"
    exit 2
 fi
```

Задать права:

```
# chmod +x /usr/lib/nagios/plugins/check_logindata
```

Добавить проверку события превышения пользователями заданного порогового значения ввода неверной аутентифицирующей информации при входе в ОС, добавив строку объявления команд в файл `/etc/nagios/nrpe.cfg`:

```
command[check_logindata]=/usr/lib/nagios/plugins/check_logindata
```

Для настройки сигнализации о попытках несанкционированного получения твердых копий защищаемой информации (вывода и информации на печать) необходимо выполнить следующие действия.

На удаленном хосте, за которым необходимо наблюдать, создать каталоги:

```
# mkdir /var/lib/nagwad/printer
# mkdir /var/lib/nagwad/printer_archived
# mkdir /usr/lib/nagwad/printer
```

Создать файл `/usr/lib/nagwad/printer/printer.regexp`:

```
# vim /usr/lib/nagwad/printer/printer.regexp
client_error_not_authorized
```

Задать права:

```
# chmod +x /usr/lib/nagwad/printer/printer.regexp
```

В файл `/usr/lib/nagwad/nagwad.sh` дописать следущие строки:

```
(/bin/tail -f /var/log/cups/access_log | grep --line-buffered -f
/usr/lib/nagwad/printer/printer.regexp |
while read -r i
do
  grep -xs "$i" /var/lib/nagwad/printer_archived
  # do not repeat archived events
  if [ $? -ne 0 ];
  then

    NAME=`mktemp -p /var/lib/nagwad/printer/ printerXXX`
    chown nagios $NAME
    echo $i > $NAME
  fi
done
) &
```

Создать файл /usr/lib/nagios/plugins/check_printer:

```
# vim /usr/lib/nagios/plugins/check_printer
#!/bin/bash
NAME=`ls -b /var/lib/nagwad/printer | head -1 `
if [ "$NAME" == "" ]
 then
    echo "OK"
    exit 0
 else
    echo "Client not authorized"
    exit 2
 fi
```

Задать права:

```
# chmod +x /usr/lib/nagios/plugins/check_printer
```

Добавить проверку попыток несанкционированного получения твердых копий защищаемой информации (вывода и информации на печать), добавив строку объявления команд в файл /etc/nagios/nrpe.cfg:

```
command[check_printer]=/usr/lib/nagios/plugins/check_printer
```

Чтобы повторно инициализировать компонент NRPE, после добавления IP-адреса, необходимо перезагрузить систему.

7.16.3. Добавление удаленных узлов для мониторинга

Для добавления удаленных узлов, на сервере мониторинга (машине, на которой работает nagios) необходимо:

- создать определения узла и служб nagios для мониторинга удаленного хоста;

- создать определение nagios для использования плагина check_nrpe.

Прежде чем контролировать службу, сначала нужно определить хост, который связан с этой услугой. Можно поместить определения хостов в любом конфигурационном файле объекта, указанном в директиве cfg_file или помещенном в каталог, указанный в директиве cfg_dir. Лучше создать новый шаблон для каждого типа узла, который планируется контролировать.

Для задания определения рабочих станций в `/etc/nagios/objects` можно скопировать файл `/usr/share/doc/examples/nagios/server/objects/p8.cfg` в `/etc/nagios/objects/p8.cfg`:

```
cp /usr/share/doc/examples/nagios/server/objects/p8.cfg
/etc/nagios/objects/
```

В файле `/etc/nagios/objects/p8.cfg` необходимо указать ip-адрес, удаленного узла и его имя (узел p8 с адресом 192.168.7.101):

```
define host{
     ; Name of host template to use
     use       linux-server
     host_name p8
     alias          p8
     address        192.168.7.101 }
```

После того, как определение было добавлено для узла, который будет контролироваться, нужно определить службы, которые должны контролироваться, на этом узле. Как и определения хостов, определения служб могут быть помещены в любой конфигурационный файл объекта.

В файле `/etc/nagios/objects/p8.cfg` определим службы для мониторинга на удаленном узле. Эти службы будут использовать команды, которые были определены в файле `nrpe.cfg` на удаленном узле:

```
define service {
     use  generic-service
     host_name      p8
     service_description  devices_not_allowed
     check_command  check_nrpe!check_devices
}

define service {
     use  generic-service
     host_name      p8
     service_description  changes_in_system
     check_command  check_nrpe!check_osec
}
```

```
define service {
     use  generic-service
     host_name      p8
     service_description  changes_in_authdata
     check_command  check_nrpe!check_authdata
}

define service {
     use  generic-service
     host_name      p8
service_description  audit_avc_event
check_command  check_nrpe!check_audit
}
define service {
        use     generic-service
        host_name       p8
        service_description  auth
        check_command   check_nrpe!check_logindata
}

define service {
        use     generic-service
        host_name       p8
        service_description  printer
        check_command   check_nrpe!check_printer
}
```

Далее необходимо заменить содержимое `00-templates.cfg` по образцу:

```
cp /usr/share/doc/examples/nagios/server/templates/00-
templates.cfg /etc/nagios/templates/
```

7.16.4. Тестирование системы мониторинга

Необходимо убедиться, что плагин check_nrpe может обмениваться данными с демоном NRPE на удаленном узле:

```
/usr/lib/nagios/plugins/check_nrpe -H 192.168.7.101
```

где 192.168.7.101 IP-адрес удаленного хоста, на котором установлен NRPE.

Если плагин возвращает ошибку, необходимо проверить следующее:

- между удаленным узлом и сервером мониторинга нет межсетевого экрана, который блокирует связь;
- демон NRPE правильно установлен и запущен на удаленном узле;
- на удаленном узле нет правил локального брандмауэра, которые не позволяют подключаться серверу мониторинга.

Проверить состояние сигнализатора на управляемом узле, можно выполнив на нем через ssh команду:

```
# systemctl status nagwad
```

В строке Tasks должно находиться, как минимум 16 заданий.

Проверить конфигурационные файлы Nagios можно командой:

```
# /usr/sbin/nagios -v /etc/nagios/nagios.cfg
```

В случае наличия ошибок, их нужно исправить, если все в порядке, нужно перезапустить Nagios:

```
# systemctl restart nagios
```

В течение нескольких минут Nagios должен получить текущую информацию о состоянии удаленной машины.

После запуска служб можно проверить работу Nagios Core веб-сервером. Для этого в адресной строке веб-браузера, необходимо ввести адрес:

```
localhost/nagios
```

Если все настроено верно, после ввода аутентификационных данных, (по умолчанию nagios/nagios), будет загружена начальная страница Nagios (рис. 46). На странице Host Detail будут показаны узлы, за которыми ведется наблюдение и их состояние (рис. 47).

Рис. 46 – Работа с Nagios в веб-браузере

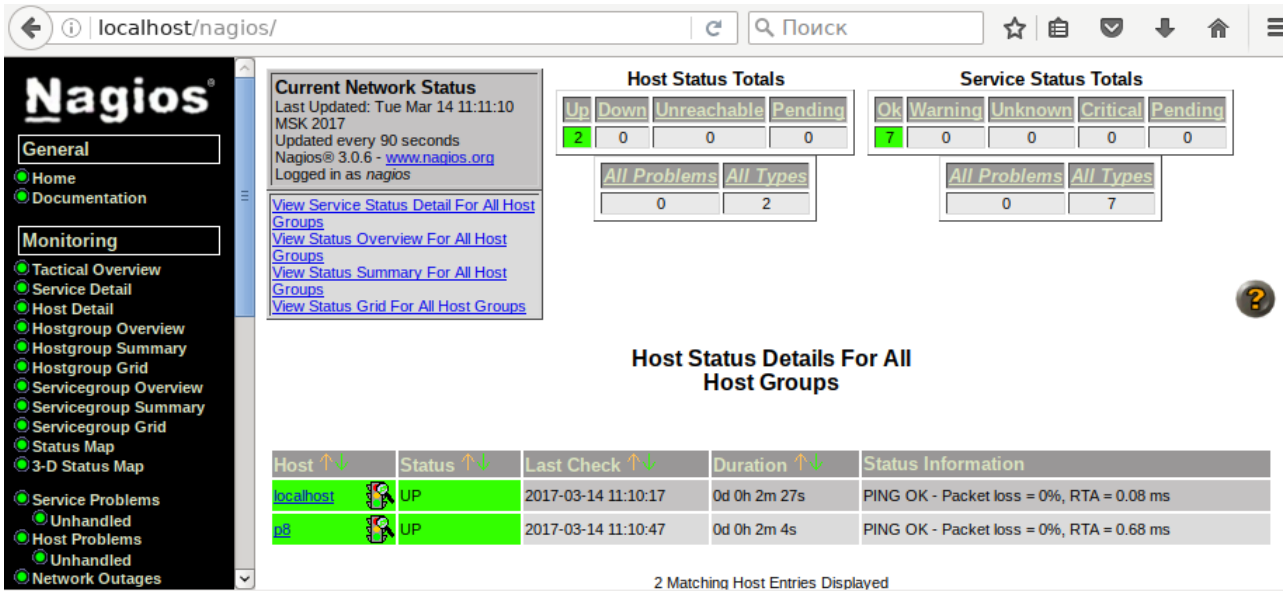

Рис. 47 – Список узлов

7.16.5. Nagstamon

Nagstamon – утилита, которая может подключаться к серверам мониторинга, например, к nagios, для того, чтобы обеспечить в режиме реального времени информацию о состоянии узлов и служб. Nagstamon в виде небольшой

настраиваемой строчки может висеть в любом месте экрана, отображая количество проблем в сети. При наведении на нее мышкой, выпадает список проблем.

Пакет nagstamon (если он еще не установлен) следует установить на сервере мониторинга:

```
# apt-get install nagstamon
```

При первом запуске Nagstamon («Приложения» → «Системные» →«Nagstamon») появляется диалоговое окно, в котором необходимо настроить хотя бы один монитор для проверки (рис. 48):

- тип сервера мониторинга: Nagios;
- URL-адрес главной страницы монитора: `http://localhost/nagios/`;
- URL-адрес монитора CGI: `http://localhost/nagios/`;
- имя пользователя: nagios;
- пароль: nagios;
- прокси, если необходимо.

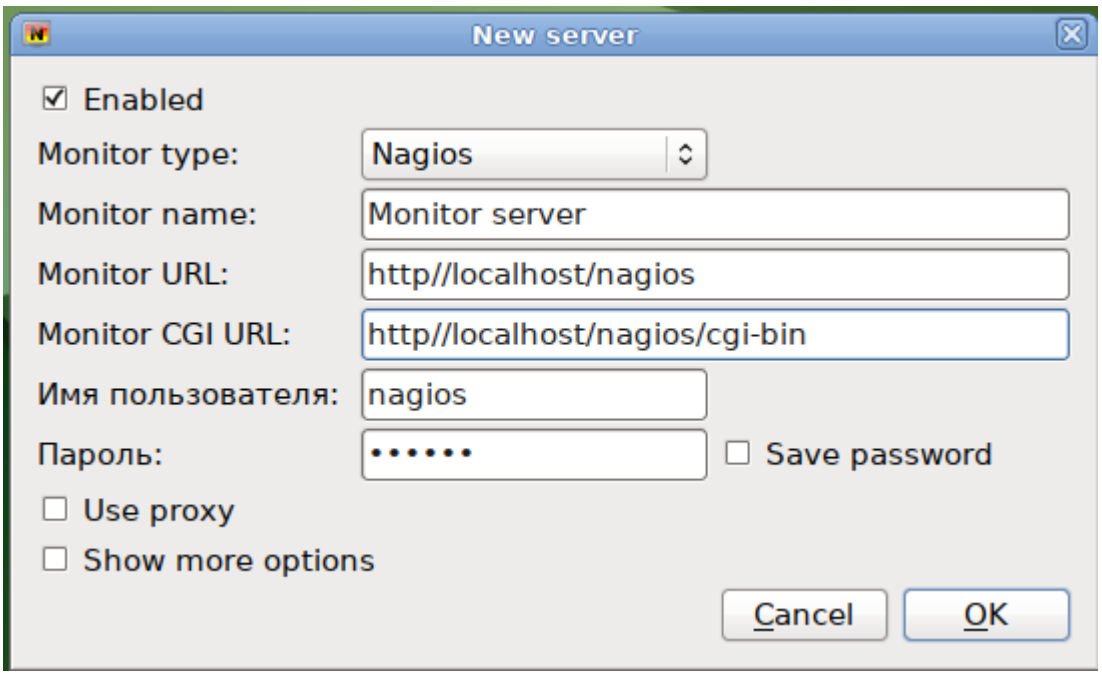

Рис. 48 – Настройка сервера мониторинга

Каталог config по умолчанию находится в `$HOME/.nagstamon`.

Nagstamon находится на рабочем столе, в виде перемещаемой строки состояния или полноэкранного режима, где представлено краткое описание (рис. 49)

критических, предупреждающих, неизвестных, недостижимых и недоступных узлов и сервисов. При касании указателем мыши уведомления, выводится подробный отчет о состоянии (рис. 50). Пользователи также могут получать звуковые сигналы.

Рис. 49 – Уведомление о критической ошибке

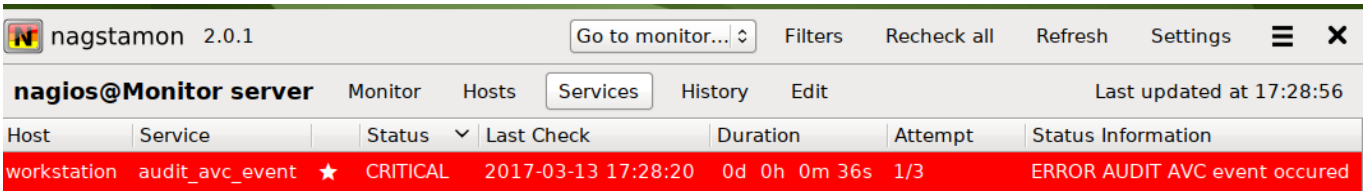

Рис. 50 – Просмотр отчета об ошибке

Nagstamon позволяет пользователю определять действия, предпринимаемые для отказавших узлов и служб. Также есть встроенные действия:

- Monitor – открыть страницу узла/службы в веб-интерфейсе монитора;
- Recheck – снова проверить состояние узла/службы;
- Acknowledge – позволяет признать проблему с узлом/службой;
- Downtime – позволяет настроить обслуживание службы/узла.

С удаленными узлами и службами можно устанавливать соединение через SSH, RDP, VNC или выполнить любые самоопределяемые действия.

В качестве примера создать действие, которое будет проверять доступность узла, командой ping. Для этого из контекстного меню выбрать команду «Edit action» (Редактировать действие) (рис. 51). В открывшемся окне необходимо нажать кнопку «New action…» (Новое действие).

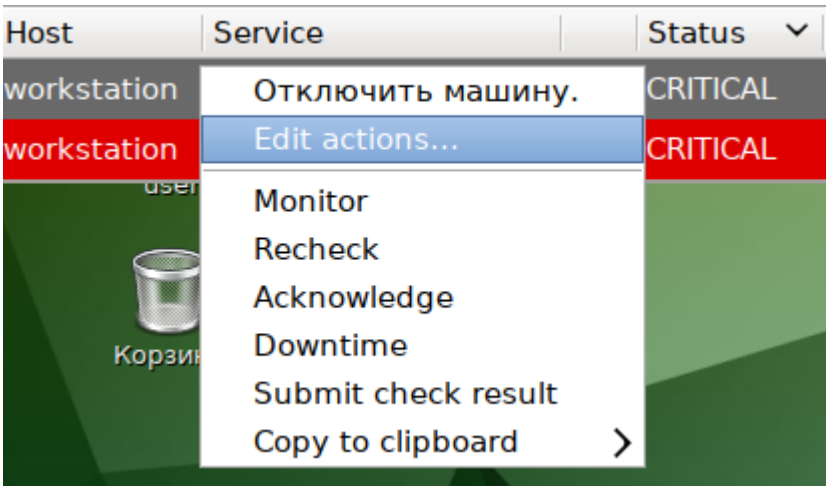

Рис. 51 – Контекстное меню Nagstamon

Существует три типа действий:

- Browser – открыть браузер с определенным URL-адресом;
- Command – вызов внешней команды с некоторыми связанными аргументами;
- URL – вызывать любой URL в фоновом режиме с аргументами, например, действие CGI.

Команды и URL-вызовы могут быть построены с использованием некоторых переменных-заполнителей.

Необходимо выбрать в поле «тип действия»: command. Далее необходимо указать уникальное имя, например, PING. Содержимое поля «Строка» будет передано как внешний вызов (рис. 52):

```
mate-terminal -e "ping $ADDRESS$"
```

Регулярными выражениями можно отфильтровать узлы и службы, чтобы меню действий оставалось как можно более удобным. Для сохранения изменений необходимо нажать кнопку OK.

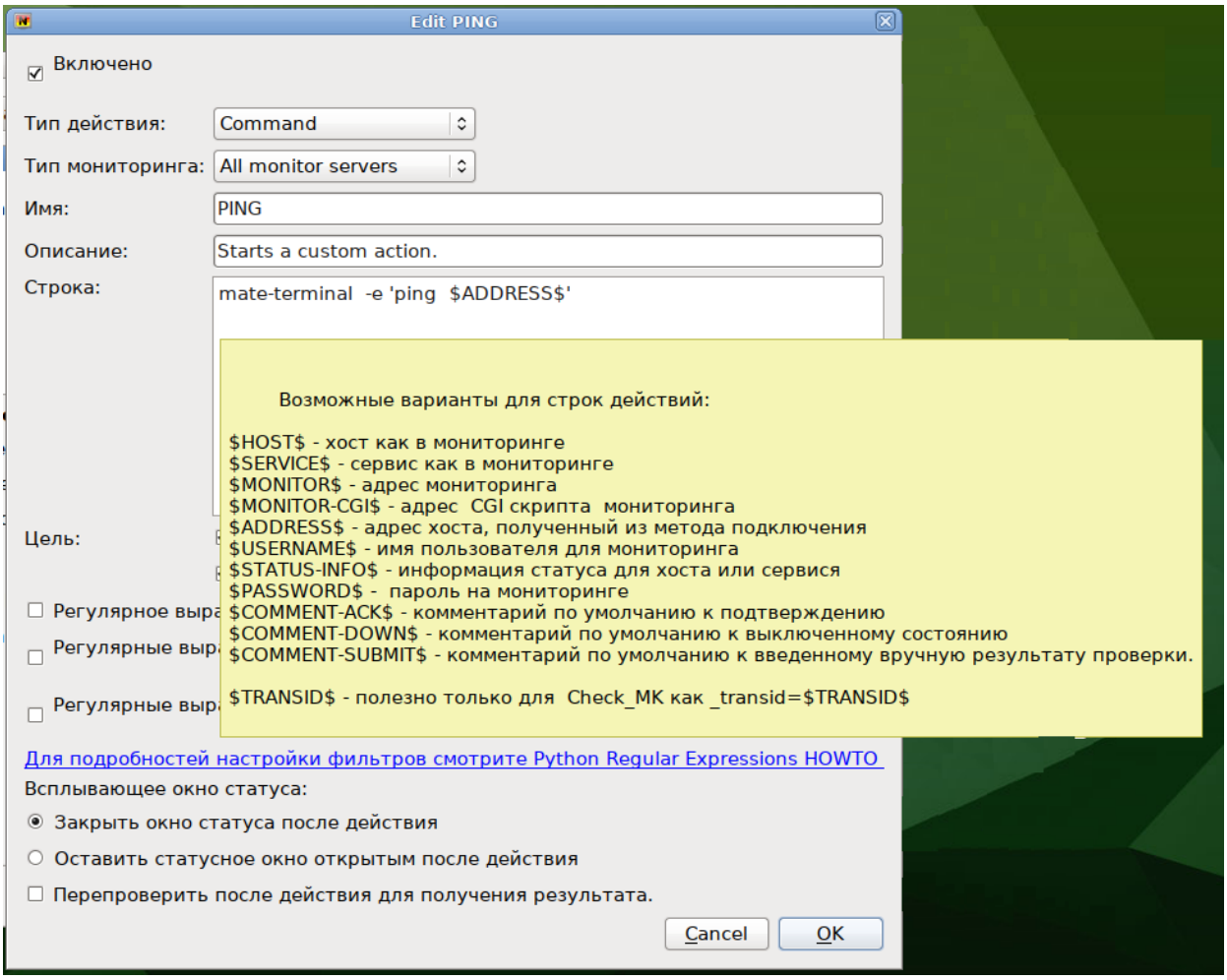

Рис. 52 – Добавление нового действия

7.16.6. Реагирование на сообщения системы сигнализации

Для блокировки рабочих станций на них должны быть настроены соответствующие разрешения:

```
# chsh –s /bin/bash nagios
# echo «nagios ALL = (root) NOPASSWD: /bin/openvt» >>/etc/sudoers
# control sudo public
```

Сгенерировать ключ пользователя-контролера на управляющей машине при помощи ssh-keygen, положить ssh key пользователя-контролера в authorized_keys пользователя nagios на рабочих станциях.

Возможные типы событий:

- check_devices – реагирование на вставление неавторизованного устройства (авторизованность устройства задается в утилите alterator-ports-access);
- check_osec – реагирование на возникновение события osec;

- check_authdata –реагирование на несанкционированное изменение файлов паролей и групп;
- check_audit – реагирование на события системы аудита;
- check_logindata – реагирование на события превышения пользователями заданного порогового значения ввода неверной аутентифицирующей информации при входе в ОС;
- check_printer – реагирование на попытки несанкционированного получения твердых копий защищаемой информации (вывода и информации на печать).

По событию сигнализации в меню правой кнопкой можно выделить пункт «заблокировать компьютер». При этом указанный мышью компьютер с сигналом будет заблокирован и разблокировать его можно будет только с консоли, введя пароль root.

Для сброса сигнала оповещения, администратор должен устранить причину, зайдя на удаленную машину по ssh или непосредственно зарегистрировавшись на ней. После административной реакции на машине, просмотреть содержимое `/var/lib/nagwag/<authdata | device | osec | audit | logindata | printer>` и переместить содержимое этого каталога в соответсвующий каталог с постфиксом `_archived`.

При удалении логов journalctl рекомендуется удалять содержимое `/var/lib/nagwad/*archived/`. То же действие необходимо делать при получении события слишком долгого выполнения NRPE plugin.

7.17. Управление печатью

В ОС Альт 8 СП используется система печати CUPS, которая позволяет выполнять следующие действия:

- управляет заданиями на печать;
- исполняет административные команды;
- предоставляет информацию о состоянии принтеров локальным и удаленным программам;
- информирует пользователей, если это требуется.

Система печати CUPS решает задачу монопольной постановки задания в очередь на печать. Данная функция предполагает невозможность вывода документа на печать в обход системы печати.

Существует два способа настройки принтера:

- утилита «Настройка принтера» (пакет system-config-printer);
- веб-интерфейс CUPS (Common UNIX Printing System) (пакет cups).

7.17.1. Устройство CUPS

В состав файлов конфигурации CUPS входят следующие файлы:

- файл конфигурации сервера CUPS (`/etc/cups/cupsd.conf`);
- файлы определения принтеров и классов (`/etc/cups/printers.conf`, `/etc/cups/classes.conf`);
- файлы типа MIME и файлы правил преобразования;
- файлы описания PostScript-принтеров (PPD).

7.17.1.1. Файл конфигурации сервера CUPS

Конфигурационный файл сервера очень похож на файлы конфигурации веб-сервера и определяет все свойства управления доступом. Настраивать CUPS можно либо непосредственно редактируя файл конфигурации `/etc/cups/cupsd.conf`, либо в веб-интерфейсе CUPS (рис. 53). Веб-интерфейс CUPS можно запустить следующими способами:

- в графической среде MATE: «Приложения» → «Системные» → «Настройка печати»;

в браузере: `http://localhost:631`.

Если файл `cupsd.conf` редактируется в консоли для применения изменения, необходимо перезапустить службу cups, выполнив команду:

```
# systemctl restart cups
```

Если файл `cupsd.conf` редактируется в веб-интерфейсе, то служба cups автоматически перезапускается после нажатия на кнопку «Сохранить изменения».

Файл конфигурации `cupsd.conf` начинается с ряда глобальных директив, которые оформлены в виде пар имя – значение.

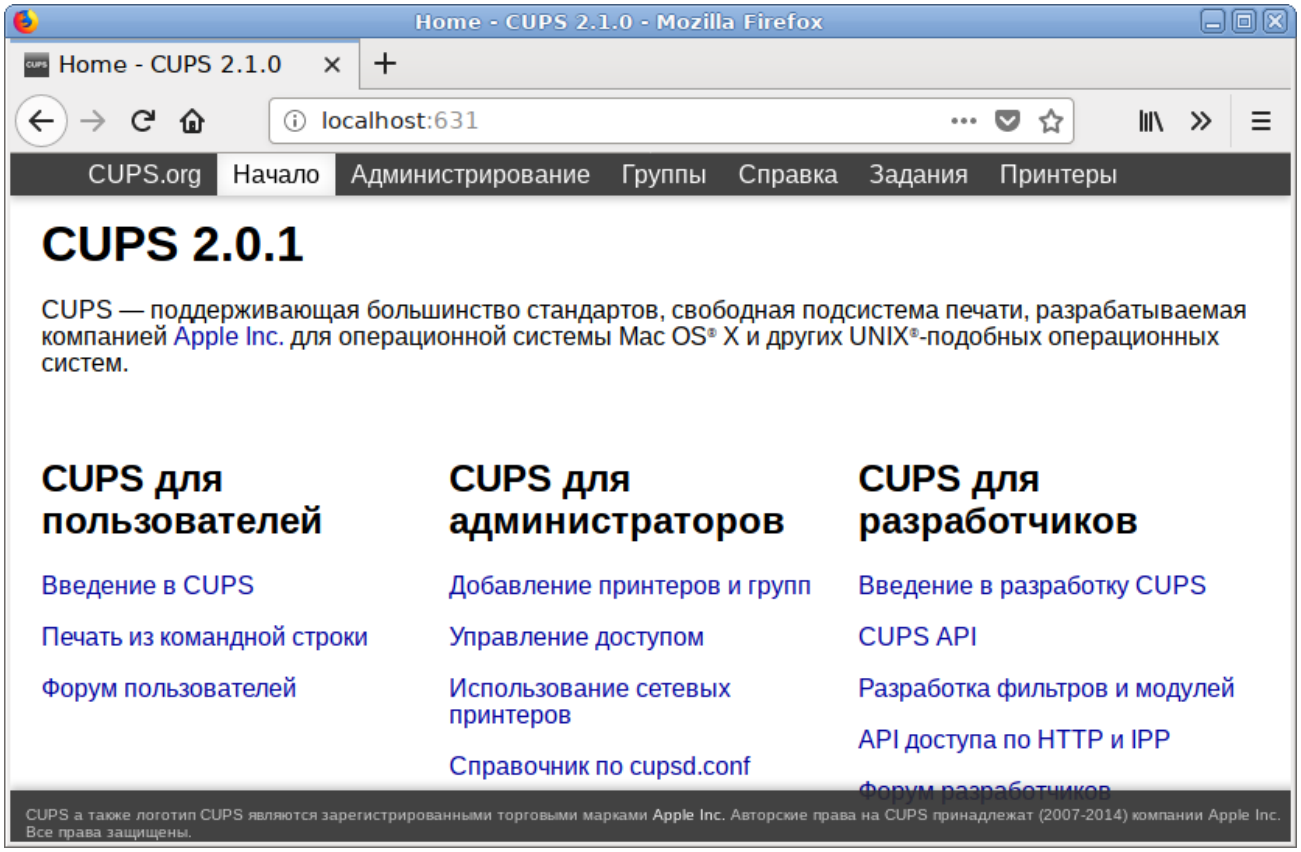

Рис. 53 – Веб-интерфейс CUPS

`LogLevel` указывает подробность журналирования. Доступные значения: `none` (не записывать логи), `emerg`, `alert`, `crit`, `error`, `warn` (по умолчанию), `notice`, `info`, `debug`, `debug2` (подробный вывод).

PageLogFormat определяет формат строк журнала печати (файл `/var/log/cups/page_log`). Последовательности начинающиеся со знака процента (%), заменяются соответствующей информацией:

- `% {name}` – значение указанного атрибута IPP;
- `% C` – количество копий для текущей страницы;
- `% P` – номер текущей страницы;
- `% T` – текущую дату и время в общий формат журнала;
- `% j` – идентификатор задания;
- `% p` – имя принтера;
- `% u` – имя пользователя.

По умолчанию строка `PageLogFormat` пустая (журнал печати не пишется). Для ведения журнала печати можно изменить эту строку:

```
PageLogFormat "%p %u %j %T %P %C %{job-billing} %{job-originating-host-name} %{job-name} %{media} %{sides}"
```

`MaxLogSize` задает максимальный размер журналов до их ротации. Значение 0 отключает ротацию.

`Listen` позволяет указать на каком IP-адресе будет доступен веб-интерфейс (по умолчанию `localhost:631`), а так же прослушиваемый сокет.

Параметры Browsing задают настройки возможности CUPS обнаруживать принтеры в сети. Данная возможность поддерживается на уровне протокола IPP. Обнаружение происходит посредством широковещательных рассылок, что при большом количестве серверов CUPS или при частом отключении/подключении принтеров может порождать дополнительную нагрузку на сеть. Browsing – указывает CUPS предоставлять свои серверы в общий доступ, либо нет. Значения может принимать `On` или `Off` соответственно.

Директива `DefaultAuthType` указывает механизм аутентификации, который будет использоваться для организации доступа (по умолчанию Basic – использовать логины/пароли от локальной системы).

`BrowseAllow` и `BrowseDeny` – указывают CUPS на стороне клиента адреса, от которых может приниматься или отвергаться, соответственно, информация о принтерах. Формат директив соответствует директивам Allow и Deny. В качестве аргумента для данной директивы может быть как отдельный IP, так и подсеть в формате 10.0.0.0/24 или 10.0.0.0/255.255.255.0 или 10.0.0.0-10.0.0.255, так и значение `@LOCAL` – обозначающее локальную сеть, а так же имена хостов. Возможно использование нескольких данных директив.

Директива `Order` определяет порядок предоставления доступа к CUPS по умолчанию. Значение `allow,deny` определяет что доступ запрещен, если право на доступ не указано явно. Если директива имеет значение `deny,allow`, то доступ будет разрешен, если явно не запрещен.

Далее идут параметры, сгруппированные в разделы `<Location /...>`. Такие директивы определяют доступ к определенным функциям сервера:

- `<Location />` – доступ к серверу;
- `<Location /admin>` – доступ к странице администрирования;
- `<Location /admin/conf>` – доступ к конфигурационным файлам;
- `<Location /jobs>` – доступ к заданиям;
- `<Location /printer>` – доступ к принтерам.

7.17.1.2. Управление политиками операций

Политики операций – это правила, используемые для каждой операции IPP в CUPS. Правила могут включать такие опции, как «пользователь должен предоставить пароль», «пользователь должен находиться в системной группе», «разрешать только из локальной системы» и т. д.

CUPS позволяет полностью переопределить правила для каждой операции и/или принтера. Каждая политика имеет название и определяет правила контроля доступа для каждой операции IPP.

Политики операций используются для всех запросов IPP, отправленных в планировщик заданий, и оцениваются после правил управления доступом на основе местоположения. Таким образом политики операций могут только добавлять дополнительные ограничения безопасности к запросу, а не ослаблять их. Для ограничений на уровне сервера необходимо использовать правила управления доступом на основе местоположения, а для ограничений на отдельные принтеры, задачи или службы – политики операций.

Политики хранятся в файле `cupsd.conf` в разделах Policy. Каждая политика имеет название, которое используется для ее выбора. Внутри раздела политики находятся один или несколько подразделов Limit, в которых перечислены операции, на которые влияют правила внутри него.

Каждая политика имеет название. В названии политики можно использовать те же символы, что и в названии принтера, в частности все печатные символы, кроме пробела, слэша (/) и решетки (#).

В разделах `< Limit …>` определяется, какие ограничения должна содержать политика. Директивы внутри подраздела `Limit` могут использовать любую из директив ограничения: `Allow`, `AuthType`, `Deny`, `Encryption`, `Require` и `Satisfy`. В таблице 11 перечислены основные примеры для разных правил контроля доступа.

Т а б л и ц а 11 – Правила контроля доступа

Уровень доступа	Директива
Разрешить всем	`Order allow,deny` `Allow from all`
Разрешить всем в локальной сети	`Order allow,deny` `Allow from @LOCAL`
Запретить всем/Отклонить операции	`Order allow,deny`
Требовать аутентификацию пользователя (Логин, Пароль)	`AuthType Basic`
Требовать CUPS аутентификацию CUPS (lppasswd) Password	`AuthType BasicDigest`
Требовать Kerberos	`AuthType Negotiate`
Только владелец	`Require user @OWNER`
Только администратор	`Require user @SYSTEM`
Члены группы foogroup	`Require user @foogroup`
Пользователи test или test1	`Require user test test1`
Требовать шифрование	`Encryption Required`

Пример политики, которая разрешает доступ только из подсети 10.110.1.x:

```
<Policy mypolicy>
# Операции, связанные с заданиями доступны только владельцам
# членам группы lab999 и администратору...
     <Limit Send-Document Send-URI Hold-Job Release-Job Restart-Job Purge-Jobs Set-Job-Attributes Create-Job-Subscription Renew-Subscription Cancel-Subscription Get-Notifications Reprocess-Job Cancel-Current-Job Suspend-Current-Job Resume-Job Cancel-My-Jobs Close-Job CUPS-Move-Job>
          Require user @OWNER @lab999 @SYSTEM
          Order allow,deny
          Allow from 10.110.1.0/24
     </Limit>
```

```
        # Все административные операции доступны только
администратору и членам группы lab999, также необходима процедура
аутентификации...
        <Limit Pause-Printer Resume-Printer Set-Printer-Attributes
Enable-Printer Disable-Printer Pause-Printer-After-Current-Job
Hold-New-Jobs Release- Held-New-Jobs Deactivate-Printer Activate-
Printer Restart-Printer Shutdown-Printer Startup-Printer Promote-Job
Schedule-Job-After CUPS- Accept-Jobs CUPS-Reject-Jobs CUPS-Set-Default>
            AuthType Default
            Require user @lab999 @SYSTEM
            Order allow,deny
            Allow from 10.110.1.0/24
        </Limit>

        # Все остальные операции доступны из подсети 10.110.1.0/24 с
обязательной аутентификацией пользователей...
        <Limit All>
            AuthType Default
            Order allow,deny
            Allow from 10.110.1.0/24
        </Limit>
    </Policy>
```

После создания политики ее можно использовать двумя способами.

Первый способ – назначить ее в качестве политики по умолчанию для всей системы, используя директиву `DefaultPolicy` в файле `cupsd.conf`. Например:

```
DefaultPolicy mypolicy
```

Второй способ – связать политику с одним или несколькими принтерами. Для этого можно воспользоваться командой lpadmin (8) или веб-интерфейсом для изменения политики операций для каждого принтера. Например:

```
# lpadmin -p HP_LaserJet_M1536dnf_MFP -o printer-op-
policy=mypolicy
```

7.17.1.3. Файлы описания принтеров и классов

Файлы описания принтеров и классов перечисляют доступные очереди печати и классы. Классы принтеров – наборы принтеров. Задания, посланные классу принтеров, направляются к первому доступному принтеру данного класса. Для редактирования фалов `/etc/cups/printers.conf` и `/etc/cups/classes.conf` можно использовать утилиту lpadmin.

Пример настройки для локального принтера:

```
<DefaultPrinter laserjet>
UUID urn:uuid:7efaaede-819d-3d9a-6270-3fe957597756
Info laserjet
Location host-15.localdomain
MakeModel HP LaserJet m1537dnf MFP pcl3, hpcups 3.19.1
DeviceURI
usb://HP/LaserJet%20M1536dnf%20MFP?serial=00CND9D8YC9C&interface=1
State Idle
StateTime 1553167952
ConfigTime 1553167952
Type 36892
Accepting Yes # принтер принимает задания
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer # остановить принтер при ошибке
Option job-hold-until indefinite
</DefaultPrinter>
```

7.17.1.4. Очередь печати

Очередь печати – механизм, который позволяет буферизовать и организовать задания, посылаемые на принтер. Необходимость организации такого механизма обуславливается тем, что принтер является медленно действующим устройством, и задания не могут быть распечатаны мгновенно.

Очевидно, что в многопользовательской среде возникает конкуренция со стороны пользователей при доступе к принтерам, поэтому задания необходимо располагать в очереди. Для этого используется буферный каталог `/var/spool/cups/`.

Файлы типов MIME перечисляют поддерживаемые MIME-типы (text/plain, application/postscript) и правила для автоматического обнаружения формата файла. Они используются сервером для определения поля Content-Type для GET- и HEAD-запросов и обработчиком запросов протоколов сетевой печати IPP (Internet Printing Protocol), чтобы определить тип файла.

Правила преобразования MIME перечисляют доступные фильтры. Фильтры используются, когда задание направляется на печать, таким образом, приложение может послать файл удобного (для него) формата системе печати, которая затем преобразует документ в требуемый печатный формат. Каждый фильтр имеет относительную «стоимость», связанную с ним, и алгоритм фильтрования выбирает набор фильтров, который преобразует файл в требуемый формат с наименьшей общей «стоимостью».

Файлы PPD описывают возможности всех типов принтеров. Для каждого принтера имеется один PPD-файл. Файлы PPD для не-PostScript-принтеров определяют дополнительные фильтры посредством атрибута cupsFilter для поддержки драйверов принтеров.

В ОС стандартным языком описания страниц является язык PostScript. Большинство прикладных программ (редакторы, браузеры) генерируют программы печати на этом языке. Когда необходимо напечатать ASCII-текст, программа печати может быть ASCII-текстом. Имеется возможность управления размером шрифтов при печати ASCII-текста.

Управляющая информация используется для контроля доступа пользователя к принтеру и аудита печати. Также имеется возможность печати изображений в форматах GIF, JPEG, PNG, TIFF и документов в формате PDF.

Фильтр – программа, которая читает из стандартного входного потока или из файла, если указано его имя. Все фильтры поддерживают общий набор опций, включающий имя принтера, идентификатор задания, имя пользователя, имя задания, число копий и опции задания. Весь вывод направляется в стандартный выходной поток.

Фильтры предоставлены для многих форматов файлов и включают, в частности, фильтры файлов изображения и растровые фильтры PostScript, которые поддерживают принтеры, не относящиеся к типу PostScript. Иногда несколько фильтров запускаются параллельно для получения требуемого формата на выходе.

Программа backend – это специальный фильтр, который отправляет печатаемые данные устройству или через сетевое соединение. В состав системы печати включены фильтры для поддержки устройств, подключаемых с помощью параллельного и последовательного интерфейсов, а также шины USB.

Клиентские программы используются для управления заданиями и сервером печати.

Управление заданиями включает выполнение следующих действий:

- формирование;
- передачу серверу печати;
- мониторинг и управление заданиями в очереди на печать.

Управление сервером включает выполнение следующих действий:

- запуск/остановку сервера печати;
- запрещение/разрешение постановки заданий в очередь;
- запрещение/разрешение вывода заданий на принтер.

Основные пользовательские настройки содержатся в файлах конфигурации `client.conf` и `~/.cups/lpoptions`.

Для удаленного использования сервера печати необходимо от имени пользователя с идентификатором root выполнить следующие команды:

```
cupsctl --remote-admin --remote-printers --remote-any
cupsctl ServerAlias=*
```

В случае использования сервера печати в едином пользовательском пространстве (далее – ЕПП) необходимо задание соответствующего типа аутентификации: для работы в ЕПП значение параметра должно быть `DefaultAuthType Negotiate`, без использования ЕПП значение параметра должно быть `DefaultAuthType Basic`.

В файле конфигурации клиента `client.conf` должен быть задан один параметр ServerName, определяющий имя сервера печати, например:

```
ServerName computer.domain
```

В общем случае вывод данных на принтер происходит следующим образом:

1) программа формирует запрос на печать задания к серверу печати;
2) сервер печати принимает подлежащие печати данные, формирует в буферном каталоге файлы с содержимым задания и файлы описания задания, при этом задание попадает в соответствующую очередь печати;
3) сервер печати просматривает очереди печати для незанятых принтеров, находит в них задания и запускает конвейер процессов, состоящий из фильтров и заканчивающийся выходным буфером, информация из которого поступает в принтер посредством драйверов ОС;
4) контроль и мониторинг процесса печати выполняется с помощью программ lpq, lpc, lprm, lpstat, lpmove, cancel.

7.17.2. Установка принтера

Перед началом установки необходимо убедиться в том, что в случае локального подключения принтер присоединен к соответствующему порту компьютера и включен, а в случае сетевого подключения принтер корректно сконфигурирован для работы в сети.

Окно «Настройки принтера» можно запустить следующими способами:

- в графической среде MATE: «Система» → «Администрирование» → «Настройки принтера»;
- из командной строки: командой system-config-printer.

Для добавления принтера в диалоговом окне «Настройки принтера» необходимо нажать кнопку «Добавить» (рис. 54).

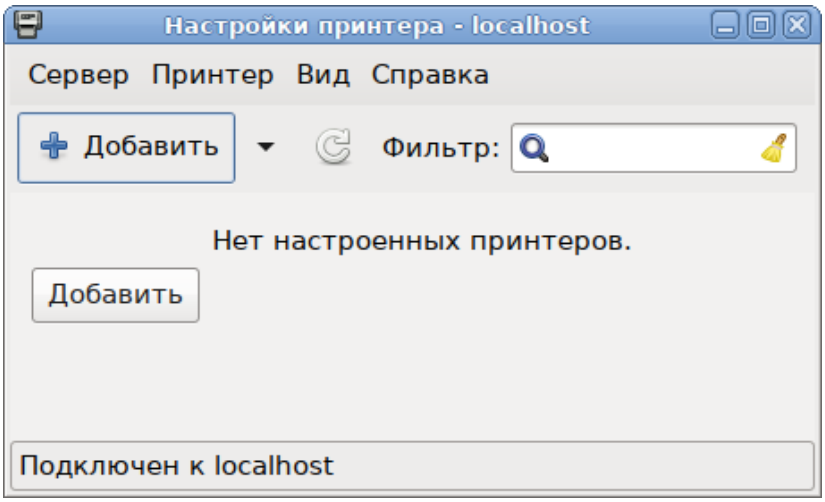

Рис. 54 – Диалоговое окно «Настройки принтера»

Примечание. Если возникает ошибка «Служба печати недоступна», необходимо имени системного администратора root выполнить команду `systemctl restart cups`. После этого следует вернуться к окну «Настройки принтера» и нажать кнопку «Обновить».

В диалоговом окне «Аутентификация» следует ввести имя, и пароль пользователя, имеющего право изменять настройки принтера, после чего нажать кнопку «ОК» (рис. 55).

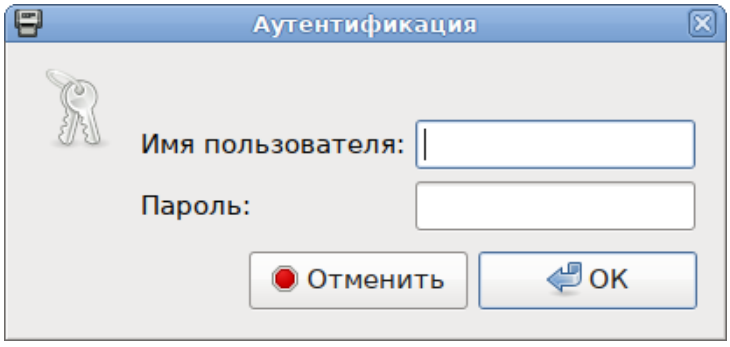

Рис. 55 – Диалоговое окно «Аутентификация»

Далее в открывшемся окне следует выбрать принтер, который необходимо подключить и нажать кнопку «Далее» (рис. 56).

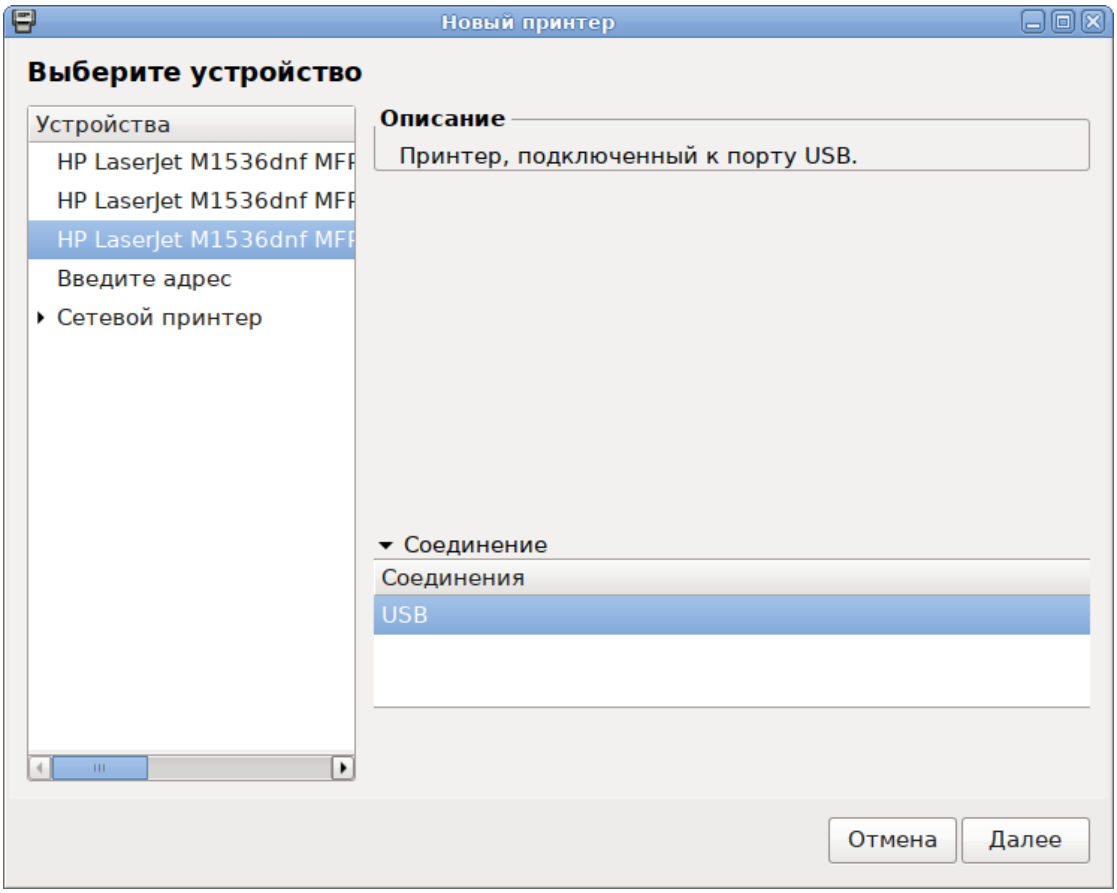

Рис. 56 – Выбор принтера

На следующих шагах настройки принтера необходимо выбрать драйвер для принтера. Драйвер можно выбрать из базы данных, содержащей различные файлы описания принтеров (PPD-файлы) от производителей или предоставить файл описания PostScript-принтера (рис. 57).

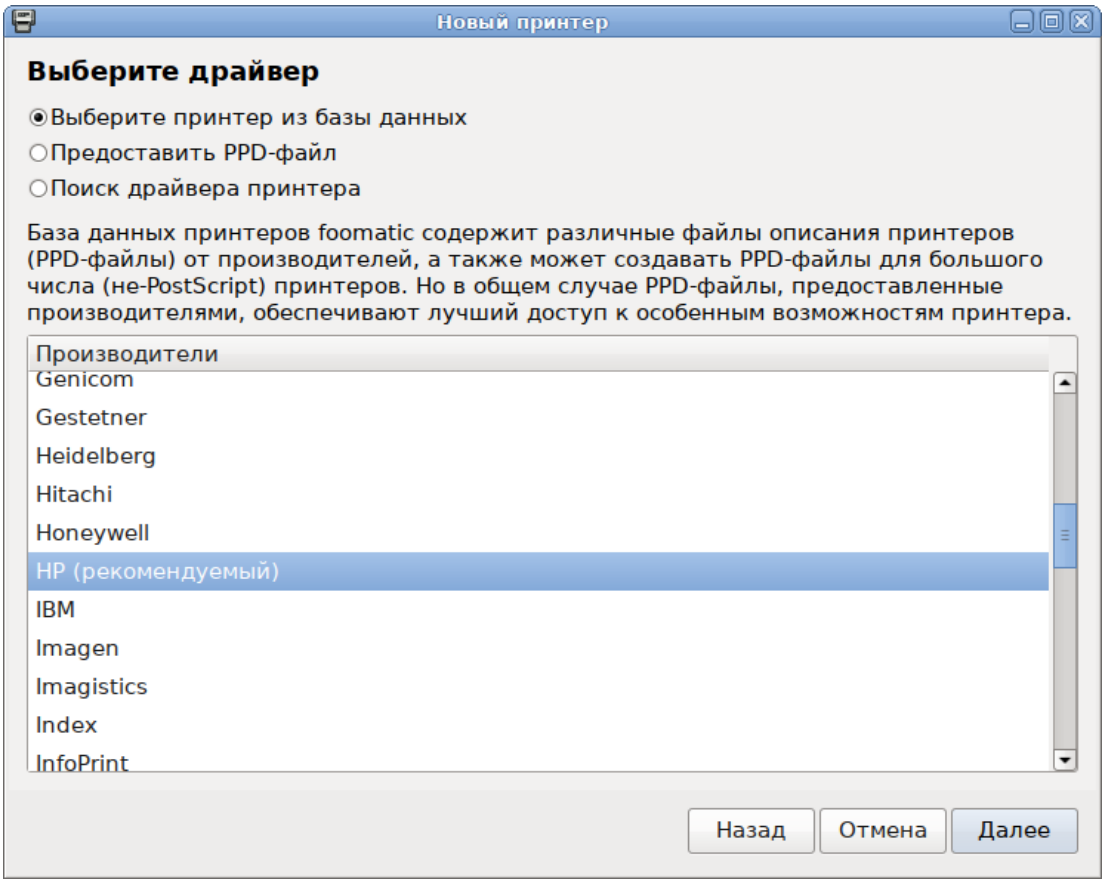

Рис. 57 – Выбор источника драйвера принтера

После выбора драйвера в окне «Новый принтер» можно изменить название и описание принтера (рис. 58).

После нажатия кнопки «Применить» установка принтера завершена, принтер станет доступным для печати (рис. 59).

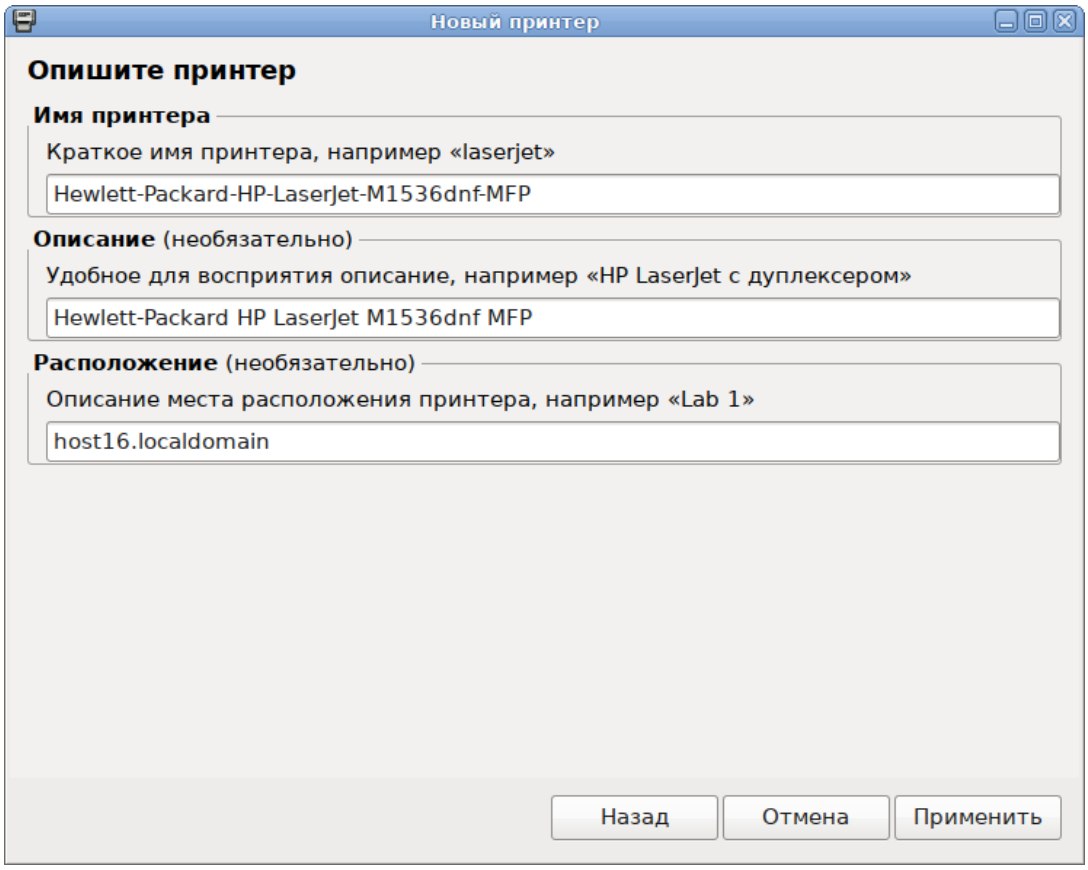

Рис. 58 – Название и описание принтера

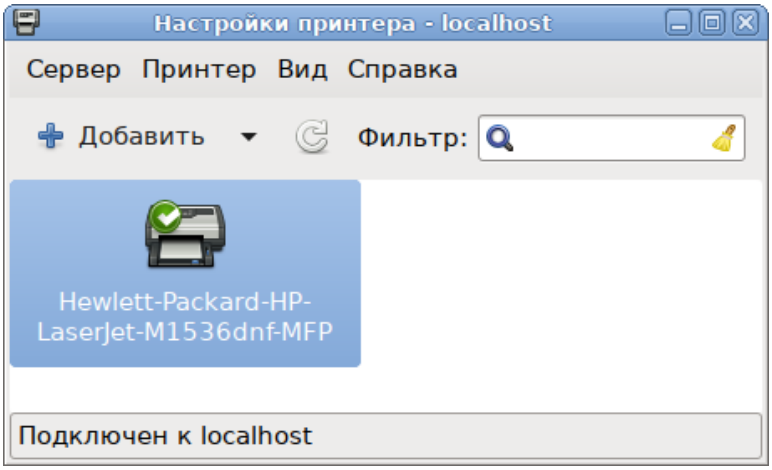

Рис. 59 – Выбор принтера

Изменить настройки принтера (разрешение, размер используемой по умолчанию бумаги, принтер по умолчанию и т. д.) можно в любой момент, выбрав в контекстном меню принтера пункт «Свойства».

7.17.3. Настройка сервера печати для сети

Если в сети имеются несколько принтеров или, когда принтеры не подключены непосредственно к тому компьютеру, на котором работает главный сервер CUPS, то целесообразно настроить сервер cupsd, так, чтобы он мог принимать задания на печать из сети.

По умолчанию сервер CUPS работает с локально установленными принтерами, для того, чтобы он мог обрабатывать задания из сети, в конфигурационный файл `/etc/cups/cupsd.conf` нужно внести следующие изменения:

- разрешить доступ к серверу – добавить в секцию `Location` директиву `Allow from`:

```
<Location />
    Order allow,deny
    Allow localhost
    Allow from ip-address/netmask
</Location>
```

- включить отображение (обнаружение) общего принтера:

```
...
Browsing On
BrowseOrder allow,deny
BrowseAllow 192.168.1.* #локальная сеть
BrowseAddress 192.168.1.*:631#локальная сеть
```

Примечание. Включить отображение (обнаружение) общего принтера можно также отметив пункт «Разрешить совместный доступ к принтерам, подключенным к этой системе» в веб-интерфейсе на вкладке «Администрирование».

После внесения изменений необходимо перезапустить службу cups:

```
# systemctl restart cups
```

На клиентах также должен быть установлен CUPS. После установки системы печати на клиенте, CUPS-принтеры, присутствующие в сети, автоматически находятся менеджерами принтеров. В качестве альтернативы, можно воспользоваться веб-интерфейсом CUPS на клиентской машине по адресу

`http://localhost:631`. Если принтер не был обнаружен автоматически, можно ввести IPP или HTTP адрес (URI) сетевого CUPS принтера:

`ipp://server-name-or-ip/printers/printername`

или

`http://server-name-or-ip:631/printers/printername`

Если CUPS клиент не находит в сети принтеры, доступные через сервер CUPS, то иногда может помочь создание или изменение файла `/usr/local/etc/cups/client.conf` с добавлением записи, подобной следующей:

```
ServerName server-ip
```

В этом случае `server-ip` необходимо заменить на IP-адрес сервера CUPS в сети.

7.17.4. Команды управления печатью

При печати через локальный сервер печати данные сначала формируются на локальном сервере, как для любой другой задачи печати, после чего посылаются на принтер, подключенный к данному компьютеру.

Вся информация, необходимая для драйвера принтера (используемое физическое устройство, удаленный компьютер и принтер для удаленной печати), содержится в файлах `/etc/cups/printers.conf` и `/etc/cups/ppd/<имя_очереди>.ppd`.

Примечание. Далее термин «принтер» в этом разделе используется для обозначения принтера, соответствующего одной записи в файле `/etc/cups/printers.conf`. Под термином «физический принтер» подразумевается устройство, с помощью которого производится печать на бумаге. В файле `/etc/cups/printers.conf` может быть несколько записей, описывающих один физический принтер различными способами.

В системе печати CUPS приняты следующие команды для управления печатью:

- `/usr/bin/lpr` – постановка заданий в очередь, совместима с командой `lpr` системы печати BSD UNIX;

- /usr/bin/lp – постановка заданий в очередь, совместима с командой lp системы печати System V UNIX;
- /usr/bin/lpq – просмотр очередей печати;
- /usr/sbin/lpc – управление принтером, является частичной реализацией команды lpc системы печати BSD UNIX;
- /usr/bin/lprm – отмена заданий, поставленных в очередь на печать;
- /usr/sbin/cupsd – сервер печати;
- /usr/sbin/lpadmin – настройка принтеров и классов принтеров;
- /usr/sbin/lpmove – перемещение задания в другую очередь;
- /usr/bin/fly-admin-printer – настройка системы печати, установка и настройка принтеров, управление заданиями.

CUPS предоставляет утилиты командной строки для отправления заданий и проверки состояния принтера. Команды lpstat и lpc status также показывают сетевые принтеры (принтер@сервер), когда разрешен обзор принтеров.

С помощью команды lp выполняется передача задачи принтеру, то есть задача ставится в очередь на печать. В результате выполнения этой команды файл передается серверу печати, который помещает его в каталог /var/spool/cups/.

Остановить работу сервиса печати можно с помощью команды:

```
# systemctl stop cups
```

Запустить сервис печати можно с помощью команды:

```
# systemctl start cups
```

7.17.4.1. Настройка принтера

Настроить принтер в ОС можно также с помощью команды lpadmin. Ее запуск с опцией -p выполняется для добавления или модификации принтера:

```
/usr/sbin/lpadmin -p printer [опции]
```

Для lpadmin существуют также опции по регулированию политики лимитов и ограничений по использованию принтеров и политики доступа к принтерам.

Для удаления принтера необходимо выполнить lpadmin с опцией -x:

```
/usr/sbin/lpadmin -x printer
```

7.17.4.2. Проверка очереди печати

Команда `lpq` предназначена для проверки очереди печати (используемой `lpd`) и вывода состояния заданий на печать, указанных при помощи номера задания либо системного идентификатора пользователя, которому принадлежит задание.

`lpq` выводит для каждого задания имя его владельца, текущий приоритет задания, номер задания и размер задания в байтах, без параметров выводит состояние всех заданий в очереди.

7.17.4.3. Удаление задания из очереди печати

Команда `lprm` предназначена для удаления задания из очереди печати. Для определения номера задания необходимо использовать команду `lpq`. Для удаления задания, необходимо быть его владельцем или пользователем с идентификатором root.

Системные каталоги, определяющие работу системы печати ОС, также содержат файлы, которые не являются исполняемыми:

- `/etc/cups/printers.conf` – содержит описания принтеров в ОС;
- `/etc/cups/ppd/<имя_очереди>.ppd` – содержит описания возможностей принтера, которые используются при печати заданий и при настройке принтеров;
- `/var/log/cups/error_log` – содержит протокол работы принтера, в этом файле могут находиться сообщения об ошибках сервера печати или других программ системы печати;
- `/var/log/cups/access_log` – содержит все запросы к серверу печати;
- `/var/log/cups/page_log` – содержит сообщения, подтверждающие успешную обработку страниц задания фильтрами и принтером.

7.17.4.4. Настройка сетевого принтера из консоли

Для настройки принтера из консоли необходимо выполнить следующие действия:

1) получить права администратора;

2) просмотреть содержимое каталога `model` на наличие необходимых драйверов:

```
ls /usr/share/cups/model
```

3) если драйвер устройства присутствует перейти к шагу № 7 (настройка нового устройства);

4) найти необходимое устройство:

```
lpinfo -m | grep название_модели
```

5) просмотреть данные о драйвере устройства:

```
foomatic-ppdfile -A | grep название_модели
```

6) сформировать файл `.ppd`:

```
foomatic-ppdfile -p ‘имя_ppd_драйвера’ > /usr/share/cups/model/имя_ppd_файла.ppd
```

7) произвести настройку нового устройства:

- если принтер подключен по сети:

```
lpadmin -p название_принтера -D еще_одно_название -m название_ppd_файла.ppd -v socket://ip_принтера -E
```

- если принтер подключен по usb:

```
lpadmin -p название_принтера -D еще_одно_название -m название_ppd_файла.ppd -v "usb://адрес_принтера" -E
```

8) печать документа:

```
lp -d название_принтера /путь_документ
```

Примечание. Список доступных устройств можно просмотреть, выполнив команду: `lpinfo –v`

Пример вывода:

```
usb://Samsung/M262x%20282x%20Series?serial=ZD1UBJCD5000LVW
```

Список установленных принтеров:

```
lpstat –p -d
```

Пример настройки сетевого принтера Kyocera Ecosys P2235dn:

1) получить права администратора;

2) просмотреть содержимое каталога `/usr/share/cups/model` на наличие необходимых драйверов:

```
ls /usr/share/cups/model
```

3) если драйвер устройства присутствует произвести настройку нового устройства (перейти к 7 шагу);
4) найти необходимое устройство:

```
lpinfo -m | grep Kyocera-P-2
```

5) просмотреть данные о драйвере устройства:

```
foomatic-ppdfile -A | grep Kyocera-P-2
```

6) сформировать файл .ppd:

```
foomatic-ppdfile -p 'Kyocera-P-2000' > /usr/share/cups/model/Kyocera.ppd
```

7) создать новое устройство:

```
lpadmin -p Kyocera -D Kyocera-P-2000 -m Kyocera.ppd -v socket://10.120.70.90 -E
```

7.18. Управление базами данных

В качестве СУБД в составе ОС Альт 8 СП используется PostgreSQL.

СУБД PostgreSQL предназначена для создания и управления реляционными БД и предоставляет многопользовательский доступ к расположенным в них данным. Данные в реляционной БД хранятся в отношениях (таблицах), состоящих из строк и столбцов. При этом единицей хранения и доступа к данным является строка, состоящая из полей, идентифицируемых именами столбцов. Кроме таблиц, существуют другие объекты БД (виды, процедуры), которые предоставляют доступ к данным, хранящимся в таблицах.

Для работы СУБД на НЖМД выделяется область для хранения БД, называемая «кластером БД». Кластер БД является набором БД, управляемых одним экземпляром сервера СУБД. Настройка работы отдельного экземпляра сервера СУБД так же определяется в рамках кластера соответствующими конфигурационными файлами.

7.18.1. Состав

СУБД PostgreSQL состоит из нескольких компонентов:

- postgresql – сервисная служба, реализующая непосредственно сервер БД;
- libpq – клиентская библиотека, предоставляющая доступ к серверу СУБД;

- набор серверных утилит для управления работой сервера и создания кластеров БД;
- набор клиентских утилит для создания и управления БД.

7.18.2. Настройка

Настройка сервера СУБД осуществляется установкой параметров в конфигурационном файле `postgresql.conf`. В дополнение к файлу `postgresql.conf` в PostgreSQL используется еще два конфигурационных файла, которые контролируют аутентификацию клиента.

По умолчанию все эти три файла находятся в каталоге данных кластера БД или в соответствующем кластеру конфигурационном каталоге, например `/etc/postgreslq/x.x/main`. За расположение указанных файлов отвечают конфигурационные параметры, описанные ниже:

- `data_directory` – определяет каталог для хранения данных;
- `config_file` – определяет основной конфигурационный файл сервера (`postgresql.conf`), значение этого параметра может быть задано только в командной строке postgres;
- `hba_file` – определяет конфигурационный файл для аутентификации по узлам (`pg_hba.conf`);
- `ident_file` – определяет конфигурационный файл для аутентификации по методу `ident` (`pg_ident.conf`);
- `external_pid_file` – определяет имя дополнительного файла с идентификатором процесса, который сервер создает для использования программами администрирования сервера.

8. КОНТРОЛЬНЫЕ ХАРАКТЕРИСТИКИ РАЗВЕРНУТОЙ ОС АЛЬТ 8 СП

После установки необходимо проверить корректность развертывания ОС Альт 8 СП путем подсчета и сличения контрольных характеристик установленных файлов.

В качестве контрольной характеристики файла выступает контрольная сумма.

Подробнее об интегральных контрольных суммах ПИ, расположении пофайловых отчетов подсчета, алгоритме подсчета контрольных сумм приведено в документе «Формуляр. ЛКНВ.11100-01 30 01».

В случае изменения контрольных сумм после применения критических обновлений ОС Альт 8 СП перечень измененных файлов и новые контрольные суммы необходимо внести в раздел «Особые отметки» документа «Формуляр. ЛКНВ.11100-01 30 01».

9. СООБЩЕНИЯ АДМИНИСТРАТОРУ

При возникновении проблем в процессе функционирования ОС Альт 8 СП появляются диагностические сообщения трех типов: информационные, предупреждающие и сообщения об ошибках.

Администратор должен проанализировать диагностические сообщения и принять меры по устранению появившихся проблем.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ЕПП – единое пользовательское пространство;

ВУ – вычислительный узел;

КСЗ – комплекс средств защиты;

ИРС – интерактивная рабочая среда;

ОС – операционная система;

ПИ – программное изделие;

ПО – программное обеспечение;

ПЭВМ – персональная электронная вычислительная машина;

СУ – сетевой узел;

УУ – управляющий узел;

ФС – файловая система;

ЦУС – центр управления системой.

Лист регистрации изменений									
Номера листов (страниц)					*Всего листов (страниц) в докум.*	*№ документа*	*Входящий № сопроводительного докум. и дата*	*Подп.*	*Дата*
Изм	*измененных*	*замененных*	*новых*	*анулированных*					