УТВЕРЖДЕН

ФЛИР.90001-01 34 01-ЛУ

ОС ОН «СТРЕЛЕЦ»

Руководство оператора

ФЛИР.90001-01 34 01

Листов 186

2019

Литера $О_1$

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

АННОТАЦИЯ

Настоящий документ является руководством оператора операционной системы общего назначения «Стрелец» (ОС ОН «Стрелец») (далее по тексту – ОС) ФЛИР.90001-01.

В документе описывается назначение ОС, условия выполнения, порядок выполнения и сообщения оператору.

Документ предназначен для ознакомления пользователей с работой ОС.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ОС

1.1. Назначение

ОС ОН «Стрелец» предназначена для создания автоматизированных систем в защищенном исполнении, обрабатывающих информацию ограниченного доступа, в том числе содержащую сведения, составляющие государственную тайну со степенью секретности до «совершенно секретно» включительно.

ОС функционирует на отечественных аппаратных платформах и совместима со средствами защиты информации (СЗИ), существующими в ведомственных органах ОС.

ОС может применяться с целью замещения импортных ОС, входящих в состав АС, обрабатывающих информацию ограниченного доступа, а также с целью объединения уже существующих разработок отечественных производителей программного обеспечения.

ОС разработана как операционная система семейства Debian Linux, функционирующая на аппаратной платформе с процессорной архитектурой x86-64.

1.2. Возможности

Основные возможности ОС:

– функционирование на аппаратной платформе с процессорной архитектурой x86-64;

– работа в режиме liveCD;

– наличие средств интеграции с доменом Windows;

– наличие замкнутой программной среды;

– СЗИ, позволяющие обрабатывать информацию ограниченного доступа;

– поддержка новейшего периферийного оборудования;

– поддержка основных сетевых протоколов;

– наличие средств автоматизации повседневной деятельности (офисные средства, работа с электронной почтой, гипертекстовыми данными, словари, работа с графикой, мультимедиа и т.д.).

1.3. Состав ОС

В состав ОС входят:

– средства установки ОС;

– загрузчик;

– набор ядер Linux и модулей ядра с механизмами защиты информации;

– системные компоненты и библиотеки;

– системные компоненты и библиотеки СЗИ;

– средства управления программными пакетами;

– системный менеджер;

– сетевые службы;

– защищенная подсистема печати;

– графическая подсистема (графический сервер, система управления сеансами пользователей и графический менеджер окон);

– средства организации домена;

– программное обеспечение (ПО) работы с мультимедиа;

– средства работы с офисными документами;

– средства виртуализации;

– система управления базами данных (СУБД);

– средства работы с интернетом (браузеры, почтовые клиенты, мессенджеры);

– средства резервного копирования и восстановления.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ

2.1. Технические средства

ОС функционирует на отечественных аппаратных платформах и совместима с СЗИ ОС, существующих в ведомственных органах.

Базовыми техническими средствами для функционирования ОС являются технические средства изделия КИ8603 ЦАВМ.461263.152.

2.2. Программные средства

Средства установки ОС:

– реализованы в виде загрузочного DVD диска, обеспечивающего загрузку в режиме Live;

– обеспечивают возможность установки ОС на компьютеры с архитектурой x86_64, с объемом оперативной памяти 2 ГБ и более, объемом жесткого диска (ЖД) 16 ГБ и более;

– поддерживают установку как в режиме BIOS, так и в режиме UEFI;

– поддерживают установку на диски с разбиением MBR и GPT;

– позволяют проводить первоначальные настройки компьютера, необходимые для инсталляции, разбиение ЖД;

– позволяют проводить установку даты и времени, конфигурирование сетевых интерфейсов, создание пользовательского аккаунта.

2.2.1. Базовые средства ОС

Базовые средства ОС:

– обеспечивают загрузку ОС в текстовом или графическом режиме по выбору пользователя;

– включают стандартные для Linux компоненты графического окружения, такие как xorg, lxqt, xfce, предоставляющие пользователю графический рабочий стол, содержащие файловый менеджер, эмулятор терминала, редактор текста, панель задач, средство просмотра изображений, индикаторы раскладки клавиатуры, монитор системных ресурсов (памяти, процессора, ЖД, процессов), программные средства (ПС) с открытым исходным кодом для записи, создания и копирования CD/DVD дисков с данными, CDDA, работы с образами дисков;

– включают средства интеграции с доменом Windows (SMB/CIFS), сервер службы каталогов и средства доступа к серверу по протоколу LDAP;

– включают средства, осуществляющие контроль и фильтрацию сетевого трафика в

соответствии с заданными правилами (iptables);

– обеспечивают базовые сетевые сервисы – DNS, DHCP, SSH, FTP, NTP, NFS, SNMP.

2.2.2. Средства установки и удаления приложений

Средства установки и удаления приложений обеспечивают:

– установку и удаление программ, поставляемых на установочном диске;

– обновление программ, поставляемых на диске с обновлениями;

– установку и удаление дополнительных программ из он-лайн репозиториев Debian stretch.

2.2.3. Средства разработки

Средства разработки включают в себя комплекс ПС, необходимых для создания прикладного и системного ПО:

– компиляторы и интерпретаторы для наиболее популярных языков (C++, Java, PHP, Python);

– интегрированные среды разработки (Qt Creator), отладчики (gdb);

– средства сборки deb пакетов (debhelper).

2.2.4. СЗИ

СЗИ ОС соответствуют требованиям документов:

– «Требования безопасности информации к операционным системам» (ФСТЭК России, 2016);

– «Профиль защиты операционных систем типа «А» второго класса защиты» ИТ.ОС.А2.П3 (ФСТЭК России, 2017).

СЗИ ОС реализуют следующие функции безопасности:

– идентификация и аутентификация;
– управление доступом;
– регистрация событий безопасности;
– ограничение программной среды;
– изоляция процессов;
– защита памяти;
– контроль целостности;
– обеспечение надежного функционирования;
– фильтрация сетевого потока;
– маркирование документов.

Дополнительно осуществляется совместимость формата передачи мандатных атрибутов в поле опций IP-пакетов при сетевом взаимодействии по протоколам стека

TCP/IPv4 с ОС МСВС 3.0, ОС МСВС 5.0 и Astra Linux Special Edition (соответствие ГОСТ Р 58256-2018).

Полный набор функций безопасности представлен в документе ФЛИР.90001-01 97 01 «ОС ОН «Стрелец». Задание по безопасности».

2.2.5. ПО работы с мультимедиа

ПО работы мультимедиа включает:

– средства воспроизведения видеофайлов в наиболее популярных форматах, таких как MPEG-4, AVC, H.265/HEVC и т.д. (SMPlayer);

– средства создания и воспроизведения аудиофайлов в форматах FLAC, mp3, WAV и т. п. (Audacious Media Player).

2.2.6. Средства работы с офисными документами

Средства работы с офисными документами реализованы на основе офисного пакета (LibreOffice) с открытым исходным кодом, содержащего следующие компоненты:

– текстовый редактор (Writer);

– табличный процессор (Calc);

– программу для подготовки и просмотра презентаций (Impress);

– векторный графический редактор (Draw);

– систему управления базами данных (Base);

– редактор формул (Math).

Основным поддерживаемым форматом файлов является открытый международный формат OpenDocument (ODF) с поддержкой форматов Office Open XML, DOC, XLS, PPT, CDR.

Средства работы с устройствами сканирования изображений обеспечивают поддержку режима пакетного сканирования и графический интерфейс пользователя (SANE, XSane).

ПС для работы с растровой и векторной графикой реализованы на основе ПО с открытым исходным кодом (GIMP).

2.2.7. ПС виртуализации и управления

ПС виртуализации и управления реализованы на основе ПО с открытым исходным кодом (libvirt, qemu/kvm, virt-manager) и обеспечивают функционал, необходимый для создания и управления виртуальной средой.

2.2.8. Система управления базами данных

СУБД включает в себя следующие компоненты:

– объектно-реляционную СУБД PostgreSQL;

– библиотеки поддержки доступа к СУБД PostgreSQL для средств разработки (Qt5,

php, python);

– графическую утилиту для работы с БД – pgAdmin.

Более подробно настройка СУБД приведена в 3.14.

2.2.9. Средства работы с интернетом

Средства работы с интернетом содержат:

– веб-сервер, обеспечивающий поддержку работы веб-протоколов (Apache);

– прокси-сервер (nginx), обеспечивающий кэширование и сжатие данных;

– почтовый сервер (Exim), почтовый клиент (Mozilla Thunderbird);

– веб-браузеры Mozilla Firefox и Chromium.

2.2.10. Средства, обеспечивающие вывод информации на печать

Средства, обеспечивающие вывод информации на печать, реализованы на основе ПО с открытым исходным кодом и поддерживают сетевую печать и управление заданиями печати (CUPS).

Защищенный сервер печати и средства маркировки обеспечивают в соответствии с требованиями РД:

– мандатное разграничение доступа при выводе документов на печать;

– управление мандатными атрибутами устройств печати;

– идентификацию и аутентификацию пользователей при попытках вывода документов на печать и управления атрибутами устройств печати;

– автоматическую маркировку документов при выводе на печать;

– управление параметрами автоматической маркировки документов при выводе на печать;

– регистрацию событий безопасности при выводе документов на печать.

2.2.11. Средства резервного копирования и восстановления данных

Средства резервного копирования и восстановления данных обеспечивают возможность создания и хранения резервных копий (в т. ч., сетевых).

К таким средтсвам относятся:

– утилита удаленного копирования (резервного копирования) или синхронизации файлов и каталогов rsync;

– утилита архивирования файлов и каталогов tar;

– система централизованного резервирования информационных ресурсов Bacula.

2.2.12. Справочное руководство

В состав ОС входят средства получения информации о доступных командах. Команда man (от англ. manual – руководство) предназначена для форматирования и вывода справочных страниц. Каждая страница справки является самостоятельным документом и пишется разработчиками соответствующего программного обеспечения.

Команда man выполняется как обычная команда Linux в консольном режиме, например в окне терминала (пункт меню «Системные/Qterminal»).

Чтобы получить справочное руководство по какой-либо команде (или программе, предусматривающей возможность запуска из терминала), необходимо ввести команду:

```
man <command_name>
```

где <command_name> – название команды.

Например, чтобы посмотреть справку по команде ls, нужно ввести:

```
man ls
```

Для навигации в справочной системе man можно использовать клавиши со стрелками для построчного перехода и «Page Up», «Page Down» для постраничного перехода вверх и вниз соответственно.

При просмотре больших страниц удобно пользоваться поиском, для этого следует нажать </> (слэш и строка поиска отобразятся в нижней части экрана), затем набрать строку поиска и нажать <Enter>. Для перехода к следующему подсвеченному совпадению нужно нажать <n> (Next – следующий). Для показа предыдущего совпадения нажать <shift+N> (заглавную, то есть «N»).

Для получения краткой справки по командам и горячим клавишам справочной системы следует нажать <H> (Help – помощь).

Для выхода из справочной системы нажать <Q> (Quit – выход).

Для получения детальной инструкции по использованию справочной системы следует использовать команду:

```
man man
```

Справочные страницы поделены на девять разделов. Каждый из разделов соответствует той или иной тематике в рамках установленной ОС:

1 – исполняемые программы и команды оболочки (shell);

2 – системные вызовы ядра (функции, предоставляемые ядром);

3 – библиотечные вызовы (функции, предоставляемые программными библиотеками;

4 – специальные файлы (находящиеся обычно в каталоге /dev);

5 – форматы файлов и соглашения, например о /etc/passwd;

6 – игры;

7 – разное (включает пакеты макросов и соглашения), например man(7), groff(7);

8 – команды администрирования системы (обычно запускаются только суперпользователем);

9 – процедуры ядра (нестандартный раздел).

Номер раздела в команде man указывается вторым аргументом, перед названием справочной страницы. Если номер раздела опущен, то поиск справочной страницы ведется по всем разделам по порядку:

```
man passwd #раздел 1
man 1 passwd #раздел 1
man 5 passwd #раздел 5
```

3. УСТАНОВКА И НАСТРОЙКА

3.1. Общие положения

Установка ОС осуществляется с ФЛИР.90001-01 12 01 «ОС ОН «Стрелец». Текст программы. Загрузочный модуль», который обеспечивает загрузку в режиме «Live».

После загрузки пользователь имеет возможность запускать приложения, установленные на диске, осуществлять подготовительные действия перед установкой ОС на ЖД. Например, протестировать работоспособность устройств ПЭВМ, удалить существующие разделы ЖД и прочее. А также запустить процесс установки ОС на ЖД компьютера.

Установщик ОС обеспечивает:

– установку ОС на компьютеры с архитектурой x86_64, с объемом оперативной памяти не менее 2 ГБ, объемом ЖД не менее 16 ГБ;

– установку ОС как в режиме BIOS, так и в режиме UEFI;

– установку ОС на ЖД с разбиением MBR и GPT;

– проведение первоначальных настроек компьютера, необходимых для инсталляции;

– разбиение ЖД на разделы;

– установку даты и времени;

– конфигурирование сетевых интерфейсов;

– создание учетной записи пользователя.

3.2. Описание процесса установки

Для запуска процесса установки следует вставить компакт-диск ФЛИР.90001-01 12 01 «ОС ОН «Стрелец». Текст программы. Загрузочный модуль» в устройство чтения компакт дисков и загрузиться с него в режиме «Live».

Далее следует запустить установщик системы, выбрав нажатием левой кнопки «мыши» (ЛКМ) пункт главного меню «Системные – Установщик ОС Стрелец» (рис. 1).

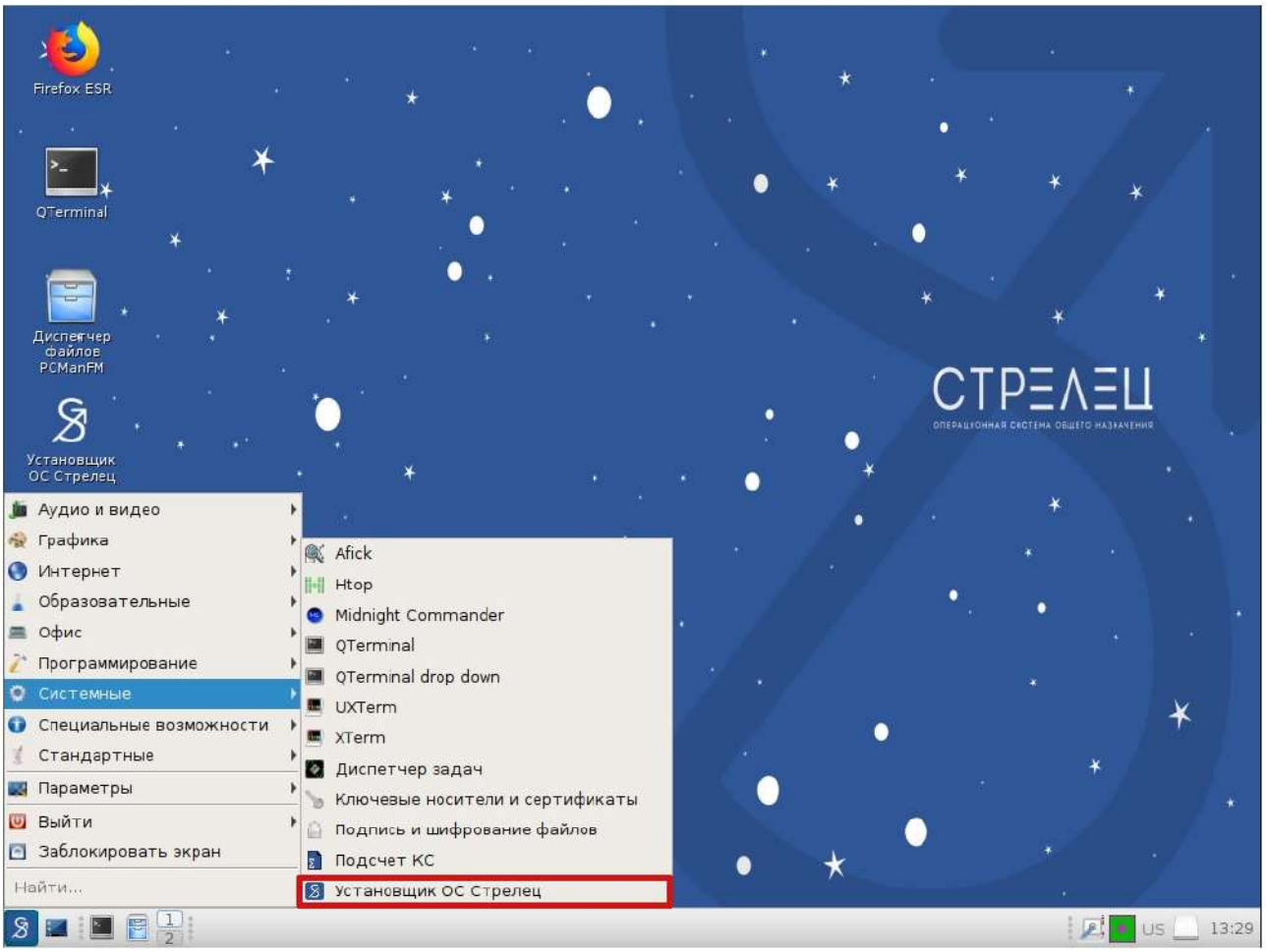


Рис. 1

После этого на экране появится окно программы (рис. 2).

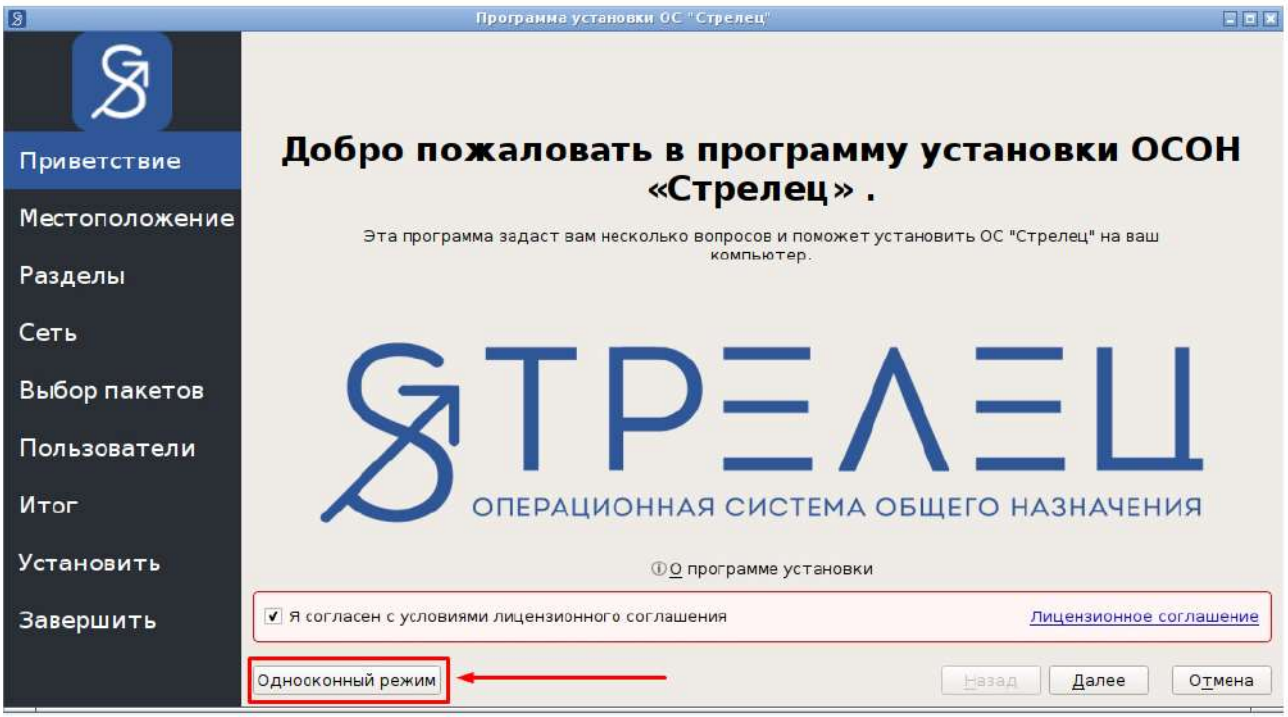


Рис. 2

Для продолжения установки необходимо согласиться с условиями лицензионного соглашения.

Далее следует выбрать режим установки – однооконный или пошаговый.

ВНИМАНИЕ! В однооконном режиме все параметры, требуемые ОС для ее установки, пользователь вводит в одном окне. В пошаговом режиме пользователь вводит паметры установки в наборе окон, последовательно появляющихся друг за другом.

3.2.1. Установка ОС в однооконном режиме

Для выбора однооконного режима установки следует нажать ЛКМ кнопку [Однооконный режим] (рис. 3).

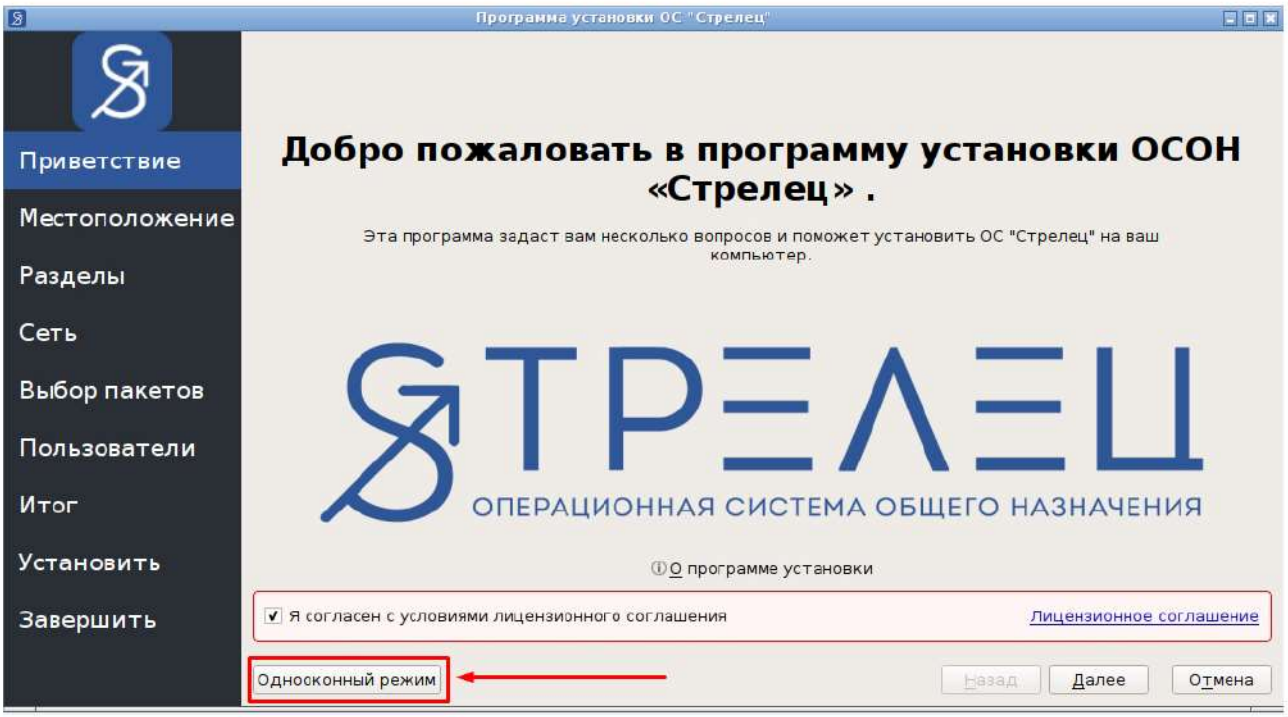


Рис. 3

После этого на экране появится окно программы (рис. 4).

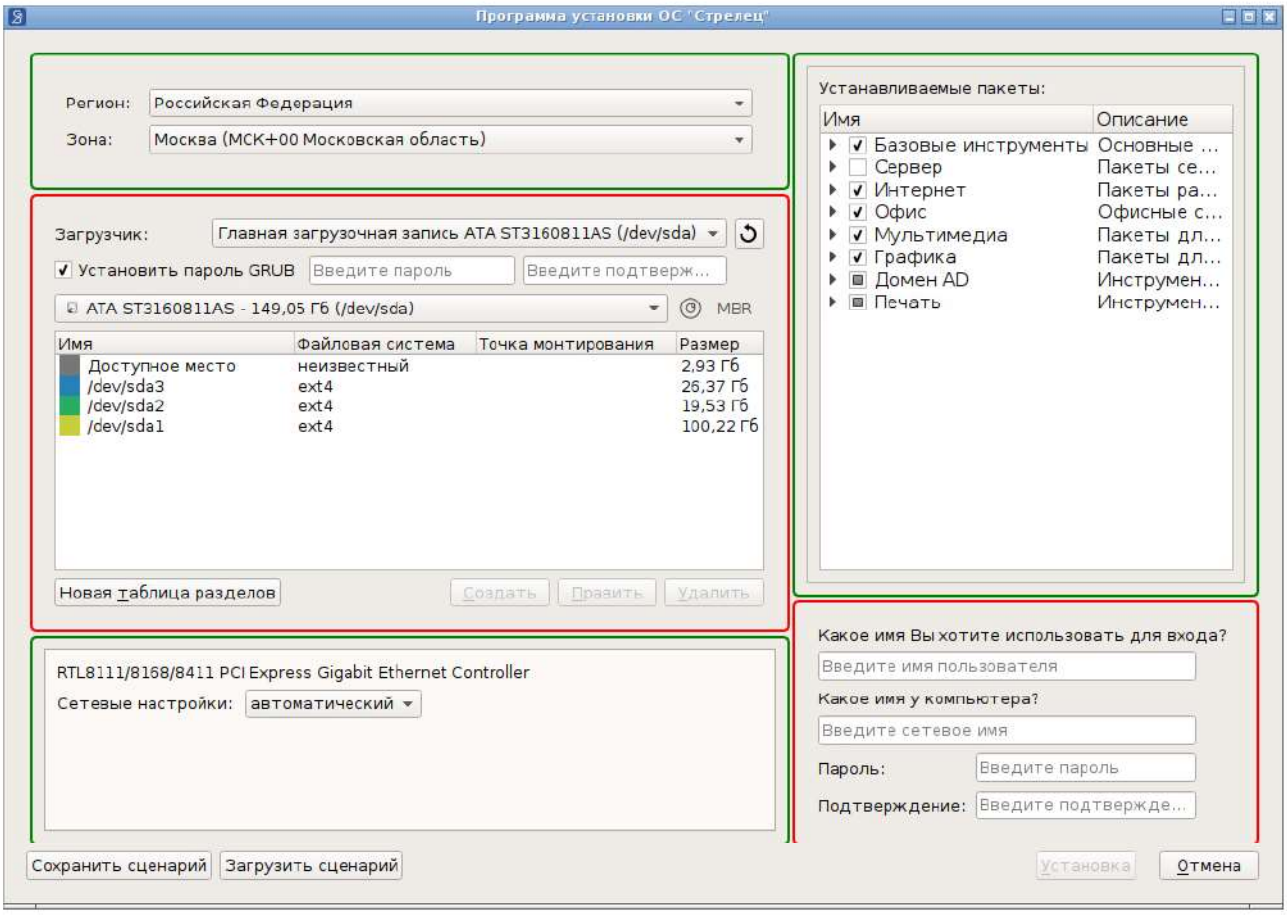


Рис. 4

Данное окно логически разделено на 5 областей ввода параметров установки (рис. 5).

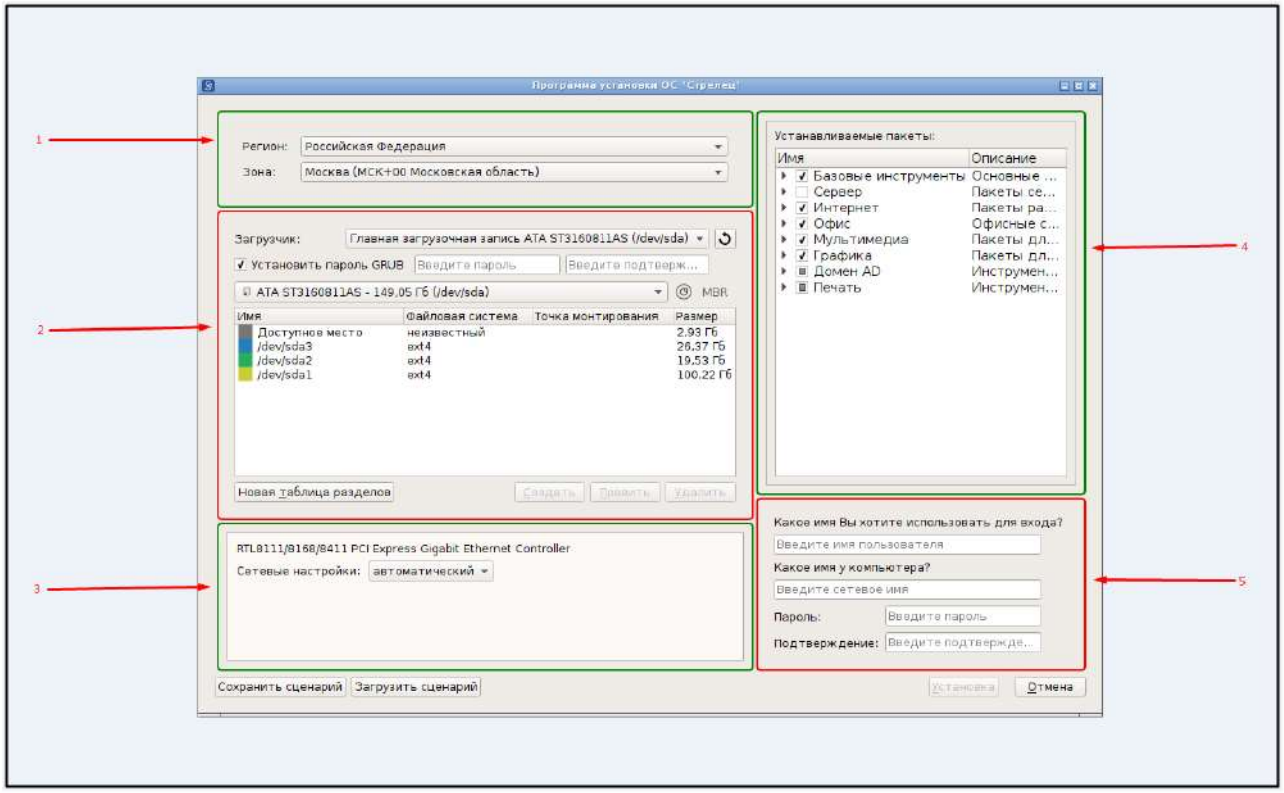


Рис. 5

Описание областей ввода параметров установки:

1) область ввода параметров местоположения, времени и даты;

2) область ввода параметров загрузки и разметки диска;

3) область ввода параметров сети;

4) область ввода параметров устанавливаемых пакетов ПО;

5) область ввода параметров идентификации и аутентификации пользователя.

ВНИМАНИЕ! В случае, если параметры установки введены полностью и корректно, область ввода подсвечивается рамкой зеленого цвета. В противном случае – область ввода подсвечивается рамкой красного цвета.

После успешного заполнения всех параметров становится доступна кнопка [Установка] (рис. 6).

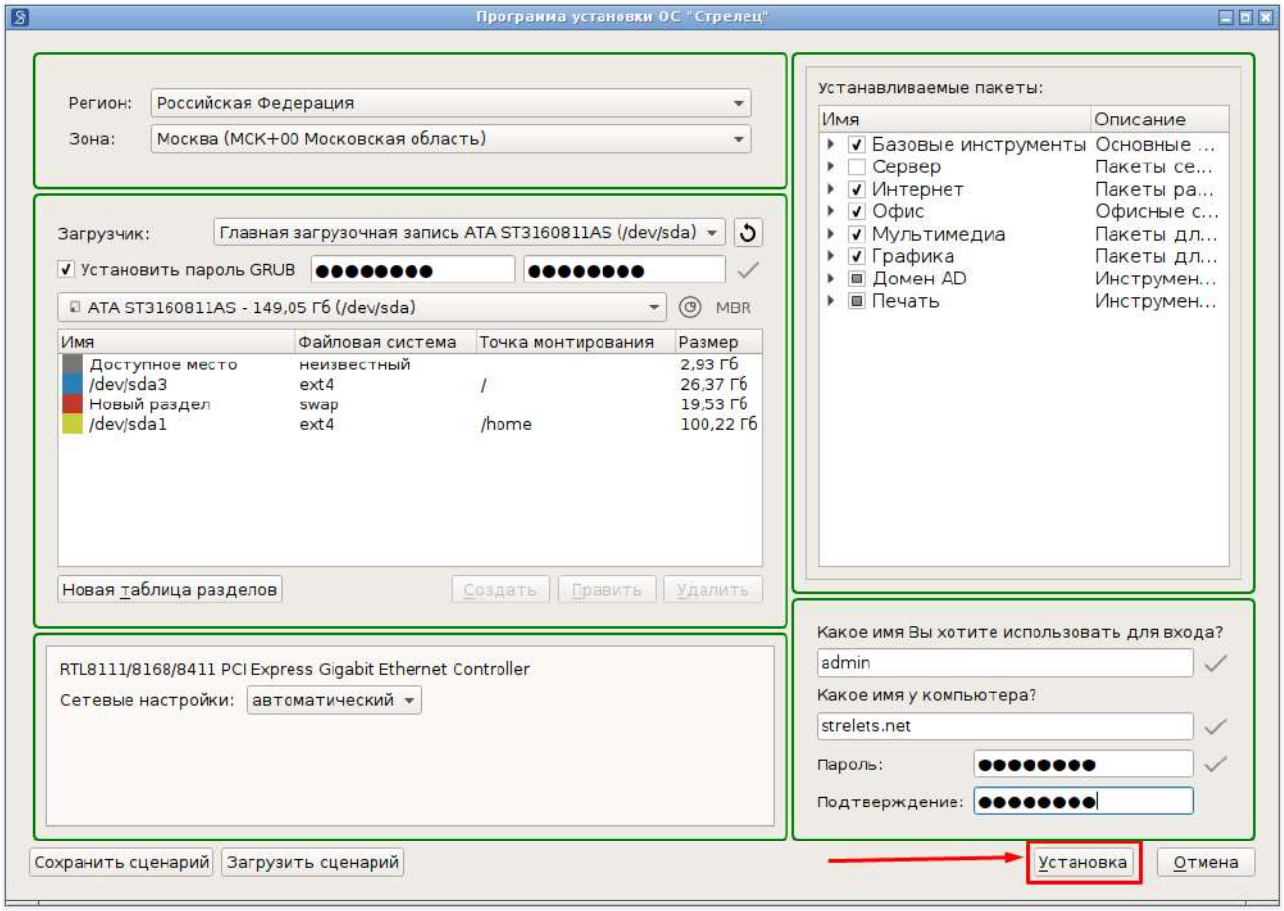


Рис. 6

Далее следует нажать ЛКМ кнопку [Установка].

После этого на экране появится окно программы (рис. 7).

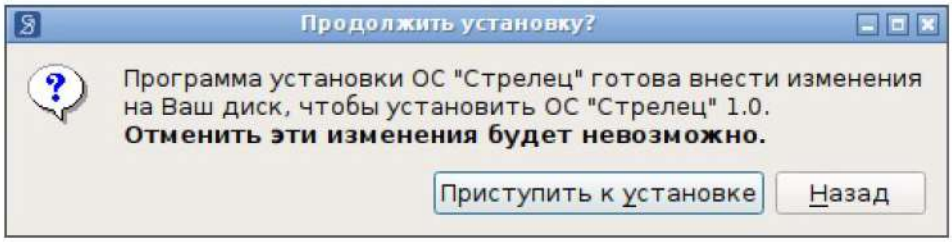


Рис. 7

В данном окне следует нажать ЛКМ кнопку [Приступить к установке].

После этого начинается процесс записи файлов ОС на жесткий диск компьютера. Данный процесс происходит в автоматическом режиме и не требует дальнейшего вмешательства.

После завершения процесса установки на экране появится окно программы (рис. 8).

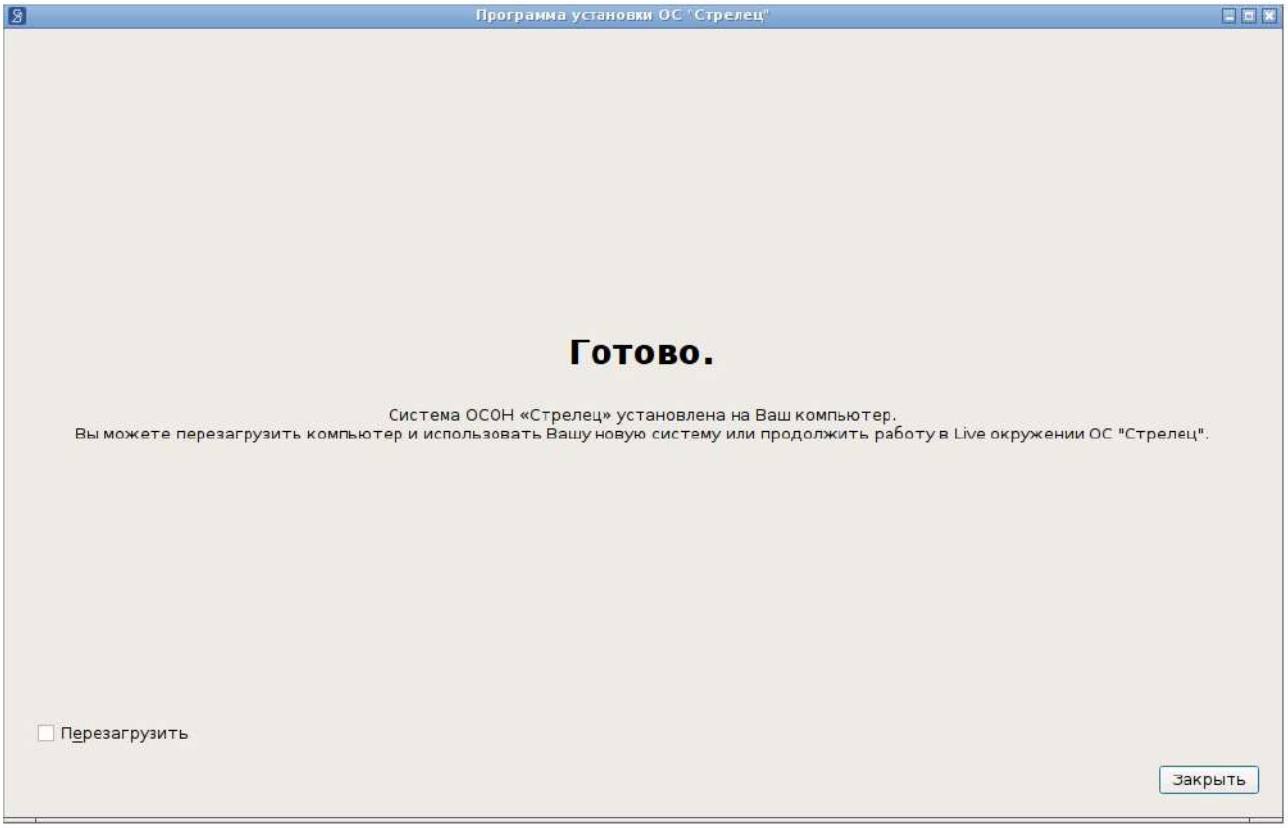


Рис. 8

В данном окне следует ЛКМ установить флажок «Перезагрузить», после чего нажать ЛКМ кнопку [Закрыть].

После перезагрузки компьютера следует извлечь диск из устройства чтения компакт-дисков.

3.2.2. Установка ОС в пошаговом режиме

Для выбора пошагового режима установки следует нажать ЛКМ кнопку [Далее] (рис. 9).

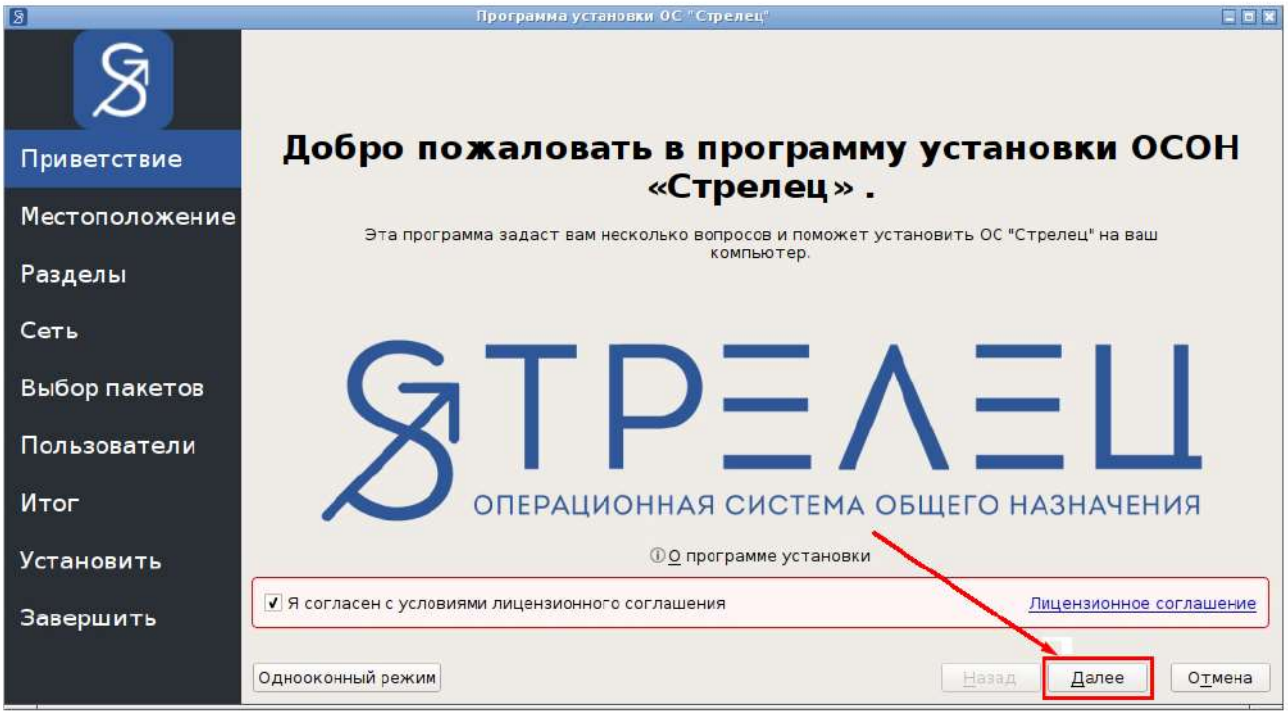


Рис. 9

После этого на экране появится окно программы (рис. 10).

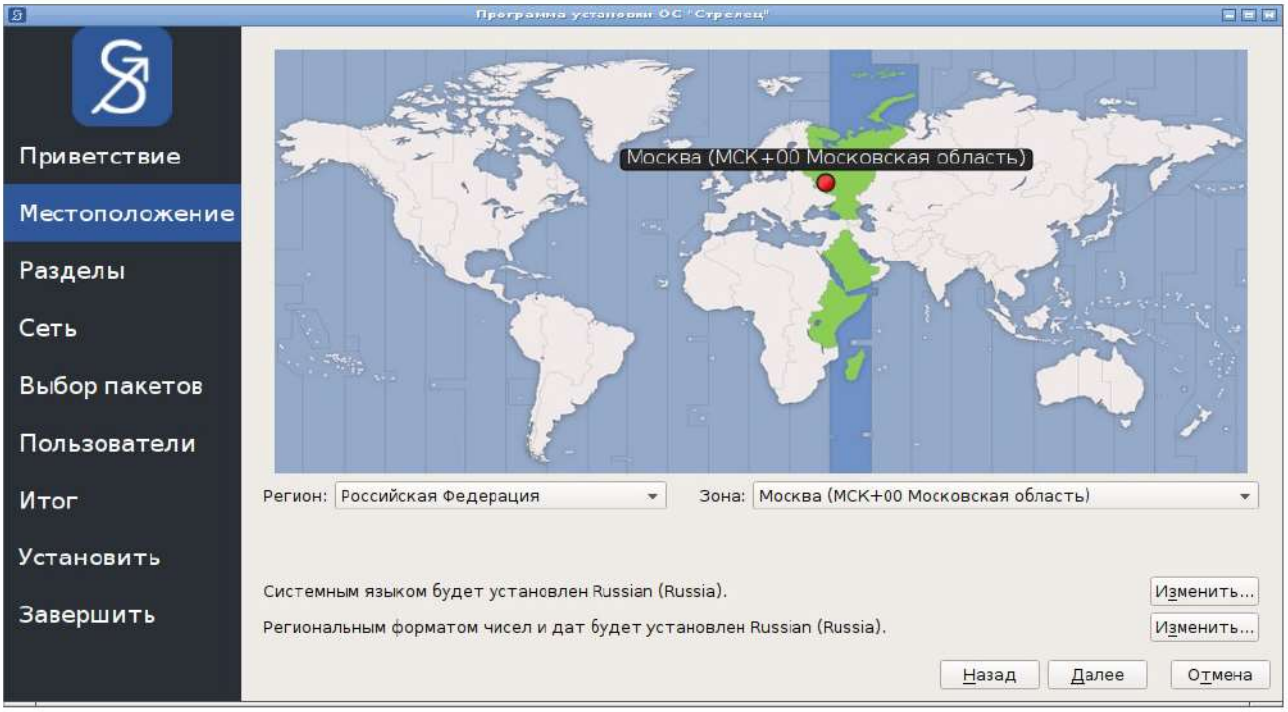


Рис. 10

В данном окне следует ввести параметры местоположения, времени и даты, затем нажать ЛКМ кнопку [Далее].

После этого на экране появится окно программы (рис. 11).

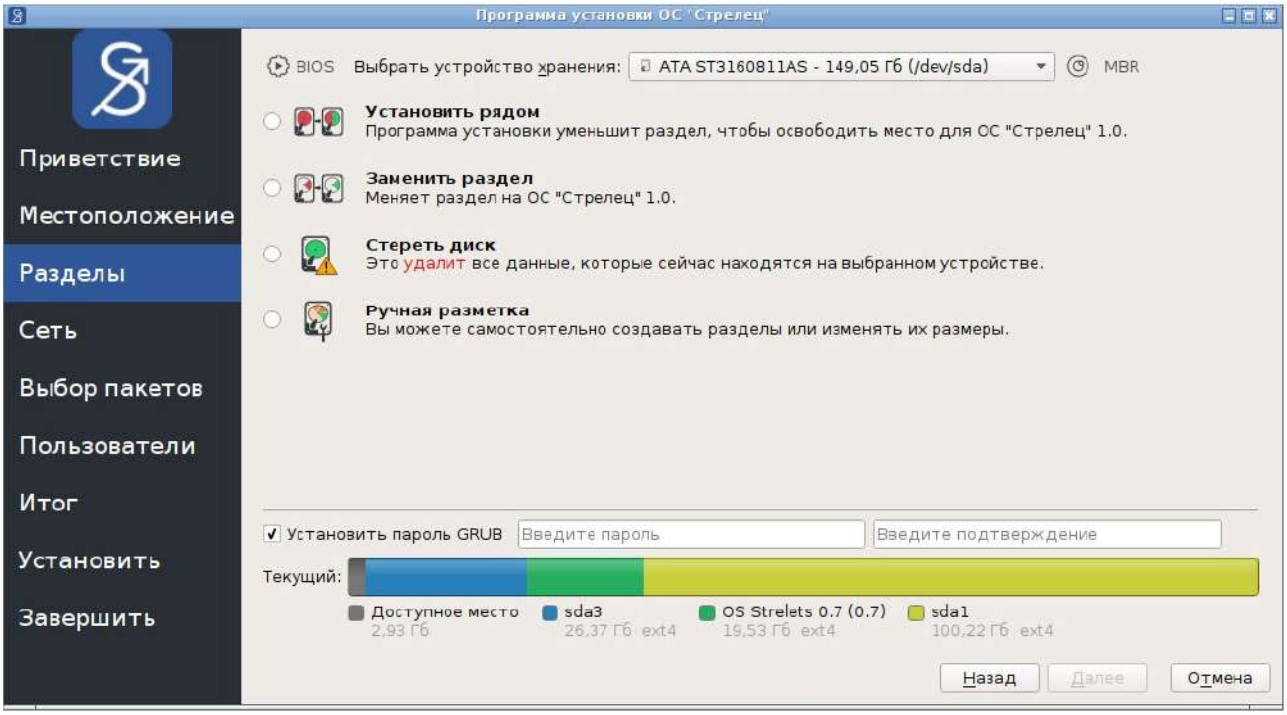


Рис. 11

В данном окне следует выбрать требуемый способ разметки диска, ввести пароль загрузчика, затем нажать ЛКМ кнопку [Далее] (рис. 12).

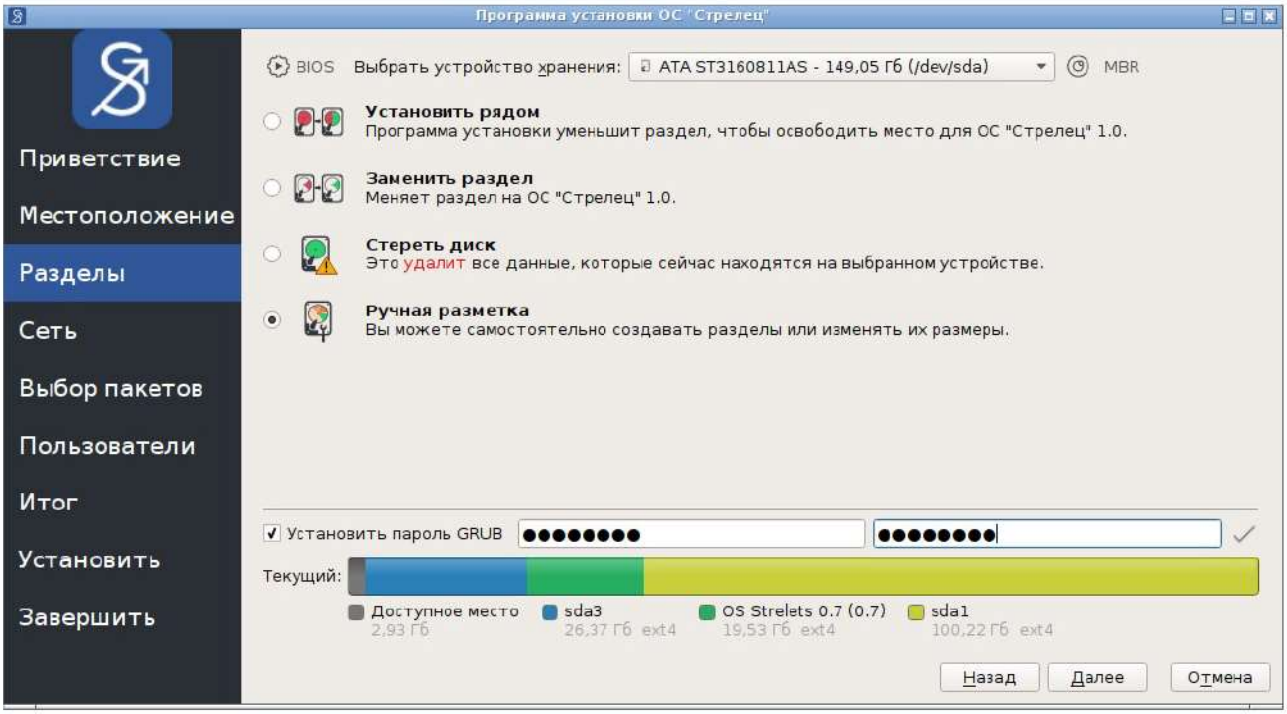


Рис. 12

После этого на экране появится окно программы (рис. 13).

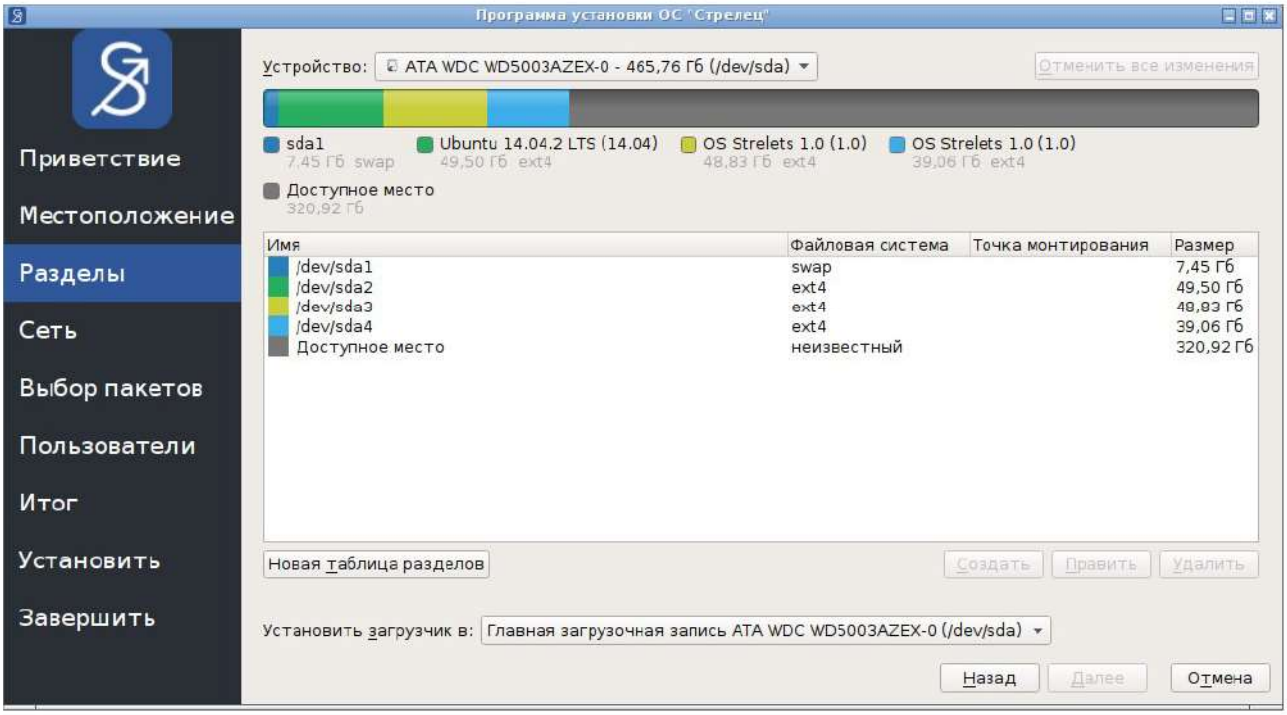


Рис. 13

В данном окне следует ввести параметры загрузки и разметки диска, затем нажать ЛКМ кнопку [Далее] (рис. 14).

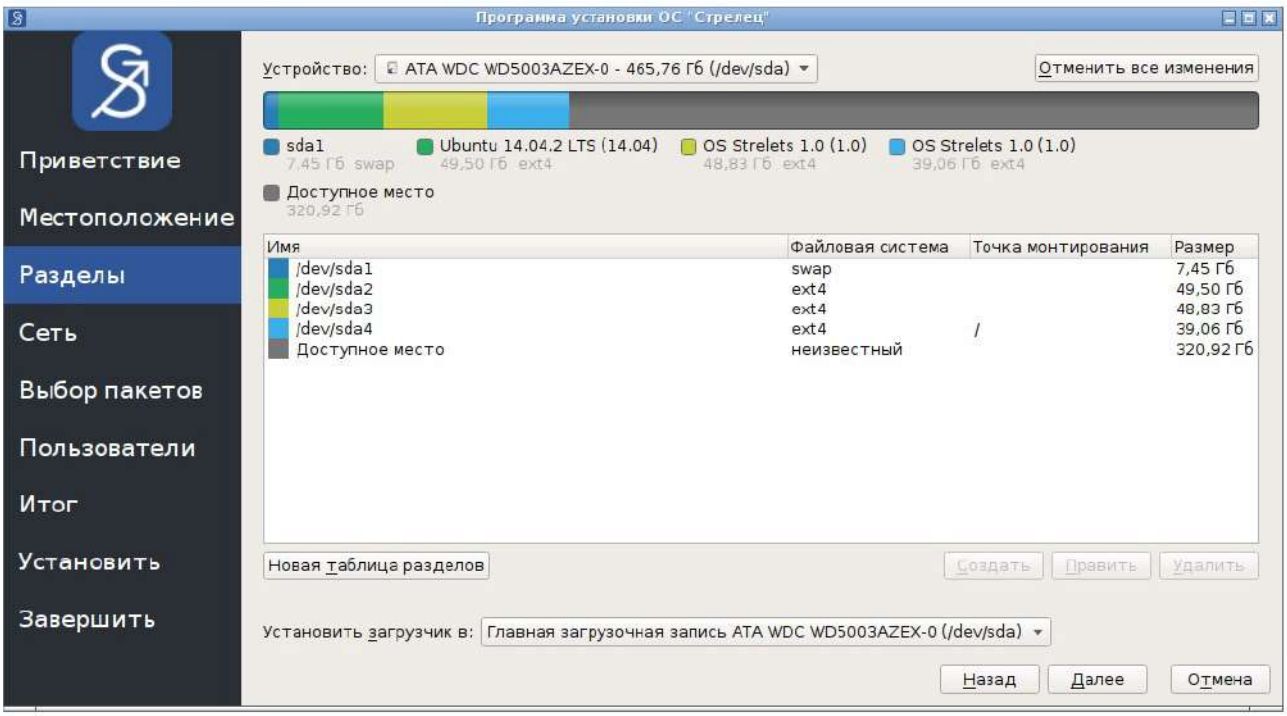


Рис. 14

После этого на экране появится окно программы (рис. 15).

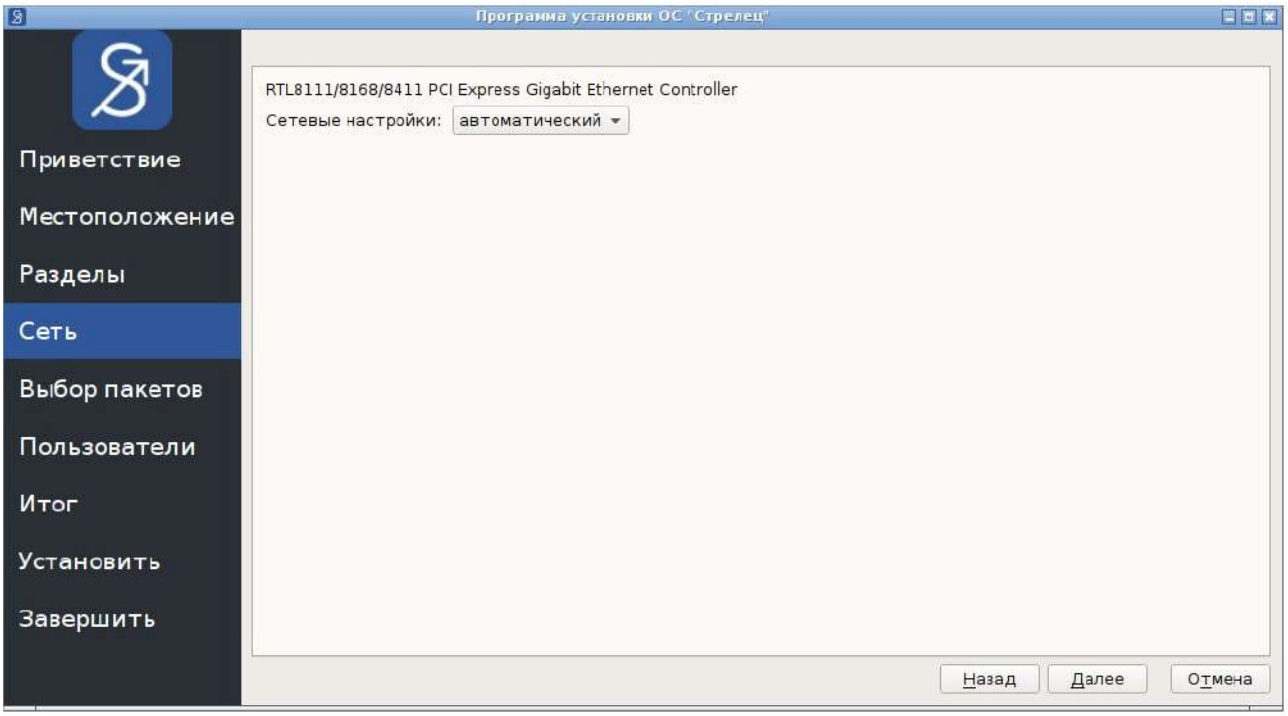


Рис. 15

В данном окне следует ввести параметры сети, затем нажать ЛКМ кнопку [Далее].

После этого на экране появится окно программы (рис. 16).

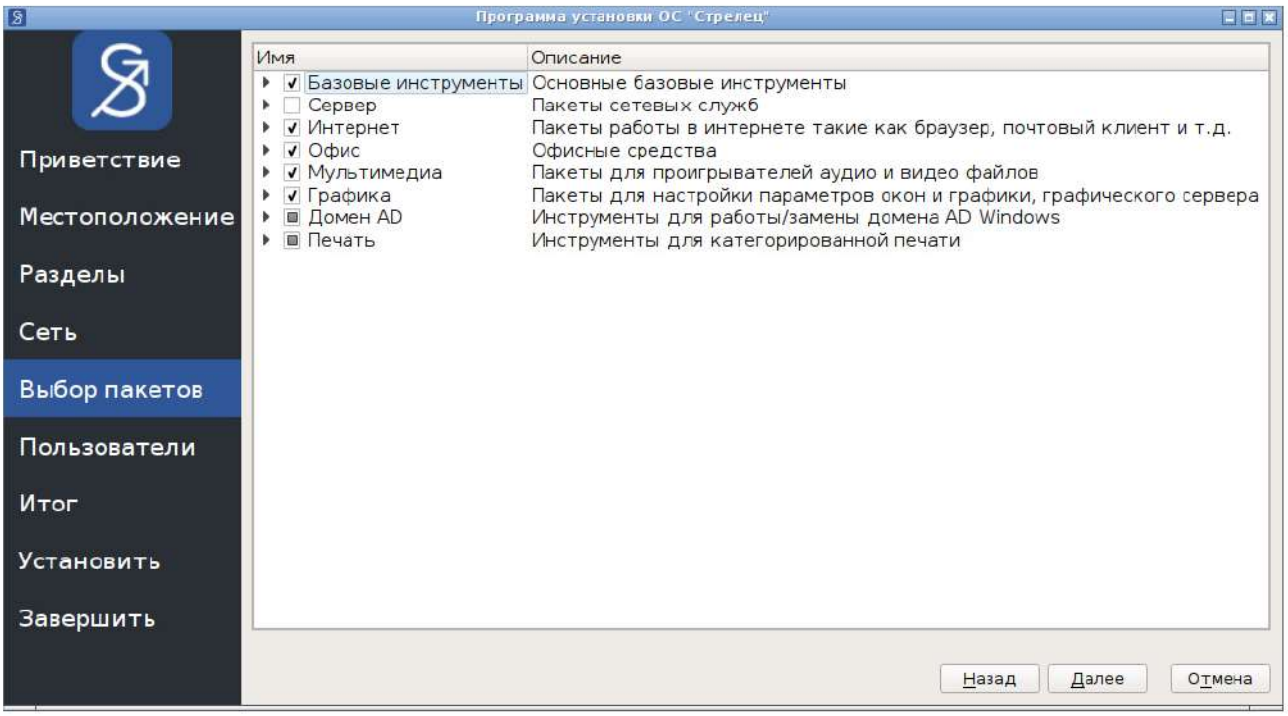


Рис. 16

В данном окне следует ввести параметры устанавливаемых пакетов ПО, затем нажать ЛКМ кнопку [Далее].

После этого на экране появится окно программы (рис. 17).

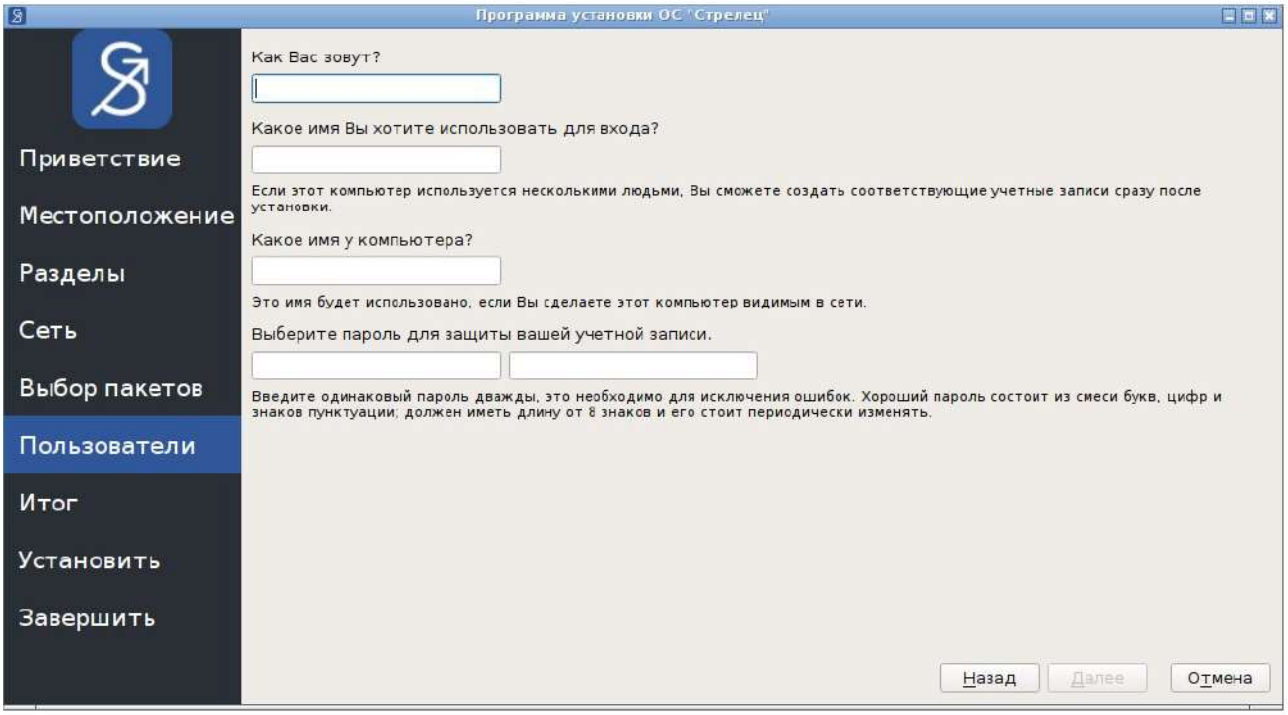


Рис. 17

В данном окне следует ввести параметры идентификации и аутентификации пользователя, затем нажать ЛКМ кнопку [Далее] (рис. 18).

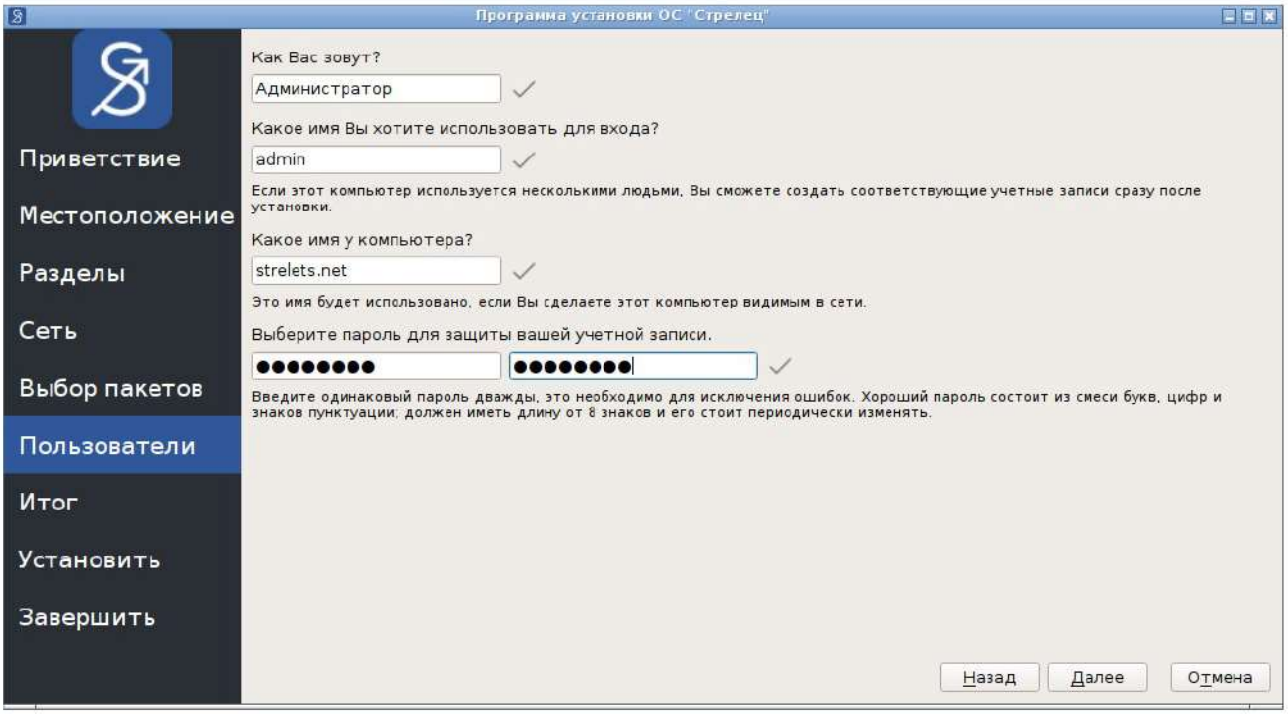


Рис. 18

После этого на экране появится окно программы (рис. 19).

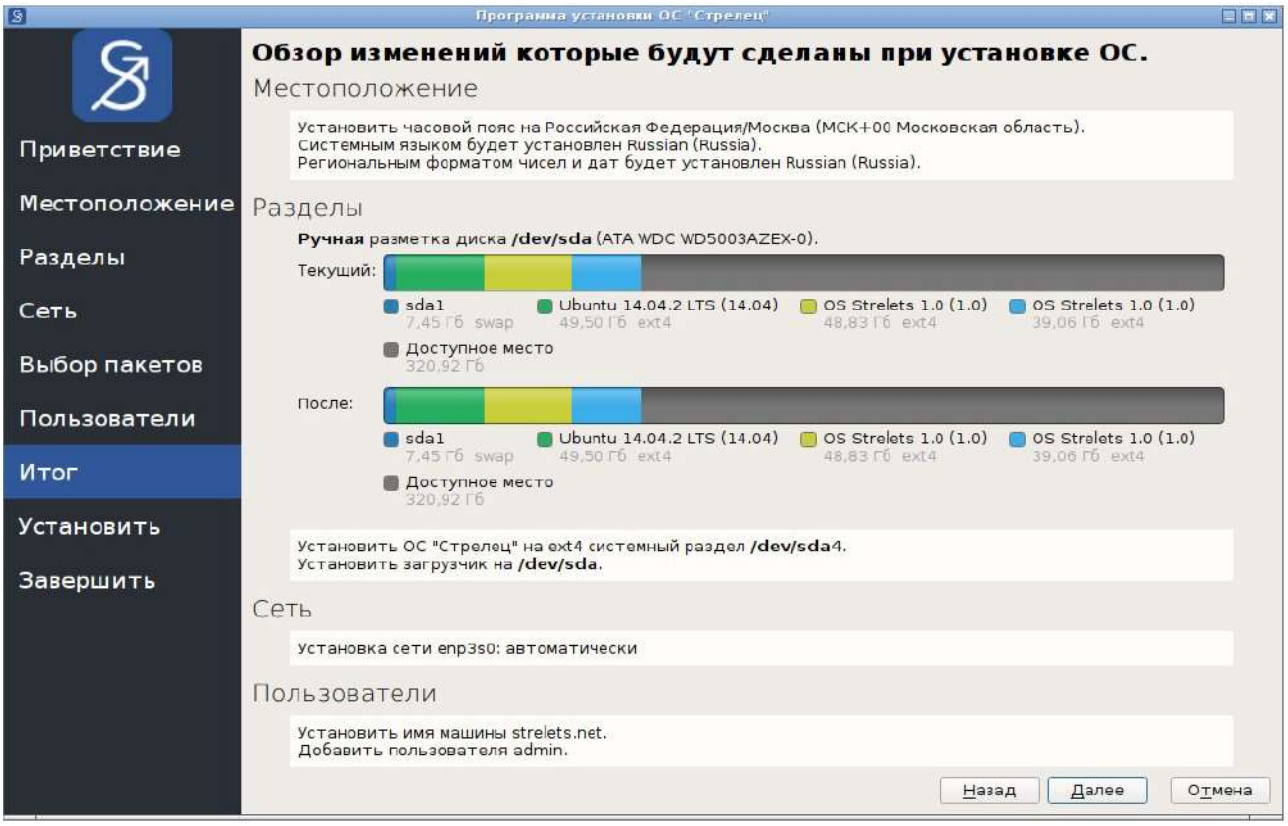


Рис. 19

В данном окне отображаются параметры установки ОС, выбранные пользователем. После просмотра параметров следует нажать ЛКМ кнопку [Далее].

После этого на экране появится окно программы (рис. 20).

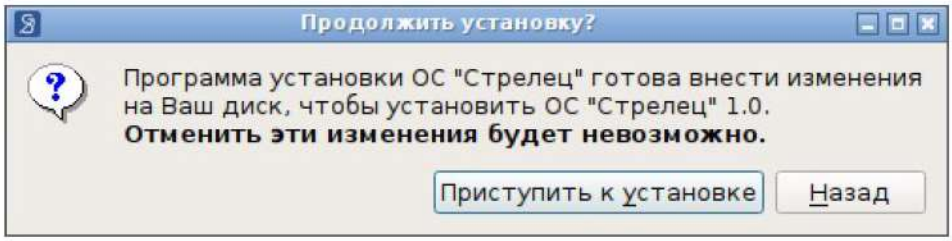


Рис. 20

В данном окне следует нажать ЛКМ кнопку [Приступить к установке].

После этого начинается процесс записи файлов ОС на жесткий диск компьютера. Данный процесс происходит в автоматическом режиме и не требует дальнейшего вмешательства.

После завершения процесса установки на экране появится окно программы (рис. 21).

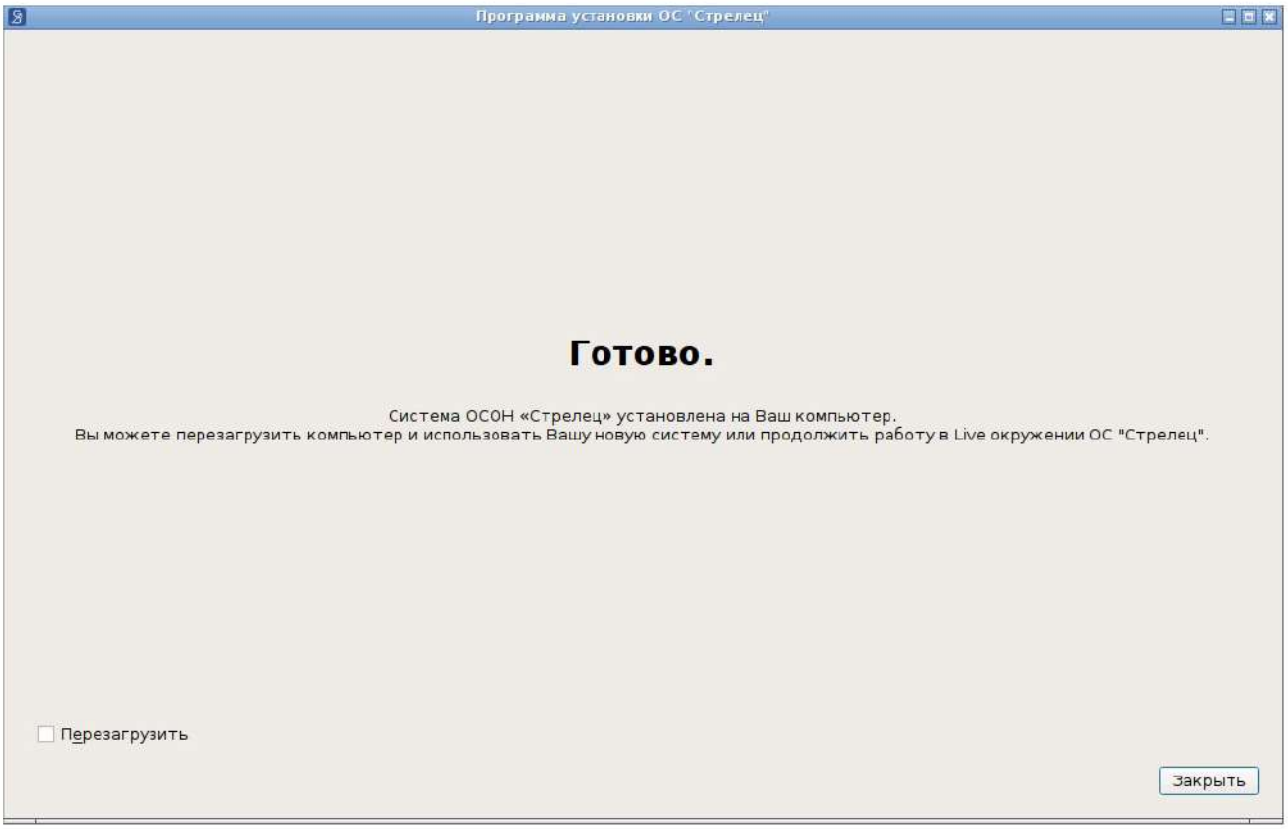


Рис. 21

В данном окне следует ЛКМ установить флажок «Перезагрузить», после чего нажать ЛКМ кнопку [Закрыть].

После перезагрузки компьютера следует извлечь диск из устройства чтения компакт-дисков.

3.2.3. Установка ОС без участия пользователя (сценарии установки)

В однооконном режиме есть возможность сохранять/загружать настройки устанавливаемой ОС ОН «Стрелец» в файл сценария установки (см. рис 6). Для этого необходимо нажать кнопку [Сохранить сценарий] (при сохранении) или кнопку [Загрузить сценарий] (при загрузке) и в появившемся диалоговом окне задать имя сценария установки.

Файл сценария установки можно использовать в качестве аргумента к установщику ОС ОН «Стрелец», при этом установка будет произведена без участия пользователя. Для выполнения такого вида установки необходимо выполнить команду в командной строке:

```
$ strelets-installer -s [имя_сценария].iscn
```

где [имя_сценария].iscn – имя файла сценария, созданного в графическом однооконном режиме установки ОС ОН «Стрелец».

3.3. Администрирование

Административное управление в ОС отделено от общего доступа пользователей.

Большинство операций по настройке и администрированию ОС требуют привилегий суперпользователя root, например:

– управление процессами и сервисами;

– управление устройствами;

– монтирование и размонтирование файловой системы (ФС);

– задание имени системы и настройка сетевых интерфейсов;

– изменение программно-аппаратных системных настроек (например, установка системных часов);

– изменение корневого каталога процесса командой `chroot`;

– управление доступом (конфигурирование сценариев входа, изменение принадлежности файлов).

В ОС применяется принцип минимальных полномочий, при котором пользователям делегируется ограниченный набор привилегий и ресурсов для выполнения конкретных действий по администрированию.

Настройка запрета интерактивного входа в систему суперпользователя (root) осуществляется посредством настройки механизма sudo.

ВНИМАНИЕ! ДЕЙСТВИЯ ПО АДМИНИСТРИРОВАНИЮ ОС НЕОБХОДИМО ВЫПОЛНЯТЬ В МАНДАТНОМ КОНТЕКСТЕ БЕЗОПАСНОСТИ СУБЪЕКТА С НУЛЕВЫМ УРОВНЕМ И ПУСТЫМ НАБОРОМ КАТЕГОРИЙ. К ПАРОЛЮ ПОЛЬЗОВАТЕЛЕЙ, ОБЛАДАЮЩИХ АДМИНИСТРАТИВНЫМ ДОСТУПОМ, ПРЕДЪЯВЛЯЮТСЯ ПОВЫШЕННЫЕ ТРЕБОВАНИЯ К КАЧЕСТВУ И НАДЕЖНОСТИ.

3.3.1. Получение полномочий администрирования – sudo

Для получения полномочий администрирования применяется системная команда `sudo`, позволяющая делегировать те или иные привилегированные ресурсы пользователям с ведением протокола работы. Целью является предоставление пользователям ограниченного числа прав, достаточного для решения поставленных задач.

Команда `sudo` предоставляет возможность пользователям выполнять команды от имени суперпользователя либо других пользователей. Правила, используемые `sudo` для

принятия решения о предоставлении доступа, находятся в файле /etc/sudoers.

ВНИМАНИЕ! ДЛЯ РЕДАКТИРОВАНИЯ ФАЙЛА ИСПОЛЬЗОВАТЬ СПЕЦИАЛЬНЫЙ РЕДАКТОР VISUDO, ЗАПУСКАЕМЫЙ ИЗ КОМАНДНОЙ СТРОКИ БЕЗ ПАРАМЕТРОВ, В ТОМ ЧИСЛЕ БЕЗ УКАЗАНИЯ ПУТИ К ФАЙЛУ.

В качестве аргументов `sudo` принимает командную строку, которую следует выполнить с правами другого пользователя. В случае допустимости запуска команды, указанной в командной строке, текущему пользователю, ему предлагается ввести свой пароль. Таким образом, для каждого пользователя установлен набор команд, которые он может исполнять от имени суперпользователя. При этом ведется регистрация с помощью системного журнала syslog выполненных команд, вызвавших их лиц, каталогов, из которых вызывались команды, и времени их вызова.

Описание команд `sudo`, `visudo` и формата файла /etc/sudoers приведено в соответствующих страницах man.

3.3.2. Механизмы разделения полномочий

В ОС применяется несколько механизмов разделения полномочий между системными администраторами и пользователями ОС:

– механизм привилегий;

– механизм повышения полномочий на время выполнения (setuid, setgid);

Механизм привилегий используется для регламентирования выполнения привилегированных операций ядра ОС процессами. При этом привилегии наследуются процессами от своих «родителей» и не могут быть переданы сторонним процессам. Для использования унаследованных привилегий процесс явным образом должен их задействовать.

Механизм повышения полномочий на время выполнения (атрибуты setuid и setgid) позволяет пользователю, запускающему этот файл на исполнение, получить повышение прав до пользователя-владельца файла (обычно суперпользователя) в рамках запущенного процесса. После получения повышенных прав приложение может выполнять задачи, выполнение которых обычному пользователю недоступно. Установленный атрибут setgid дает повышение прав группы.

3.3.3. Управление программными пакетами

В ходе настройки параметров установки задать набор пакетов, предлагаемых для установки ОС.

Предлагается следующее разбиение на функциональные роли устанавливаемой

ОС:

– сервер;

– рабочая станция;

– специализированная рабочая станция;

– планшет.

Каждый вариант установки может быть использован с включенными функциями СЗИ и с выключенными функциями СЗИ (режим разработчика/отладки).

3.3.4. Системные команды

Основные системные команды, используемые для администрирования и управления функционированием, приведены в таблице 1.

Таблица 1

Команда	Описание
addgroup	Создание учетной записи группы
adduser	Создание учетной записи пользователя
ar	Создание и работа с библиотечными архивами
awk	Язык обработки строковых шаблонов
bash	Командный интерпретатор bash
bc	Строковый калькулятор
bzip2	Архиватор файлов BZIP
chacl	Изменением списка контроля доступа файла
chage	Управление временем жизни пароля пользователя
chattr	Изменение атрибутов файла
chfn	Изменение описательной части учетной записи пользователя
chgrp	Изменение группы файла
chmod	Изменение режима доступа к файлу (прав доступа)
chown	Изменение владельца файла
chsh	Выбор командного интерпретатора (по умолчанию — для учетной записи пользователя)
cmp	Побайтовое сравнение файлов
cp	Копирование файлов
cut	Разбивка файла на секции, задаваемые контекстными разделителями
dash	Командный интерпретатор Dash
date	Вывод и изменение системных даты и времени
delgroup	Удаление учетной записи группы
deluser	Удаление учетной записи пользователя
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 регулярных выражений
fdisk	Утилита управлением таблицей разделов дисков
fgrep	Поиск в файлах содержимого согласно фиксированным шаблонам

Продолжение таблицы 1

Команда	Описание
file	Определение типа файла
find	Поиск файла по различным признакам в ФС
fsck	Проверка и исправление ошибок ФС
fuser	Определение процессов, использующих файл или сокет
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 и состава групп
gunzip	Распаковка файла ZIP
gzip	Упаковка файла ZIP
halt	Полный останов ОС
hostname	Вывод и задание имени хоста
id	Вывод действительных и эффективных идентификаторов пользователя и группы
install	Копирование файла с установкой атрибутов
ipcrm	Удаление IPC ресурса
ipcs	Вывод характеристик IPC ресурса
kill	Отправка сигнала процессу (по умолчанию — SIGTERM)
killall	Удаление процессов по имени
ls	Вывод содержимого каталога
m4	Макропроцессор
mkfs	Создание ФС
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	Запуск процесса с заданным уровнем приоритета
nohup	Запуск процесса с игнорированием прерываний по сигналу HUP
od	Вывод содержимого файла в восьмеричном и других видах

Окончание таблицы 1

Команда	Описание
passwd	Смена пароля учетной записи

patch	Применение файла описания изменений к оригинальному файлу
pidof	Вывод идентификатора процесса по его имени
poweroff	Выключение системы
ps	Вывод информации о процессах
pstree	Вывод информации о дереве процессов
pwd	Вывод текущего рабочего каталога
reboot	Перезагрузка системы
renice	Изменение уровня приоритета процесса
scp	Копирование по сети с использованием ssh
sed	Строковый редактор
sendmail	Транспорт системы электронных сообщений
sh	Командный интерпретатор Sh
shutdown	Команда останова системы
stat	Вывод статусной информации о файле
su	Изменение идентификатора пользователя для имперсонации или получения полномочий суперпользователя
sync	Сброс системных буферов на носители
tar	Файловый архиватор TAR
time	Замер времени работы процесса
top	Вывод списка процессов
touch	Изменение меток времени файла
umount	Размонтирование ФС
uname	Вывод системной информации
useradd	Создание учетной записи пользователя
userdel	Удаление учетной записи пользователя
usermod	Изменение учетной записи пользователя
users	Вывод списка имен активных пользователей системы
w	Вывод списка активных пользователей системы и их текущих процессов
wc	Подсчет слов и строк в файле
which	Определение пути к команде
who	Вывод списка активных пользователей системы

3.3.5. Управление функционированием

Цикл функционирования ОС включает в себя:

– процесс инициализации при включении оборудования;

– штатный режим фоновой работы системных сервисов и служб;

– режим интерактивной работы пользователя;

– процесс сохранения данных и подготовка к выключению оборудования.

Единицей функционирования системы является процесс ОС, характеризующийся следующими свойствами:

– исполняемым модулем, содержащим инструкции к исполнению;

– собственным изолированным адресным пространством, предоставленным ядром ОС;

– набором открытых объектов ОС (файлов, сокетов и других объектов межпроцесного взаимодействия);

– набором привилегий ОС;

– информацией об учетной записи пользователя, от имени которого функционирует процесс.

ОС является многозадачной и представляет собой систему параллельных взаимодействующих процессов с древовидной структурой. При этом каждый процесс может порождать дочерние, становясь для них родительским.

В процессе старта системы существует один системный процесс «init», запускающий другие системные процессы, в т. ч. системные сервисы и сеансы работы пользователей, являясь, таким образом, общим прародителем всех процессов.

Примечания:

1. Ядро ОС с помощью механизмов защиты страниц памяти и трансляции виртуального адреса в физический гарантирует изоляцию собственных адресных пространств процессов. При этом процесс не может несанкционированным образом получить доступ к адресному пространству других процессов, в том числе и ядра ОС.

2. Управление функционированием процессов, распределением времени между ними и организацией межпроцесного взаимодействия осуществляется ядром ОС.

3.3.6. Управление процессами

В ОС различают системные процессы, порожденные в ходе инициализации и запуска системы или ими, и пользовательские процессы, запущенные в сессии пользователя. Управление системными процессами может быть выполнено только при наличии привилегий суперпользователя, пользовательские процессы могут управляться владельцем (т. е. создавшим их пользователем).

3.3.7. Получение информации и просмотр списка процессов

Для просмотра списка процессов ОС применяются следующие системные команды:

– ps – отображение снимка текущих процессов с информацией о них (в случае запуска пользователем — выводятся только процессы текущей сессии);

– pstree – отображение снимка дерева текущих процессов с информацией о них;

– top – отображение процессов ОС в реальном времени с сортировкой по тому или иному параметру, в том числе использованию процессора и других ресурсов ОС.

Перечисленные команды обладают широким набором опций и параметров для управления составом и видом отображаемой информации.

Подробное описание команд приводится в соответствующем руководстве man ps, man pstree и man top.

3.3.8. Сигналы

Управление процессами выполняется с помощью отправки им сигналов, основные из которых приведены в таблице 2. За обработку сигналов отвечает непосредственно программная реализация процесса за исключением сигналов SIGKILL и SIGSTOP, которые не могут быть пойманы, блокированы или проигнорированы.

ВНИМАНИЕ! ПО УМОЛЧАНИЮ НЕ ОБРАБОТАННЫЕ ПРОЦЕССОМ СИГНАЛЫ ПРИВОДЯТ К ЕГО ПРИНУДИТЕЛЬНОМУ ЗАВЕРШЕНИЮ.

Таблица 2

Наименование	Значение	Описание
SIGHUP	1	Потеря соединения с управляющим терминалом
SIGINT	2	Прерывание с клавиатуры
SIGQUIT	3	Прекратить работу с клавиатурой
SIGILL	4	Некорректная инструкция от процессора
SIGABRT	6	Сигнал о прекращении работы, выданный процессом
SIGFPE	8	Неправильная операция с плавающей запятой
SIGKILL	9	Принудительное завершение работы процесса
SIGSEGV	11	Некорректное обращение к памяти
SIGPIPE	13	Запись в канале, не имеющем считывающих процессов
SIGALRM	14	Сигнал таймера
SIGTERM	15	Сигнал завершения работы процесса
SIGCHLD	20,17,18	Дочерний процесс остановлен или прерван
SIGCONT	19,18,25	Продолжить в случае остановки
SIGSTOP	17,19,23	Процесс остановлен

Для завершения процессов им посылается набор сигналов SIGHUP, SIGQUIT и SIGTERM. Отправка процессу сигнала SIGKILL вызывает его принудительное завершение. Сигналы процессам отправляются непосредственно ядром ОС в процессе ее жизненного цикла функционирования, например при завершении работы.

Для отправки сигналов процессам (например, для их завершения) по инициативе пользователя применяются системные команды `kill` и `killall`.

Команда `kill` отправляет заданный сигнал указанному процессу, а `killall` либо всем процессам с таким именем, либо при указании пути — всем процессам, выполняющим указанный файл. Сигналы могут задаваться именем или значением с дефисом перед ними:

```
$ kill -KILL 6348
$ kill -9 6348
$ sudo killall -utest
```

По умолчанию отправляется сигнал SIGTERM.

Примечание. Без привилегий суперпользователя сигналы могут быть отправлены пользователем только процессам, владельцем которых он является.

Запуск процессов, продолжающих работать без связи с управляющим терминалом, возможен с помощью команды `nohup`. В этом случае вывод результатов работы программы может быть перенаправлен в указанный файл. При этом порожденный

процесс игнорирует посылаемый ему сигнал SIGHUP.

3.3.9. Управление уровнями приоритета

Каждый процесс имеет свое значение приоритета, которое используется для разделения процессорного времени. Диапазон возможных уровней простирается от минус 20 (наивысший приоритет) до плюс 19 (низший приоритет). Большее абсолютное значение уровня означает большую склонность процесса отдавать кванты процессорного времени другим процессам. Обычные пользовательские процессы по умолчанию запускаются с уровнем 0.

Уровень приоритета может быть отображен или задан при запуске процесса с помощью системной команды `nice`.

Для изменения уровня приоритета выполняющегося процесса используется системная команда `renice`.

Примечание. Повышение приоритета (уменьшение его уровня) требует привилегий суперпользователя.

3.3.10. Управление сервисами (systemd)

В ОС для управления сервисами применяется системный менеджер systemd, пришедший на замену используемого ранее механизма System V init. При этом применяется механизм ограничения привилегий и возможностей сервисов, что повышает защищенность всей системы в целом. Менеджер systemd выполняет запуск сервисов по возможности параллельно на основе зависимостей между сервисами, что уменьшает время загрузки системы.

Системный менеджер systemd оперирует специально оформленными файлами конфигурации – юнитами (unit). Каждый юнит отвечает за отдельно взятую службу, точку монтирования, подключаемое устройство, файл подкачки, виртуальную машину и т.п. Существуют специальные типы юнитов, которые не несут функциональной нагрузки, но позволяют задействовать дополнительные возможности systemd. К ним относятся юниты типа target, slice, automount и др. Поддерживает следующие типы юнитов:

– .target – позволяет группировать юниты, воплощая концепцию уровней запуска (runlevel);

– .service – отвечает за запуск сервисов (служб), также поддерживает вызов интерпретаторов для исполнения пользовательских скриптов;

– .mount – отвечает за монтирование файловых систем;

– .automount – позволяет отложить монтирование файловых систем до фактического обращения к точке монтирования;

– .swap – отвечает за подключение файла или устройства подкачки;

– .timer – позволяет запускать юниты по расписанию;

– .socket – предоставляет службам поддержку механизма сокет-активации;

– .slice – отвечает за создание контейнера cgroups;

– .device – позволяет реагировать на подключение устройств;

– .path – управляет иерархией ФС.

По сравнению с System V init systemd имеет следующие преимущества:

– контроль состояния службы, реакция на изменения состояния;

– сокет-активные и шина-активные службы, которые иногда приводят к лучшему распараллеливанию взаимозависимых служб;

– cgroups используется для отслеживания служебных процессов, вместо идентификаторов процессов (PID). Это означает, что демоны не будут потеряны даже после разветвления в другие процессы.

Помимо простого запуска и контроля сервисов systemd предлагает некоторые другие удобные функции, для использования которых ранее системным администраторам приходилось прибегать к помощи дополнительных программ-демонов. Среди таких функций:

– сокет-активация служб (заменяет inetd);

– запуск сервисов по расписанию (заменяет CRON);

– работа с аппаратным сторожевым таймером (заменяет WatchDog);

– смена корня ФС (заменяет chroot);

– автомонтирование разделов дисков и сетевых ресурсов (заменяет mount и fstab).

Юниты представляют собой специально созданные файлы конфигурации запуска той или иной службы.

Для управления службами в systemd применяется системная команда `systemctl`. Просмотр журнала systemd выполняется с помощью системной команды `journalctl`.

Используемые для управления команды приведены в таблице 3.

Таблица 3

Команда	Описание
`systemctl start <unit>`	Запуск юнита
`systemctl stop <unit>`	Остановка юнита
`systemctl restart <unit>`	Перезапуск юнита
`systemctl kill <unit>`	Принудительная остановка юнита
`systemctl status`	Отображение состояния служб
`systemctl show-environment`	Вывод окружения
`systemctl set-environment <param>=<value>`	Установка переменной окружения
`systemctl unset-environment <param>`	Удаление переменной окружения
`systemctl list-units`	Вывод списка известных юнитов
`systemctl list-dependencies –all`	Отображение зависимостей между юнитами
`systemctl enable <unit>`	Включение юнита в автозагрузку
`systemctl disable <unit>`	Выключение юнита из автозагрузки
`systemctl mask <unit>`	Запретить запуск юнита
`systemctl unmask <unit>`	Разрешить запуск юнита
`journalctl –u <unit>`	Просмотр журнала запуска юнита

Используемые ранее в механизме System V init утилиты (service) и конфигурационные файлы (например, в /etc/init.d/) поддерживаются ретроспективно.

3.3.11. Изменение состояния системы – shutdown, init

В каждый момент времени система находится на одном из восьми возможных уровней выполнения. Каждое состояние определяет особенности функционирования (таблица 4).

Таблица 4

Уровень	Описание
0	Остановка системы для безопасного отключения питание. Если возможно, требует автоматически выключить питание
1	Режим системного администрирования. Все локальные файловые системы смонтированы. Работает только небольшой набор существенных процессов ядра. Предназначен для решения административных задач, например, установки дополнительных пакетов. Все файлы доступны, и никакие пользователи в системе не зарегистрированы

Окончание таблицы 4

Уровень	Описание
2	Многопользовательский режим. Запускаются все необходимые для работы многопользовательской среды процессы и службы
3	Расширенный многопользовательский режим. Предоставляется доступ по сети к локальным ресурсам
4	Альтернативная конфигурация многопользовательской среды. Не обязателен для работы системы и обычно не используется
5	Расширенный многопользовательский режим. Предоставляется доступ по сети к локальным ресурсам. Используется для графического входа
6	Остановить операционную систему и перезагрузить ее в состояние, задаваемое записью initdefault в файле /etc/inittab
S, s	Однопользовательский режим. Единственный уровень выполнения, не требующий наличия файла /etc/inittab соответствующего формата. Все пользовательские процессы останавливаются, а файловые системы, необходимые для многопользовательской работы, демонтируются. После этого доступ к системе возможен только с консоли

Останов, выключение и перезагрузка ОС выполняются с помощью изменения ее уровня выполнения. Для этой операции предназначены системные команды `shutdown`, `init`, `halt`, `poweroff` и `reboot`.

Команда `shutdown` используется как универсальный способ инициирования останова, перезагрузки или возврата в однопользовательский режим. При указании паузы перед отработкой команды утилита посылает зарегистрированным пользователям через постепенно укорачивающиеся промежутки времени сообщения, предупреждая их о приближающемся останове (по умолчанию в сообщениях указывается время, оставшееся до останова).

Выполнение команды осуществляется:

```
$ sudo shutdown [flags] time [warning-message]
```

где `[warning-message]` – посылаемое всем пользователям сообщение;

`time` представляет собой время выполнения отключения системы.

Значение может быть также задано в формате +m, где m – количество мин ожидания до остановки системы. Значение +0 может быть заменено словом now.

В таблице 5 перечислены основные опции команды `shutdown`.

Таблица 5

Опция	Описание
-H, --halt	Останов системы
-P, --poweroff	Выключение системы (по умолчанию)
-r, --reboot	Перезагрузка системы
-h	Эквивалентно – poweroff, если не задано – halt
-k	Послать предупреждение без реального завершения работы системы
--no-wall	Не посылать предупреждения
-c	Отказаться от уже запущенного процесса завершения работы. Опция time при этом не может быть использована

Системная команда `init` применяется для инициации перехода к заданному уровню выполнения.

Системные команды `halt`, `poweroff` и `reboot` инициируют переход к соответствующему уровню выполнения. Являются альтернативой вызова shutdown с соответствующей опцией, но могут содержать свои расширенные опции.

3.4. Управление устройствами

3.4.1. Типы устройств

В ОС существует два типа устройств: блочные с прямым доступом (например, ЖД) и символьные (например, последовательные порты), некоторые из них могут быть последовательными, а некоторые – с прямым доступом. Каждое поддерживаемое устройство представляется в ФС файлом устройства. При выполнении операций чтения или записи с подобным файлом происходит обмен данными с устройством, на которое указывает этот файл. Такой способ доступа к устройствам позволяет не использовать специальные программы (а также специальные методы программирования, такие как работа с прерываниями).

Так как устройства отображаются как файлы в ФС (в каталоге /dev), их можно обнаружить с помощью команды ls. После выполнения команды:

```
$ ls -l
```

на экран монитора выводится список файлов, причем в первой колонке содержится тип файла и права доступа к нему. Например, для просмотра файла, соответствующего звуковому устройству, используется следующая команда:

```
$ ls -l /dev/dsp
crw-rw---T+ 1 root audio 14, 3 Июл 1 13:05 /dev/dsp
```

Первый символ в первой колонке (c) показывает тип файла, в данном случае –

символьное устройство. Для обычных файлов используется символ «-» (дефис), для каталогов - d, для блочных устройств - b (описание команды приведено в man ls).

Наличие большого количества файлов устройств не означает, что эти устройства на самом деле установлены. Наличие файла /dev/sda ни о чем не говорит и совсем не означает, что на компьютере установлен ЖД SCSI. Это предусмотрено для облегчения установки программ и нового оборудования (нет необходимости искать нужные параметры и создавать файлы для новых устройств).

3.4.2. Управление разделами

Для управления разделами дисков в состав ОС включены ПС gparted, parted, fdisk, cfdisk, sfdisk.

Утилиты parted, fdisk, cfdisk, sfdisk не имеют графического интерфейса, утилита gparted является графической оболочкой к программе parted.

Для создания файловой системы используется графическая утилита gparted или консольная утилита mkfs.

3.4.3. Программная организация разделов в RAID

Для управления программными RAID-массивами используется утилита mdadm, реализующая программный RAID-массив различных типов.

3.5. Управление ФС

ФС определяет формат содержимого и способ физического хранения информации, сгруппированной в виде файлов и каталогов. Конкретная ФС определяет размер имен файлов и каталогов, максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые ФС предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

ФС не обязательно напрямую связана с физическим носителем информации. Существуют виртуальные ФС, а также сетевые ФС, которые являются лишь способом доступа к файлам, находящимся на удаленном компьютере.

ОС поддерживает следующие типы ФС (состав может быть расширен с помощью дополнительного программного обеспечения):

– носители с произвольным доступом – ext2, ext3, ext4, FAT, NTFS;

– оптические носители – ISO9660, UDF;

– носители флеш-памяти – exFAT;

– виртуальные ФС– tmpfs, ramfs;

– сетевые ФС– NFS, CIFS.

Конкретная ФС перед использованием должна быть инициализирована и

подключена как часть общей ФС ОС.

3.5.1. Структура ФС

ФС имеет древовидную структуру, корень которой обозначается как «/». Относительно корня задаются абсолютные пути к файлам и каталогам.

Типовая структура ФС после установке ОС приведена в таблице 6. Отдельные элементы структуры ФС ОС могут размещаться на разных физических разделах, иметь разный тип, в том числе размещаться на удаленных сетевых ресурсах.

Таблица 6

Путь	Описание
/	Корень ФС, размещается на основном системном разделе
/bin	Каталог исполняемых программ, доступных пользователю
/boot	Содержит необходимые для загрузки системы файлы: ядра, загрузочный образ initrd, файлы загрузчика
/cdrom	Временная точка монтирования оптических носителей
/dev	Каталог файлов устройств
/etc	Каталог системных конфигурационных файлов, необходимых для работы ОС и ее компонентов
/NESS	Каталог системных файлов подсистемы безопасности NESS (0)
/home	Каталог домашних каталогов пользователей, в зависимости от назначения системы и количества пользователей может размещаться на отдельном разделе или подключаться удаленно
/lib /lib64	Каталоги основных и системных разделяемых библиотек, необходимых для функционирования ОС и ее начальной загрузке
/media	Каталог точек автоматического монтирования ФС, как правило носителей информации и сетевых дисков
/mnt	Каталог точек временного ручного монтирования ФС, как правило носителей информации и сетевых дисков
/opt	Каталог дополнительного ПО, повторяет структуру /usr
/proc	Точка монтирования виртуальной ФС, предоставляющей информацию о процессах ОС
/root	Домашний каталог суперпользователя

Окончание таблицы 6

Путь	Описание
/run	Каталог файлов состояния приложений
/sbin	Каталог системных привилегированных программ
/srv	Каталог данных сервисных служб

/sys	Точка монтирования виртуальной ФС sysfs, предоставляющей информацию о присутствующих в системе устройствах и драйверах
/tmp	Каталог временных файлов, в зависимости от назначения системы может располагаться на отдельном носителе или размещаться в оперативной памяти (ОП)
/usr	Каталог пользовательских программ и данных, используемых только для чтения
/var	Каталог изменяемых данных, БД, файлы журналов и т.п.

3.5.2. Инициализация — mkfs

Инициализация ФС заключается в создании служебной структуры хранений информации об иерархии ее объектов в физической среде хранения. Структура зависит от типа ФС и требует часть доступного физического пространства устройства хранения.

Для создания ФС предназначена утилита mkfs, имеющая формат вызова:

```
$ sudo mkfs [-t fstype] [fs-options] device [size]
```

Данная команда является оболочкой для вызова утилиты инициализации, соответствующей указанному с помощью опции -t fstype типу ФС, например mkfs.ext4 (при mkfs -t ext4).

В качестве места создания ФС может быть указана любая точка ФС, как файл блочного устройства ЖД (например, /dev/hda1), точка монтирования или даже существующий файл.

При этом с помощью дополнительных опций fs-options можно задать параметры, специфичные для выбранного типа ФС.

3.5.3. Монтирование – mount

Каждая ФС перед использованием должна быть примонтирована как часть общей ФС ОС. При этом выполняются необходимые действия, обеспечивающие подключение указанной ФС в заданную точку монтирования, являющуюся каталогом общей ФС ОС. При необходимости выполняются процедуры аутентификации для доступа к сетевым ФС или иные подготовительные действия.

Доступ к подключенным ФС осуществляется с помощью обращений к каталогу общей структуры каталогов ОС. Такой каталог называется точкой монтирования и должен существовать на момент монтирования.

Просмотр информаций по принадлежности каталога ФС может быть выполнен с помощью команды `df`, например:

```
$ df -h
```

При этом будет выведен состав каталогов первого уровня с указанием типов ФС, их

размеров и используемого места в удобочитаемом виде.

При монтировании ФС в каталог, уже содержащий файлы, доступ к ним будет невозможен до размонтирования.

Для монтирования ФС предназначена команда `mount`, имеющая обобщенный формат вызова:

```
$ sudo mount [options] [-t fstype] [-o fs-options] device dir
```

где: `fstype`– тип ФС;

`device`– устройство;

`dir`– точка монтирования (существующий каталог).

Тип ФС указывается с помощью опции fstype,

Для неподдерживаемых самой командой типов ФС, она вызывает соответствующую указанному с помощью опции -t fstype типу ФС утилиту монтирования, например mount.cifs (при mount -t cifs).

Команда `mount` предоставляет широкий функционал по операциям, связанным с монтированием ФС:

– монтирование ФС;

– перемонтирование ФС с другими опциями и режимами доступа;

– перемонтирование каталога ФС в другое место (создание синонима для каталога);

– изменение режима работы точки монтирования.

Список часто используемых опций команды приведен в таблице 7.

Таблица 7

Опция	Описание
-a, --all	Подключить все ФС, перечисленные в /etc/fstab
-f, --fake	Имитация подключения ФС. Выполняются все действия, кроме непосредственно монтирования
--bind	Создание синонима для каталога
-i, --internal-only	Использовать только поддерживаемые командой типы ФС без вызова внешних обработчиков
-n, --no-mtab	Не фиксировать факт подключения в файле /etc/mtab
-o <опции>, --options <опции>	Специфичные для конкретного типа ФС параметры, разделенные запятой
-O, --test-opts <опции>	Используется с -a для ограничения списка ФС
-r, --read-only	Подключение в режиме только для чтения, аналог -o
-t <тип>, --types <тип>	Список типов подключаемых ФС
--source <источник>	Явное указание источника
--target <цель>	Явное указание точки монтирования
-v, --verbose	Подробный отчет о выполняемых действиях
-w, --rw, --read-write	Подключение в режиме чтения/записи (по умолчанию)
-h, --help	Справка по способу вызова и опциям команды

Информация по монтируемым ФС отображается в специальном системном файле динамической конфигурации ФС/etc/mtab. При выполнении команды `mount` информация о монтировании ФС отражается в этом файле, если это не отключено специально.

ВНИМАНИЕ! МОНТИРОВАНИЕ ПРОИЗВОЛЬНЫХ ФС ДОСТУПНО ТОЛЬКО СУПЕРПОЛЬЗОВАТЕЛЮ. ПОЛЬЗОВАТЕЛЬ ИМЕЕТ ВОЗМОЖНОСТЬ ВЫПОЛНИТЬ С ПОМОЩЬЮ ЭТОЙ КОМАНДЫ ТОЛЬКО ОПЕРАЦИИ, СВОЙСТВА КОТОРЫХ ЗАДАНЫ В ФАЙЛЕ СТАТИЧЕСКОЙ КОНФИГУРАЦИИ ФС /ETC/FSTAB С ОПЦИЕЙ USER.

3.5.4. Конфигурация ФС– fstab

Для описания автоматически подключаемых ФС предназначен системный конфигурационный файл /etc/fstab.

Статическая конфигурация ФС содержит информацию о подключении необходимых для функционирования ОС ФС и ФС для автоматического монтирования. При этом указывается их назначение, точки и опции монтирования.

Каждая строка файла имеет следующий формат (поля разделяются пробелом или табуляцией и описаны таблице 8):

```
<file system><dir><type><options><dump><pass>
```

Таблица 8

Поле	Описание
file system	Имя ФС или файла устройства, может содержать UUID раздела жесткого диска
dir	Точка монтирования в ФС, может указываться none для системных ФС или файлов подкачки
type	Тип ФС
options	Опции монтирования через запятую без пробелов, значение defaults определяет опции по умолчанию. Состав опций зависит от типа ФС и может включать следующие: auto – автоматическое монтирование (или по mount -a); noauto – только ручное монтирование; exec – исполнять файлы с ФС (по умолчанию); ro (rw) – режим только для чтения (чтения/записи); sync (async) – синхронное (асинхронное) выполнение операций; user – разрешение монтировать пользователем (включает noexec, nosuid, nodev – если они не переопределены); nouser – монтирование только суперпользователем (по умолчанию); defaults – опции по умолчанию (rw, suid, dev, exec, auto, nouser, async); suid (nosuid) – использование (запрет) операций с suid и sgid битами; nodev – запрет использования файлов устройств; atime (noatime) – включает (выключает) запись времени доступа; size – задание размера файловой системы (только для tmpfs).
dump	Флаг резервного копирования утилитой dump, если 1 – выполняется резервное копирование, по умолчанию – 0
pass	Порядок проверки целостности ФС при загрузке командой `fsck`: 0 – не проверять; 1 – корневая ФС; 2 – остальные ФС.

В тексте могут использоваться комментарии стандартного вида с начальным символом «#».

3.5.5. Размонтирование – umount

Подключенные ФС могут быть отключены или могут отключаться автоматически. Ручное отключение требуется перед физическим изъятием носителя, например USB флеш-диска, или при необходимости выполнить профилактическое обслуживание, например проверки и восстановления с помощью команды `fsck`.

Сетевые ФС могут отключаться в случае потери связи с удаленным источником.

Перед отключением ФС завершить все работающие с ней процессы (например, выйти из каталога точки монтирования).

ВНИМАНИЕ! ФС НЕ МОЖЕТ БЫТЬ ОТКЛЮЧЕНА, ЕСЛИ СУЩЕСТВУЮТ ПРОЦЕССЫ, ОБРАЩАЮЩИЕСЯ К ПРИНАДЛЕЖАЩИМ ЕЙ ФАЙЛАМ (ДЕРЖАТ

ОТКРЫТЫМИ ДЕСКРИПТОРЫ). В ЭТОМ СЛУЧАЕ ФС СЧИТАЕТСЯ ЗАНЯТОЙ («BUSY»), И МОЖЕТ БЫТЬ ОТКЛЮЧЕНА ЛИБО ПРИНУДИТЕЛЬНО, ЛИБО С ОПЦИЕЙ ОТЛОЖЕННОГО ОТКЛЮЧЕНИЯ.

Для определения занимающих ФС процессов может быть использована служебная команда `fuser`, например:

```
$ fuser -v <каталог>
```

При этом будет выведен список файлов и идентификаторов процессов, которые их открыли.

При необходимости с помощью опций команды `fuser` перечисленные процессы могут быть завершены.

Для отключения (размонтирования) подключенный ФС предназначена команда `umount`, имеющая следующий обобщенный формат вызова:

```
$ sudo umount [options] {<directory>|<device>}
```

где в качестве аргумента может указываться устройство или точка монтирования.

Список часто используемых опций команды приведен в таблице 9.

Таблица 9

Опция	Описание
-a, --all	Отключить все ФС, перечисленные в /etc/mtab, за исключением виртуальной ФС /proc
--fake	Имитация отключения ФС. Выполняются все действия, кроме непосредственно размонтирования
-f, --force	Принудительное отключение ФС
-i, --internal-only	Использовать только поддерживаемые командой типы ФС без вызова внешних обработчиков
-l, --lazy	Отложенное отключение занятых ФС после их освобождения
-n, --no-mtab	Не фиксировать факт отключения в файле /etc/mtab
-O, --test-opts <опции>	Используется с -a для ограничения списка ФС
-R, --recursive	Рекурсивное отключение ФС
-r, --read-only	В случае неудачи отключения ФС, переподключение в режиме только для чтения

Окончание таблицы 9

Опция	Описание
-t <тип>, --types <тип>	Список типов отключаемых ФС
-v, --verbose	Подробный отчет о выполняемых действиях
-h, --help	Справка по способу вызова и опциям команды

Для размонтирования и освобождения ФС подключенных сменных носителей информации предназначена команда `eject`.

3.5.6. Проверка и исправление – fsck

Для проверки целостности и исправления ошибок предназначена системная команда `fsck`, имеющая следующий обобщенный формат вызова:

```
$ sudo fsck [options] [-t fstype] [filesys … ] [--] [fs-specific-options]
```

где:

`fstype`–тип ФС;

`filesys`– точка монтирование или файл устройства;

`fs-specific-options`– специфичные для типа ФС опции.

Как и многие команды этой группы, может вызывать дополнительные обработчики, соответствующие типу ФС (например, fsck.vfat).

ВНИМАНИЕ! ДЛЯ ВОЗМОЖНОСТИ ИСПРАВЛЕНИЯ ТРЕБУЕТСЯ МОНОПОЛЬНЫЙ ДОСТУП К ФС. ПЕРЕД ВЫПОЛНЕНИЕМ КОМАНДЫ FSCK ФС ДОЛЖНА БЫТЬ РАЗМОНТИРОВАНА.

В таблице 10 приведены основные опции команды `fsck`.

Таблица 10

Опция	Описание
-A	Проверить все ФС, перечисленные в /etc/fstab
-M	Не проверять примонтированные ФС
-N	Эмуляция без реального применения на ФС
-P	Используется с -A для указания проверки корневой ФС
-R	Используется с -A для исключения проверки корневой ФС
-s	Сериализация операций
-t <тип>	Список типов проверяемых ФС
-V	Подробный отчет о выполняемых действиях
-h, --help	Справка по способу вызова и опциям команды

В качестве специфичных типу ФС опций зачастую используются следующие:

– -a – автоматическое исправление выявленных проблем без запроса подтверждения у пользователя;

– -n – отключение исправления выявленных проблем, вывод только информации о них;

– -r – интерактивный режим исправления выявленных проблем с запросом подтверждения у пользователя;

– -y – во многих случая указывает автоматически исправлять все выявленные ошибки.

Код, возвращаемый командой `fsck`, является суммой следующих условий:

– 0 – нет ошибок;

– 1 – ошибки ФС исправлены;

– 2 – необходима перезагрузка системы;

– 4 – ошибки ФС не исправлены;

– 8 – в процессе проверки произошли ошибки;

– 16 – неверное использование команды либо синтаксическая ошибка;

– 32 – fsck была прервана пользователем;

– 128 – ошибка разделяемых объектов.

При загрузке системы выполняется проверка ФС согласно опциям статической конфигурации ФС в /etc/fstab.

3.6. Управление учетными записями и параметрами аутентификации

3.6.1. Управление учетными записями пользователей и групп

В ОС существуют следующие системные файлы учетных записей:

– /etc/passwd – учетные записи пользователей;

– /etc/group – учетные записи групп с указанием членства в них пользователей;

– /etc/shadow, /etc/gshadow – недоступные для чтения непривилегированным пользователям скрытые части учетных записей (пароли).

В процессе управления учетными записями наряду с созданием каталогов и другими необходимыми операциями производится модификация перечисленных файлов.

Управление учетными записями включает следующие административные действия:

– управление учетными записями пользователей (adduser/deluser);

– изменение атрибутов учетной записи пользователя (usermod/chfn/chfs);

– управление учетными записями групп (addgroup/delgroup);

– изменение атрибутов и состава пользователей группы (gpasswd);

Выполнение указанных действий требует прав администратора.

3.6.1.1. Управление учетными записями пользователей – adduser/deluser

Создание пользователей ОС выполняется с помощью команды `adduser`.

Данная команда представляет собой скрипт командной оболочки, позволяющий создать учетную запись пользователя, сформировать домашний каталог из шаблона в /etc/skel, задать параметры учетной записи пользователя и назначить пароль в интерактивном режиме. Помимо этого реализована возможность создания пользователя в

не интерактивном режиме.

Для создания пользователя ввести команду:

```
$ sudo adduser <пользователь>
```

Опции --system и --group позволяют задать создание системного пользователи и указать группу пользователя.

Для удаления учетной записи пользователя используется команда `deluser`. По умолчанию данная команда не удаляет домашний каталог и почтовый ящик пользователя. Если необходимо удалить домашний каталог, используется опция --remove-home. При указании опции --remove-all-files удаляются все принадлежащие пользователю файлы, включая почтовый ящик и домашний каталог.

Формат вызова:

```
$ sudo deluser <пользователь> [опции]
```

3.6.1.2. Изменение атрибутов учетной записи пользователя – usermod

Атрибуты учетной записи пользователя (расположение домашнего каталога, комментарий к учетной записи, группа пользователя, командная оболочка и др.) изменяются с помощью команды `usermod`.

Формат вызова команды usermod:

```
$ sudo usermod [опции] <имя_пользователя>
```

В таблице 11 приведены наиболее часто используемые опции данной команды.

Таблица 11

Ключ	Описание
-c <комментарий>	Изменить комментарий к учетной записи
-d<путь>	Создать домашний каталог по указанному пути

Окончание таблицы 11

Ключ	Описание
-m	Переместить домашний каталог (новое расположение указывается с помощью опции -d)
-s <оболочка>	Установить командную оболочку
-g <идентификатор_группы>	Изменить первичную группу

Для управления отдельными атрибутами учетной записи могут применяться команды:

– `cnfn`– изменение описательной части учетной записи;

– `chsh`– изменение командной оболочки пользователя, предоставляемой ему при

консольном входе.

3.6.1.3. Управление учетными записями групп – addgroup/delgroup

Создание учетных записей групп ОС выполняется с помощью команды `addgroup`.

Пример использования:

```
$ sudo addgroup [опции] <группа>
```

Опция --system позволяет задать создание системной группы.

Для удаления учетной записи группы пользователей используется команда `delgroup`:

```
$ sudo delgroup <группа>
```

ВНИМАНИЕ! ПЕРВИЧНАЯ ГРУППА СУЩЕСТВУЮЩЕГО ПОЛЬЗОВАТЕЛЯ НЕ МОЖЕТ БЫТЬ УДАЛЕНА.

3.6.1.4. Изменение атрибутов и состава пользователей в группе – gpasswd

Утилита gpasswd используется для изменения атрибутов группы и ее состава пользователей. Наряду с членством пользователей позволяет установить пароль на группу или изменить состав ее администраторов.

Утилита управляет только локальными группами и вторичным членством пользователей. Первичная группа пользователей является атрибутом его учетной записи и изменяется с помощью утилит изменения учетной записи пользователя.

Для добавления пользователя в группу используется команда:

```
$ sudo gpasswd --add <пользователь><группа>
```

Для удаления пользователя из группы используется команда:

```
$ sudo gpasswd --remove <пользователь><группа>
```

3.6.2. Управление параметрами аутентификации

Управление параметрами аутентификации включает административные действия:

– установка пароля пользователя (passwd);

– настройка срока действия пароля пользователя (chage);

– настройка блокировки учетной записи пользователя (usermod, passwd);

– учет неуспешных попыток входа в систему (faillog).

Выполнение указанных действий требует прав администратора.

3.6.2.1. Установка пароля пользователя – passwd

Изменение пароля пользователя выполняется с помощью утилиты passwd.

Смена пароля может выполняться по инициативе пользователя после истечения его времени действия или прямой установкой или сбросом администратором.

Вызов команды `passwd` запускает сценарий смены пароля для текущего

пользователя. Для изменения пароля конкретного пользователя требуются права администратора:

```
$ sudo passwd <пользователь>
```

В ходе сценария смены пароля пользователю будет предложено ввести старый пароль, после проверки которого пользователь должен ввести новый пароль и подтвердить повторным вводом. В случае запуска с правами администратора старый пароль не запрашивается. При этом к паролю пользователя применяется существующая в системе политика паролей.

Типовая парольная политика включает следующие правила: пароль должен иметь не менее шести символов (предпочтительно – восемь символов), содержать как прописные, так и строчные буквы, знаки препинания и цифры.

Данная команда модифицирует файл /etc/shadow, сохраняя в нем зашифрованное представление нового пароля.

ВНИМАНИЕ! ПАРОЛЬ РЕКОМЕНДУЕТСЯ СОЗДАВАТЬ СПОСОБОМ, МАКСИМАЛЬНО ЗАТРУДНЯЮЩИМ ЕГО ПОДБОР. НАИБОЛЕЕ БЕЗОПАСНЫЙ ПАРОЛЬ СОСТОИТ ИЗ СЛУЧАЙНОЙ (ПСЕВДОСЛУЧАЙНОЙ) ПОСЛЕДОВАТЕЛЬНОСТИ БУКВ, ЗНАКОВ ПРЕПИНАНИЯ, СПЕЦИАЛЬНЫХ СИМВОЛОВ И ЦИФР.

ПАРОЛЬ ХРАНИТСЯ В ЗАШИФРОВАННОМ ВИДЕ, ЧТО ИСКЛЮЧАЕТ ЕГО ВОССТАНОВЛЕНИЕ ПУТЕМ НАПОМИНАНИЯ. ЕДИНСТВЕННЫМ СПОСОБОМ ВОССТАНОВЛЕНИЯ ЯВЛЯЕТСЯ УСТАНОВКА НОВОГО ПАРОЛЯ.

3.6.2.2. Настройка срока действия пароля пользователя – chage

Для управления сроком действия пароля пользователя используется команда `chage`. Команда изменяет количество дней между датой смены пароля и датой последней смены пароля. Эта информация используется системой для определения момента, когда пользователь должен сменить свой пароль.

Формат вызова:

```
$ sudo chage [опции] <пользователь>
```

Список наиболее часто используемых опций приведен в таблице 12.

Таблица 12

Ключ	Описание
-E, --expiredate ДАТА	Установить дату окончания действия пароля (в формате ГГГГ-ММ-ДД)
-I, --inactive КОЛ_ДНЕЙ	Установить неактивность пароля после устаревания в значение КОЛ_ДНЕЙ
-m, --mindays МИН_ДНЕЙ	Установить минимальное число дней перед сменой пароля

Ключ	Описание
-M, --maxdays МАКС_ДНЕЙ	Установить максимальное число дней перед сменой пароля
-W, --warndays ДНЕЙ	Установить количество дней с выдачей предупреждения о необходимости смены пароля

3.6.2.3. Настройка блокировки учетной записи пользователя – usermod, passwd

Блокировка учетной записи может быть произведена следующими способами:

– ограничением срока жизни учетной записи;

– сбросом пароля пользователя;

– блокировкой пароля пользователя.

Ограничение срока жизни учетной записи производится командой `usermod`:

```
$ sudo usermod --expiredate <дата><пользователь>
```

В указанной команде дата указывается в формате ГГГГ-ММ-ДД. Если необходимо выполнить блокировку учетной записи немедленно, то в качестве даты указывается 0. Если требуется сделать учетную запись бессрочной – -1.

Операции сброса пароля пользователя и его блокировка осуществляется с помощью команды `passwd`.

Для сброса пароля ввести:

```
$ sudo passwd --delete <пользователь>
```

Для блокировки пароля ввести:

```
$ sudo passwd --lock <пользователь>
```

Для разблокировки пароля используется следующая команда:

```
$ sudo passwd --unlock <пользователь>
```

3.6.2.4. Регистрация неуспешных попыток входа в систему – faillog

В случае ввода неверного пароля производится запись о факте неуспешного входа в систему в журнал неудачных попыток /var/log/faillog.

Команда `faillog` показывает содержимое журнала неудачных попыток (файл/var/log/faillog) входа в систему. Также она может быть использована для управления счетчиком неудачных попыток и их ограничением. При запуске faillog без параметров выводятся записи faillog только тех пользователей, у которых имеется хотя бы одна неудачная попытка входа.

Предельное число попыток входа для каждой учетной записи равно 10. Для сброса неудачных попыток входа применяется опция -r.

3.6.3. Графическая утилита управления учетными записями – muser

Для управления группами пользователей, пользователями и атрибутами безопасности в состав ОС входит графическая утилита muser.

Запуск утилиты выполняется с помощью пункта «Пользователи и группы» из меню ОС, либо с помощью команды:

```
$ muser-runner
```

На главной экранной форме представлены основные свойства пользователей, групп и мандатных атрибутов (рис. 22).

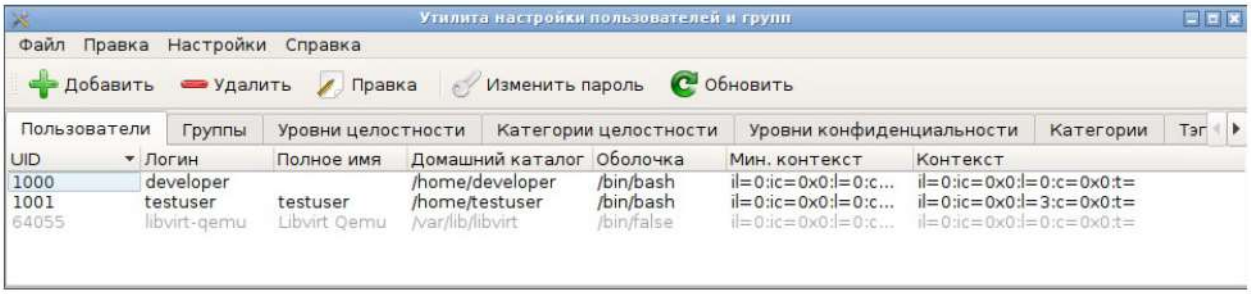


Рис. 22

В верхней части представлены меню и панель инструментов для быстрого доступа к функциям управления учетными записями. При создании учетных записей применяется ряд системных предустановок, доступ к которым возможен с помощью пунктов меню «Настройки» (см. рис. 22).

Меню «Настройки» содержит пункты:

– «Показать панель инструментов» – влияет на скрытие панели инструментов приложения;

– «Показать системных пользователей/группы» – если активно, то во вкладках «Пользователи» и «Группы» отображаются все пользователи и группы, включая системные;

– «Разрешить редактирование атрибутов безопасности» – если активно, то разрешается вносить изменения в состав атрибутов безопасности на соответствующих вкладках;

– «Настройки по умолчанию» – вызывает диалог просмотра настроек создания пользователей и групп, применяемых по умолчанию (берутся из файла /etc/adduser.conf) (рис. 23).

– «Сложность пароля» – вызывает диалог редактирования сложности пароля (рис. 24);

– «Сложность PIN (GuardantID)» – вызывает диалог редактирования параметров сложности PIN GuardantID (рис. 25).

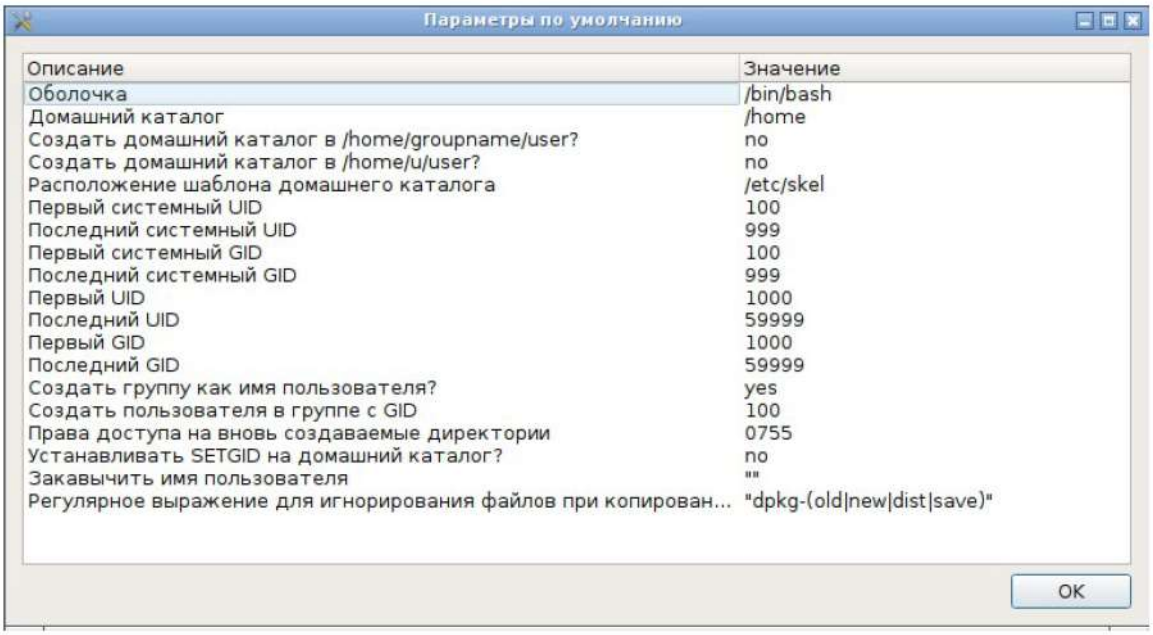

Параметры по умолчанию
Описание
Значение
Оболочка
/bin/bash
Домашний каталог
/home
Создать домашний каталог в /home/groupname/user?
no
Создать домашний каталог в /home/u/user?
no
Расположение шаблона домашнего каталога
/etc/skel
Первый системный UID
100
Последний системный UID
999
Первый системный GID
100
Последний системный GID
999
Первый UID
1000
Последний UID
59999
Первый GID
1000
Последний GID
59999
Создать группу как имя пользователя?
yes
Создать пользователя в группе с GID
100
Права доступа на вновь создаваемые директории
0755
Устанавливать SETGID на домашний каталог?
no
Закавычить имя пользователя
""
Регулярное выражение для игнорирования файлов при копирован...
"dpkg-(old|new|dist|save)"
OK

Рис. 23

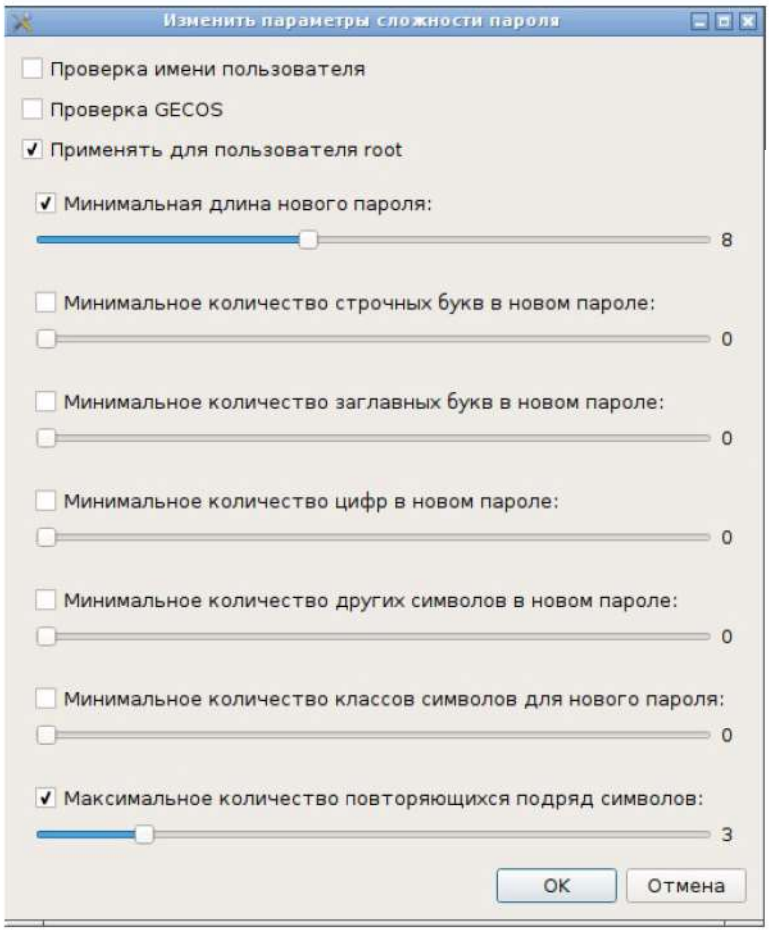

Изменить параметры сложности пароля
Проверка имени пользователя
Проверка GECOS
Применять для пользователя root
Минимальная длина нового пароля:
8
Минимальное количество строчных букв в новом пароле:
0
Минимальное количество заглавных букв в новом пароле:
0
Минимальное количество цифр в новом пароле:
0
Минимальное количество других символов в новом пароле:
0
Минимальное количество классов символов для нового пароля:
0
Максимальное количество повторяющихся подряд символов:
3
OK
Отмена

Рис. 24

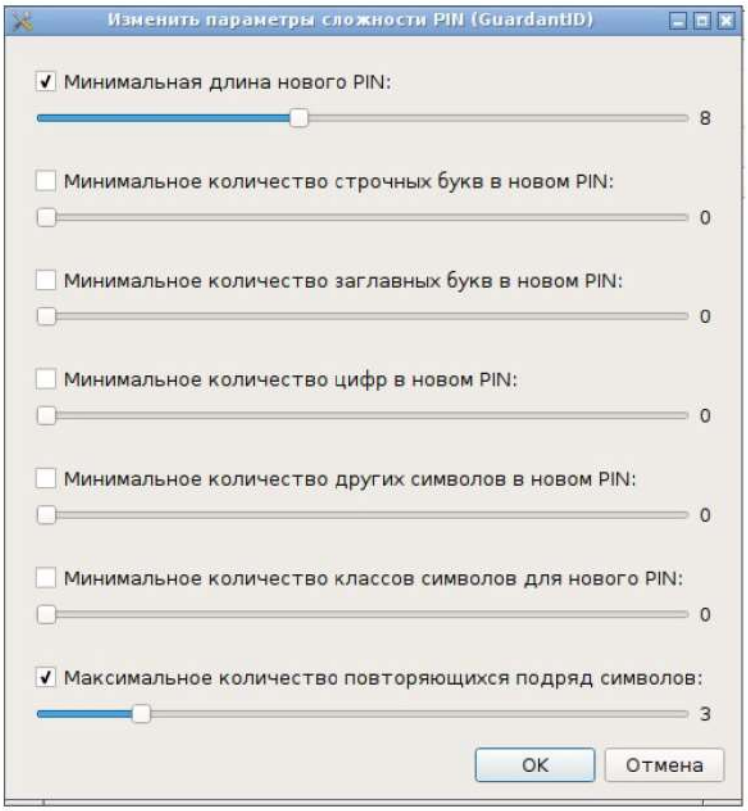


Рис. 25

Для выполнения операций по добавлению, модификации, удалению учетной записи пользователя и изменению пароля необходимо выбрать вкладку «Пользователи» главной экранной формы.

Нажатие кнопки [Добавить] на панели инструментов запускает диалог создания нового пользователя (рис. 26).

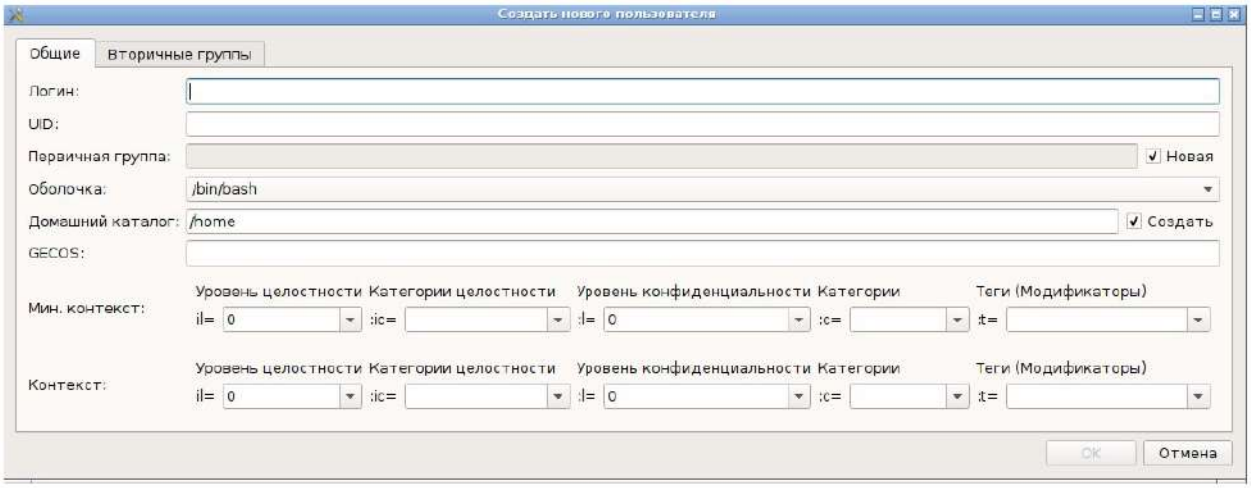


Рис. 26

На первой вкладке диалога заполняются основные свойства учетной записи: логин, идентификатор пользователя, командная оболочка, домашний каталог и описание GECOS. При необходимости, администратор может указать минимальный контекст

безопасности и контекст для пользователя с помощью выпадающих списков. На вкладке «Вторичные группы» могут быть указаны группы, в которые будет входить пользователь после добавления в ОС (рис. 27).

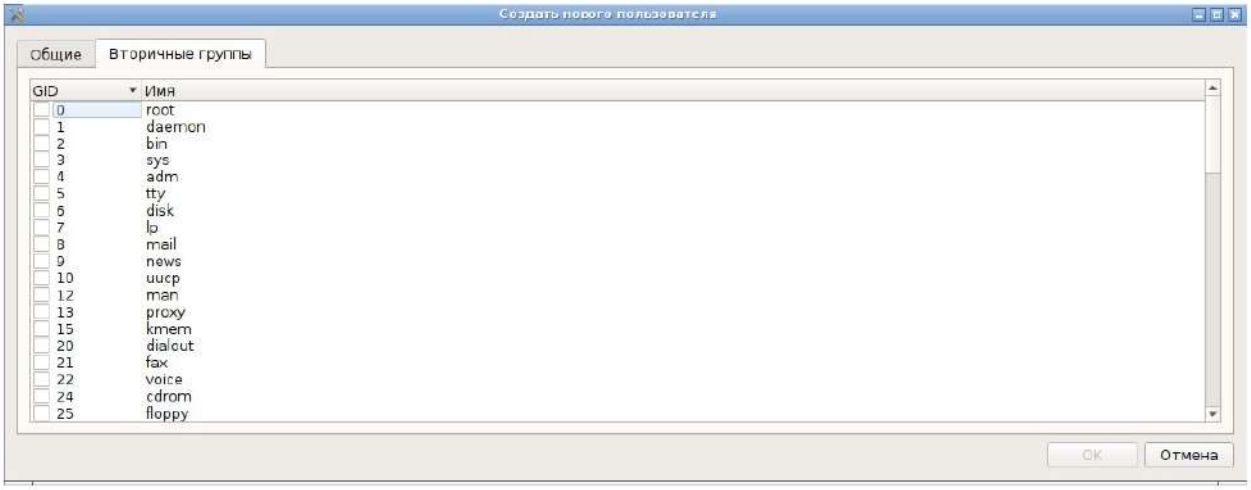


Рис. 27

Смена пароля уже созданного пользователя выполняется с помощью кнопки [Изменить пароль] на панели инструментов. При этом используется системный сценарий смены пароля, в ходе которого будут запрошены необходимые данные.

Для изменения свойств учетной записи пользователя, срока ее действия или выполнения ее блокировки требуется выбрать пользователя и нажать [Правка]. Далее будет запущен диалог редактирования свойств выбранной учетной записи (рис. 28).

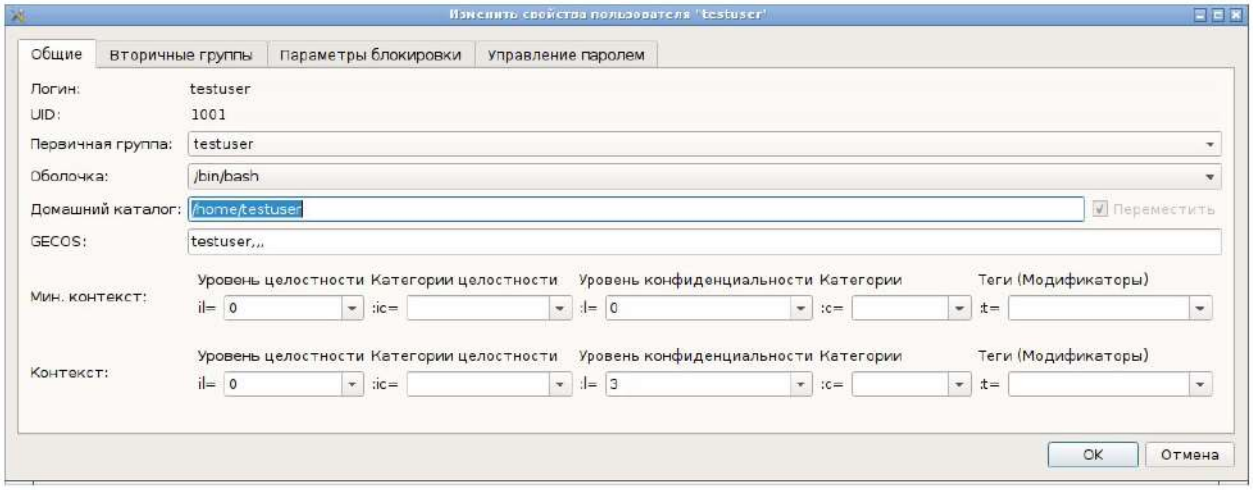


Рис. 28

В отличие от процедуры создания, часть обязательных атрибутов недоступна для модификации.

На основной вкладке диалога доступны для изменения: командная оболочка, первичная группа пользователя, домашний каталог, описание учетной записи (GECOS), а также с помощью выпадающих списков задаются контексты безопасности.

На вкладке «Вторичные группы» настраиваются дополнительные группы пользователя.

На вкладке «Параметры блокировки» доступны для редактирования счетчик неудачных попыток, максимальное число неуспешных попыток, а также параметры блокировки учетной записи (рис. 29).

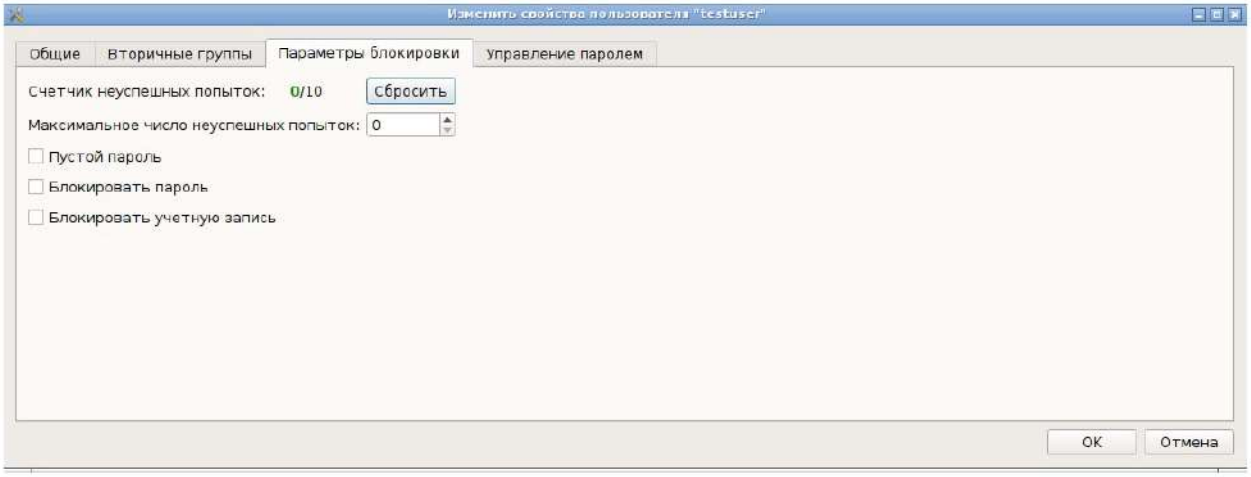


Рис. 29

При необходимости сброс счетчика неуспешных попыток выполняется нажатием [Сбросить].

На вкладке «Управление паролем» указываются настройки времени жизни учетной записи (рис. 30).

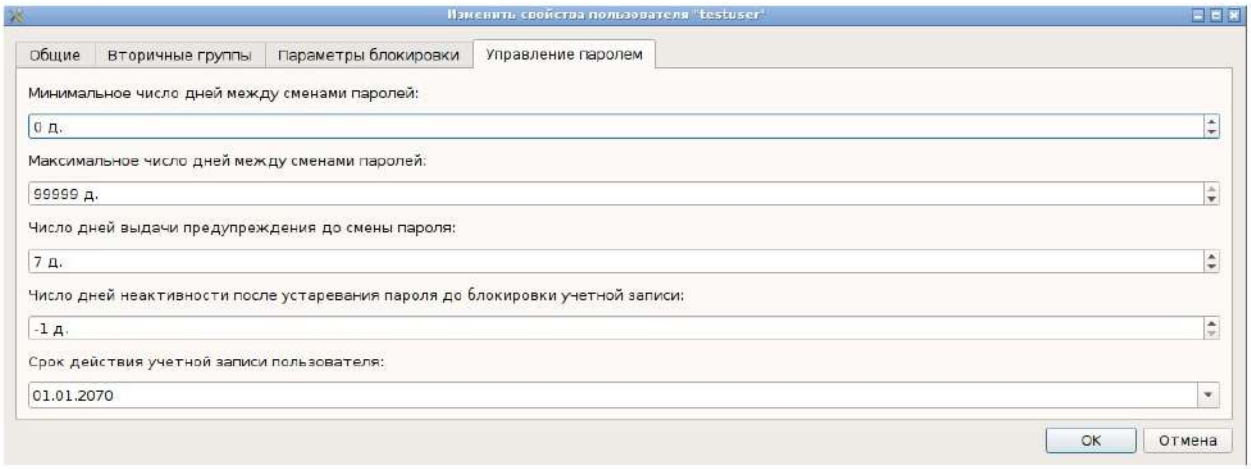


Рис. 30

Параметры времени жизни задаются в днях. Ввод срока действия выполняется с помощью выпадающего календаря.

Для удаления пользователя из ОС требуется выбрать пользователя и нажать кнопку [Удалить] на панели инструментов, при этом будет запрошено подтверждение выполнения операции.

Для выполнения операций по добавлению, модификации, удалению учетной записи групп пользователей выбрать вкладку «Группы» главной экранной формы (рис. 31).

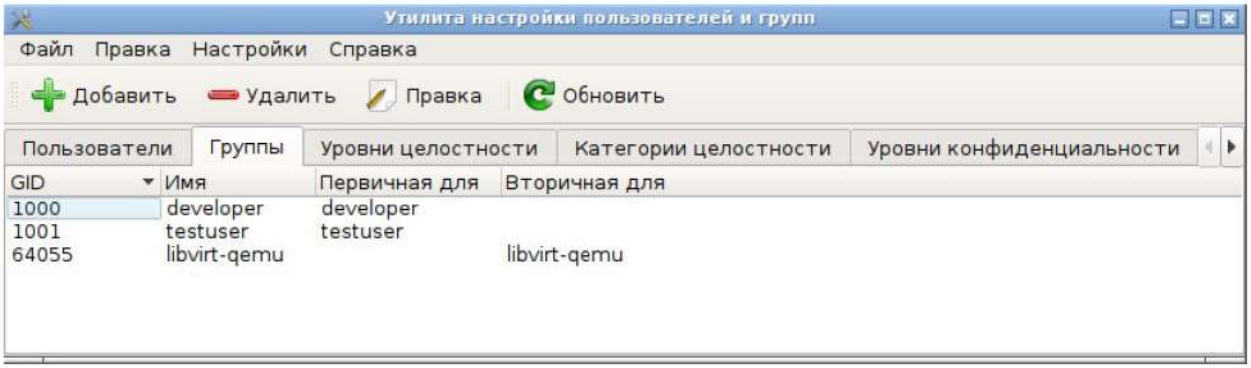


Рис. 31

Нажатие [Добавить] на панели инструментов запускает диалог создания новой группы (рис. 32).

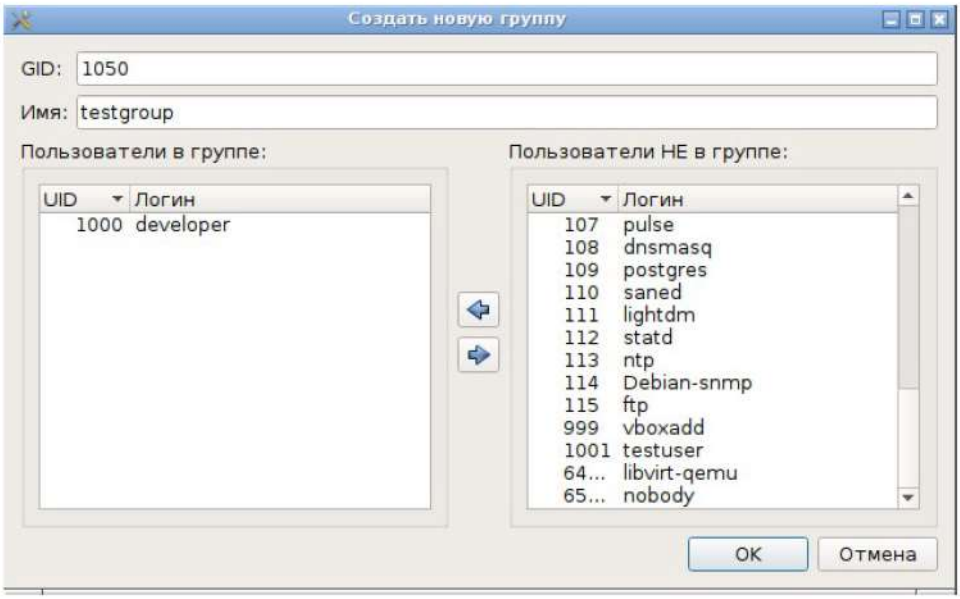


Рис. 32

В диалоге указывается имя группы и по желанию администратора идентификатор группы (GID). Также в создаваемую группу могут быть добавлены выбранные пользователи. Редактирование состава группы выполняется с помощью кнопки [Правка].

Для удаления группы из ОС требуется выделить группу и нажать кнопку [Удалить] на панели инструментов, при этом будет запрошено подтверждение выполнения операции.

Доступные уровни целостности, категории целостности, модификаторы доступа (тэги) приведены на вкладках «Уровни целостности» (рис. 33), «Категории целостности» (рис. 34) и «Тэги (модификаторы)» (рис. 35) соответственно.

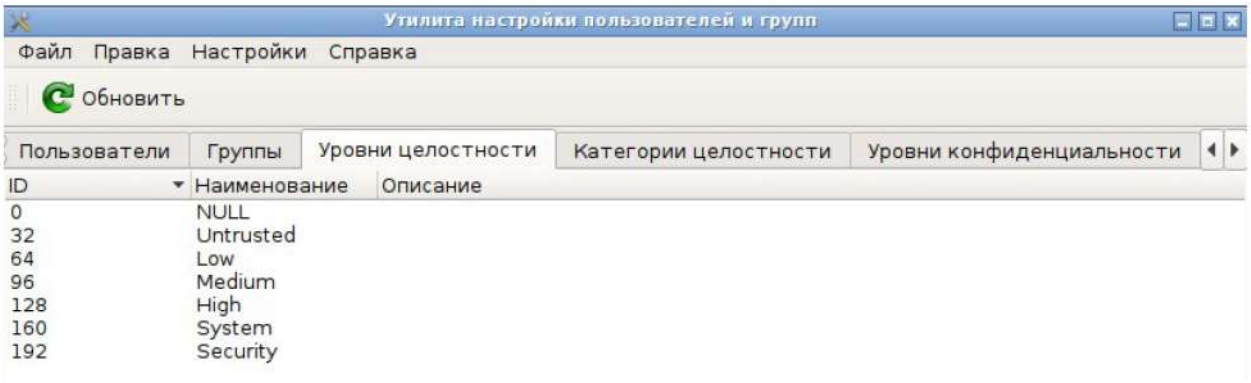


Рис. 33

Рис. 34

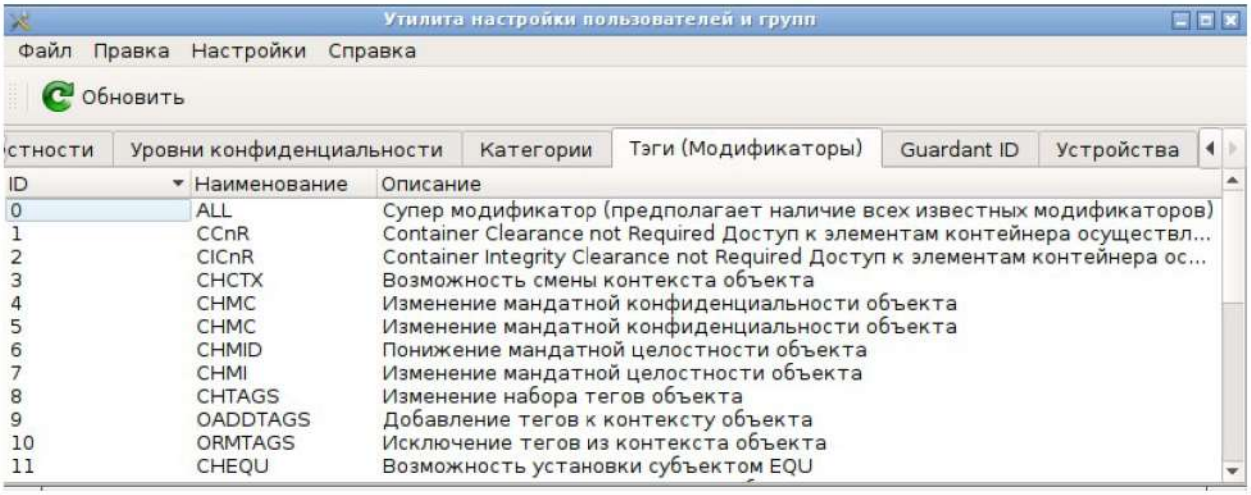


Рис. 35

Для выполнения операций по добавлению, модификации, удалению уровней конфиденциальности выбрать вкладку «Уровни конфиденциальности» главной экранной формы (рис. 36) и убедиться, что флаг «Разрешить редактирование атрибутов безопасности» установлен в настройках программы.

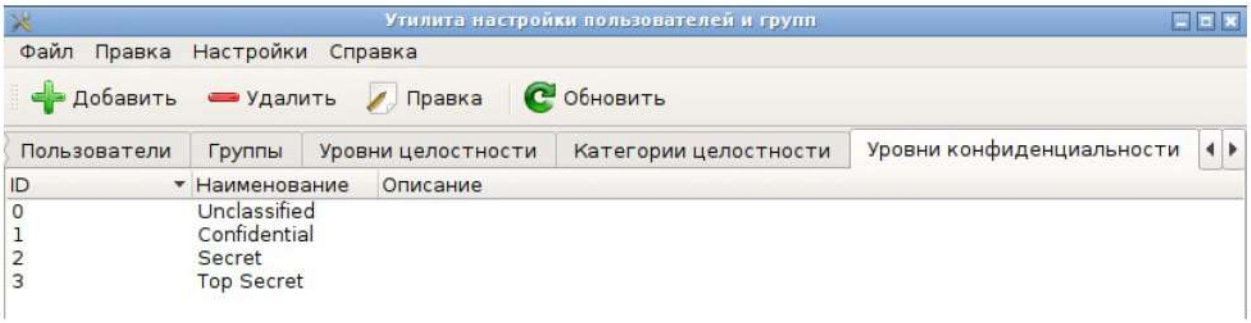


Рис. 36

Нажатие кнопки [Добавить] запускает диалог создания нового уровня конфиденциальности (рис. 37).

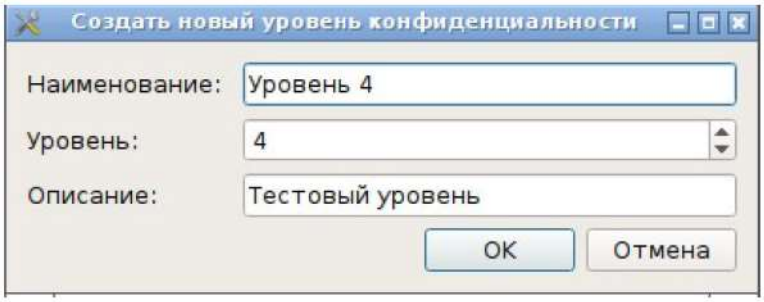


Рис. 37

В диалоге указывается наименование уровня (текстовое значение), его числовое значение и описание при необходимости.

Для редактирования наименования и описания уровня требуется выбрать желаемый уровень в списке уровней конфиденциальности и нажать кнопку [Правка].

Для удаления уровня конфиденциальности из ОС требуется выбрать уровень и нажать кнопку [Удалить] на панели инструментов, при этом будет запрошено подтверждение выполнения операции.

Для выполнения операций по добавлению, модификации, удалению категорий необходимо выбрать вкладку «Категории» главной экранной формы (рис. 38) и убедиться, что флаг «Разрешить редактирование мандатных уровней и категорий» поставлен в настройках программы.

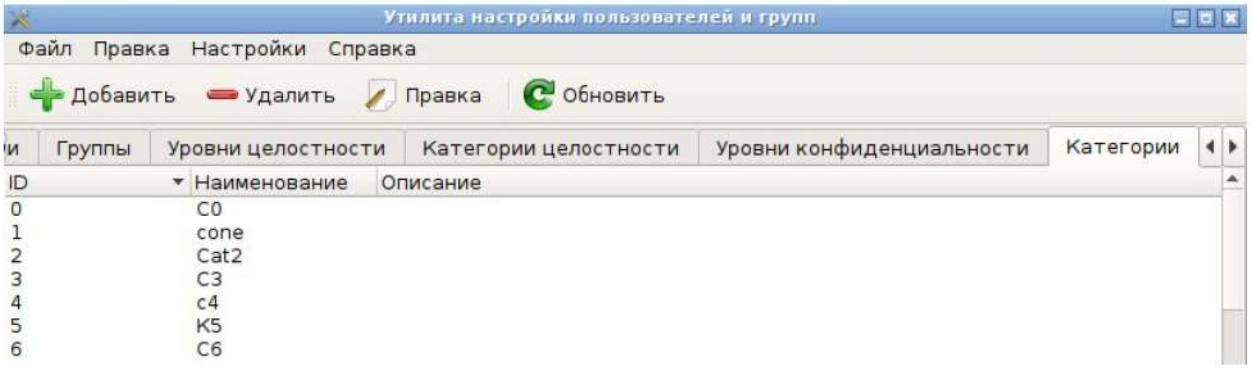


Рис. 38

Нажатие кнопки [Добавить] запускает диалог создания новой категории (рис. 39).

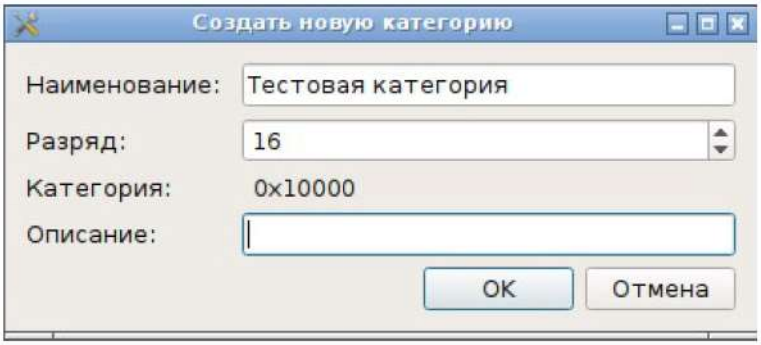


Рис. 39

В диалоге указывается наименование категории и ее разряд. При изменении номера разряда категории автоматически пересчитывается ее числовое значение. Дополнительно администратором может быть указано описание создаваемой категории.

Для редактирования наименования и описания категории требуется выбрать желаемую категорию в списке категорий и нажать кнопку [Правка].

Для удаления категории из ОС требуется выбрать категорию и нажать кнопку [Удалить] на панели инструментов, при этом будет запрошено подтверждение выполнения операции.

Для выполнения операций по регистрации, очистке и изменению свойств персональных идентификаторов GuardantID выбрать вкладку «GuardantID» главной экранной формы программы (рис. 40).

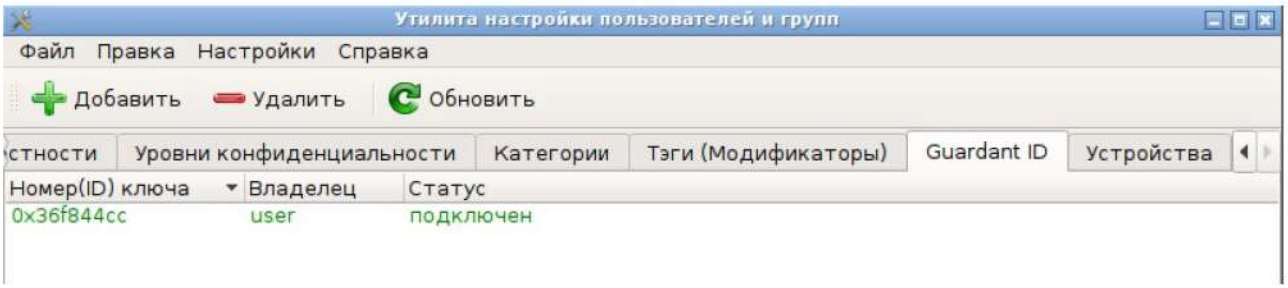


Рис. 40

При этом будет выведен список зарегистрированных (выданных) пользователям идентификаторов. Подключенный идентификатор подсвечивается цветом.

Для выполнения операции по регистрации нового персонального идентификатора необходимо подключить GuardantID и нажать кнопку [Добавить] на панели инструментов (рис. 41).

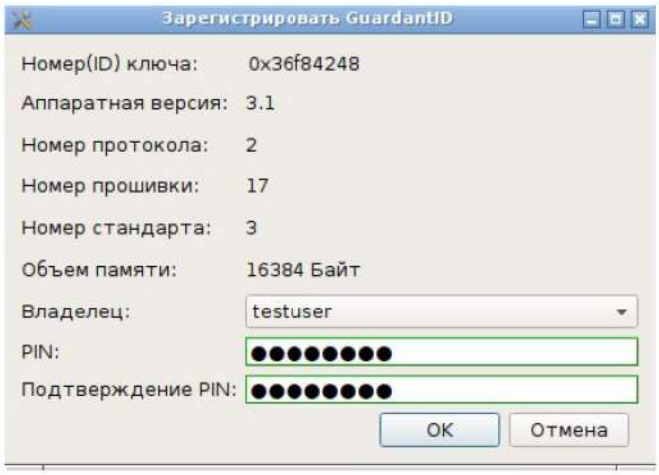


Рис. 41

В диалоге необходимо выбрать владельца с помощью выпадающего списка [Владелец], а также ввести в поля [PIN] и [Подтверждение PIN] желаемый PIN. При совпадении PIN и его подтверждения данные поля ввода подсвечиваются зеленым цветом, в противном случае – данные поля подсвечиваются красным.

Для очистки персонального идентификатора требуется выбрать персональный идентификатор в списке зарегистрированных и нажать на кнопку [Удалить] на панели инструментов. Если выбранный персональный идентификатор не был подключен, то будет выведен диалог очистки с информацией о необходимости подключения устройства (рис. 42).

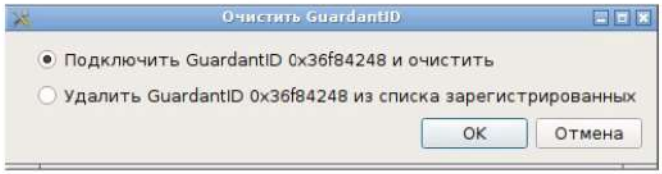


Рис. 42

Опция «Подключить GuardantID и очистить» требует подключения указанного идентификатора, после чего происходит его физическая очистка.

Опция «Удалить GuardantID из списка зарегистрированных» может быть применена, когда указанный идентификатор не может быть подключен (например, в случае его повреждения или утери). В данном случае информация о регистрации удаляется из системного файла /etc/grdid/grdid.conf.

В случае, когда выбранный персональный идентификатор уже подключен, выводится диалог подтверждения очистки персонального идентификатора.

Для выполнения операций по регистрации, отмены регистрации и изменению свойств регистрации USB-устройств требуется выбрать вкладку «Устройства» главной экранной формы программы (рис. 43).

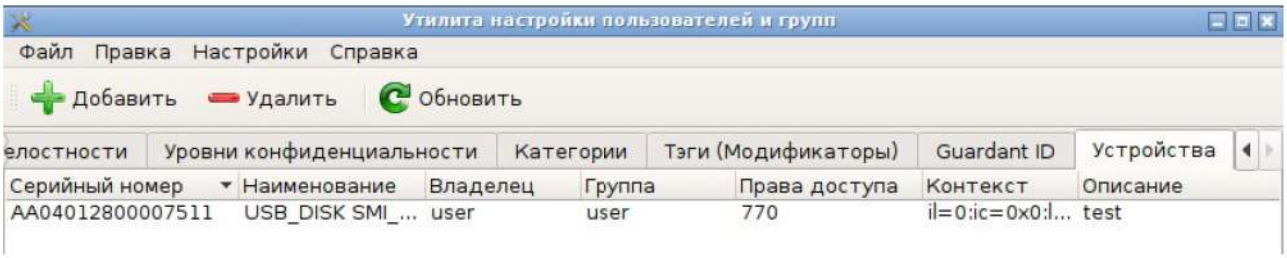


Рис. 43

При этом будет выведен список зарегистрированных (учтенных) USB-устройств.

Для выполнения операции по регистрации нового персонального идентификатора необходимо нажать кнопку [Добавить] на панели инструментов. Далее по запросу программы требуется подключить USB-устройства к ПК для их определения программой (рис. 44).

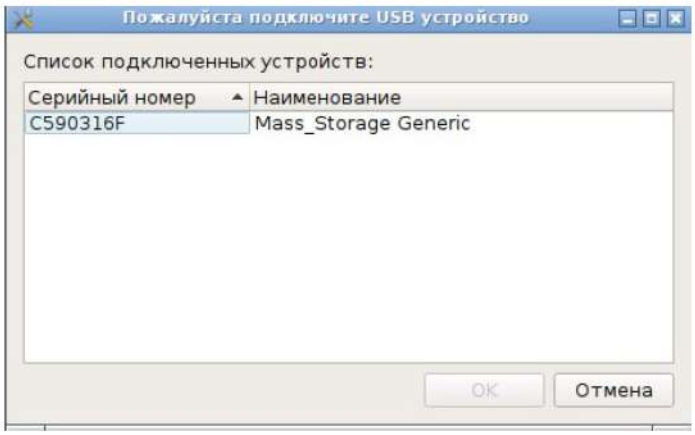


Рис. 44

Выбор устройства для регистрации подтверждается нажатием на кнопку [ОК]. При необходимости администратор может дополнительно выполнить операцию форматирования устройства в ФС ext4, после чего вызывается диалог регистрации выбранного устройства (рис. 45).

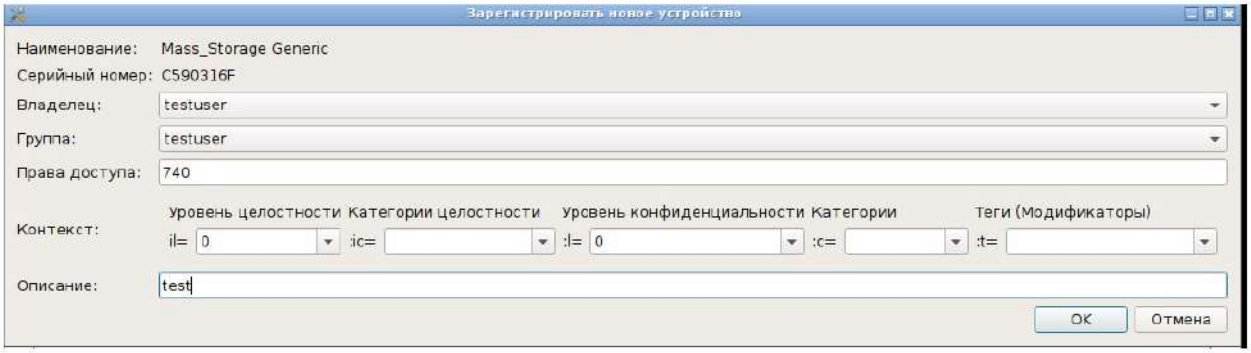


Рис. 45

Для редактирования доступны поля [Владелец], [Группа], [Права доступа] (задаются в числовом виде), [Контекст] и [Описание].

Для отмены регистрации требуется выбрать устройство в списке зарегистрированных и нажать на кнопку [Удалить] на панели инструментов, при этом будет запрошено подтверждение выполнения операции.

3.7. Управление сетевыми настройками и фильтрацией

3.7.1. Сетевое взаимодействие (TCP/IP)

В начале каждого пакета есть заголовок, в котором содержится служебная информация – IP-адреса отправителя и получателя, для которого предназначен данный пакет, к какому сетевому протоколу относится (TCP, UDP и т.п.), в некоторых случаях (для протоколов UDP и TCP) – номера портов отправителя и получателя, а также другая специфическая информация.

3.7.2. Фильтр сетевых пакетов

Задачей фильтра сетевых пакетов является фильтрация и обработка пакетов, проходящих через компьютер. При анализе входного пакета фильтр принимает решение о том, какое действие следует произвести для этого пакета: выбросить пакет, принять или выполнить другое действие.

Функции фильтра реализуются модулем ядра netfilter. Интерфейсом для управления правилами, по которым этот модуль обрабатывает пакеты, служит утилита iptables для IPv4 и утилита ip6tables для IPv6.

К основным возможностям iptables относится:

– фильтрация трафика на основе адресов отправителя и получателя пакетов с учетом портов;

– перенаправление пакетов по определенным параметрам;

– организация доступа в сеть (SNAT);

– проброс портов из глобальной сети в локальную (DNAT);

– ограничение числа подключений;

– установление квот трафика;

– выполнение правил по расписанию.

В настоящее время для удобства управления применяются различные средства более высокого уровня, например ufw, описанный в 3.7.5.

3.7.2.1. Порядок фильтрации и правила обработки пакетов

Сетевые пакеты, поступившие на сетевой интерфейс, проходят последовательность цепочек фильтра сетевых пакетов. Пакет обязательно проходит первоначальную цепочку PREROUTING. После цепочки PREROUTING, в соответствии с

таблицей маршрутизации, проверяется отправитель пакета и, в зависимости от назначения пакета, определяется, в какую цепочку он попадет дальше. Если пакет адресован не локальной системе (в TCP пакете поле «адрес получателя» указана не локальная система), то он направляется в цепочку FORWARD, если пакет адресован локальной системе – он отправляется в цепочку INPUT и после ее прохождения отдается локальным службам/процессам. После обработки локальной программой при необходимости формируется ответ. Ответный пакет направляется на соответствующий маршрут (хост из локальной сети или адрес маршрутизатора) и направляется в цепочку OUTPUT. После цепочки OUTPUT (или FORWARD, если пакет был проходящий) пакет снова сверяется с правилами маршрутизации и отправляется в цепочку POSTROUTING.

Каждая цепочка, которую проходит пакет, состоит из набора таблиц. Таблицы в разных цепочках имеют одинаковое наименование, но между собой не связаны. Например, таблица nat в цепочке PREROUTING не связана с таблицей nat в цепочке POSTROUTING. Каждая таблица состоит из упорядоченного набора правил. Каждое правило содержит условие, которому должен соответствовать проходящий пакет, и набор действий с пакетом, подходящим данному условию.

Проходя через серию цепочек, пакет последовательно проходит каждую таблицу и в каждой таблице последовательно сверяется с каждым набором условий в правиле, и если пакет соответствует какому-либо условию, то выполняется заданное действие над пакетом. В каждой таблице (кроме пользовательской) существует заданная по умолчанию политика. Данная политика определяет действие над пакетом, если пакет не соответствует ни одному из правил в таблице. Чаще всего — это действие ACCEPT, чтобы принять пакет и передать в следующую таблицу или DROP — чтобы отбросить пакет. Если пакет не был отброшен, он завершает свое путешествие по ядру системы и отправляется в сетевой интерфейс, который подходит по правилам маршрутизации.

Подсистема netfilter использует следующие цепочки таблиц:

– PREROUTING – для изначальной обработки входящих пакетов

– INPUT – для входящих пакетов, адресованных непосредственно локальному компьютеру

– FORWARD – для проходящих (маршрутизируемых) пакетов

– OUTPUT – для пакетов, создаваемых локальным компьютером (исходящих)

– POSTROUTING – для окончательной обработки исходящих пакетов

Существует возможность создавать и уничтожать собственные цепочки при помощи утилиты iptables.

Цепочки прохождения пакетов используют следующие таблицы:

– raw – пакет проходит данную таблицу до передачи системе определения состояний. Используется редко, например, для маркировки пакетов, которые НЕ должны обрабатываться системой определения состояний. Для этого в правиле указывается действие NOTRACK. Содержится в цепочках PREROUTING и OUTPUT;

– mangle – содержит правила модификации (обычно полей заголовка) IP-пакетов. Среди прочего поддерживает действия TTL, TOS, и MARK (для изменения полей TTL и TOS и для изменения маркеров пакета). Содержится во всех пяти стандартных цепочках;

– nat – предназначена для подмены адреса отправителя или получателя. Данную таблицу проходит только первый пакет из потока, трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT и POSTROUTING;

– filter – основная таблица, используется по умолчанию, если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD и OUTPUT.

Непосредственно для фильтрации пакетов используются таблицы filter. Для пакетов, предназначенных данному узлу, необходимо модифицировать таблицу filter цепочки INPUT, для проходящих пакетов – цепочки FORWARD, для пакетов, созданных данным узлом – OUTPUT. Цепочки и таблицы обработки пакетов приведены в таблице 13.

Таблица 13

Цепочка	Таблица		
	filter	nat	mangle
INPUT	+		+
FORWARD	+		+
OUTPUT	+	+	+
PREROUTING		+	+
POSTROUTING		+	+

3.7.2.2. Таблица mangle

Основное назначение таблицы mangle – внесение изменений в заголовок пакета. В этой таблице могут производиться следующие действия:

– установка бита Type Of Service;

– установка поля Time To Live;

– установка метки на пакет, которая может быть проверена в других правилах.

Цепочки в таблице mangle:

– PREROUTING – используется для внесения изменений в пакеты на входе в iptables, перед принятием решения о маршрутизации;

– POSTROUTING – используется для внесения изменений в пакеты на выходе из iptables, после принятия решения о маршрутизации;

– INPUT – используется для внесения изменений в пакеты, перед тем как они будут переданы локальному приложению;

– OUTPUT – используется для внесения изменений в пакеты, поступающие от приложения внутри iptables;

– FORWARD – используется для внесения изменений в транзитные пакеты.

3.7.2.3. Таблица nat

Таблица используется для преобразования сетевых адресов (Network Address Translation) и когда встречается пакет, устанавливающий новое соединение. В этой таблице могут производиться следующие действия:

– DNAT (Destination Network Address Translation) – преобразование адреса назначения в заголовке пакета;

– SNAT (Source Network Address Translation) –изменение исходного адреса пакета;

– MASQUERADE – используется в тех же целях, что и SNAT, но позволяет работать с динамическими IP-адресами.

Цепочки в этой таблице:

– PREROUTING – используется для внесения изменений в пакеты на входе в iptables;

– OUTPUT – используется для преобразования адресов в пакетах перед дальнейшей маршрутизацией;

– POSTROUTING – используется для преобразования пакетов перед отправкой их в сеть.

3.7.2.4. Таблица filter

Таблица используется для фильтрации пакетов. В этой таблице есть три цепочки:

– INPUT – цепочка для входящих пакетов;

– FORWARD – цепочка для пересылаемых (транзитных) пакетов;

– OUTPUT – цепочка для исходящих пакетов.

Пакет, проходящий через эти цепочки, может подвергаться действиям: ACCEPT, DROP, REJECT, LOG.

3.7.3. Управление фильтром сетевых пакетов (iptables)

Управление фильтром сетевых пакетов осуществляется путем загрузки в него

правил фильтрации. Далее описываются операции по сохранению и загрузке правил iptables.

ВНИМАНИЕ! Администратор должен обеспечить загрузку соответствующих правил при загрузке операционной системы с помощью доступных средств запуска служебных скриптов во время загрузки.

Примечание. Средство управления фильтром ufw (см. 3.7.5) самостоятельно обеспечивает автоматическую загрузку правил iptables, построенных с помощью упрощенного синтаксиса.

Сохранение правил:

```
$ sudo sh -c "iptables-save > /etc/iptables/iptables.rules"
```

Восстановление правил из файла:

```
$ iptables-restore < firewall-config
```

Каждое правило в iptables– это отдельная строка, сформированная по определенным правилам и содержащая критерии и действия. В общем виде правило имеет такой формат:

```
iptables [-t table] command [match] [target/jump]
```

– `t table`– задает имя таблицы, для которой будет создано правило;

– `command`– команда, которая определяет действие iptables – добавить правило, удалить правило и т. д.;

– `match`– задает критерии проверки, по которым определяется, попадает ли пакет под действие правила или нет;

– `target/jump`– какое действие должно быть выполнено при выполнении критерия.

Команды iptables:

– `-A`– добавление правила в цепочку, правило будет добавлено в конец цепочки;

– `-D`– удаление правила из цепочки;

– `-R`– заменить одно правило другим;

– `-I`– вставить новое правило в цепочку;

– `-L`– вывод списка правил в заданной цепочке;

– `-F`– сброс всех правил в заданной цепочке;

– `-Z`– обнуление всех счетчиков в заданной цепочке;

– `-N`– создание новой цепочки с заданным именем;

– `-X`– удаление цепочки;

– `-P`– задает политику по умолчанию для цепочки;

– -E – переименование пользовательской цепочки.

Для указания действия (цели) с пакетом служит опция -j. Основные действия:

– ACCEPT – разрешить пакет;

– DROP – уничтожить пакет;

– REJECT – будет отправлено ICMP сообщение, что порт недоступен;

– LOG – информация об этом пакете будет добавлена в системный журнал. Не прерывает цепочку;

– RETURN – возвращает пакет в ту цепочку, из которой он прибыл;

– SNAT – применить source NAT ко всем удовлетворяющим условию пакетам. Может использоваться только в цепочках POSTROUTING и OUTPUT таблицы NAT;

– DNAT – применить destination NAT ко всем удовлетворяющим условию пакетам. Может использоваться только в цепочке POSTROUTING таблицы NAT*;

– MASQUERADE – может применяться только в цепочке POSTROUTING таблицы NAT. Используется при наличии соединения с динамическим IP. Похож на SNAT, однако имеет свойство забывать про все активные соединения при потере интерфейса. Это полезно при наличии соединения, на котором время от времени меняется IP-адрес, но при наличии постоянного IP это только доставит неудобства. В том числе поэтому рекомендуется для статических IP использовать SNAT.

В качестве действия можно указать и имя пользовательской цепочки. Например, «перекинуть» все пакеты с локальной сети в цепочку, где будет производиться дополнительная проверка:

```
$ iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT
```

3.7.3.1. Примеры фильтрации сетевого уровня

Для фильтрации по источнику используется опция -s. Например, запретим 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
```

Также 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 хождение трафика по localhost:

```
$ iptables -A INPUT -i lo -j ACCEPT
```

Логируем попытки спуфинга с префиксом «IP_SPOOF A:» и завершаем 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
```

Для использования в качестве фильтрации адреса получателя 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:

```
$ iptables -A OUTPUT -d vk.com -j REJECT
```

Как и в случае с источником пакета можно использовать адреса подсети и доменные имена. Отрицание также работает.

3.7.3.2. Примеры фильтрации транспортного уровня

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
```

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
```

При указании порта необходимо указать транспортный 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
```

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 --stateNEW,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 -jACCEPT
```

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
```

Разрешить торенты:

```
$ iptables -A INPUT -p TCP -i eth0 --dport 51413 -j ACCEPT
$ iptables -A INPUT -p UDP -i eth0 --dport 51413 -j ACCEPT
$ iptables -A INPUT -p TCP -i eth0 --dport 6881 -j ACCEPT
$ iptables -A INPUT -p UDP -i eth0 --dport 6881 -j ACCEPT
```

Разрешить исходящий Google Talk:

```
$ iptables -A OUTPUT -p TCP -o eth0 --dport 5222 -j ACCEPT
```

Разрешить TeamViewer:

```
$ iptables -A OUTPUT -p UDP -o eth0 --dport 5938 -j ACCEPT
```

Проброс портов:

Для примера направим трафик с порта 442 на 22, это значит, что входящие ssh-соединения могут быть приняты 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
```

3.7.3.3. Пример базового набора правил

В большинстве случаев конечному пользователю (рабочая станция) достаточно выполнить такую последовательность команд:

```
$ iptables -P FORWARD DROP
$ iptables -P OUTPUT ACCEPT
$ iptables -A INPUT -i lo -j ACCEPT
$ iptables -A INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
$ iptables -P INPUT DROP
```

3.7.3.4. Расширение функциональности iptables подключаемыми модулями

В iptables имеется возможность подключать модули, для этого используется опция -m.

Модуль limit предназначен для ограничения нагрузки, например:

```
$ iptables -A INPUT -p icmp -m limit --limit 4/second -j ACCEPT
```

Разрешить поддерживать открытыми уже установленные соединения:

```
$ iptables -A OUTPUT -o eth0 -m state --state ESTABLISHED,RELATED
-j ACCEPT
```

3.7.3.5. Управление правилами

Список текущих правил:

```
iptables -nvL —line-numbers
```

где L – показать список правил;

v – отображать дополнительную информацию;

*n – отображать ip адрес и порт числами (не используя DNS сервера для определения имен. Это ускорит отображение);

line-numbers – вывод номеров строк.

Очистка всех правил:

```
$ iptables -F
```

Очистка правил в цепочке:

```
$ iptables -F INPUT
```

Удаления пятого правила в цепочке INPUT:

```
$ iptables -D INPUT 5
```

Удалить правило, в котором адрес источника (192.168.1.15)

```
$ iptables -D INPUT -s 192.168.1.15 -j DROP
```

3.7.4. Пример защиты от разных видов забивания полосы пропускания

3.7.4.1. ICMP-флуд

Очень примитивный метод забивания полосы пропускания и создания нагрузок на сетевой стек через монотонную посылку запросов ICMP ECHO (пинг). Легко обнаруживается с помощью анализа потока трафика в обе стороны: во время атаки типа ICMP-флуд они практически идентичны. Почти безболезненный способ абсолютной защиты основан на отключении ответов на запросы ICMP ECHO:

```
$ iptables -A INPUT -p icmp -j DROP --icmp-type 8
```

3.7.4.2. SYN-флуд

Один из распространенных способов не только забить канал связи, но и ввести сетевой стек операционной системы в такое состояние, когда он уже не сможет принимать новые запросы на подключение. Основан на попытке инициализации большого числа одновременных TCP-соединений через посылку SYN-пакета с несуществующим обратным адресом. После нескольких попыток отослать ответный ACK-пакет на недоступный адрес большинство ОС ставят неустановленное соединение в очередь. И только после n-ой попытки закрывают соединение. Так как поток ACK-пакетов очень велик, вскоре очередь оказывается заполненной, и ядро дает отказ на попытки открыть новое соединение. Наиболее умные DoS-боты еще и анализируют систему перед началом атаки, чтобы слать запросы только на открытые жизненно важные порты. Идентифицировать такую атаку просто: достаточно попробовать подключиться к одному из сервисов.

Оборонительные мероприятия обычно включают в себя:

Увеличение очереди «полуоткрытых» TCP-соединений:

```
$ sysctl -w net.ipv4.tcp_max_syn_backlog=1024
```

Уменьшение времени удержания «полуоткрытых» соединений:

```
$ sysctl -w net.ipv4.tcp_synack_retries=1
```

Включение механизма TCP syncookies:

```
$ sysctl -w net.ipv4.tcp_syncookies=1
```

Ограничение максимального числа «полуоткрытых» соединений с одного IP к конкретному порту:

```
$ iptables -I INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 10 -j DROP
```

3.7.4.3. UDP-флуд

Типичный метод забивания полосы пропускания. Основан на бесконечной посылке UDP-пакетов на порты различных UDP-сервисов. Легко устраняется за счет отрезания таких сервисов от внешнего мира и установки лимита на количество соединений в единицу времени к DNS-серверу на стороне шлюза:

```
$ iptables -I INPUT -p udp --dport 53 -j DROP -m connlimit --connlimit-above 1
```

3.7.4.4. HTTP-флуд.

Один из самых распространенных на сегодняшний день способов флуда. Основан на бесконечной посылке HTTP-сообщений GET на 80-ый порт с целью загрузить web-сервер настолько, чтобы он оказался не в состоянии обрабатывать все остальные запросы. Часто целью флуда становится не корень web-сервера, а один из скриптов, выполняющих ресурсоемкие задачи или работающий с БД. В любом случае, индикатором начавшейся атаки будет служить аномально быстрый рост логов web-сервера.

Определившись с IP виновника, начинаем удалять по IP-адресам:

```
$ iptables -A INPUT -s xxx.xxx.xxx.xxx -p tcp --destination-port http -j DROP
```

Или сразу по подсетям:

```
$ iptables -A INPUT -s xxx.xxx.0.0/16 -p tcp --destination-port http -j DROP
```

Для ограничения количества одновременных подключений к серверу для клиента по IP используется модуль connlimit. Ограничим количество параллельных подключений по SSH до трех для одного клиента:

```
$ iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT
```

Ограничить количество параллельных подключений по HTTP до трех для одного клиента:

```
$ iptables -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 --connlimit-mask 24 -j DROP
```

где: connlimit-above 10 – условие для проверки одновременных подключений не более 10;

connlimit-mask 24 – группировка хостов по длине префикса, иначе говоря «маска» (для IPv4 это число должно быть в диапазоне от 0 до 32 включительно).

Также ограничить количество подключений в единицу времени можно с помощью модуля limit.

```
$ iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute
--limit-burst 100 -j ACCEPT
```

где: m limit – подключаем модуль limit;

limit 25/minute – порог в 25 подключений в мин.;

limit-burst 100 – условие включения порога: после достижения 100 подключений.

3.7.5. Управление фильтром сетевых пакетов (ufw)

По умолчанию для управления правилами фильтрации в системе используется простой командный интерфейс ufw. Он разработан для легкой настройки iptables и предоставляет дружественный способ создания сетевой защиты для IPv4 и IPv6. Упрощенные правила ufw транслируются в расширенные правила iptables и применяются при загрузке системы автоматически (в случае включенного по умолчанию ufw).

Изначально сервис ufw выключен.

Простота использования ufm может быть показана на следующих примерах:

Включение сетевого фильтра:

```
$ sudo ufw enable
```

Открытие порта (в данном примере SSH):

```
$ sudo ufw allow 22
```

Правила могут быть добавлены с использованием нумерованного формата:

```
$ sudo ufw insert 1 allow 80
```

Подобным образом можно закрыть открытый порт:

```
$ sudo ufw deny 22
```

Для удаления правила используется команда delete:

```
$ sudo ufw delete deny 22
```

Также можно разрешить доступ к порту с определенных компьютеров или сетей. Следующий пример разрешает на этом компьютере доступ по SSH с адреса 192.168.0.2 на любой IP адрес:

```
sudo ufw allow proto tcp from 192.168.0.2 to any port 22
```

Добавление опции --dry-run команде ufw выведет список правил, но не применит их. Например, далее показано, что было бы применено, если открыть HTTP порт:

```
sudo ufw --dry-run allow http
```

```
*filter
:ufw-user-input - [0:0]
:ufw-user-output - [0:0]
:ufw-user-forward - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
### RULES ###

### tuple ### allow tcp 80 0.0.0.0/0 any 0.0.0.0/0
-A ufw-user-input -p tcp --dport 80 -j ACCEPT

### END RULES ###
-A ufw-user-input -j RETURN
-A ufw-user-output -j RETURN
-A ufw-user-forward -j RETURN
-A ufw-user-limit -m limit --limit 3/minute -j LOG --log-prefix
"[UFW LIMIT]: "
-A ufw-user-limit -j REJECT
-A ufw-user-limit-accept -j ACCEPT
COMMIT
Rules updated
```

ufw можно выключить командой:

```
sudo ufw disable
```

Чтобы посмотреть статус сетевой защиты:

```
sudo ufw status
```

Для более полного отображения информации следует ввести:

```
sudo ufw status verbose
```

Для отображения в виде формата numbered:

```
sudo ufw status numbered
```

Для управления правилами и состоянием ufw в системе используется графическая утилита ufw-gtk (пункт основного меню «Параметры/Управление межсетевым экраном»).

На рис. 46 приведено основное окно программы, позволяющее выполнять запуск/останов межсетевого экрана и управление составом правил фильтрации с использованием графического интефейса задания правил, приведенного на рис. 47.

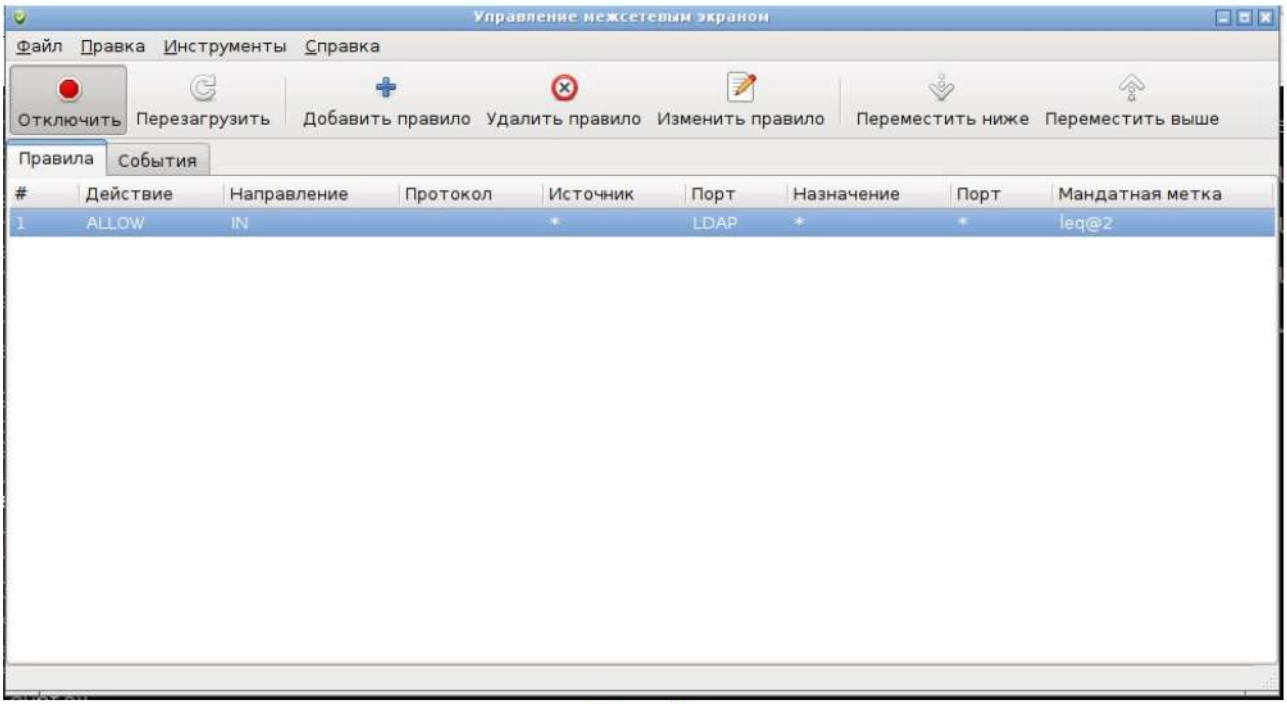


Рис. 46

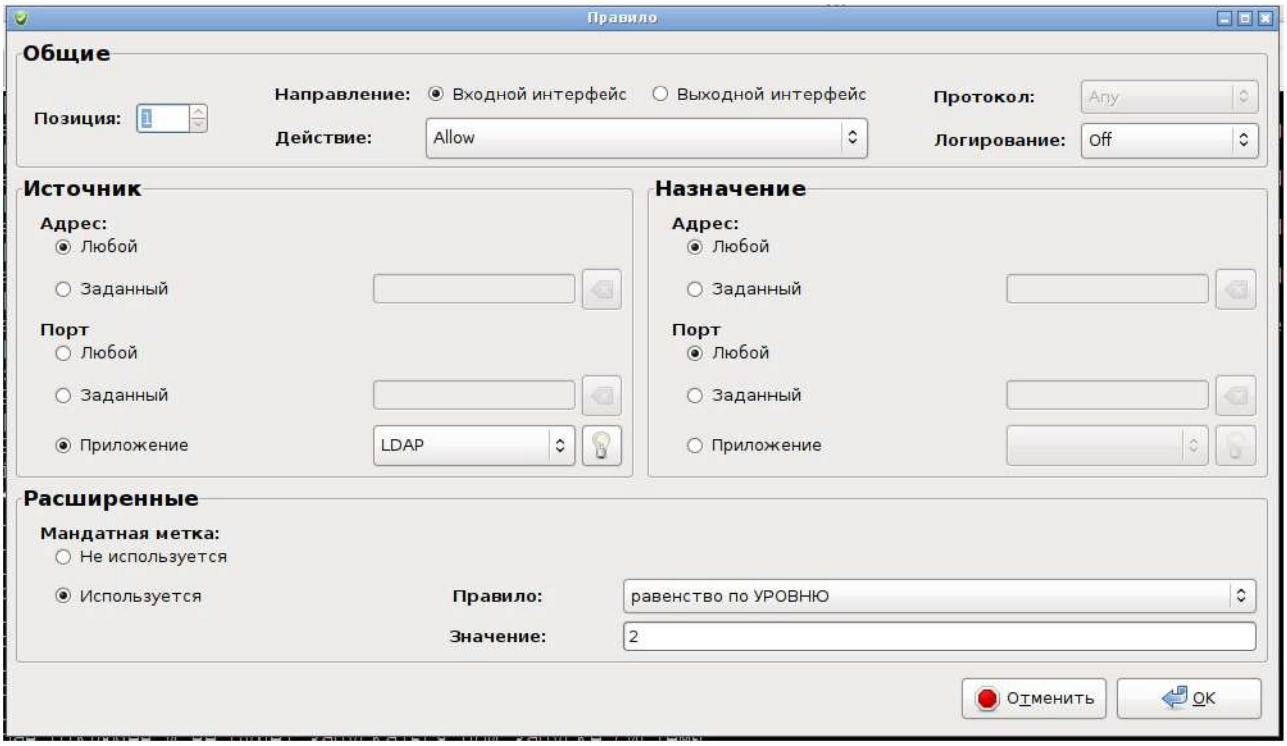


Рис. 47

3.8. Управление средствами регистраций событий

Подсистема регистрации событий представляет собой службу (auditd), фиксирующую события системных вызовов ядра ОС и классифицирующих их в зависимости от контекста появления:

– task – события, связанные с созданием новых процессов;

– entry – события, которые имеют место при входе в системный вызов;

– exit – события, которые имеют место при выходе из системного вызова;

– user – события, использующие параметры пользовательского пространства.

Каждое событие представляет собой набор значений вида ключ-значение, где ключом могут выступать различные поля. В состав обязательной информации входят следующие поля:

– надежная метка времени;

– тип события;

– результат события (успех или отказ).

Состав полей события может отличаться в зависимости от его типа, например, при регистрации события входа пользователя (USER_LOGIN) присутствуют поля идентификатора процесса (pid), номер терминала (tty) и др.

Данная служба устанавливается следующей командой:

```
$ sudo apt-get install auditd
```

В состав пакета auditd входят следующие утилиты:

– служба ведения событий аудита – auditd;

– управление правилами аудита – auditctl;

– просмотр отчетов о событиях аудита – aureport;

– поиск событий безопасности – ausearch.

3.8.1. Служба ведения событий аудита – auditd

Служба ведения событий аудита auditd представляет собой службу, осуществляющую запись событий безопасности в лог файл согласно правилам аудита.

Данная служба не запускается автоматически. Для ее запуска при старте ОС требуется добавить данную службу в список автозагрузки:

```
$ sudo systemctl enable auditd
```

Для запуска службы используется следующая команда:

```
$ sudo systemctl start auditd
```

Для остановки и перезапуска необходимо выполнить:

```
$ sudo systemctl stop auditd
$ sudo systemctl restart auditd
```

Служба auditd использует следующие файлы:

– /etc/audit/auditd.conf – конфигурационный файл службы аудита;

– /etc/audit/audit.rules – файл правил, применяемых при запуске службы;

– /etc/audit/rules.d – каталог, содержащий файлы с правилом или наборов правил;

– /var/log/audit/audit.log –журнал службы auditd.

3.8.2. Управление правилами аудита – auditctl

Для управления правилами аудита используется утилита auditctl, имеющая следующий формат вызова: auditctl [опции].

Утилита имеет большое число опций, основные из которых:

– l – вывести список основных правил;

– a – добавить новое правило;

– d – удалить правило;

– D – удалить все имеющиеся правила.

Для просмотра списка применяемых правил используется следующая команда:

```
$ sudo auditctl -l
```

Для добавления нового правила используется команда вида:

```
$ sudo auditctl -a <список>, <действие> -S <имя системного вызова> -F <фильтры>
```

Опции выбора списка при управлении правилами приведены в таблице 14

Таблица 14

Список	Описание
task	Список правил процесса. Этот список правил используется только во время создания процесса – когда родительский процесс вызывает `fork()` или `clone()`
entry	Список правил точек входа системных вызовов. Этот список применяется, когда необходимо создать событие для аудита, привязанное к точкам входа системных вызовов.
exit	Список правил точек выхода из системных вызовов. Этот список применяется, когда необходимо создать событие для аудита, привязанное к точкам выхода из системных вызовов.
user	Список фильтрации пользовательских сообщений. Этот список используется ядром, чтобы отфильтровать события, приходящие из пользовательского пространства, перед тем как они будут переданы демону аудита.
exclude	Список фильтрации событий определенного типа. Этот список используется, чтобы отфильтровывать ненужные события. Например, если вы не хотите видеть avc сообщения, вы должны использовать этот список. Тип сообщения задается в поле msgtype.

Список доступных действий приведен в таблице 15.

Таблица 15

Действие	Описание

Действие	Описание
never	Не генерировать никаких записей. Это может быть использовано для подавления генерации событий.
always	Установить контекст аудита. Всегда заполнять его во время входа в системный вызов и всегда генерировать запись во время выхода из системного вызова.

В качестве `имени системного вызова` может быть использовано как его имя, так и его номер.

Указание фильтров позволяет определить случаи фиксации событий аудита. Общий синтаксис для указания фильтров:

```
<поле><операция><значение>
```

где `<операция>` является логическим оператором: `=, !=, <, >, <=, >=, &, &=`

Например, для задания фильтра для директории /home используется следующая строка:

```
dir=/home
```

Список доступных атрибутов для задания фильтров указан в man auditctl.

Для удаления существующего правила используется следующая команда:

```
$ sudo auditctl -d <список>, <действие>
```

Например, правило на отслеживание системных вызовов open создается следующим образом:

```
$ sudo auditctl -a exit,always -S open
```

Создание правила для контроля создания суперпользователем файлов в каталоге /home:

```
$ sudo auditctl -a always,exit -F dir=/home/ -F uid=0 -C
auid!=obj_uid
```

Для удаления предыдущего правила используется команда:

```
$ sudo auditctl -d always,exit -F dir=/home/ -F uid=0 -C
auid!=obj_uid
```

3.8.3. Просмотр отчетов о событиях аудита – aureport

Утилита aureport используется для просмотра отчетов безопасности, зарегистрированных службой auditd. При запуске без аргументов выводит общую статистику по событиям безопасности. Опции выбора типа отчета утилиты aureport представлены в таблице 16.

Таблица 16

Опция	Название отчета
-a, --avc	Отчет AVC
-au, -auth	События использования механизма аутентификации
-cr, --crypto	События криптографии
-e, --event	Все сообщения
-h, --host	События удаленного хоста
-l, --login	События авторизации пользователей
-m, --mods	События изменения аккаунтов
-p, --pid	События по процессам
-u, --user	События пользователей

Окончание таблицы 16

Опция	Название отчета
-x, -- executable	События запуска программ

Для сортировки событий по типу результата (успех, отказ) используются ключи командной строки --success, --failure. Для задания интервала времени для отчета используются ключи --start, --end (задаются в формате ДД.ММ.ГГГГ).

Например, просмотр событий использования механизма аутентификации:

```
$ sudo aureport -au
```

Просмотр событий изменения учетных записей пользователей за декабрь 2017 года:

```
$ sudo aureport -m --start 01.12.2017 --end 31.12.2017
```

Подробное описание утилиты aureport приведено в man aureport.

3.8.4. Поиск событий безопасности – ausearch

Для просмотра детальной информации о событии используется утилита ausearch:

$ sudo ausearch -a <номер события> [опции]

Сообщения аудита выводятся в формате списка ключ-значение. Для поиска событий безопасности задаются условия поиска с помощью одного или нескольких ключей из таблицы 17.

Таблица 17

Опция	Описание
-a, --event <номер события аудита>	Поиск по номеру события
--arch <архитектура>	Поиск по архитектуре процессора
-c, --comm <команда>	Поиск по исполняемой команде
-e, --exit <код ошибки>	Поиск по коду возврата или номера ошибки

Опция	Описание
-f, --file <файл>	Поиск по имени файла
-gi, --gid <идентификатор>	Поиск по идентификатору группы
-hn, --host <хост>	Поиск по имени хоста
-p, --pid <идентификатор>	Поиск по идентификатору процесса
-sc, --syscall <вызов>	Поиск по названию системного вызова
-ue, --uid-effective <эффективный UID>	Поиск по эффективному идентификатору пользователя
-ui, --uid <UID>	Поиск по UID

Окончание таблицы 17

Опция	Описание
-ul, --loginuid <идентификатор логина>	Поиск по идентификатору логина пользователя
-x, --executable <исполняемый файл>	Поиск по исполняемому файлу

Аналогично команде aureport для поиска по заданному интервалу времени используются опции --start и --end.

В рассматриваемой подсистеме представлено большое число типов сообщений, список которых может быть получен запуском утилиты с опцией -m без аргументов. Указанная опция с аргументом (типом сообщения) может быть использована для отбора сообщений с заданным типом.

Примеры:

Поиск событий использования механизма повышения привилегий sudo:

```
$ sudo ausearch -x /usr/bin/sudo
```

Поиск событий входа пользователя (LOGIN) за декабрь 2017:

```
$ sudo ausearch -m LOGIN --start 01.12.2017 --end 31.12.2017
```

3.8.5. Графическая утилита управления аудитом – maudit

Для просмотра событий безопасности и управления правилами ведения журнала безопасности используется графическая утилита maudit. Данная утилита входит в состав пакета maudit.

Для ее запуска требуется выбрать пункт «Аудит» в панели запуска или запустить из командной строки командой:

```
$ maudit-runner
```

Основная экранная форма приложения представлена на рис. 48. На ней представлена статистическая информация по событиям аудита, зарегистрированным на

текущий момент.

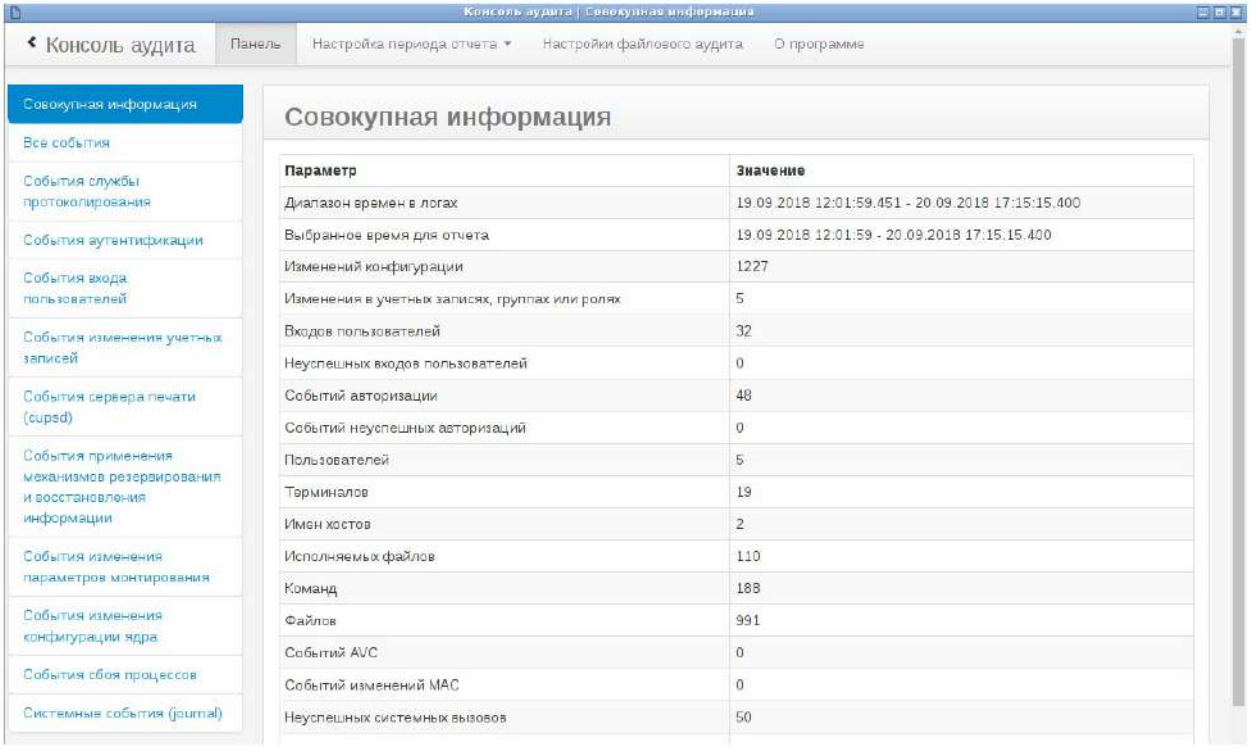


Рис. 48

В левой части представлена панель выбора типа отчета по категориям. При выборе конкретной категории в правой части отображается список с основными свойствами событий (дата, время, субъект, объект, операция и статус операции). Например, категория «Все события» (рис. 49) содержит список всех событий безопасности, а категория «События входа пользователей» – зарегистрированные события входа пользователей.

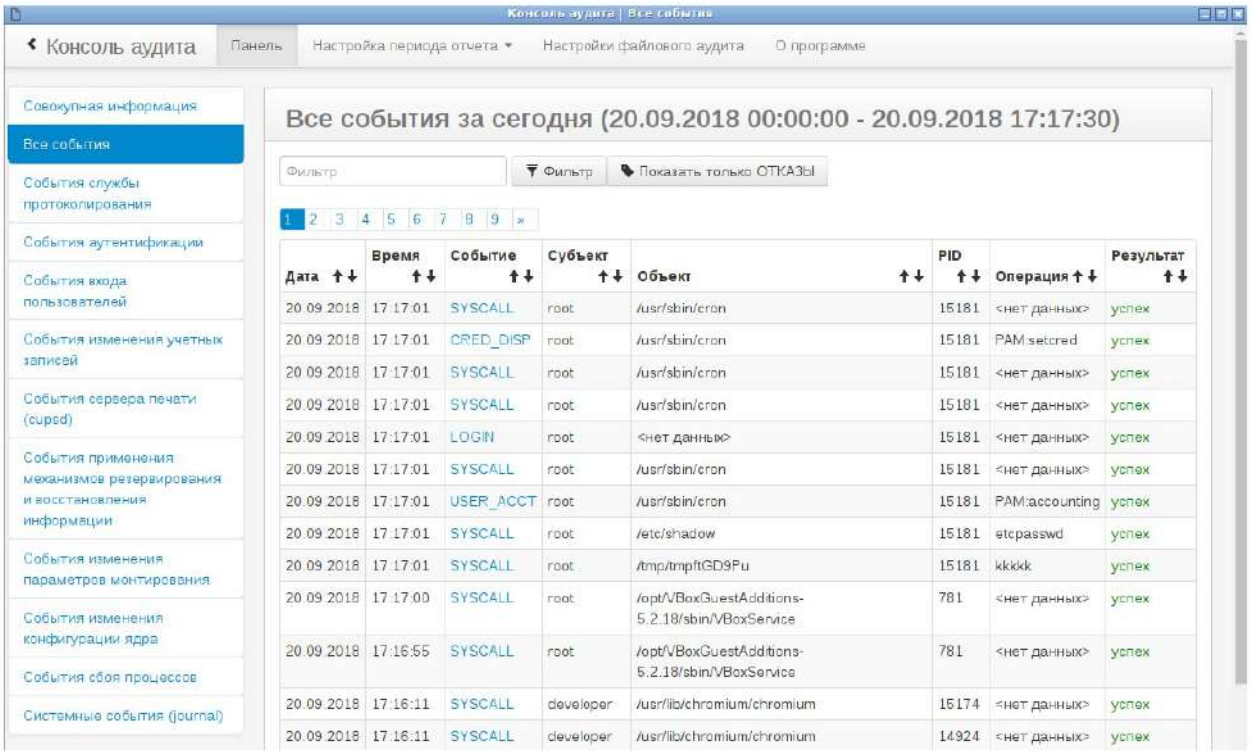


Рис. 49

Нажатие на тип события в столбце «Тип события» позволяет посмотреть понятную для администратора информацию о событии (рис. 50).

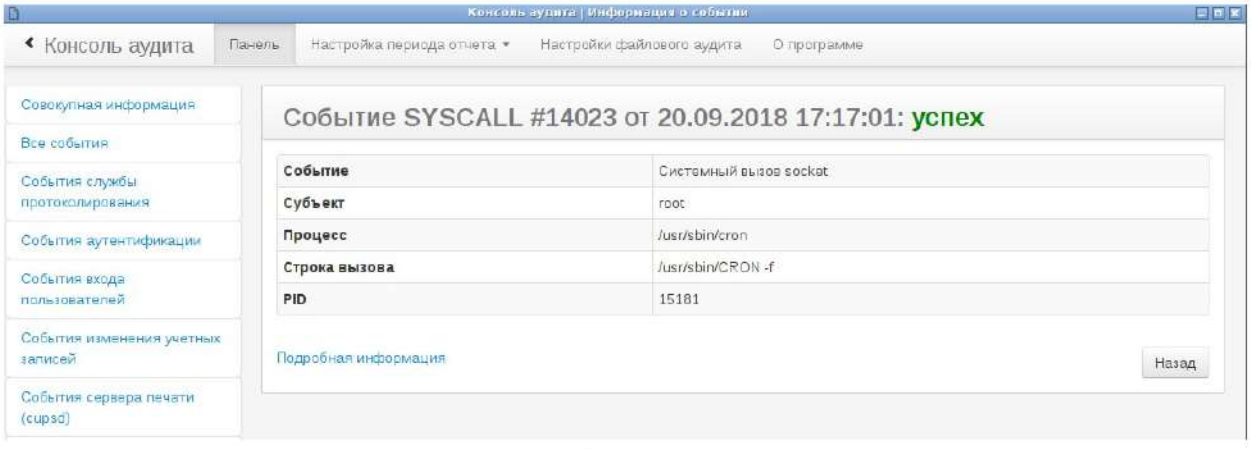


Рис. 50

Нажатие на ссылку «Подробная информация» (рис. 51) выводит все атрибуты события.

Параметр	Значение
Дата	20.09.2018
Время	17:17:01
fsuid	root
ses	83
uid	root
syscall	socket
pid	15181
sgid	root
serial	14023
auid	root
suid	root
gid	root

Рис. 51

Для отображения событий безопасности за выбранный администратором период времени используются пункты меню «Настройка периода отчета»:

– Сегодня — события с 00.00 текущего дня до настоящего времени;

– Последние 24 часа — события за 24 ч до настоящего времени;

– Текущая неделя — события с начала недели до настоящего дня;

– Выбранный день — позволяет указать день, за который будут отражены события;

– Выбранный период — позволяет указать диапазон дат, за которые будут отражены события.

Дополнительно предоставляется возможность событий со статусом «ОТКАЗ», для этого нажать кнопку [Показать только ОТКАЗЫ]. Нажатие кнопки [Показать все события] приведет к сбросу данного фильтра.

Для активизации контекстного фильтра желаемая строка фильтрации вводится в поле ввода «Фильтр» и выбор подтверждается нажатием кнопки [Фильтр]. Для сброса контекстного фильтра используется кнопка [Сбросить фильтр].

Настройка файлового аудита производится нажатием на пункт меню «Настройка файлового аудита» (рис. 52).

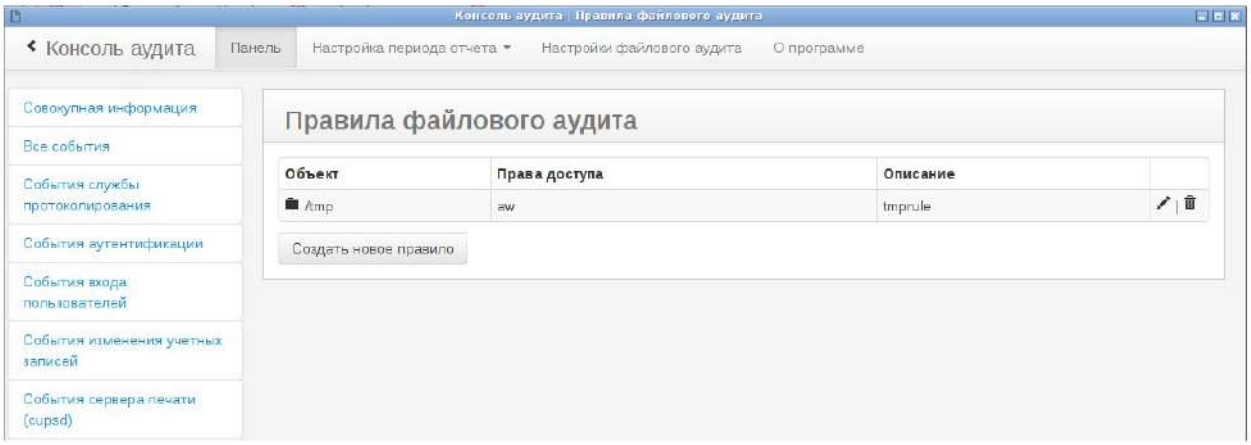


Рис. 52

На странице приводится список текущих правил файлового аудита, отображается путь файловой системы, права доступа и ключ-описание для фильтрации событий.

Для создания нового правила требуется нажать на кнопку [Создать новое правило]. Далее в диалоге выбирается путь файловой системы для аудита, в следующем окне – указываются права доступа и описание правила (рис. 53).

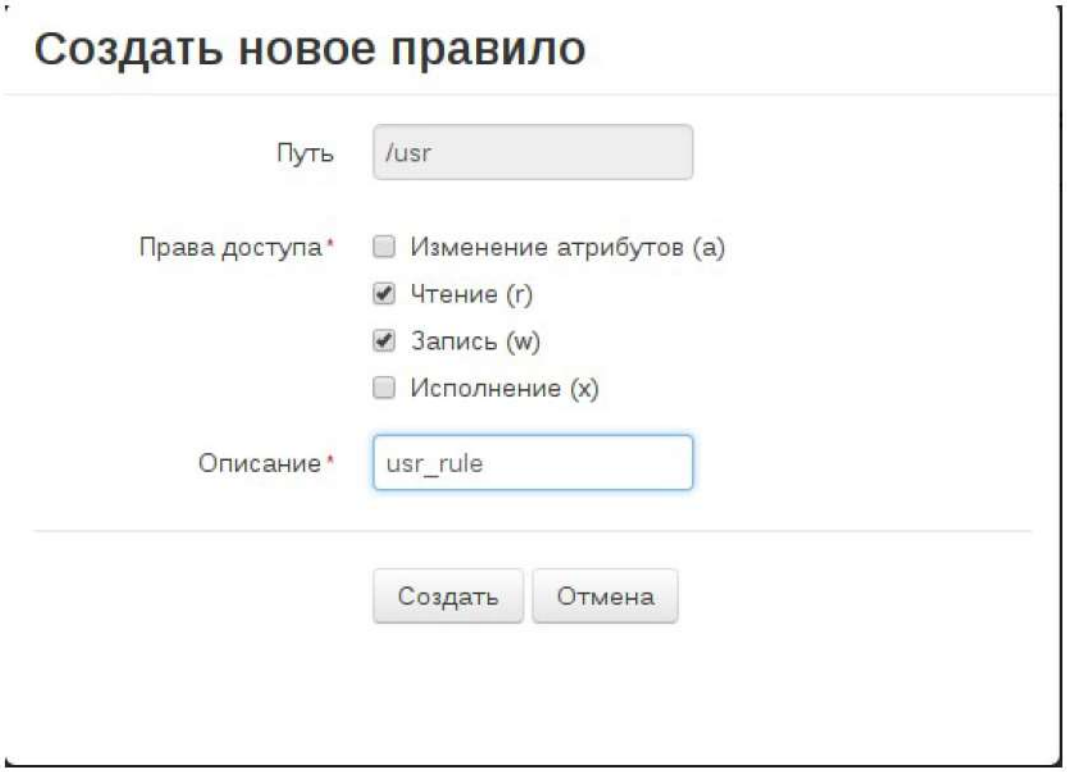


Рис. 53

Для изменения прав доступа или описания правила требуется нажать на кнопку [Изменить правило] в списке правил.

Для удаления правила используется кнопка [Удалить правило]. Для удаления правила требуется подтвердить свое действие в сплывающем диалоге.

3.9. Управление сервером печати (CUPS)

CUPS (Common UNIX Printing System) – модульная система печати для UNIX-подобных операционных систем. Компьютер с запущенным сервером CUPS представляет собой сетевой узел, который принимает задания на печать от клиентов, обрабатывает их и отправляет на соответствующий принтер.

В ОС подсистема печати CUPS обеспечивает выполнение следующих задач:

– регистрация и управление устройствами печати (очередями, принтерами и т. п.);

– предоставление информации о состоянии устройств печати локальным и удаленным программам;

– управление заданиями, выдаваемыми на печать;

– выполнение команд администратора печати;

– маркировка выводимых на печать документов в соответствии с требованиями РД для документов, подлежащих маркировке.

Операции по установке и первоначальной настройке подсистемы печати выполняются с привилегиями суперпользователя. Штатный режим администрирования подразумевает выполнение операций по добавлению и удалению принтеров, изменению их политики и мандатных атрибутов от имени пользователя, входящего в группу администраторов печати, указанную в значении параметра SystemGroup (по умолчанию lpadmin) в файле /etc/cups/cups-files.conf.

Операции по маркировке выполняются от имени пользователя, входящего в специальную группу, указанную в значении параметра MarkGroup (по умолчанию lpmark) в файле /etc/cups/cups-files.conf.

Дополнительная информация о настройке сервера CUPS находится в документе ФЛИР.90001-01 93 01 «ОС ОН «Стрелец». Пояснительная записка».

Примечание. Операции по маркировке выполняются от имени пользователя, входящего в специальную группу (по умолчанию marker).

3.9.1. Архитектура и принципы функционирования

Состав CUPS:

– диспетчер печати (scheduler);

– система фильтрации, преобразующая данные печати в формат, поддерживаемый конкретным принтером (mime-filters);

– система взаимодействия с оборудованием, отправляющая данные на устройства печати (backends).

3.9.1.1. Диспетчер печати

Диспетчер печати CUPS предоставляет веб-интерфейс для управления заданиями печати, конфигурирования сервера и предоставления справочной документации с помощью протокол IPP (Internet Printing Protocol), в основе которого использован протокол HTTP 1.1.

По умолчанию используется TCP-порт 631.

Модуль авторизации контролирует, какие запросы могут обрабатываться в системе. После прохождения авторизации запрос пересылается клиентскому модулю для обработки.

Клиентский модуль также отвечает за исполнение внешних CGI-программ, нужных для поддержки сетевых принтеров, классов и мониторинга состояний заданий печати.

Модуль конфигурации считывает конфигурационные файлы, инициализирует структуры данных CUPS, управляет запуском и остановкой вспомогательных программ CUPS. Модуль конфигурации также приостанавливает службы CUPS в течение обработки файлов конфигурации и возобновляет их работу после обновления параметров.

Модуль ведения журналов осуществляет запись событий, возникающих в процессе работы. Доступны следующие виды журналов:

– журнал заданий (access);

– журнал информации о функционировании и возможных ошибках (error);

– журнал распечатанных страниц (page).

MIME-модуль обрабатывает протокол MIME (Multipurpose Internet Mail Extensions), используется в системе фильтрации для преобразования различных типов данных в формат, поддерживаемый устройством печати.

PPD-модуль обрабатывает список файлов описания принтеров PPD (Postscript Printer Description).

Модуль устройств обрабатывает список устройств печати, доступных в системе.

Модуль принтеров, который работает с принтерами и PPD-файлами внутри CUPS.

3.9.1.2. Система фильтрации

Система печати CUPS позволяет отправлять различные данные на сервер CUPS, где они преобразуются в формат, поддерживаемый используемым принтером.

Это осуществляется с помощью последовательного применения различных фильтров. Для идентификации форматов данных и правил работы с ними используются следующие файлы конфигурации: mime.types, в котором определены известные типы данных MIME, поддерживаемые системой CUPS, и mime.convs, в котором определены правила преобразования этих типов.

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. Как правило такие программы используют единый формат опций вызова: имя принтера, идентификатор задания, имя пользователя, имя задания, число копий и параметры задания печати. Передача информации осуществляется с помощью стандартных потоков ввода/вывода.

3.9.1.3. Система взаимодействия с оборудованием

Система взаимодействия с оборудованием определяет способ, с помощью которого данные отправляются на конкретный принтер. Этот интерфейс взаимодействия включает в себя параллельные, последовательные и USB порты, виртуальный принтер cups-pdf для печати в файл формата PDF, сетевые принтеры, которые работают через протоколы IPP, JetDirect (AppSocket), Line Printer Daemon (LPD) и SMB.

3.9.1.4. Рабочие файлы и каталоги

В системе CUPS используются следующие файлы и рабочие каталоги (таблица 18).

Таблица 18

Путь	Описание
/etc/cups/	Файлы параметров конфигурации сервера печати
./cupsd.conf	Основные параметры сервера печати и политики
./cups-files.conf	Дополнительные параметры сервера печати
./printers.conf	Описания используемых в системе печати принтеров и их параметры
./subscriptions.conf	Список подписок
./client.conf	Общие параметры подключения клиента по сети к серверу печати
/etc/cups/ppd	Описания принтеров PPD
/etc/cups/ssl	Сертификаты и ключи для доступа по протоколу SSL
/var/spool/cups/	Файлы заданий печати
./aXXXXX	Данные аутентификации пользователя для задания XXXXX
./cXXXXX	Атрибуты задания XXXXX

Окончание таблицы 18

Путь	Описание
./dXXXXXX-NNN	Файл задания XXXX файла NNN
/var/spool/cups/tmp/	Временный каталог для фильтров
/var/cache/cups/	Каталог, содержащий оперативную информацию о заданиях печати
/var/log/cups/	Журналы работы сервера печати

./error_log	Основной журнал работы, содержит информацию о событиях, происходящих на сервере печати
./access_log	Журнал заданий, содержит информацию о состояниях операций с сервером печати
./page_log	Журнал страниц, который содержит информацию о распечатанных страницах

3.9.1.5. Политики контроля операций

В системе CUPS для проверки доступа при выполнении различных операций применяются различные политики контроля, которые могут быть назначены как для сервера печати, так и для конкретного принтера.

Описания этих политик содержатся в основном конфигурационном файле /etc/cups/cupsd.conf.

Изначально доступны политики по умолчанию с именами default, authenticated, kerberos и NESS.

3.9.1.6. Очереди печати

Работа с отправляемыми на печать заданиями реализуется с помощью очередей печати — механизм организации (определения порядка) и буферизации заданий, выводимых на печать. Данный механизм необходим по причине низкой скорости обработки данных устройствами печати и для обеспечения конкурентного доступа к ним со стороны пользователей. Для хранения очередей заданий используется каталог /var/spool/cups/.

В качестве очередей печати могут выступать поименованные устройства печати (принтеры) или группы принтеров. При печати данные сначала формируются на сервере печати, после чего посылаются на подключенный к нему или зарегистрированный на нем сетевой принтер.

Состояние очереди печати может предоставляться пользователю для отображения хода процесса печати.

3.9.2. Администрирование

Управление подсистемой печати CUPS включает следующие административные действия:

– изменение параметров конфигурации подсистемы печати cupsd.conf;

– просмотр списка очередей печати (группы принтеров, принтеры) и управление ими;

– просмотр списка текущих заданий и управления ими.

Для изменения параметров конфигурации подсистемы печати требуется открыть

файл /etc/cups/cupsd.conf с правами администратора текстовым редактором, выполнить редактирование настроек, сохранить файл и перезапустить подсистему печати:

```
$ sudo systemctl restart cups
```

3.9.2.1. Команды управления печатью

Для управления подсистемой печати используются следующие системные команды:

– `lpadmin` – управление принтерами и классами принтеров;

– `lp, lpr` – постановка заданий в очередь;

– `lpc, lpstat` – проверка текущего состояния очередей печати и планировщика;

– `lpinfo` – просмотр доступных устройств и драйверов;

– `lpmove` – перемещение заданий между очередями печати;

– `lprm` – отмена заданий, поставленных в очередь на печать;

– `lpq` – просмотр очередей печати.

Управление принтерами (добавление и настройка) может быть выполнена с помощью команды `lpadmin`:

```
$ sudo lpadmin -p printer [опции]
```

Основные опции команды lpadmin приведены в таблице 19.

Таблица 19

Опция	Описание
-c class	Добавляет названный принтер к классу принтеров class. Если класс не существует, то он создается
-m model	Задает стандартный драйвер принтера, обычно файл PPD. Файлы PPD обычно хранятся в каталоге /usr/share/cups/model/. Список доступных моделей можно вывести командой lpinf
-o param=value	Задание опций и параметров принтера

Окончание таблицы 19

Опция	Описание
-r class	Удаляет указанный принтер из класса class. Если в результате класс становится пустым, он удаляется
-u rule	Управление доступом пользователей к принтеру
-v device-uri	Указывает адрес устройства (URI) для связи с принтером. Список доступных адресов (URI) можно вывести командой lpinfo
-D info	Выдает текстовое описание принтера
-E	Разрешает использование принтера и включает прием заданий
-L location	Выводит расположение принтера

-P ppd-file	Указывает локальный файл PPD для драйвера принтера

Для отправки файлов на печать и управления заданиями печати предназначены команды `lp` и `lpr`. Предоставляют схожий набор опций и позволяют отправлять на печать файлы с указанием необходимых параметров задания печати: пользователя, имени, количества копий и т. п.

Для просмотра текущего состояния очередей печати и планировщика предназначены команды `lpc`, `lpstat` и `lpq`. Наиболее простой командой является команда просмотра заданий печати `lpq`, другие команды предоставляют большой набор опций для просмотра статуса принтеров и очередей печати с возможностью отбора по заданным критериям.

Отмена задания из очереди печати выполняется командой `lprm` с указанием идентификатора задания, который может быть определен, например, с помощью команды lpq. Без указания аргумента отменяется текущее задание.

С помощью команды `lpmove` задание может быть перенесено в очередь печати другого принтера.

Команда `lpinfo` позволяет получить список доступных устройств печати и драйверов для них.

3.9.3. Графический интерфейс управления подсистемой печати

Управление списком текущих заданий и очередями печати осуществляется с помощью встроенного веб-сервера. Для его функционирования требуется установить следующие параметры конфигурации в конфигурационном файле /etc/cups/cupsd.conf:

Listen localhost:<порт>

WebInterface: Yes

По умолчанию веб-сервер функционирует на порту 631.

Для запуска графического средства управления подсистемой печати требуется подключиться веб-браузером по адресу:http://localhost:<порт>/admin.

Пример графического окна приложения приведен на рис. 54.

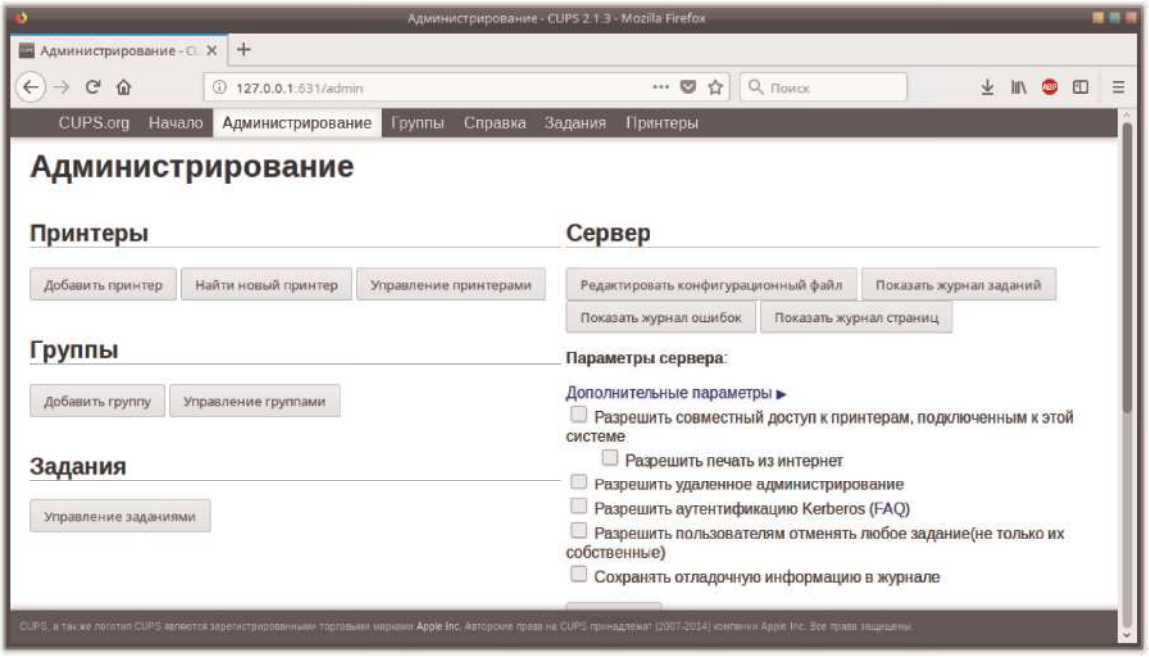


Рис. 54

Для управления очередями печати выполнить «Администрирование – Принтеры»:

– «Добавить принтер» – запускает диалог добавления принтера, позволяет указать расположение, драйвер и прочие параметры;

– «Найти новый принтер» – запускает поиск принтеров в сети и запускает диалог добавления принтера среди найденных;

– «Управление принтерами» – отображение уже добавленных в подсистему печати принтеров с возможностью изменения их параметров.

Операция добавления принтера реализована с помощью диалога, в котором администратор указывает следующие сведения (рис. 55):

– протокол работы принтера и его IP-адрес или выбранный принтер среди найденных автоматически;

– дополнительную информацию о принтере (название принтера для пользователя, описание, расположение);

– производителя и драйвер PPD принтера;

– настройки параметров печати.

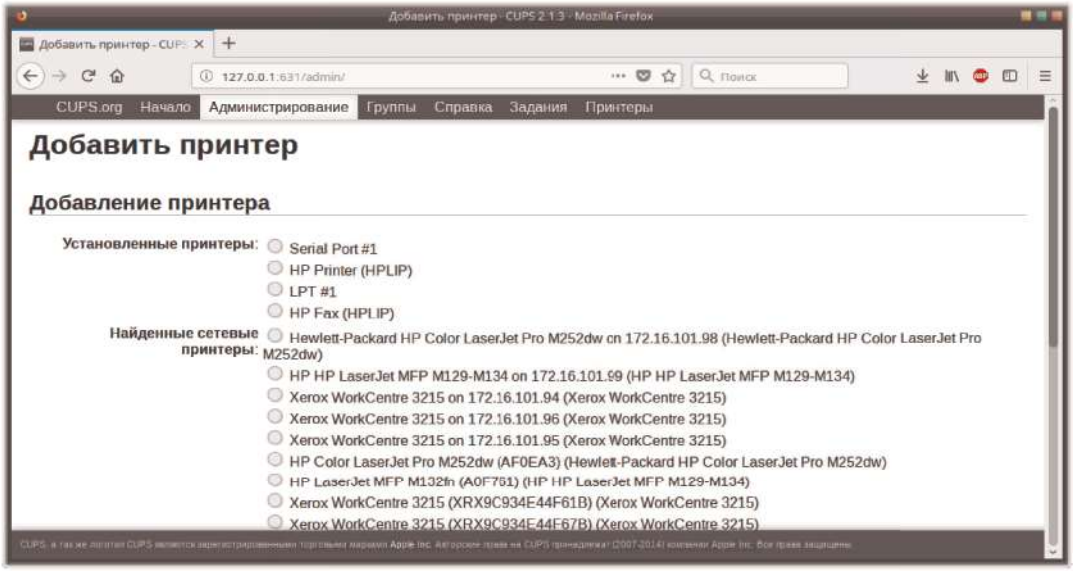


Рис. 55

Операция поиска нового принтера реализована схожим образом (рис. 56).

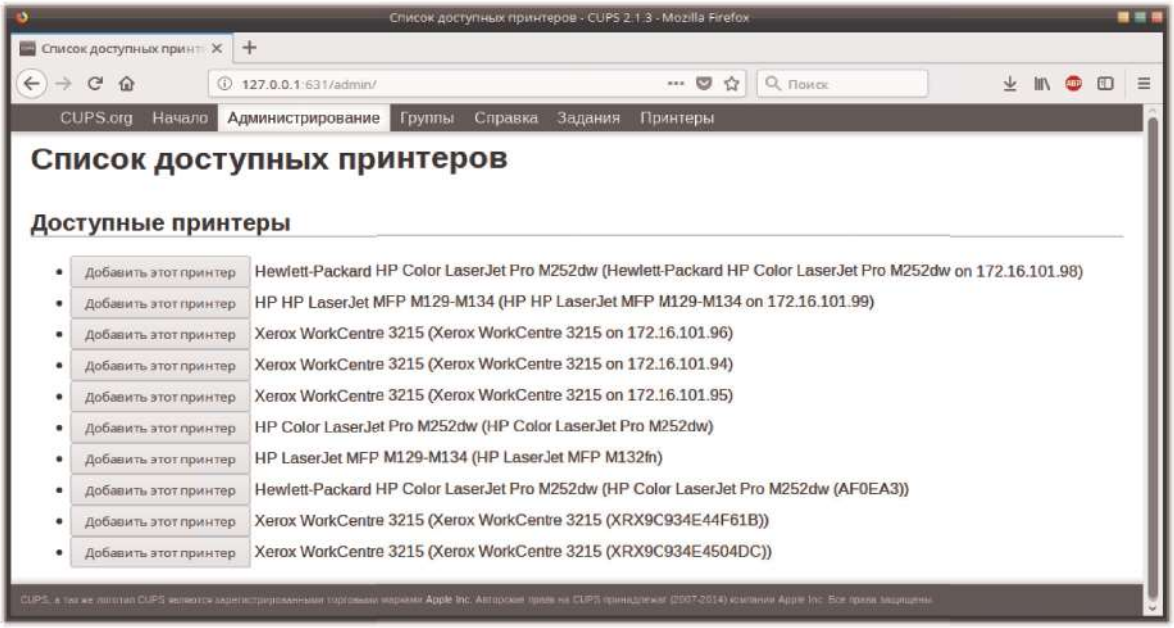


Рис. 56

При этом предлагается выбрать принтер среди найденных по сети подсистемой печати без возможности указания произвольного адреса расположения принтера.

Нажатие кнопки [Управление принтерами] открывает окно со списком всех добавленных в подсистему печати принтеров. Выбор принтера в списке доступных принтеров запускает диалог управления состоянием принтера и изменением его параметров (рис. 57).

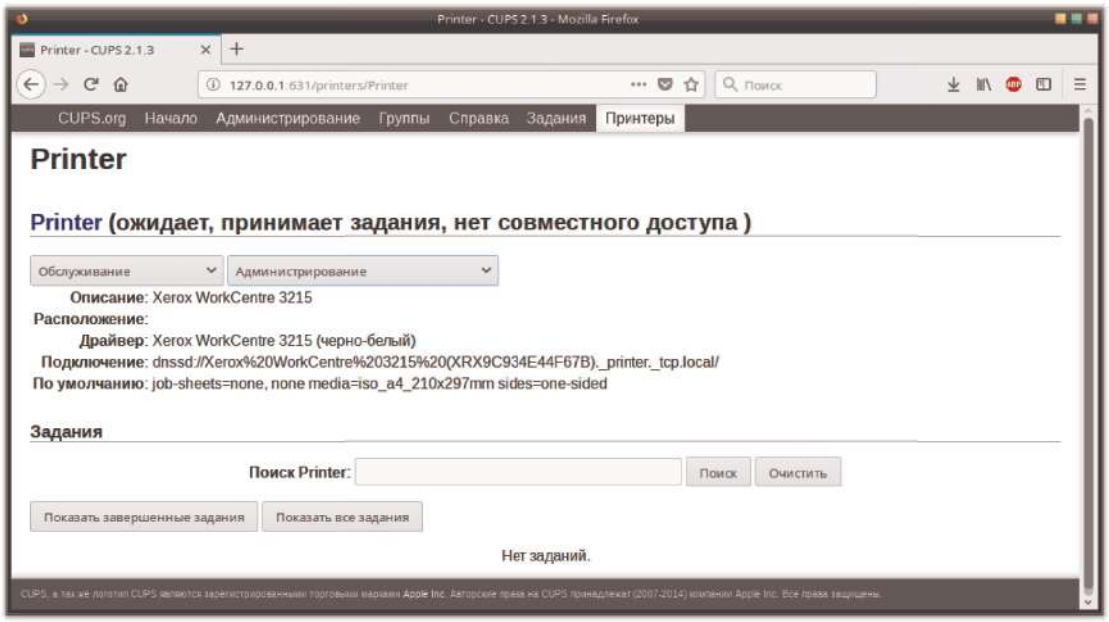


Рис. 57

В выпадающих списках «Обслуживание» и «Администрирование» выбрать необходимые действия, например, печать пробной страницы, удаление принтера, установка его по умолчанию.

В секции «Обслуживание» приведены операции по изменению состояния, в секции «Администрирование» – операции по изменению атрибутов принтера.

Для изменения политики безопасности принтера выбрать «Администрирование – Установить параметры по умолчанию». В открывшемся окне в секции «Политики» указать желаемую политику (рис. 58).

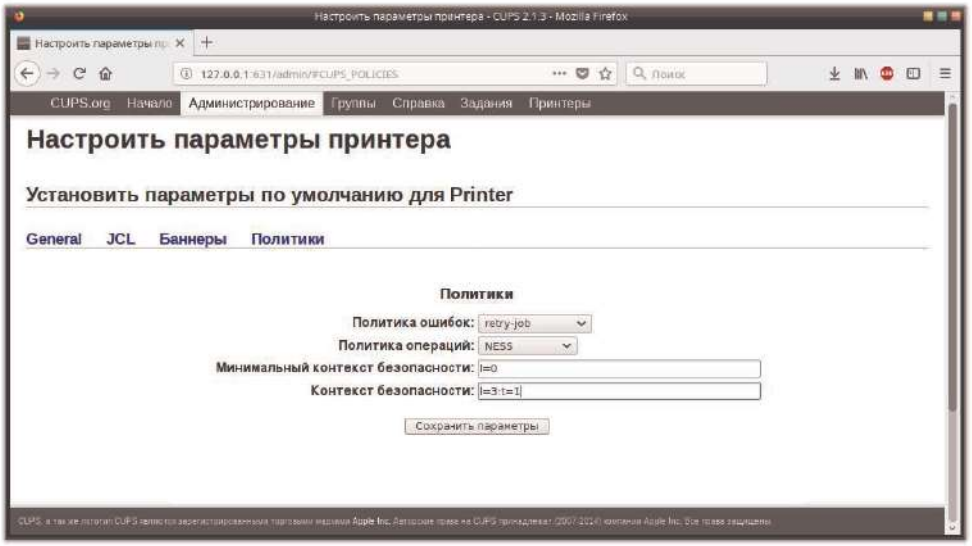


Рис. 58

Для просмотра и управления текущими заданиями нажать кнопку [Управление

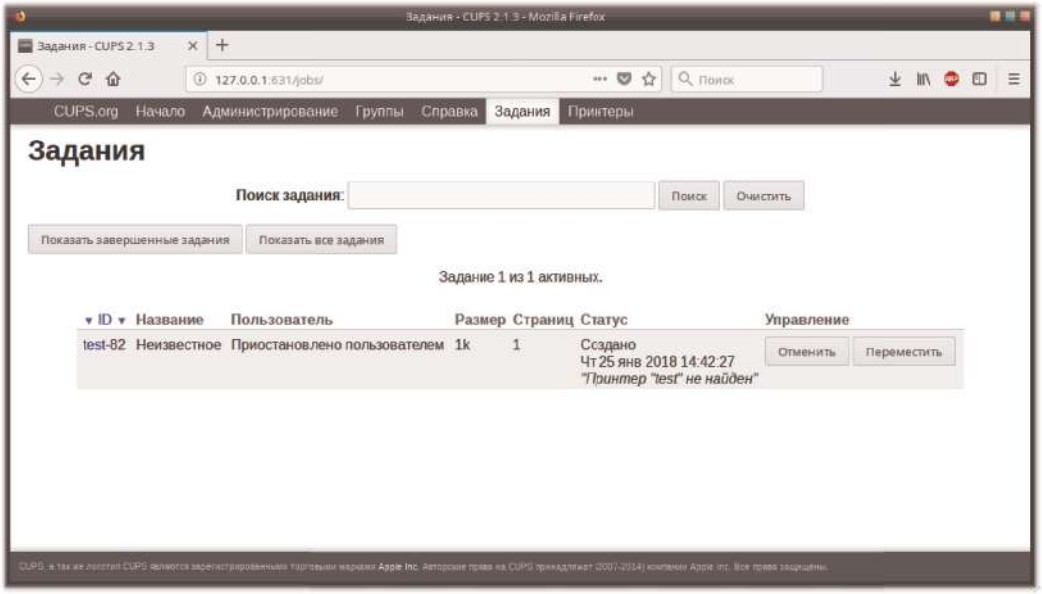


Рис. 59

В открывшемся окне отобразится список активных задач и элементы управления для сортировки по параметрам.

3.9.4. Маркировка документов

Все задания, которые получают контекст безопасности с ненулевым значением уровня конфиденциальности или непустым набором категорий, задерживаются на сервере печати и требуют маркировки.

Такие задания могут включать в себя только один документ (файл).

В составе ОС представлены следующие средства для маркировки:

– /usr/bin/lpattr – клиентское приложение для получения атрибутов задания;

– /usr/share/cups/mark/ – шаблоны и сценарии на языке PostScript для маркировки документов;

– /usr/bin/markjob-bin – приложение для маркировки документов;

– /usr/lib/cups/mark/psmark – вспомогательное приложение, выполняющее маркировку документа печати;

– /usr/lib/cups/mark/psmarker – сценарий, выполняющий запуск приложения psmarker с необходимыми параметрами;

– /usr/lib/cups/mark/pssplash – вспомогательное приложение, выполняющее генерацию так называемого фонарика для документа печати;

– /usr/lib/cups/mark/splash – сценарий, выполняющий запуск приложения pssplash с необходимыми параметрами;

– /usr/lib/cups/mark/cupspasswd – вспомогательное приложение графического

Маркировка документа осуществляется с помощью консольного приложения markjob-bin, запускаемого от имени специального пользователя, входящего в группу lpmark.

Приложение markjob-bin производит следующие действия:

– выводит список заданий, требующих маркировки, с помощью запроса IPP_OP_NESS_MARK_GET_JOBS;

– запрашивает у пользователя номер маркируемого задания и список следующих атрибутов:

– ness-inv-num – инвентарный номер;

– ness-owner-phone – телефон исполнителя;

– ness-workplace-id – идентификатор рабочего места;

– ness-distribution – список рассылки;

– отправляет запрос установки атрибутов IPP_OP_NESS_MARK_SET_JOB_ATTRIBUTES с введенными значениями вышеуказанных атрибутов на сервер печати;

– отправляет запрос IPP_OP_NESS_MARK_JOB на сервер печати;

– выводит номера заданий для промаркированного документа и соответствующего ему фонарика.

В ходе выполнения запроса IPP_OP_NESS_MARK_JOB на сервере печати выполняется преобразование оригинального документа в формат языка PostScript с использованием MIME-фильтра «vnd.cups-postscript».

Затем запускается на выполнение bash-скрипт /usr/lib/cups/mark/psmarker, принимающий в качестве входных параметров имя файла документа для маркировки на языке PostScript, имя файла шаблона маркировки /usr/share/cups/mark/marker.defs, имя файла переменных значений маркировки и имя выходного файла, который будет содержать промаркированный документ на языке PostScript.

В ходе выполнения этого bash-скрипта запускается на выполнение вспомогательная программа /usr/lib/cups/mark/psmarker, которая занимается подготовкой исходных данных на языке PostScript.

В процессе ее работы используются данные файла шаблона /usr/share/cups/mark/marker.template.

Затем происходит запуск программы Ghostscript /usr/bin/gs, в результате работы которой получается файл на языке PostScript, представляющий собой промаркированный документ.

В качестве исходных данных для программы Ghostscript используются следующие

файлы на языке PostScript:

– /usr/share/cups/mark/encoding.ps – данные шрифтов в кодировке Unicode, используемые для отображения текста маркировки;

– /usr/share/cups/mark/inject.ps – реализация обработчика EndPage, используемого для нанесения маркировки на каждую страницу документа;

– /usr/share/cups/mark/utils.ps – заготовки для отображения текста маркировки в различных частях листа.

Затем, после отработки bash-скрипта /usr/lib/cups/mark/psmarker на базе результирующего файла создается задание на печать, которое является дочерним по отношению к исходному заданию.

Это задание добавляется в очередь для печати в приостановленном состоянии.

Аналогично, для создания фонарика используется bash-скрипт /usr/lib/cups/mark/splash. Его параметры: имя файла шаблона маркировки /usr/share/cups/mark/splash.defs, имя файла переменных значений маркировки, имя выходного файла на языке PostScript, количество копий отправленного на печать документа.

В процессе выполнения этого bash-скрипта запускается на выполнение вспомогательная программа /usr/lib/cups/mark/pssplash, которая занимается подготовкой исходных данных на языке PostScript. В процессе ее работы используются данные файла шаблона/usr/share/cups/mark/marker.template.

Затем происходит запуск программы Ghostscript /usr/bin/gs, в результате работы которой получается файл фонарика на языке PostScript.

В качестве исходных данных для программы Ghostscript используются следующие файлы на языке PostScript:

– /usr/share/cups/mark/blank.ps – пустой лист формата А4;

– /usr/share/cups/mark/encoding.ps – данные шрифтов в кодировке Unicode, используемые для отображения текста маркировки;

– /usr/share/cups/mark/inject.ps – реализация обработчика EndPage, используемого для нанесения маркировки;

– /usr/share/cups/mark/utils.ps – заготовки для отображения текста маркировки в различных частях листа.

Затем, после отработки bash-скрипта /usr/lib/cups/mark/splash на базе результирующего файла создается задание на печать, которое является дочерним по отношению к исходному заданию. Это задание добавляется в очередь для печати в приостановленном состоянии.

Процесс создания дочерних заданий на печать повторяется по количеству копий, запрошенных при отправке документа на печать.

Для вывода промаркированного документа и фонарика на печать следует возобновить выполнение соответствующих заданий.

3.9.5. Графический интерфейс маркировки документов

Для маркировки документов также можно использовать приложение «Маркировка заданий».

Запуск приложения осуществляется через меню «Пуск» – «Системные» – «Маркировка заданий печати» или через командную строку:

```
$ markjob-app
```

Основная экранная форма приложения представлена на рис. 60. На ней представлена основная информация о заданиях, находящихся в очереди на маркировку.

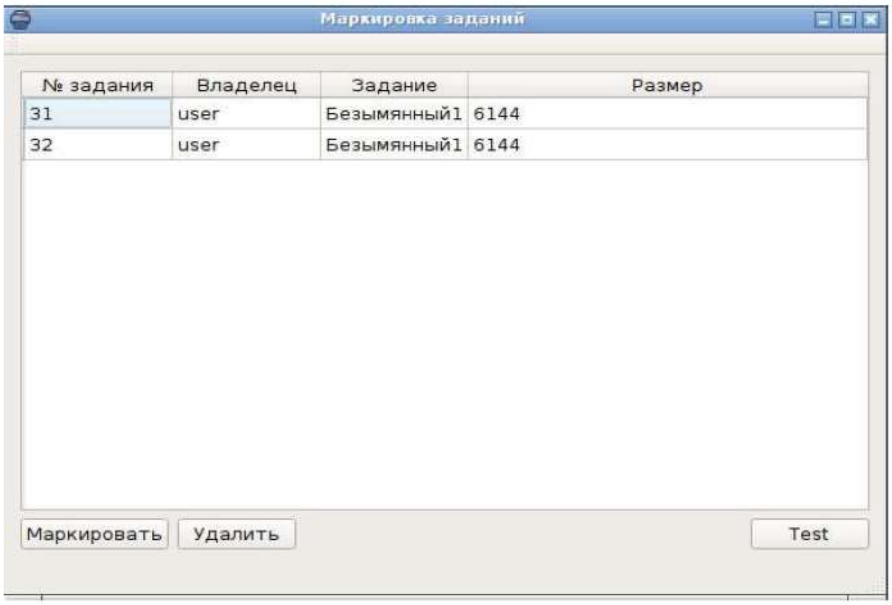


Рис. 60

В приложении отображаются задания принтера, установленного по умолчанию.

Нажатие на элемент в списке выделяет его для маркировки, после чего необходимо нажать на кнопку [Маркировать]. Появится форма для заполнения атрибутов (рис. 61).

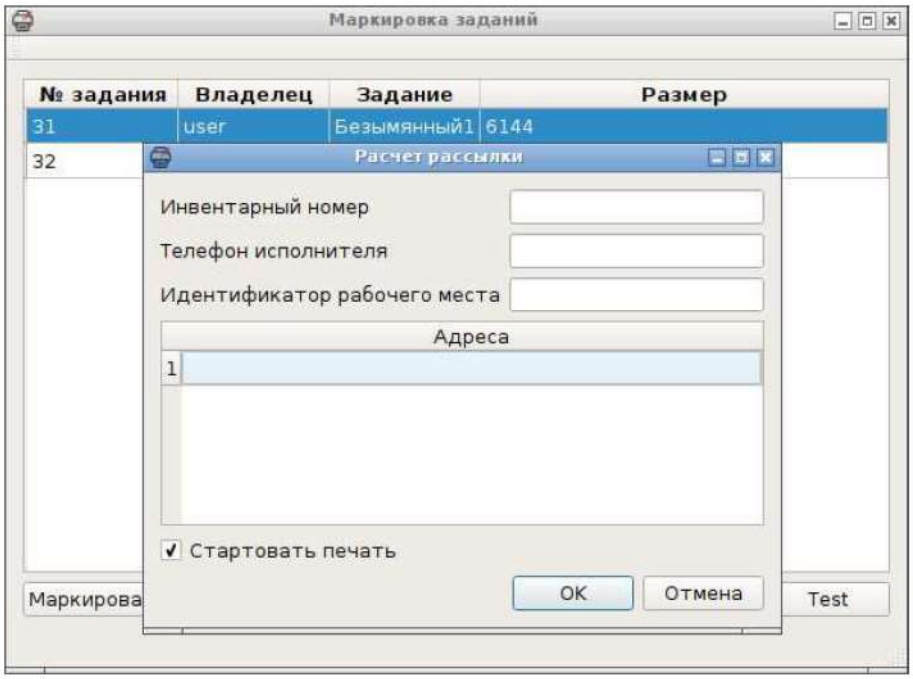


Рис. 61

После заполнения всех атрибутов можно выделить пункт «Стартовать печать», что убирает необходимость запускать дочерние задания через веб-интерфейс CUPS.

В случае если необходимо распечатать более одной копии документа, в окне заполнения атрибутов появляются дополнительные строки в списке «Адреса» (рис. 62).

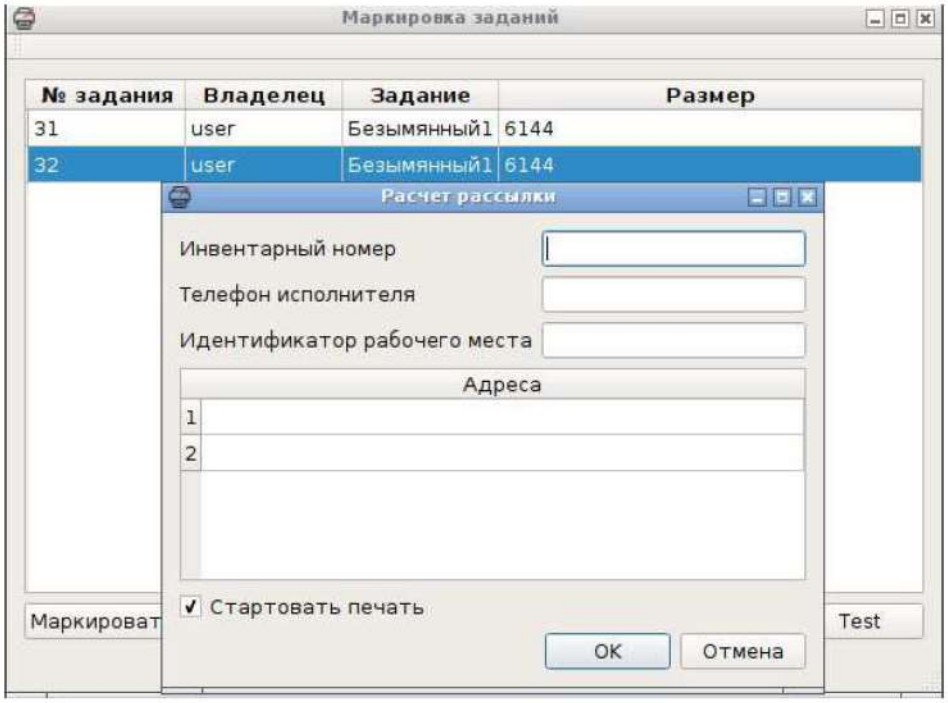


Рис. 62

Через приложение возможно удалить задание на печать. Для этого необходимо выделить задание и нажать на кнопку [Удалить], после чего появится окно с подтверждением операции (рис. 63).

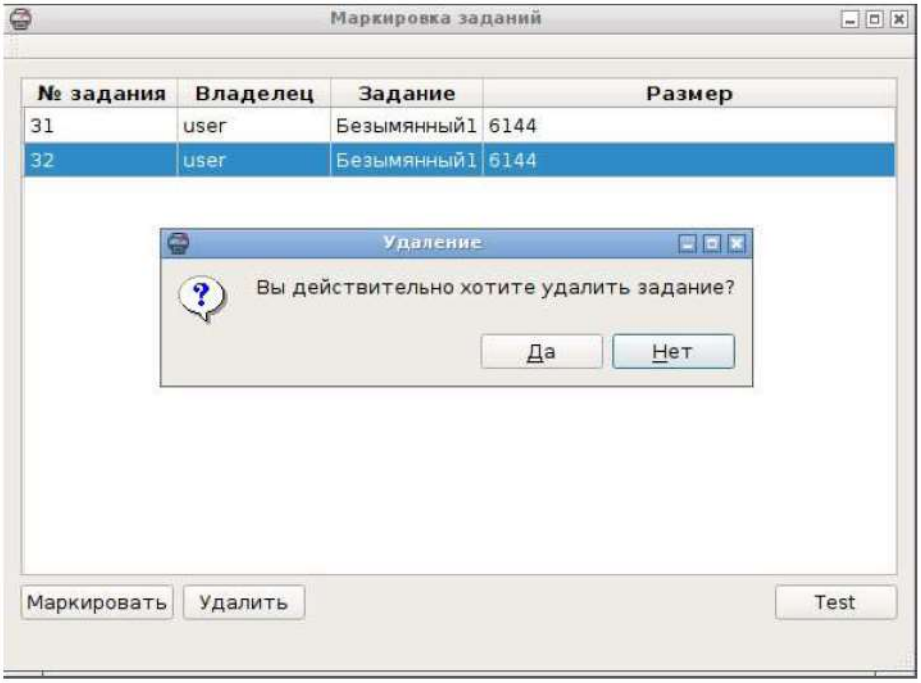


Рис. 63

3.9.6. Графический интерфейс просмотра истории заданий печати

Для просмотра истории заданий печати существует приложение «Обзор печати», которое открывается с помощью командной строки:

```
$ printlog-app
```

Основная экранная форма приложения представлена на рис. 64.

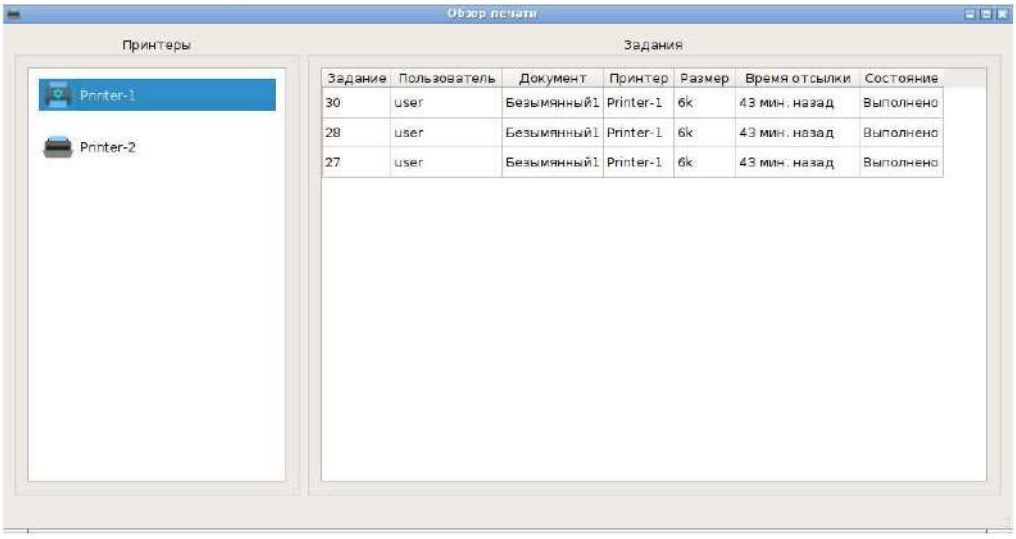


Рис. 64

В левой части окна выбирается принтер для просмотра истории заданий печати. В правой части отображается история завершенных заданий (со статусом «Выполнено» или «Отменено»).

Двойное нажатие по заданию открывает окно с параметрами задания.

Для просмотра истории заданий нескольких принтеров, необходимо выбрать требуемые принтеры в левой части окна (рис. 65). В столбце «Принтер» указано, к какому принтеру относится определенное задание.

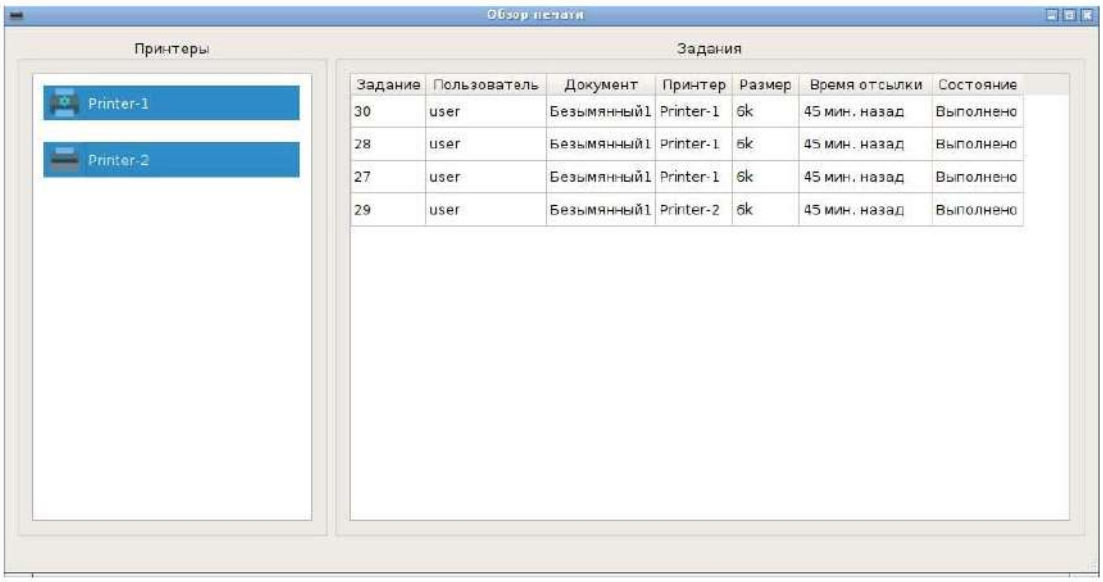


Рис. 65

3.9.7. Алгоритм печати документов, требующих маркировки

Общий алгоритм печати документов, требующих маркировки, выглядит следующим образом:

1) пользователь заходит в систему с требуемым контекстом безопасности (3.15.6.1);

2) документ с ненулевым контекстом безопасности или непустым набором категорий отправляется пользователем на печать (3.9.2.1);

3) документ автоматически задерживается на сервере печати;

4) администратор, входящий в группу marker, указывает атрибуты документа при помощи приложения «Маркировка заданий» и отправляет документ на печать (3.9.5).

Администратор может проверить выполнение задания печати при помощи приложения «Обзор печати» (3.9.6).

3.10. Контроль целостности

Целостность среды исполнения ОС обеспечивается следующими механизмами:

– средства регламентного контроля целостности – afick;

– средство подсчета контрольных сумм файлов и носителей – gostsum;

– средства создания замкнутой программной среды.

Каждый из механизмов нацелен на решения конкретных задач обеспечения контроля целостности.

3.10.1. Средства регламентного контроля целостности – afick

В качестве регламентного контроля целостности файлов ОС применяется набор ПС на основе программного пакета afick («Another File Integrity Checker»).

Принцип функционирования заключается в ведении эталонной (и обновляемой) БД контрольных сумм помещенных под контроль файлов. Процесс проверки может быть запущен вручную либо с помощью системного планировщика заданий (cron). В ходе проверки производится вычисление контрольных сумм файлов с последующим сравнением вычисленных значений с эталонными.

Основные параметры функционирования средства регламентного контроля целостности указываются в конфигурационном файле по умолчанию (/etc/afick.conf), например, путь к файлу БД.

Для описания способа вычисления и контроля целостности отдельных путей и каталогов применяется система правил. Например, правило для каталогов может выглядеть следующим образом:

DIR = p+i+n+u+g

где перечисленные ключи задают контроль прав доступа, метаданных, количества ссылок и стандартных атрибутов. Список основных ключей контролируемых атрибутов приведен в таблице 20.

Таблица 20

Ключ	Описание
a	Метка времени последнего доступа файла (atime)
b	Размер в блоках (blocks)
c	Метка времени изменения inode (ctime)
d	Составной номер устройства или ФС (device number)
g	gid — группа
i	Номер inode
m	Метка времени последнего изменения файла (mtime)
md5	Контрольная сумма MD5 (по умолчанию)
n	Число жестких ссылок
p	Права доступа (владелец, группа, другое) (permissions)

Ключ	Описание
u	uid — владелец (user)
s	Размер файла (size)

Окончание таблицы 20

Ключ	Описание
sha1 sha256 sha512	Один из вариантов контрольной суммы SHA. Может быть задан только один из вариантов
t	Расширенные права доступа (ACL)
z2 z5	Один из вариантов контрольной суммы по алгоритму ГОСТ Р 34.10-2012 (z2 - с длиной хэш-суммы 256 бит, z5 - с длиной хэш-суммы 512 бит).
e	Контекст безопасности
all	b+c+d+g+i+m+md5+n+p+u+s
R	p+d+i+n+u+g+s+m+c+md5
L	p+d+i+n+u+g
E	Пустой набор

Дополнительно представлены специальные правила.

Правило NESS выглядит следующим образом:

```
NESS = p+d+i+n+u+g+s+b+md5+m+e+t
```

где p+d+i+n+u+g+s+b+md5+m означает слежение за всеми стандартными атрибутами файла и использование хэш-функции MD5-Digest для слежения за целостностью содержимого файлов;

+e+t означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно.

Контроль ACL осуществляется при установке флага +g.

Правило GOST выглядит следующим образом:

```
GOST = p+d+i+n+u+g+s+b+gost+m+e+t
```

где p+d+i+n+u+g+s+b+gost+m означает слежение за всеми стандартными атрибутами файла и использование хэш-функции ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит для слежения за целостностью содержимого файлов;

+e+t означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно.

Контроль ACL осуществляется при установке флага +g.

Контроль файлов и каталогов в файле конфигурации выполняется:

```
/boot GOST
/bin GOST
/etc/security NESS
/etc/pam.d NESS
/etc/fatab NESS
```

```
/lib/modules NESS
/lib64/security NESS
/lib/security NESS
/sbin NESS
/usr/bin NESS
/usr/lib NESS
/usr/sbin NESS
```

Создание БД эталонных значений контрольных сумм и атрибутов выполняется при помощи следующей команды:

```
$ sudo afick –i
```

При этом будет создан файл /var/lib/afick/afick в формате ndbm.

Пакет содержит конфигурацию по умолчанию для системного планировщика заданий cron.

ВНИМАНИЕ! КОНФИГУРАЦИЯ СИСТЕМНОГО ПЛАНИРОВЩИКА ЗАДАНИЙ CRON ДЛЯ ЗАПУСКА РЕГЛАМЕНТНОГО КОНТРОЛЯ ЦЕЛОСТНОСТИ ДОЛЖНА БЫТЬ РАЗРАБОТАНА С УЧЕТОМ СОВМЕСТНОГО ПРИМЕНЕНИЯ С ПОДСИСТЕМОЙ РЕГИСТРАЦИИ СОБЫТИЙ.

Подробное описание утилиты afick и формата конфигурационного файла приведено в руководстве man afick, man afick.conf.

В состав средств регламентного контроля целостности входит графическая утилита afick-gui (afick-tk).

3.10.2. Средство подсчета контрольных сумм файлов – gostsum

Для подсчета контрольный сумм файлов и носителей в состав ОС включены утилиты командной строки gost12sum и gost94sum. Утилита gost12sum использует алгоритм вычисления хеш-суммы ГОСТ Р 34.11-2012. Утилита gost94sum использует устаревший алгоритм вычисления хеш-суммы ГОСТ Р 34.11-94.

Для вывода информации о синтаксисе необходимо выполнить команду

```
gost12sum –h
```

3.10.2.1. Графический интерфейс работы с gost12sum

Для проверки контрольных сумм дисков и образов в ОС ОН «Стрелец» необходимо:

1) запустить программу «Подсчет КС» (рис. 66).

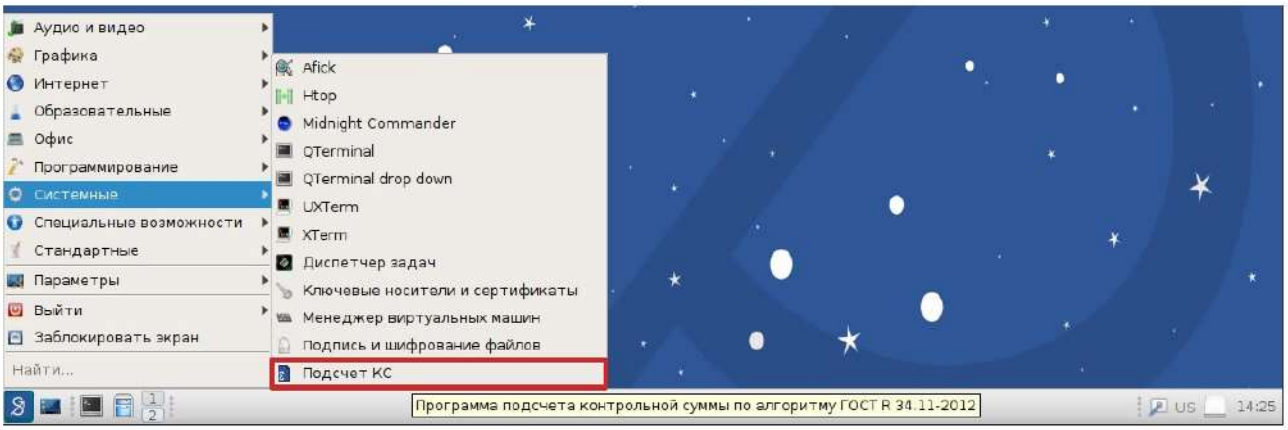


Рис. 66

Откроется окно программы «Подсчет контрольной суммы» (рис. 67).

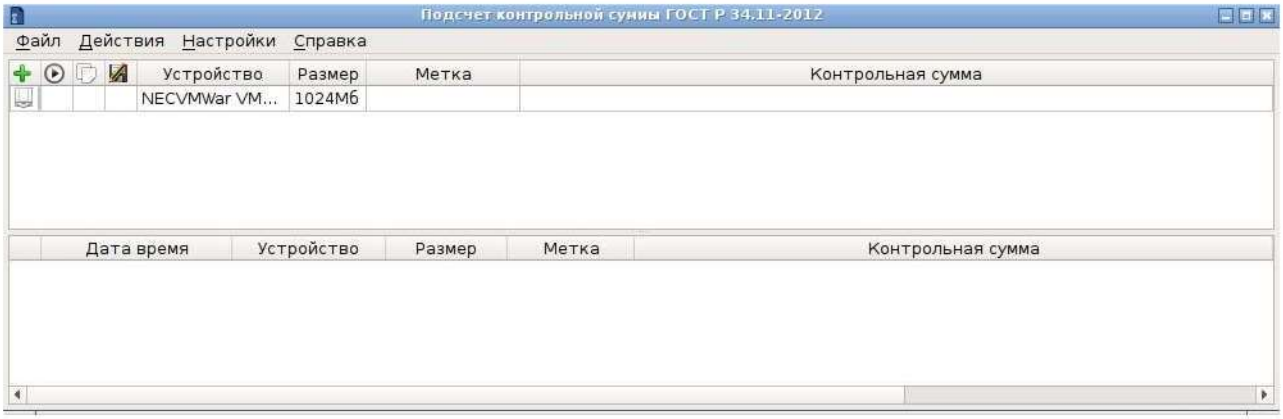


Рис. 67

Контрольные суммы дисков, вставленных в дисковод, начинают подсчитываться автоматически (можно отключить в пункте «Настройки»);

2) чтобы добавить образ для проверки контрольной суммы, нужно выбрать пункт «Файл» – «Добавить файл»;

3) для начала подсчета контрольной суммы добавленного образа необходимо выбрать пункт меню «Действия» – «Запустить все» либо нажать на значок напротив необходимого образа в списке.

Начнется подсчет контрольной суммы с указанием прогресса и оставшегося времени в столбце «Контрольная сумма» (рис. 68, рис. 69).

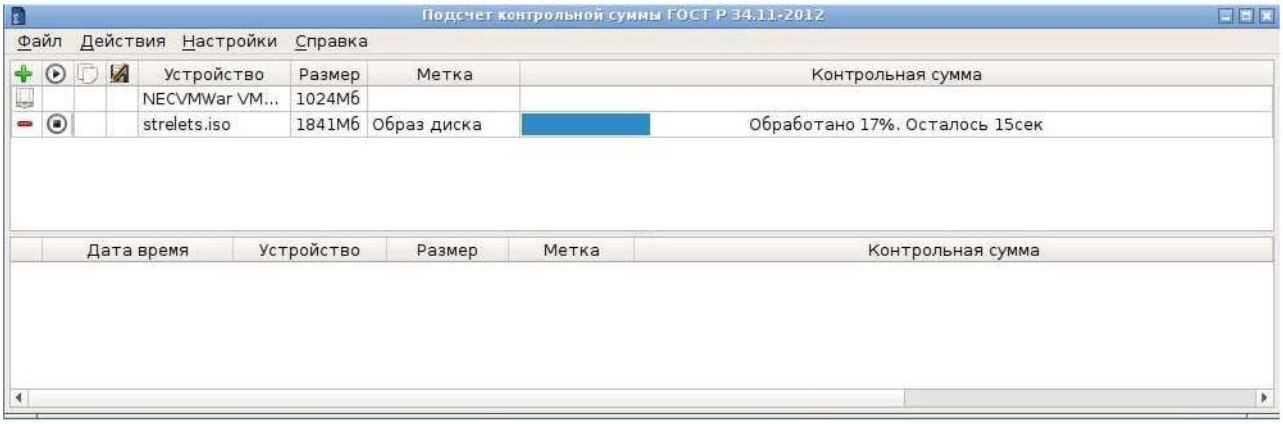


Рис. 68

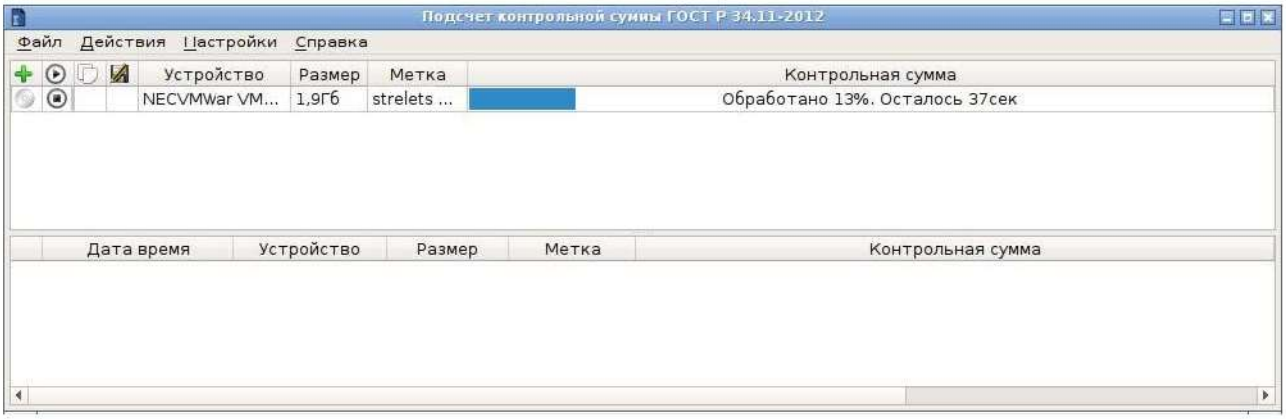


Рис. 69

После окончания подсчета контрольная сумма будет указана в столбце «Контрольная сумма» (рис. 70);

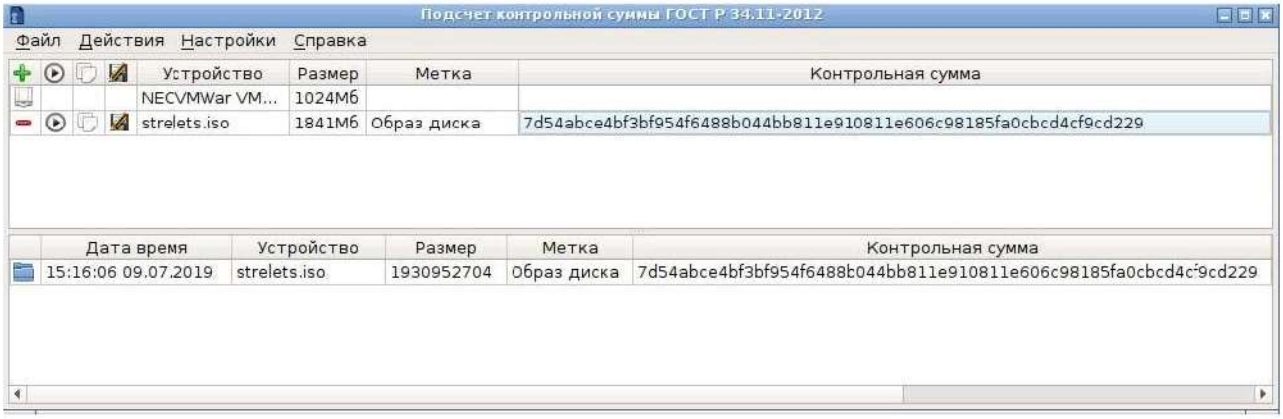


Рис. 70

Во второй половине окна отображается история контрольных сумм образов и дисков.

3.10.3. Средства создания замкнутой программной среды

Для реализации замкнутой программной среды используется механизм ядра

Integrity Measurement Architecture (IMA). Его реализация дополнена поддержкой функций хэширования в соответствии с ГОСТ Р 34.11-2012 и формирования и проверки цифровой подписи в соответствии с ГОСТ Р 34.10-2012.

Механизм IMA предоставляет возможность внедрения контрольной суммы или цифровой подписи в расширенные атрибуты файлов. При обращении к файлу проводится проверка корректности подписи и в случае несоответствия доступ к такому файлу запрещается.

На этапе компиляции в ядро ОС добавляется корневой сертификат разработчика ОС (длина ключа 512 бит) и подписанный на нем сертификат для проверки подписей модулей ядра (длина ключа 512 бит). Подпись модулей ядра формируется в ходе компиляции и хранится в конце файлов модулей в формате PKCS7.

В тестовых целях возможно отключение проверки подписей модулей ядра. Для этого надо указать в параметрах загрузки ядра module.sig_enforce=0. Сделать это можно в интерфейсе загрузчика GRUB или внести эти изменения в файле /etc/default/grub и выполнить команду:

```
$ sudo upgrade-grub
```

Содержимое бинарных пакетов подписывается во время сборки и подписи добавляются в расширенные атрибуты файлов пакета. Сертификат, на котором проверяются подписи файлов, содержится в файле /etc/ima/x509_ima.der (длина ключа 256 бит). Этот сертификат также подписан на корневом сертификате разработчика ОС.

Есть возможность выпуска дополнительных сертификатов для разработчиков ПО, которые подписываются на корневом сертификате производителя и также могут использоваться для подписи и последующей проверки подписей объектов файловой системы. Использование для проверки подписей сертификатов, не подписанных на корневом сертификате разработчика ОС, невозможно.

Для проверки подписей используются сертификаты разработчиков, находящиеся в кейринге ima. При загрузке системы в кейринг ima подпадают все ключи (подписанные на корневом сертификате разработчика ОС) из директории /etc/ima/certs, включая вложенные директории.

В тестовых целях возможно добавить свой ключ в кейринг вручную, используя команды:

```
$ sudo evmctl -iv import <certificate.file> 0x<nnnnnn>
```

где nnnnnn это идентификатор кейринга (изменяется динамически при загрузке системы).

Получить его можно, используя команду:

```
$ sudo grep .ima /proc/keys
```

0bb0e31e I------ 1 perm 1f0f0000 0 0 keyring,→ .ima_blacklist: empty

2c264a9e I------ 1 perm 1f0f0000 0 0 keyring .ima: 5

В данном примере идентификатор кейринга ima — 2c264a9e.

Для выполнения операций по управлению ключами и ручной проверки подписей файлов в состав ОС входит утилита evmctl. Далее приведен пример основных операций утилиты, а подробное описание доступно в руководстве man evmctl.

Примеры:

1) Установка подписи IMA:

```
$ sudo evmctl -ivk <private.key> ima_sign <filename>
```

2) Проверка подписи IMA:

```
$ sudo evmctl -ivk <certificate.file> ima_verify <filename>
```

3) Просмотр информации о подписи IMA:

```
$ sudo evmctl show <filename>
```

4) Просмотр информации о подписи IMA в формате XML:

```
$ sudo evmctl show_xml <filename>
```

Для осуществления подписи своих пакетов в ОС разработчикам предоставляются несколько скриптов:

– /usr/share/doc/ima-evm-utils/sign-deb.sh – осуществляет подпись deb-пакета, внедряя информацию о подписи в расширенные атрибуты файлов внутри пакета, контрольная сумма пакета изменяется.

Использование:

```
$ /usr/share/doc/ima-evm-utils/sign-deb.sh <secretkey> <source deb> <destination deb>
```

где <secretkey> – закрытая часть ключа разработчика, <source deb> – исходный deb-пакет, <destination deb> – целевой deb-пакет;

– /usr/share/doc/ima-evm-utils/sign-deb-ko.sh – осуществляет подпись deb-пакета, внедряя информацию о подписи в расширенные атрибуты файлов внутри пакета, контрольная сумма пакета изменяется. Помимо этого подписываются все .ko файлы модулей ядра, находящиеся в пакете.

Использование:

```
$ /usr/share/doc/ima-evm-utils/sign-deb-ko.sh <secretkey> <source deb> <destination deb>
```

где <secretkey> – закрытая часть ключа разработчика, <source deb> – исходный deb-пакет, <destination deb> – целевой deb-пакет;

– /usr/share/doc/ima-evm-utils/sign-foreign-deb.sh – создает файл, содержащий информацию о подписях всех файлов пакета. Сам пакет при этом не изменяется.

Использование:

```
$ /usr/share/doc/ima-evm-utils/sign-foreign-deb.sh <secretkey> <source deb> <package-name>.dump
```

где <secretkey> – закрытая часть ключа разработчика, <source deb> – исходный deb-пакет, <package-name>.dump – файл, содержащий информацию о подписях пакета.

Для установки пакетов, подготовленных таким образом, предварительно необходимо поместить файл <package-name>.dump в директорию /etc/dpkg/ima.d/. После установки каждого пакета dpkg проверяет наличие файла /etc/dpkg/ima.d/<package-name>.dump и, при его существовании, переносит информацию о подписях файлов пакета в расширенные атрибуты файлов пакета.

Предустановленные политики проверки подписей находятся в директории /etc/ima/policy.d:

– empty – не осуществлять проверку подписей файлов;

– appraise – осуществлять проверку подписей у исполняемых файлов и скриптов;

– appraise_audit – осуществлять проверку подписей у исполняемых файлов и скриптов, а также производить логирование всех проверок подписей.

Выбор политики осуществляется через установку символической ссылки /etc/ima/policy на нужную политику.

Примеры:

1) Для выключения проверки подписей необходимо выполнить команды:

```
$ sudo rm /etc/ima/policy
$ sudo ln -s /etc/ima/policy.d/empty /etc/ima/policy
$ sudo update-initramfs -u -k all
```

2) Для включения проверки подписи выполнить:

```
$ sudo rm /etc/ima/policy
$ sudo ln -s /etc/ima/policy.d/appraise /etc/ima/policy
$ sudo update-initramfs -u -k all
```

Администратор системы может создавать свою собственную политику проверки подписей. Для ее включения необходимо поменять ссылку на активную политику, как указано ранее.

Политика представляет из себя текстовый файл, каждая строка в котором содержит одно правило. При применении политики правила проверяются по порядку. В случае если подходит правило dont_measure или dont_appraise, проверка прекращается и операция

разрешается.

Нет возможности привязывания правил к путям файловой системы, например «проверять подписи у всех файлов /sbin». Это связано с тем, что не для всех контекстов выполнения возможно получение исходного пути файла, например при системном вызове mmap().

Каждое правило в политике представляет из себя отдельную строку. Правило должно начинаться с одного из действий:

- measure: подписать файл в случае отсутствия IMA подписи;
- dont_measure: не подписывать файл;
- appraise: проверять IMA подпись файла;
- dont_appraise: не проверять IMA подпись;
- audit: добавить сообщение в файл аудита.

После действия приводится список условий, при выполнении которых это действие должно применяться (таблицы 21, 22).

Таблица 21

Условие	Описание
fsmagic=<magic>	Файл должен располагаться на заданной файловой системе. Возможные значения magic доступны по команде man 2 statfs
fsuuid=<id>	Файл должен располагаться на файловой системе с заданным UUID
fowner=0, fowner <100	Файл должен принадлежать пользователю с заданным uid либо с uid больше или меньше заданного
uid=0, euid=0	Процесс выполняется с заданным (эффективным) uid
mask=MAY_READ	Ограничение по возможному режиму доступа процесса к файлу
appraise_type=imasig	Указывает, что для операций appraise обязательно наличие подписи, а не просто хэша
func=<context>	Ограничение по типу операции, проводящейся с файлом, возможные значения приведены в таблице 20

Таблица 22

Значение	Описание
BPRM_CHECK	Файл запускается на выполнение
MMAP_CHECK	Файл отображается в память процесса
FILE_CHECK	Открытие файла
MODULE_CHECK	Файл загружается в качестве модуля ядра
FIRMWARE_CHECK	Файл загружается в ядро в качестве бинарной прошивки

Управление модулем IMA осуществляется с помощью опций загрузки ядра ima_apprise, ima_hash и ima_audit.

ima_apprise=off|enforce|fix|log, параметры имеют следующий смысл:

– off: не проводить проверку никаких файлов;

– enforce: отклонять доступ к файлам, не имеющим подписей или хэшей, соответствующих действующей политике;

– fix: выставлять значения хэшей файлов в соответствии с политикой (только для файлов, не имеющих подписи в расширенных атрибутах);

– log: разрешать доступ к файлам с некорректными подписями, но проводить аудит таких обращений.

ima_hash=md5|sha1|rmd160|sha256|sha384|md_gost94 – хэш по умолчанию для правил measure.

ima_audit=0|1 – если выставлено в 1, в аудит добавляется расширенная информация.

Пример файла политики:

```
# PROC_SUPER_MAGIC
dont_measure fsmagic=0x9fa0
dont_appraise fsmagic=0x9fa0
# SYSFS_MAGIC
dont_measure fsmagic=0x62656572
dont_appraise fsmagic=0x62656572
# DEBUGFS_MAGIC
dont_measure fsmagic=0x64626720
dont_appraise fsmagic=0x64626720
# TMPFS_MAGIC
dont_measure fsmagic=0x01021994
dont_appraise fsmagic=0x01021994
# RAMFS_MAGIC
dont_appraise fsmagic=0x858458f6
# DEVPTS_SUPER_MAGIC
dont_measure fsmagic=0x1cd1
dont_appraise fsmagic=0x1cd1
# BINFMTFS_MAGIC
dont_measure fsmagic=0x42494e4d
dont_appraise fsmagic=0x42494e4d
# SECURITYFS_MAGIC
dont_measure fsmagic=0x73636673
dont_appraise fsmagic=0x73636673
```

```
# SELINUX_MAGIC
dont_measure fsmagic=0xf97cff8c
dont_appraise fsmagic=0xf97cff8c
# CGROUP_SUPER_MAGIC
dont_measure fsmagic=0x27e0eb
dont_appraise fsmagic=0x27e0eb
# CGROUP2_SUPER_MAGIC
dont_measure fsmagic=0x63677270
dont_appraise fsmagic=0x63677270
# NSFS_MAGIC
dont_measure fsmagic=0x6e736673
dont_appraise fsmagic=0x6e736673
appraise func=BPRM_CHECK fowner=0 appraise_type=imasig
appraise func=BPRM_CHECK uid=0 appraise_type=imasig
appraise func=FILE_MMAP fowner=0 mask=MAY_EXEC appraise_type=imasig
appraise func=FILE_MMAP uid=0 mask=MAY_EXEC appraise_type=imasig
appraise func=MODULE_CHECK appraise_type=imasig
appraise func=FIRMWARE_CHECK appraise_type=imasig
```

В данном наборе правил сначала исключаются из проверки содержимое /proc, /sys, удаленные ФС и прочие подобные. После этого указано, что должны проверяться:

– все принадлежащие пользователю с uid=0 файлы, которые запускаются или отображаются в память процессов;

– все файлы, которые запускают или отображают в память процессы, выполняющиеся в контексте пользователя с uid=0;

– все файлы, загружаемые в ядро в качестве модуля или бинарной прошивки.

3.11. Резервное копирование и восстановление данных

Резервное копирование – процесс создания копии данных на носителе (ЖД, дискете и т. д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Примечание. Резервное копирование влияет на работоспособность системы. Резервное копирование и восстановление увеличивает текущую нагрузку на систему, что может вызывать замедление работы системы или недовольство пользователей. Кроме того, в зависимости от вида резервного копирования и восстановления, может требоваться монопольный доступ к системе или даже полная остановка ее работы.

Для обеспечения надежного резервного копирования и восстановления в реальных системах применяется четкое планирование указанных процессов, учитывающее все аспекты построения и функционирования системы.

ВНИМАНИЕ! РАБОТА С МАНДАТНЫМИ АТРИБУТАМИ И АТРИБУТАМИ АУДИТА ПРИ ИСПОЛЬЗОВАНИИ РАЗЛИЧНЫХ УТИЛИТ СОЗДАНИЯ РЕЗЕРВНЫХ КОПИЙ ТРЕБУЕТ ОПЦИЙ СОХРАНЕНИЯ РАСШИРЕННЫХ АТРИБУТОВ (ВИДА –XATTRS, ВОЗМОЖНО С УКАЗАНИЕМ ДОПОЛНИТЕЛЬНЫХ ПАРАМЕТРОВ).

3.11.1. Утилита rsync

Утилита rsync предназначена для удаленного копирования (резервного копирования) или синхронизации файлов и каталогов с минимальными затратами трафика.

Все действия выполняются от имени учетной записи администратора с использованием механизма sudo.

В таблице 23 приведены некоторые наиболее часто используемые опции команды rsync.

Таблица 23

Опция	Назначение
-v, --verbose	Подробный вывод
-z, --compress	Сжимать трафик
-r, --recursive	Выполнять копирование рекурсивно
-p, --perms	Сохранять дискретные права доступа
-t, --times	Сохранять время доступа к файлам
-g, --group	Сохранять группу
-o, --owner	Сохранять владельца
-A, --acls	Сохранять списки контроля доступа ACL (включает -p)
-X, --xattrs	Сохранять расширенные атрибуты (в том числе мандатные атрибуты)

Следующая команда сделает копию домашней директории:

```
$ sudo runcon t=CHCTX,SETXATTR_OMITMAC rsync -vzrptgoAX /home/ /tmp/home_bak
```

ВНИМАНИЕ! НЕ РЕКОМЕНДУЕТСЯ ИСПОЛЬЗОВАТЬ ОПЦИЮ -L ДЛЯ КОПИРОВАНИЯ СИМВОЛИЧЕСКИХ ССЫЛОК ПРИ СОЗДАНИИ РЕЗЕРВНОЙ КОПИИ ДОМАШНИХ КАТАЛОГОВ ПОЛЬЗОВАТЕЛЕЙ.

3.11.2. Утилита tar

Утилита tar предназначена для архивирования файлов и каталогов.

Все действия выполняются от имени учетной записи администратора с использованием механизма sudo.

Подробное описание команды приведено в man tar.

Далее приведены примеры создания и восстановления резервных копий с использованием утилиты tar.

Создание администратором архива домашнего каталога пользователя может быть выполнено с помощью команды:

```
$sudo tar --xattrs --acls --xattrs-include=security.NESSCTX \
-cvvvzf /var/backups/home.tgz /home/
```

Опция `--xattrs` означает включение поддержки расширенных атрибутов. Опция `--xattrs-include=security.NESSCTX` определяет подключаемый шаблон восстановления расширенных атрибутов для ключа `xattrs`. Опция `--acls` означает включение поддержки POSIX ACL. Опции `-cvvvzf` необходимы для создания архива (create), включения режима отображения обрабатываемых файлов (verbose), применения метода сжатия (gzip), указания файла (file) соответственно.

Путь /var/backups/home.tgz задает место расположения созданного архива и его имя, путь /home/ определяет, что именно будет вложено в архив.

Восстановление выполняется с помощью команды:

```
$sudo runcon t=CHCTX,SETXATTR_OMITMAC,OMIT tar --xattrs \
-xattrs-include=security.NESSCTX --acls -xvvvf \
/var/backups/home.tgz -C /
```

Опции `-xvvvf` необходимы для извлечения из архива (extract), включения режима отображения обрабатываемых файлов (verbose), указания файла (file) соответственно.

3.11.3. Система резервного копирования Bacula

Bacula – это система централизованного резервирования информационных ресурсов, основными преимуществами которой являются:

– возможность функционирования в гетерогенных средах;

– централизованное управление как в текстовом, так и в графическом режимах;

– обеспечение высокого уровня безопасности резервных копий;

– поддержка широкого спектра оборудования (ленточные библиотеки и т.д.).

Реализована как сетевая клиент-серверная программа для резервного копирования, архивирования и восстановления. Предлагая широкие возможности для управления хранилищами данных, облегчает поиск и восстановление потерянных или поврежденных файлов. Благодаря модульной структуре, Bacula масштабируется и может работать как на маленьких, так и на крупных системах, состоящих из сотен компьютеров, расположенных в большой сети.

Система резервирования данных Bacula состоит из четырех основных элементов: Director Daemon, Storage Daemon, File Daemon и Bacula Console. Все эти элементы реализованы в виде самостоятельных приложений.

Director Daemon (DD) – это центральный элемент системы, осуществляющий управление ее остальными компонентами. В его задачи входит управление процессом резервирования/восстановления данных, обеспечение интерфейса управления для администраторов и многое другое. Говоря проще – это диспетчер, который инициирует все процессы и отслеживает ход их выполнения.

Storage Daemon (SD) – приложение, отвечающее за чтение/запись данных непосредственно на устройства хранения информации. Принимает управляющие команды от DD, а также резервируемые данные от/к File Daemon.

File Daemon (FD) – этот элемент еще можно назвать Агентом. Ведь именно он работает в рамках ОС, данные которой необходимо резервировать. File Daemon выполняет всю рутину, осуществляя обращение к резервируемым файлам и их дальнейшую передачу к SD. Также на стороне FD выполняется шифрование резервных копий, если это определено конфигурацией.

Bacula Console (BC) – интерфейс администратора системы. По своей сути, это командный интерпретатор для управления Bacula. Строго говоря, Bacula Console может быть расширена с помощью графических систем управления, которые, как правило, являются всего лишь надстройкой над BC. К таким системам можно отнести Tray Monitor и Bat. Первая устанавливается на компьютере администратора системы и осуществляет наблюдение за работой системы резервирования, а вторая обеспечивает возможность управления посредством графического интерфейса.

Bacula Catalog – БД, в которой хранятся сведения обо всех зарезервированных файлах и их местонахождении в резервных копиях. Каталог необходим для обеспечения эффективной адресации к требуемым файлам. Поддерживаются MySql, PostgreSql и SqLite.

Такое структурное деление позволяет организовать очень гибкую систему резервирования, когда Storage Daemon разворачивается на выделенном сервере с несколькими устройствами хранения данных. Также Bacula Director может управлять несколькими экземплярами SD, обеспечивая резервирование части данных на одно устройство хранения, а части – на другое.

Все указанные компоненты могут находиться как на одном компьютере, так и на нескольких, объединенных в сеть.

Программа Bacula обеспечивает поддержку сохранения расширенных атрибутов каталогов и файлов и, при необходимости, их последующее восстановление.

Далее описан пример настройки комплекса программ Bacula.

Все действия выполняются от имени учетной записи администратора с использованием механизма sudo.

В примере использована следующая инфраструктура:

– выделенный сервер dd.ex.net с IP-адресом 10.0.0.10 (на нем будет функционировать Director Daemon – это главный сервер, осуществляющий резервное копирование);

– выделенный сервер sd.ex.net с IP-адресом 10.0.0.20 (на нем будет функционировать Storage Daemon – это машина, на которой будут размещаться резервные копии данных);

– персональный компьютер fd.ex.net с IP-адресом 10.0.0.30 (на нем будет функционировать File Daemon – это машина, с которой будут копироваться данные и на которую будут восстанавливаться резервные копии данных).

Подготовка инфраструктуры для управления системой резервного копирования выполняется следующим образом:

1) установить Postgresql на сервер, где будет работать Director Daemon:

```
$ sudo aptitude install postgresql-9.6
```

2) предполагается, что на всех машинах изначально установлены все пакеты, касающиеся Bacula, из состава ОС. С помощью менеджера пакетов Synaptic по ключевому слову «bacula» установить все пакеты, кроме тех, где в названии фигурирует «-sqlite3»:

– bacula-bscan

– bacula-common

– bacula-common-pgsql

– bacula-console

– bacula-director

– bacula-director-pgsql

– bacula-fd

– bacula-sd

– bacula-server

Также должен быть установлен пакет attr:

```
$ sudo aptitude install attr
```

При настройке Bacula появится интерфейс для настройки совместимости с БД, в качестве имени базы указать bacula и пароль bacula.

При настройке базы Bacula может произойти ошибка, на данном этапе необходимо ее игнорировать, база будет настроена позже.

П р и м е ч а н и е. Далее в любом случае созданная БД будет перенастроена.

3) подготовить БД для Bacula, для чего:

– в файле /etc/postgresql/9.6/main/postgresql.conf параметру listen_addresses= присвоить значение '*';

– в файле /etc/postgresql/9.6/main/pg_hba.conf внести необходимые изменения, для простоты можно указать метод trust для всех соединений, удалить любую дополнительную конфигурацию после метода типа mod=;

– обязательно добавить host с IP-адресом, где будет работать Director Daemon. В случае если все демоны Bacula будут установлены на одну машину, указывать IP-адрес не обязательно, т.к. работа будет идти через localhost.

Пример файла pg_hba.conf:

```
local     all  postgres       trust
local     all  all       trust
host all  all  127.0.0.1/32   trust
host all  all  10.0.0.1/24    trust
```

– присвоить пароль postgres:

```
$ sudo passwd postgres
```

– присвоить для Bacula пароль bacula:

```
$ sudo passwd bacula
```

– выполнить перезапуск БД:

```
$ sudo pg_ctlcluster 9.6 main restart
```

– создать пользователя БД для работы с Bacula и настроить ему привилегии (выполнять не от имени учетной записи администратора):

```
# psql template1 -U postgres -h 10.0.0.10
template1=# CREATE ROLE bacula;
```

```
template1=# ALTER USER bacula PASSWORD 'bacula';
template1=# ALTER USER bacula LOGIN SUPERUSER CREATEDB
CREATEROLE;
```

– не завершая работу с интерпретатором psql создать БД и присвоить ей владельца:

```
template1=# CREATE DATABASE bacula;
template1=# ALTER DATABASE bacula OWNER TO bacula;
```

4) на сервере dd.ex.net запустить скрипты, которые создадут все необходимые таблицы и привилегии:

– в скрипте /usr/share/bacula-director/make_postgresql_tables внести изменения:

```
db_name=${db_name:-bacula}
psql -U bacula -h 10.0.0.10 -f - -d ${db_name} $* <<END-OF-DATA
```

– в скрипте /usr/share/bacula-director/grant_postgresql_privileges внести изменения:

```
db_user=${db_user:-bacula}
bindir=/usr/bin
db_name=${db_name:-bacula}
db_password=bacula
if [ "$db_password" != "" ]; then
   pass="password '$db_password'"
fi
$bindir/psql -U bacula -h 10.0.0.10 -f - -d ${db_name} $*  <<END-
OF-DATA
```

– сохранить изменения и выполнить скрипты:

```
make_postgresql_tables
grant_postgresql_privileges
```

5) на машине, где будет работать Storage Daemon, необходимо создать каталог /back, в котором будут храниться резервные копии данных, и присвоить каталогу владельца bacula:

```
$ sudo mkdir /back
$ sudo chown -R bacula /back
```

6) на машине, где будет работать File Daemon, необходимо создать каталог /etc2, в который будут восстанавливаться данные из резервной копии:

```
$ sudo mkdir /etc2
```

Если подготовительные настройки выполнены корректно, БД стартует без ошибок и скрипты выполнились без ошибок, то можно приступить к настройке Bacula.

Подготовка Bacula к работе заключается в настройке каждого компонента в отдельности и последующей настройке их взаимодействия.

Далее приступить к настройке Director Daemon в конфигурационном файле /etc/bacula/bacula-dir сервера dd.ex.net. В первую очередь определить основные параметры в секции Director. На начальном этапе важно установить параметры Name и Password. Name задает уникальное имя Director Daemon, а Password - пароль, который будет использоваться при соединениях BC с DD. Остальные параметры можно оставить в значениях по умолчанию.

```
Director { # define myself
     Name = bacula-dir
     DIRport = 9101 # where we listen for UA connections
     QueryFile = "/etc/bacula/scripts/query.sql"
     WorkingDirectory = "/var/lib/bacula"
     PidDirectory = "/var/run/bacula"
     Maximum Concurrent Jobs = 1
     Password = "1" # Console password
     Messages = Daemon
     DirAddress = 10.0.0.10
}
```

Следующей группой параметров, которые необходимо определить, является секция Catalog. Здесь необходимо указать реквизиты доступа к БД, а также назначить уникальное имя данного Bacula Catalog с помощью параметра Name.

```
Catalog {
     Name = MyCatalog
     # Uncomment the following line if you want the dbi
     PS. driver
     # dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
     dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
     DB Address = 10.0.0.10
}
```

Определить SD, на который будет производиться передача данных для дальнейшей записи на устройство хранения. Storage Daemon настроен и готов к работе, необходимо определить реквизиты доступа к нему в секции Storage файла bacula-dir.conf. Основные параметры:

1) Name – уникальное имя, использующееся для адресации секции Storage в рамках файла bacula-dir.conf;

2) Device и MediaType дублируют одноименные параметры файла bacula-sd.conf;

3) Password содержит пароль, который будет использоваться при подключении к Storage Daemon.

```
Storage {
     Name = File
     # Do not use "localhost" here
     Address = 10.0.0.20 # N.B. Use a fully qualified name here
     SDPort = 9103
     Password = "1"
     Device = FileStorage
     Media Type = File
}
```

Секция Pool определяет набор носителей информации и параметры, определяющие то, как SD будет их обрабатывать. Каждый Pool взаимодействует с устройством хранения данных, и поэтому необходимо создать столько же пулов, сколько определено устройств хранения. Фактически если для каждого File Daemon вы определяете отдельное устройство, то для каждого FD необходимо определить и Pool. Основные параметры:

1) Name определяет уникальное имя пула;

2) Pool Type определяет тип, и для резервных копий должен быть установлен в значение Backup;

3) Maximum Volume Jobs рекомендуется установить в значение 1. Это будет означать, что в рамках одного носителя данных могут быть размещены резервные данные, полученные в ходе выполнения только одного задания. Носитель данных - это устройство, на которое непосредственно записываются данные (оптические диски, магнитные ленты). Если размер созданной резервной копии много меньше размера носителя, то имеет смысл сохранить на него и другие копии, которые будут создаваться в будущем. Но если говорится о файлах, то желательно придерживаться правила «один файл - одна копия», т.е. в одном файле Bacula должны храниться резервные данные, которые были сформированы в рамках выполнения одного задания. Для каждого последующего будут создаваться новые файлы;

4) Volume Retention – время, по прошествии которого, данные о резервной копии, хранящейся на носителе, будут удалены из каталога. Для обеспечения

работоспособности Bacula необходимо помнить о том, что информация обо всех зарезервированных файлах хранится в БД, по записи на каждый файл. Если резервируются тысячи файлов, то очень скоро БД станет огромной, что может затруднить работу Bacula. Поэтому очень важно своевременно очищать базу от устаревшей информации. При этом сам носитель информации не будет очищен автоматически. Он будет промаркирован как устаревший, но всегда можно будет использовать его для восстановления данных в ручном режиме;

5) Maximum Volumes – максимальное количество носителей (в нашем случае файлов), доступных в данном пуле. Параметр Recycle указывает на необходимость повторного использования носителей, помеченных как устаревшие. При этом реальная перезапись носителя произойдет лишь в случае, когда свободных носителей не останется. Свободные носители определяются из параметра Maximum Volumes;

6) AutoPrune указывает на необходимость удаления устаревших записей из Bacula Catalog автоматически после завершения выполнения очередного задания;

7) Label Format определяет префикс, который будет использован Bacula для маркирования носителей информации, в нашем случае – для именования файлов;

8) Storage указывает на имя устройства хранения данных, указанного в параметре Name секции Storage файла bacula-dir.conf.

```
Pool {
     Name = Default
     Pool Type = Backup
     Recycle = yes # Bacula can automatically recycle Volumes
     AutoPrune = yes # Prune expired volumes
     Volume Retention = 1 month # one year
     Maximum Volume Jobs = 1
     Maximum Volumes = 32
     Storage = File
     Label Format = "volume-"
}
```

Секция FileSet позволяет предопределить несколько наборов резервируемых файлов. Например, один набор для Windows, другой – для Linux или один для серверов, а другой – для рабочих станций. Параметр Name определяет уникальное имя набора.

Секция Include содержит пути к резервируемым файлам/каталогам, а Exclude – пути к файлам и каталогам, которые необходимо исключить из списка резервируемых. В

секции Include возможна секция Options, в которой определяются параметры резервирования. Основные параметры:

1) signature указывает алгоритм вычисления контрольных сумм файлов;

2) compression указывает алгоритм компрессии файлов;

3) recurse указывает на необходимость рекурсивного резервирования, включая подкаталоги и файлы;

4) File указывает на каталог, который копируется;

5) xattrsupport указывает на возможность включения поддержки расширенных атрибутов, это обязательный параметр для работы с мандатными метками.

```
FileSet {
     Name = "Catalog"
     Include {
          Options {
               signature = MD5
               compression = GZIP
               # recurse = yes
               aclsupport = yes
               xattrsupport = yes
          }
          File = /etc
     }
}
```

Все настройки связываются воедино с помощью секции Job, в которой дается задание планировщику по выполнению резервирования данных. Основные параметры:

1) Type указывает на тип задания. Типов существует несколько. Здесь достаточно указать Backup;

2) Schedule указывает на предопределенное расписание, согласно которому будет выполняться резервирование данных. Все расписания определены здесь же, в файле bacula-dir.conf;

3) Where указывает на каталог, в котором будут восстанавливаться данные из резервной копии;

4) Write Bootstrap указывает путь к файлу, в который будет записываться информация, с помощью которой данные могут быть восстановлены из резервной копии без наличия подключения к Bacula Catalog. Вместо %n будет подставлено значение параметра Name.

```
Schedule {
     Name = "DailyCycle"
     Run = Full daily at 16:10
     # Run = Differential 2nd-5th sun at 23:05
     Run = Incremental mon-sat at 23:05
}
Job {
     Name = "RestoreFiles"
     Type = Restore
     Client= bacula-fd
     FileSet="Catalog"
     Storage = File
     Pool = Default
     Messages = Standard
     Where = /etc2
}
Job {
     Name = "BackupCilent1"
     Type = Backup
     Client = bacula-fd
     FileSet = "Catalog"
     Schedule = "DailyCycle"
     Messages = Standard
     Pool = Default
     Write Bootstrap = "/var/lib/bacula/Client1.bsr"
     Priority = 1
}
```

Указать параметры единственного Агента.

```
Client {
     Name = bacula-fd
     Address = 10.0.0.30
     FDPort = 9102
     Catalog = MyCatalog
     Password = "1" # password for FileDaemon
     File Retention = 30 days # 30 days
```

```
    Job Retention = 6 months # six months
    AutoPrune = yes # Prune expired Jobs/Files
}
```

Закомментировать все остальные секции: Job, JobDefs, Client и Console, на данном этапе они не понадобятся. Трафик данных будет идти по тем портам, которые указаны в конфигурационных файлах каждого из компонентов Bacula.

Далее настроить доступ к DD со стороны Bacula Console в файле /etc/bacula/bconsole.conf сервера dd.ex.net:

```
Director {
    Name = bacula-dir
    DIRport = 9101
    address = 10.0.0.10
    Password = "1"
}
```

На машине, где будет Director Daemon удалить пакеты bacula-sd и bacula-fd:

```
$ sudo apt-get remove bacula-sd
$ sudo apt-get remove bacula-fd
```

Конфигурационные файлы bacula-sd и bacula-fd в /etc/bacula следует либо переименовать, либо удалить. Сервисы bacula-sd и bacula-fd следует остановить:

```
$ sudo /etc/init.d/bacula-sd stop
$ sudo /etc/init.d/bacula-fd stop
```

Далее начать подготовку Storage Daemon, который будет отвечать за непосредственную работу с устройством хранения данных. Bacula поддерживает широкий спектр устройств, начиная от оптических дисков и заканчивая полнофункциональными ленточными библиотеками. Например, самый распространенный вариант – обычный жесткий диск с существующей ФС (например, ext3). Итак, на сервере sd.ex.net необходимо отредактировать файл /etc/bacula/bacula-sd.conf. В секции основных параметров – Storage необходимо определить параметр Name, который задает уникальное имя Storage Daemon. Остальные параметры можно оставить в значениях по умолчанию.

Секция Director необходима для указания уникального имени DD и пароля, с которым этот DD может подключаться к SD. Секций может быть несколько, что дает возможность использовать единый сервер хранения данных для нескольких систем резервирования. Все остальные секции Director, найденные в файле, закомментировать.

```
Storage { # definition of myself
```

```
        Name = bacula-sd
        SDPort = 9103 # Director's port
        WorkingDirectory = "/var/lib/bacula"
        Pid Directory = "/var/run/bacula"
        Maximum Concurrent Jobs = 20
        SDAddress = 10.0.0.20
}
Director {
        Name = bacula-dir
        Password = "1"
}
```

Но основные настройки, определяющие взаимодействие с устройствами хранения, находятся в секции Device. Ниже приведены параметры, необходимые для хранения резервных копий в рамках существующей ФС, подключенной в каталог /back:

1) Name определяет уникальное имя подключенного устройства. Если планируется создавать изолированные друг от друга резервные копии для каждого из File Daemon, то необходимо создать несколько секций Device с уникальными именами. В противном случае резервируемые файлы со всех FD будут размещаться в одном и том же файле, что может затруднить дальнейшее обслуживание системы;

2) Media Type определяет произвольное уникальное имя, которое будет использоваться Bacula при восстановлении данных. Согласно ему определяется устройство хранения, с которого будет производиться восстановление. Если резервные копии хранятся в файлах, то для каждой секции Device должен быть задан уникальный Media Type;

3) Archive Device указывает путь к файлу устройства в каталоге /dev или путь к каталогу, в котором будут размещаться резервные копии;

4) Device Type определяет тип устройства. Для размещения в существующей ФС указывается File;

5) Random Access указывает на возможность случайной (непоследовательной) адресации. Для файлов указывается Yes;

6) RemovableMedia указывает, возможно ли извлечение устройства хранения. Необходимо для ленточных устройств, приводов оптических дисков и т.д.

Для файлов устанавливается в значение No. Параметр LabelMedia указывает на необходимость автоматического маркирования носителей информации.

```
Device {
```

```
        Name = FileStorage
        Media Type = File
        Archive Device = /back
        LabelMedia = yes; # lets Bacula label unlabeled media
        Random Access = Yes;
        AutomaticMount = yes; # when device opened, read it
        RemovableMedia = no;
        AlwaysOpen = no;
}
```

Для базовой настройки этого достаточно.

На машине, где будет Storage Daemon, удалить пакет bacula-fd:

```
$ sudo apt-get remove bacula-fd
```

Конфигурационный файл bacula-fd в /etc/bacula следует либо переименовать, либо удалить. Сервис bacula-fd следует остановить:

```
$ sudo /etc/init.d/bacula-fd stop
```

Для настройки File Daemon на рабочей станции fd.ex.net используется файл /etc/bacula/bacula-fd, в котором для базовой настройки достаточно лишь определить параметры секции Director, где указывается пароль, который будет использовать DD при подключении к FD, а также секции FileDaemon, где указываются настройки FD. Все остальные секции Director, найденные в файле, необходимо закомментировать.

```
Director {
        Name = bacula-dir
        Password = "1"
}
```

В секции FileDaemon на данном этапе необходим только параметр Name, в котором указывается уникальное имя File Daemon:

```
FileDaemon { # this is me
        Name = bacula-fd
        FDport = 9102 # where we listen for the director
        WorkingDirectory = /var/lib/bacula
        Pid Directory = /var/run/bacula
        Maximum Concurrent Jobs = 20
        FDAddress = 10.0.0.30
}
```

На машине, где будет File Daemon, удалить пакет bacula-sd:

```
$ sudo apt-get remove bacula-sd
```

Конфигурационный файл bacula-sd в /etc/bacula следует либо переименовать, либо удалить.

Сервис bacula-sd остановить:

```
$ sudo /etc/init.d/bacula-sd stop
```

Далее запустить все компоненты соответствующими командами, данными на соответствующих серверах:

```
$ sudo /etc/init.d/bacula-director restart
$ sudo /etc/init.d/bacula-sd restart
$ sudo /etc/init.d/bacula-fd restart
```

После этого Bacula будет работать. Управление Bacula осуществляется через bconsole. Настройки каталогов, заданий, расписаний и прочие задаются в конфигурационных файлах.

Для тестовой проверки необходимо:

– выполнить bconsole;

– выполнить run;

– выбрать job 1;

– войти в меню, набрав mod;

– выбрать 1 (Level);

– выбрать 1 (Full);

– подтвердить выполнение, набрав yes.

Будет создана резервная копия данных в каталоге /back на машине с Storage Daemon.

Для восстановления объектов ФС с установленными мандатными атрибутами необходимо запустить консоль управления Bacula с необходимыми привилегиями, выполнив команду:

```
sudo runcon t=CHCTX,SETXATTR_OMITMAC,OMIT bconsole
```

Для восстановления данных из резервной копии необходимо:

– выполнить restore;

– выбрать пункт 12;

– ввести номер job id;

– указать параметр маркировки mark *;

– подтвердить выполнение командой done.

Данные из резервной копии будут восстановлены в каталоге /etc2 на машине с File Daemon.

Также управление Bacula возможно с помощью графической утилиты bacula-console-qt, ее конфигурационный файл расположен в /etc/bacula/bat.

3.12. Поддержка средств двухфакторной аутентификации

Для повышения надежности аутентификации и снижения риска компрометации паролей применяется принцип многофакторной аутентификации.

Многофакторная аутентификация является расширенным видом аутентификации, при котором используется более одного «доказательства механизма аутентификации». Как правило, выделяют 3 фактора аутентификации:

– знание – информация, которую знает субъект: ввод пароля или пин-кода.

– владение – вещь, которой обладает субъект: предоставление физического устройства или носителя (смарт-карта, USB-токен, и т.п.).

– свойство, которым обладает субъект–биометрия, природные уникальные отличия: лицо, отпечатки пальцев, радужная оболочка глаз, капиллярные узоры, последовательность ДНК.

В большинстве случаев в качестве многофакторной аутентификации применяется двухфакторная аутентификация. Для стационарных систем, не связанных с сетями общего доступа, в качестве второго фактора используется предоставление физического устройства или носителя, содержащего дополнительную аутентификационную информацию. Дополнительная информация может представлять собой размещенный на устройстве сертификат пользователя.

3.12.1. Электронный идентификатор Guardant ID

Для применения в системах, обрабатывающих информацию ограниченного доступа, предназначен электронный идентификатор Guardant ID. В отличие от других похожих устройств, только этот идентификатор сертифицирован ФСТЭК России на соответствие требованиям РД НДВ – по 2-му уровню контроля.

Электронный идентификатор Guardant ID – это устройство, подключаемое к USB-порту компьютера (непосредственно или через удлинитель) и служащее для хранения идентификационных данных пользователя в энергонезависимой памяти.

Особенностью устройства является наличие уникального неизменяемого идентификатора и области защищенной энергонезависимой памяти, для доступа к которой используется кодирование с уникальным для каждого Guardant ID паролем.

Для идентификации пользователя при выполнении команд чтения/записи может быть использован PIN-код длиной от 1 до 32 байт. PIN-код может быть изменен штатным образом в случае получения доступа к памяти устройства.

Память ключа может быть инициализирована (полностью обнулена) специальной командой, требующей предъявления особого Master PIN-кода. Master PIN-код задается не этапе производства Guardant ID и не может быть считан или изменен.

В настоящее время применяются электронные идентификаторы Guardant ID с объемом памяти 16 Кбайт.

3.12.2. Применение Guardant ID

Электронный идентификатор Guardant ID является только персональным идентификатором с защищенной паролем памятью. Для реализации на его основе механизмов двухфакторной аутентификации требуется:

– наличие средств управления электронными идентификаторами для выполнения операций инициализации, выдачи пользователям, назначения и смены пароля/PIN и проверки функционирования;

– обеспечение контроля сложности и использования пароля (PIN), поскольку пароль доступа к памяти и признак необходимости его проверки не проверяются на корректность устройством и доступны для изменения самим пользователем;

– сопоставление пользователя с устройством, так как само устройство не содержит никакой информации о своей принадлежности;

– наличие PAM модуля аутентификации, обеспечивающего применение идентификатора для контроля доступа при входе пользователя с систему.

Перечисленные выше задачи реализуются следующим образом.

Для сопоставления пользователя с устройством используется реестр выданных (зарегистрированных) идентификаторов, расположенный в файле /etc/grdid/dongles.

При инициализации идентификатора, он регистрируется в реестре, при очистке — удаляется из него.

Корректность PIN и его применения производятся утилитой администрирования и PAM модулем с использованием настроек сложности пароля, заданных в конфигурационном файле /etc/grdid/grdid.conf, например:

```
minlen=8
dcredit=0
ucredit=0
lcredit=0
ocredit=0
minclass=0
maxrepeat=0
```

```
ignoreusers=root
```

где конфигурационные параметры совпадают с аналогичными для pam_cracklib:

– minlen=N — минимальная длина для пароля учетной записи, может зависеть от остальных параметров;

– dcredit=N — количество числовых символов; значение большее нуля задает максимальное количество, при этом каждый символ уменьшает требование к минимальной длине пароля, значение меньшее нуля задает минимальное количество символов такого типа;

– ucredit=N — количество символов верхнего регистра; значение большее нуля задает максимальное количество, при этом каждый символ уменьшает требование к минимальной длине пароля, значение меньшее нуля задает минимальное количество символов такого типа;

– lcredit=N — количество символов нижнего регистра; значение большее нуля задает максимальное количество, при этом каждый символ уменьшает требование к минимальной длине пароля, значение меньшее нуля задает минимальное количество символов такого типа;

– ocredit=N — количество остальных символов; значение большее нуля задает максимальное количество, при этом каждый символ уменьшает требование к минимальной длине пароля, значение меньшее нуля задает минимальное количество символов такого типа;

– minclass=N — минимальное число разных типов символов;

– maxrepeat=N — максимально число идущих подряд одинаковых символов;

– ignoreusers=root — список пользователей через запятую, для которых разрешен вход в режиме strict, если для них не зарегистрирован идентификатор.

ВНИМАНИЕ! ПО УМОЛЧАНИЮ ПАРАМЕТР IGNOREUSERS СОДЕРЖИТ ЗНАЧЕНИЕ ROOT, ЧТО МОЖЕТ СНИЗИТЬ ЗАЩИЩЕННОСТЬ. РЕКОМЕНДУЕТСЯ ВЫДАВАТЬ ИДЕНТИФИКАТОРЫ ПОЛЬЗОВАТЕЛЯМ С ПРИВИЛЕГИЯМИ АДМИНИСТРИРОВАНИЯ И НЕ УКАЗЫВАТЬ ЭТИХ ПОЛЬЗОВАТЕЛЕЙ В ПАРАМЕТРЕ IGNOREUSERS.

При входе пользователя PAM модуль производит следующие действия:

– определяет необходимость применения Guardant ID входящим пользователем (по реестру выданных устройств);

– проверяет или предлагает предъявить устройство с соответствующим номером (если он отсутствует или предъявлен неверный);

– проверяет идентификатор устройства и введенный PIN пользователя;

– читает с устройства необходимую информацию (хеш, PIN и слово состояния);

– проверяет PIN и слово состояния (длину и сложность PIN, признак его проверки);

– при успешном выполнении ранее перечисленных действий PAM модуль разрешает пользователю вход в систему.

Таким образом, достигаются следующие цели:

– устройство сопоставляется с пользователем;

– с помощью локального реестра обеспечивается разграничение по местам входа;

– при входе пользователю предлагается предъявить конкретное устройство;

– пользователь имеет возможность смены своего PIN;

– пользователь не имеет возможности отключить проверку PIN или упростить его по длине или сложности;

– незарегистрированные устройства и устройства, для которых была сброшена проверка PIN или PIN небезопасен, не принимаются для входа.

Примечание. Существует возможность ограничить пользователя только операциями чтения с устройства без возможности записи (с помощью правил udev), что повысит безопасность, но не позволит ему самостоятельно менять PIN.

3.12.3. Управление электронными идентификаторами – grdid-tool

Для управления электронными идентификаторами Guardant ID предназначена утилита grdid-tool, имеющая следующий формат вызова:

```
$ sudo grdid-tool [опции]
```

Основным назначением утилиты является инициализация и выдача устройств пользователям. При необходимости утилита может быть использована самим пользователем для смены своего PIN. При этом у него будет запрошено старое значение PIN.

В таблице 24 приведены основные опции команды grdid-tool.

Таблица 24

Опция	Описание
-h, --help	Справка по способу вызова и опциям команды
-v, --verbose	Подробный отчет о выполняемых действиях
-f, --force	Не спрашивать подтверждение
-i, --info	Отображение информации об идентификаторе
-e, --erase	Стирание идентификатора. При успешном выполнении операции идентификатор удаляется из реестра выданных идентификаторов

--init-pin	Инициализация PIN пользователя, требует задания пользователя опцией -u. При успешном выполнении операции идентификатор регистрируется в реестре выданных идентификаторов
--verify-pin	Проверить PIN
--change-pin	Смена PIN пользователя (самим пользователем, требует старое значение PIN)
--so-pin ЗНАЧЕНИЕ	PIN администратора (Master PIN код), может быть задан как env:VARIABLE, тогда значение будет взято из соответствующей переменной окружения
--pin ЗНАЧЕНИЕ	PIN пользователя, может быть задан как env:VARIABLE, тогда значение будет взято из соответствующей переменной окружения
-u ПОЛЬЗОВАТЕЛЬ, --user ПОЛЬЗОВАТЕЛЬ	Использовать в качестве владельца идентификатора учетную запись ПОЛЬЗОВАТЕЛЬ
-l, --list	Вывести список зарегистрированных (выданных) идентификаторов
-r, --unregister	Удалить запись о ключе пользователя

3.12.4. Использование Guardant ID для доступа к системе – pam_grdid

Применение идентификатора для контроля доступа при входе пользователя в систему обеспечивается специальным PAM модулем аутентификации pam_grdid.

При попытке входа в систему у пользователя запрашивается PIN представленного электронного идентификатора Guardant ID, в случае соответствия представленного идентификатора пользователю и успешной проверки PIN процедура входа продолжается. В противном случае происходит отказ доступа к системе с соответствующей регистрацией попытки несанкционированного доступа.

Указанный модуль должен быть встроен в требуемые PAM сценарии следующим образом (на примере /etc/pam.d/common-auth):

```
# here are the per-package modules (the "Primary" block)
auth [success=3 default=die ignore=ignore]  pam_grdid.so
auth [success=2 default=ignore]    pam_unix.so nullok_secure
auth [success=1 default=ignore]    pam_winbind.so krb5_auth ...
```

При использовании модуля могут применяться опции:

– debug – вывод отладочной информации;

– try_only – попытка проведения проверки (возможен вход без предоставления устройства);

– no_prompt – не выводить предложение подключить идентификатор;

– strict – строгий режим: запрет входа без использования идентификатора Guardant ID;

– set_pass – установить введенный PIN в качестве AUTHTOK для последующих модулей аутентификации.

При необходимости с помощью ключей PAM модуля можно настроить желаемый режим использования двухфакторной аутентификации (таблица 25).

Таблица 25

Режим	Порядок настройки режима	no_prompt	Вход пользователей, не имеющих Guardant ID
Общий	Предлагается подключить идентификатор, вход по PIN, при неверном PIN отказ входа	Используется подключенный идентификатор, вход по PIN, если идентификатор не предъявлен или неверный PIN — отказ входа	Стандартным способом по паролю
try_only	Предлагается подключить идентификатор, вход по PIN, при неверном PIN переход на другие способы аутентификации (по паролю)	Используется подключенный идентификатор, вход по PIN, если идентификатор не предъявлен или неверный PIN – переход на другие способы аутентификации (по паролю)	Стандартным способом по паролю
strict	Предлагается подключить идентификатор, вход по PIN, при неверном PIN отказ входа	Используется подключенный идентификатор, вход по PIN, если идентификатор не предъявлен или неверный PIN — отказ входа	Запрещен, кроме пользователей указанных в конфигурационном файле параметром ignoreusers

Примечание. При необходимости может быть реализован сквозной вход в систему с помощью опции set_pass, поскольку в этом случае PAM модуль сохраняет PIN в PAM стеке для других модулей. При этом в конфигурационном параметре «сроки вызова модуля для опции success» должно быть задано значение ignore. В случае задания PIN равного паролю Linux или Kerberos будет осуществляться сквозной вход без дополнительных запросов паролей. При этом фрагмент PAM сценария может выглядеть следующим образом:

```
auth [success=ignore default=die ignore=ignore]  pam_grdid.so set_pass
auth [success=2 default=ignore]    pam_unix.so use_first_pass
```

3.12.5. Настройка ssh для доступа с использованием Guardant ID

Для корректной работы удаленного входа с помощью ssh необходимо выполнить следующие изменения в конфигурационном файле /etc/ssh/sshd.conf:

– включить параметр ChallengeResponseAuthentication в on;

– выключить параметр PasswordAutentication.

3.13. Средства организации домена

Средства организации домена представляют собой совокупность технологий, протоколов и сетевых служб, обеспечивающих объединение в одной сети логически связанных объектов на основе доменного принципа построения сети. Объединение может быть выполнено, например, по территориальному признаку или по принадлежности какой-либо организации или ее части. При этом предусматривается несколько категорий объектов: ресурсы (например, принтеры), службы (например, электронная почта) и учетные записи пользователей и компьютеров.

Организация домена позволяет обеспечить:

– единую среду именования связанных в одной сети объектов (компьютеров, ресурсов, учетных записей пользователей);

– сквозную аутентификацию в сети;

– централизацию хранения информации об объектах домена и окружении пользователей;

– единообразие настройки пользовательской рабочей среды;

– централизованное администрирование и управление.

3.13.1. Архитектура

Примером реализации средств организации домена являются службы каталога Active Directory корпорации Microsoft для операционных систем семейства Windows Server.

Далее описывается совместимая с Active Directory реализация на основе программного пакета с открытым исходным кодом Samba 4.

Основными компонентами являются следующие технологии и реализующие их сетевые службы:

– LDAP-совместимая служба каталогов для централизованного хранения информации о структуре домена и свойствах его объектов;

– служба доверенной аутентификации Kerberos V5;

– служба системы доменных имен DNS;

– служба файлового сервера Samba/CIFS;

– служба управления групповыми политиками.

3.13.1.1. Служба каталогов LDAP

Служба каталогов LDAP предоставляет клиенту доступ по протоколу LDAP к службе распределенного каталога X.500, являющейся иерархической БД для централизованного хранения и управления информацией обо всех именованных объектах сети (ресурсах, приложениях и пользователях). Служба предоставляет информацию об объектах, позволяет организовывать объекты, управлять доступом к ним, а также устанавливает правила безопасности.

Каждый объект каталога уникально определяется своим именем (DN – Distinguished Name), имеет набор атрибутов и может содержать иерархически подчиненные объекты. Атрибуты являются составляющей структуры объекта, зависят от его типа и определяются в схеме каталога. Схема определяет, какие типы объектов могут существовать, и состоит из двух типов объектов: объекты классов схемы и объекты атрибутов схемы. Один объект класса схемы определяет один тип объекта каталога (например, объект «Пользователь»), а один объект атрибута схемы определяет атрибут, который объект может иметь (например, атрибут «Имя пользователя»).

Служба каталогов поддерживает расширяемый состав схем, описывающих структуру информации, необходимой для функционирования других компонентов домена, и может содержать информацию о нескольких доменах, которые идентифицируются своими структурами имен DNS – пространствами имен.

В части средств организации домена служба каталогов LDAP используется в качестве источника данных для базовых системных сервисов ОС (NSS) для получения информации об объектах сети, атрибутах и свойствах учетных записей.

3.13.1.2. Служба доверенной аутентификации Kerberos V5

Служба доверенной аутентификации домена использует сетевой протокол аутентификации Kerberos, который предусматривает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними с учетом возможности перехвата и модификации нарушителем передаваемых в незащищенной среде сетевых пакетов.

Основным компонентом подобной службы является центр распределения ключей (KDC –Key Distribution Center). KDC хранит БД с информацией об учетных записях всех клиентов сети. Вместе с информацией о каждом абоненте в базе KDC хранится криптографический ключ, известный только этому абоненту и службе KDC.

При обращении клиента к серверу, клиент обращается к KDC, который направляет клиенту копии сеансового ключа, действующие в течение небольшого промежутка

времени. Копия сеансового ключа, направляемая клиенту, шифруется с помощью долговременного ключа данного клиента, а копия для сервера вместе с информацией о клиенте шифруется с помощью долговременного ключа сервера и выдается клиенту в виде сеансового билета/мандата («session ticket»), сохраняемого клиентом в своей кэш-памяти удостоверений («credentials cache»). При обращении к серверу, клиент использует полученный билет. Следует отметить, что полученный билет не содержит ключей клиента или сервера и его время действия ограничено.

ВНИМАНИЕ! ПОСКОЛЬКУ ВСЕ ВЗАИМОДЕЙСТВИЯ МЕЖДУ УЧАСТНИКАМИ ПРОТОКОЛА KERBEROS ИСПОЛЬЗУЮТ ВРЕМЕННЫЕ МЕТКИ, ОБЯЗАТЕЛЬНЫМ ТРЕБОВАНИЕМ ДЛЯ ФУНКЦИОНИРОВАНИЯ ЯВЛЯЕТСЯ СИНХРОНИЗАЦИЯ ВРЕМЕНИ НА КЛИЕНТЕ И СЕРВЕРЕ. СИНХРОНИЗАЦИЯ МОЖЕТ БЫТЬ ОБЕСПЕЧЕНА ИСПОЛЬЗОВАНИЕМ СЛУЖБЫ СИНХРОНИЗАЦИИ ВРЕМЕНИ NTP.

3.13.1.3. Служба системы доменных имен DNS

Система доменных имен DNS (Domain Name System) представляет собой иерархическую распределенную систему для получения информации о компьютерах, сервисах и ресурсах, входящих в глобальную или приватную компьютерную сеть. Чаще всего используется для получения IP-адреса по имени компьютера или устройства, получения информации о маршрутизации почты и т.п.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Распределенная БД DNS поддерживается с помощью иерархии DNS-серверов. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу, что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Служба системы доменных имен DNS является необходимым элементом средств организации домена, отвечающим за разыменование объектов домена при сетевом взаимодействии.

3.13.1.4. Служба файлового сервера Samba/CIFS

Служба файлового сервера Samba/CIFS предназначена для создания и управления сетевыми файловыми ресурсами. При этом осуществляется управление доступом на основе стандартных прав доступа к файлам, определяемым по атрибутам обращающегося клиента, после прохождения им аутентификации. В основу положена сетевая файловая система CIFS, работающая по протоколу SMB/CIFS.

Использование данной службы позволяет обеспечить:

– организацию удаленного рабочего каталога пользователя, что позволяет

пользователю работать с любого доступного ему компьютера в домене;

– создание разделяемых файловых ресурсов в гетерогенных сетях Linux/Windows;

– совместное использование принтеров в гетерогенных сетях Linux/Windows.

3.13.1.5. Служба управления групповыми политиками

Служба управления групповыми политиками предназначена для облегчения управлением множествами объектов домена за счет задания групповых параметров и настроек, что позволяет обеспечить в домене:

– единообразие настройки пользовательской рабочей среды;

– устанавливать или обновлять ПС на множестве компьютеров домена.

3.13.2. Реализация

Для обеспечения задач организации домена применяются утилиты программного пакета с открытым исходным кодом Samba 4 и набор дополнительных программных средств (в том числе с графическим пользовательским интерфейсом), облегчающих создание и администрирование домена.

3.13.2.1. Состав

Средства организации домена включают в себя пакеты, приведенные в таблице 26.

Таблица 26

Наименование	Описание
ndds-ctl	Утилита для создания/удаления и управления контроллером домена
ndds-ctl-ness	Расширение для утилиты ndds-ctl для управления мандатными атрибутами пользователей
ndds-client	Утилита для ввода/вывода хоста в домен и управления клиентской частью
ndds-admin-srv	Сервер администрирования
ndds-admin-srv-configure	Скрипт принудительного переконфигурирования сервера администрирования
ndds-admin	Графическая часть утилиты администрирования

3.13.2.2. Установка

Пакеты ndds-ctl, ndds-ctl-ness и ndds-admin-srv должны быть установлены на сервере, на котором будет развернут контроллер домена.

Пакет ndds-client устанавливается на клиентские рабочие станции, которые будут входить в домен.

Пакет ndds-admin устанавливается на рабочую станцию администратора домена.

При установке пакетов организации домена автоматически устанавливаются все необходимые зависимости.

3.13.2.3. Настройка

После успешной установки пакетов по организации домена утилиты полностью готовы к работе, дополнительные настройки не требуются.

При включенном межсетевом экране на контроллере домена необходимо в приложении «Управление межсетевым экраном» импортировать правила для корректной работы служб. Правила находятся в /usr/share/ndds-ctl/ndds_ufw_rules.sh.

Скрипт ndds-admin-srv-configure служит для принудительного переконфигурирования серверной части утилиты администрирования.

3.13.3. Администрирование домена

Управление доменом включает в себя операции по созданию домена, изменению состава учетных записей объектов домена (компьютеров, пользователей, сервисов), настройке параметров аутентификации и внешних отношений с другими доменами.

Раздел содержит описание конкретных операций по управлению доменом с помощью утилит средств организации домена.

3.13.3.1. Создание домена (инициализация контроллера домена)

Управление доменом осуществляется средствами утилиты командной строки ndds-ctl, которая включает в себя ряд команд, исполнение которых задается ключами, указанными в таблице 27.

Таблица 27

Команда	Описание
`init`	Инициализация контроллера домена
`join`	Присоединение контроллера Samba к Windows домену
`demote`	Понижение прав контроллера домена
`createkeytab`	Создание keytab файла
`remove`	Удаление контроллера домена
`start`	Запуск служб контроллера домена
`stop`	Остановка служб контроллера домена
`restart`	Перезапуск служб контроллера домена
`backup`	Создание резервной копии
`restore`	Восстановление из резервной копии
`status`	Текущий статус служб контроллера домена
`config`	Текущая конфигурация контроллера домена

Для инициализации контроллера домена следует выполнить команду:

```
$ sudo ndds-ctl init
```

Данная команда может выполняться в двух режимах, интерактивном (по

умолчанию) и «тихом» (при указании ключа --force). При выборе интерактивного режима будут запрошены необходимые для создания домена параметры. В случае выбора «тихого» режима требуется с помощью соответствующих ключей задать необходимые параметры (таблица 28).

Таблица 28

Ключ	Описание
--interface INTERFACE	Имя сетевого интерфейса
--address ADDRESS	IP адрес
--mask MASK	Маска сети
--dns DNS	Адрес DNS сервера
--domain DOMAIN	Имя домена
--partition PARTITION	Раздел для общих файлов

Окончание таблицы 28

Ключ	Описание
--id_range ID_RANGE	Диапазон идентификаторов пользователей и групп (Разделитель ",")
--adminpass ADMINPASS	Пароль администратора

В процессе инициализации контроллера домена на экране будут появляться уведомления о настраиваемом в данный момент сервисе и результат настройки. Выполняемые действия фиксируются в лог-файле. При этом конфигурируются следующие системные файлы (с сохранением исходных версий с суффиксом «.orig») (таблица 29).

Таблица 29

Файл	Описание
/etc/fstab	Файл с опциями монтирования
/etc/network/interfaces	Базовый файл с сетевыми настройками
/etc/network.d/<domain_name>_on_<interface>	Файл с сетевыми настройками контроллера домена
/etc/hosts	Файл с алиасами доменных имен
/etc/dhcp/dhclient.conf	Файл для конфигурации resolv.conf
/etc/ntp.conf	Файл конфигурации сервера синхронизации времени
/etc/samba/smb.conf	Файл конфигурации samba
/etc/krb5.conf	Файл конфигурации Kerberos5
/etc/security/limits.conf	Файл с настройками лимитов для пользователей и групп

Для удаления контроллера домена выполнить команду:

```
$ sudo ndds-ctl remove
```

При удалении контроллера домена удаляются все конфигурационные файлы контроллера домена и возвращаются к исходному состоянию конфигурационные файлы, измененные при инициализации.

Для присоединения контроллера домена Samba к существующему домену (например, построенном на Windows) следует выполнить команду:

```
$ sudo ndds-ctl join
```

Данная команда может выполняться в двух режимах, интерактивном (по умолчанию) и «тихом» (при указании ключа --force). При выборе интерактивного режима будут запрошены необходимые для присоединения домена параметры. В случае выбора «тихого» режима требуется с помощью соответствующих ключей задать необходимые параметры (таблица 30).

Таблица 30

Ключ	Описание
-f, --force	«Тихий» режим
--dc-address	IP адрес контроллера домена
--domain	Имя домена
--interface	Имя сетевого интерфейса
--address	IP адрес хоста
--mask	Маска сети
--partition	Раздел для общих файлов
--admin	Имя администратора домена
--adminpass	Пароль администратора домена

Для понижения прав контроллера домена (при этом он перестает выполнять функции контроллера и становится просто рабочей станцией в домене) необходимо выполнить команду:

```
$ sudo ndds-ctl demote
```

Во время выполнения команды запрашивается логин администратора домена и его пароль. Эти аргументы можно задать с помощью ключей «--admin» и «--adminpass» соответственно.

Для создания файла ключей для существующей учетной записи некоторого сервиса на контроллере домена следует выполнить команду:

```
$ sudo ndds-ctl createkeytab
```

Данная команда требует указания обязательного ключа «--principal», с помощью

которого указывается имя сервисной записи, для которой будет создан файл ключей. По умолчанию файл ключей сохраняется в директории /etc/ndds_keytabs, с помощью ключа «--out» можно указать произвольный путь для создания файла ключей.

Пример вызова:

```
$ sudo ndds-ctl createkeytab --principal postgres
```

В данном случае будет создан файл ключей для сервисной записи postgres.

Файл будет создан по пути /etc/ndds_keytabs/.

Предусмотрены 4 уровня журналирования действий, которые задаются соответствующими ключами:

– verbose – фиксируются все действия с подробным описанием;

– quiet – фиксируются только сообщений об ошибках;

– debug – дополнительно выводится отладочная информация.

По умолчанию выводятся информационные сообщения и сообщения об ошибках.

3.13.3.2. Управление составом компьютеров домена

Управление составом рабочих станций в домене осуществляется средствами утилиты командной строки ndds-client, которая включает в себя ряд команд, исполнение которых задается ключами, указанными в таблице 31:

Таблица 31

Команда	Описание
join	Ввод рабочей станции в домен
unjoin	Вывод рабочей станции из домена
start	Запуск служб клиента домена
stop	Остановка служб клиента домена
restart	Перезапуск служб клиента домена
status	Текущий статус служб клиента домена
config	Текущая конфигурация клиента домена
createkeytab	Создание файла ключей

Все выполняемые действия фиксируются в лог-файле в соответствии с заданным уровнем детализации.

Для корректного добавления рабочей станции в состав домена необходимо синхронизировать время с контроллером домена. Синхронизация выполняется в автоматическом режиме в процессе добавления. Для принудительной синхронизации времени в ручном режиме следует выполнить команду:

```
$ sudo net time set -S <IP-адрес контроллера домена>
```

При этом сетевой интерфейс рабочей станции должен быть настроен в соответствии с сетевыми настройками домена.

Для ввода рабочей станции в домен следует выполнить команду:

```
$ ndds-client join
```

Данная команда может выполняться в двух режимах, интерактивном (по умолчанию) и «тихом» (при указании ключа --force). При выборе интерактивного режима будут запрошены необходимые для создания домена параметры. В случае выбора «тихого» режима требуется с помощью соответствующих ключей задать необходимые параметры (таблица 32).

Таблица 32

Ключ	Описание
--interface INTERFACE	Имя сетевого интерфейса
--address HOST_ADDRESS	IP адрес рабочей станции
--mask MASK	Маска сети
--dns DNS	IP адрес DNS сервера
--domainDOMAIN	Имя домена
--id_range ID_RANGE	Диапазон идентификаторов пользователей и групп (разделитель ",")
--admin ADMIN	Имя пользователя с правами администратора
--adminpass ADMINPASS	Пароль администратора

Для вывода рабочей станции из домена выполнить команду:

```
$ ndds-client unjoin
```

3.13.3.3. Создание файла ключей сервиса

Для создания файла ключей следует выполнить команду (с помощью команды sudo):

```
$ sudo ndds-client createkeytab
```

Данная команда требует указания обязательного ключа «--principal», с помощью которого указывается имя сервисной записи, для которой будет создан файл ключей. По умолчанию файл ключей сохраняется в директории /etc/ndds_keytabs, с помощью ключа «--out» можно указать произвольный путь для создания файла ключей.

Пример вызова:

```
$ sudo ndds-client createkeytab --principal postgres
```

В данном случае будет создан файл ключей для сервисной записи postgres.

3.13.3.4. Управление учетными записями пользователей домена

Управление осуществляется средствами графической утилиты администрирования. Для запуска утилиты на рабочей станции администратора домена выбрать в меню

приложений «Администрирование домена» или из командной строки выполнить команду:

```
$ ndds-admin
```

Для отображения системных объектов нужно нажать кнопку  ([Показать системные объекты]).

Для управления учетными записями пользователей перейти во вкладку «Пользователи» (открыта по умолчанию) (рис. 71).

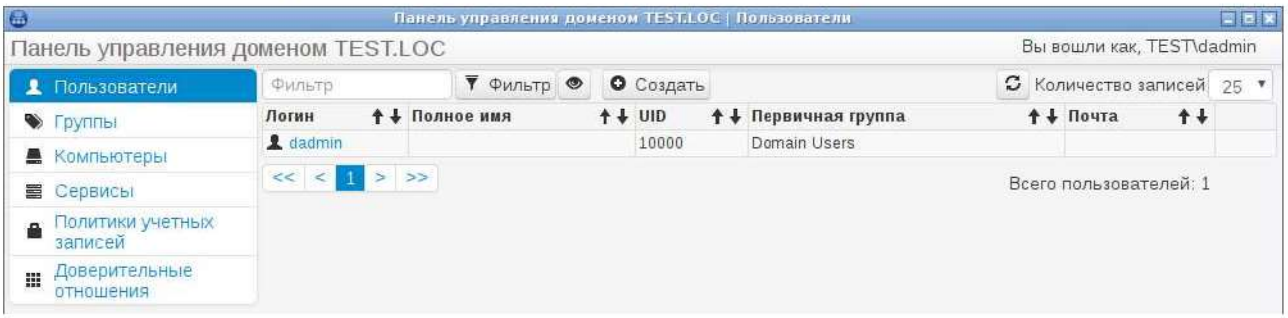


Рис. 71

Для создания учетной записи пользователя необходимо нажать ЛКМ кнопку [Создать]. Затем в появившемся диалоговом окне заполнить поля (обязательными являются «Логин», «Пароль» и «Повтор пароля») и нажать кнопку [Создать] (рис. 72).

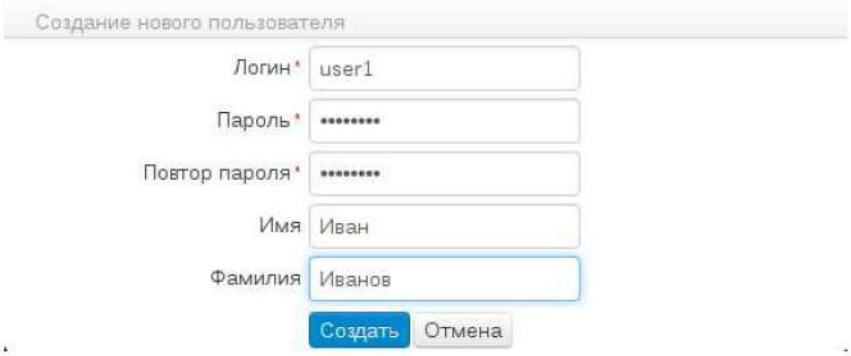


Рис. 72

Для блокировки пользователя в строке с соответствующим пользователем необходимо нажать ⊘ ([Заблокировать пользователя]), для разблокировки – нажать ⊘ ([Разблокировать пользователя]).

Для удаления учетной записи пользователя в строке с соответствующим пользователем нажать 🗑 ([Удалить пользователя]). Подтвердить удаление пользователя нажатием кнопки [Да].

Для изменения данных пользователя нажать на ссылку с соответствующим логином в столбце «Логин». Перейти в нужную вкладку, внести изменения в форму и нажать кнопку [Применить] (рис. 73-75).

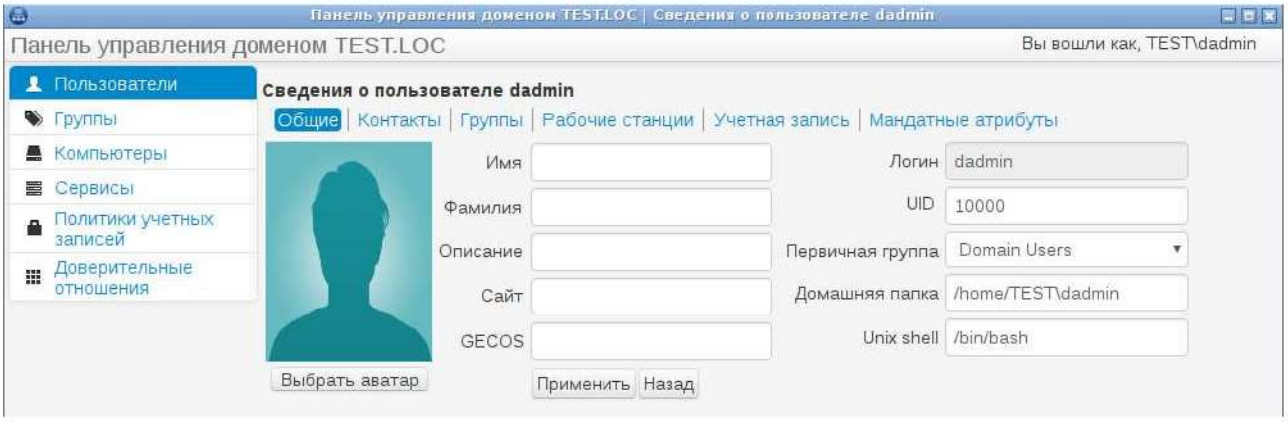

Панель управления доменом TEST.LOC | Сведения о пользователе dadmin
Панель управления доменом TEST.LOC
Вы вошли как, TEST\dadmin
Пользователи
Группы
Компьютеры
Сервисы
Политики учетных записей
Доверительные отношения
Сведения о пользователе dadmin
Общие | Контакты | Группы | Рабочие станции | Учетная запись | Мандатные атрибуты
Имя
Фамилия
Описание
Сайт
GECOS
Логин dadmin
UID 10000
Первичная группа Domain Users
Домашняя папка /home/TEST\dadmin
Unix shell /bin/bash
Выбрать аватар
Применить
Назад

Рис. 73

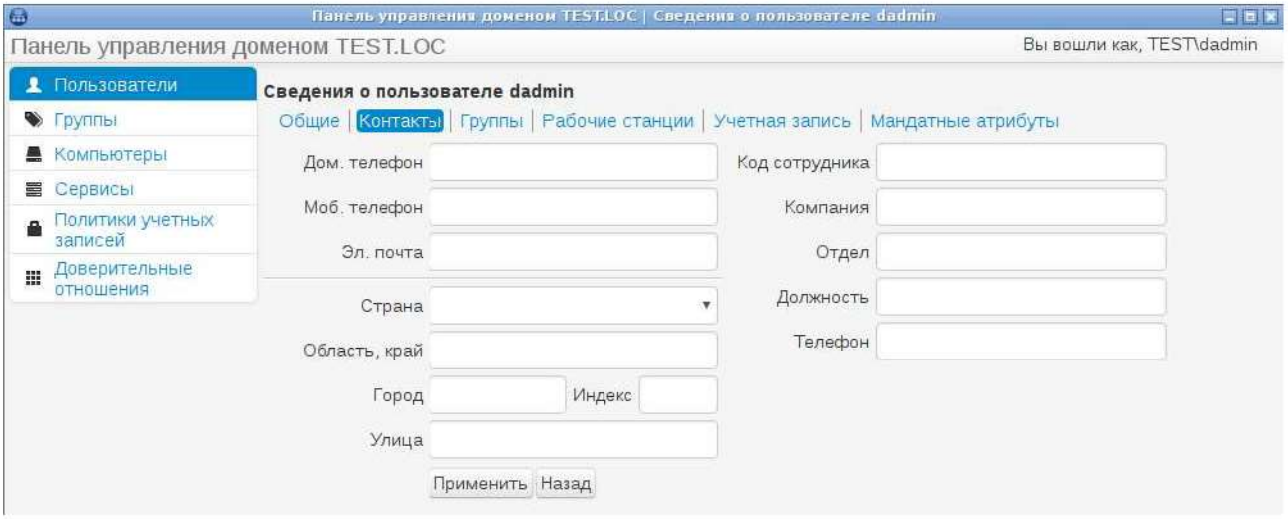

Панель управления доменом TEST.LOC | Сведения о пользователе dadmin
Панель управления доменом TEST.LOC
Вы вошли как, TEST\dadmin
Пользователи
Группы
Компьютеры
Сервисы
Политики учетных записей
Доверительные отношения
Сведения о пользователе dadmin
Общие | Контакты | Группы | Рабочие станции | Учетная запись | Мандатные атрибуты
Дом. телефон
Моб. телефон
Эл. почта
Страна
Область, край
Город
Индекс
Улица
Код сотрудника
Компания
Отдел
Должность
Телефон
Применить
Назад

Рис. 74

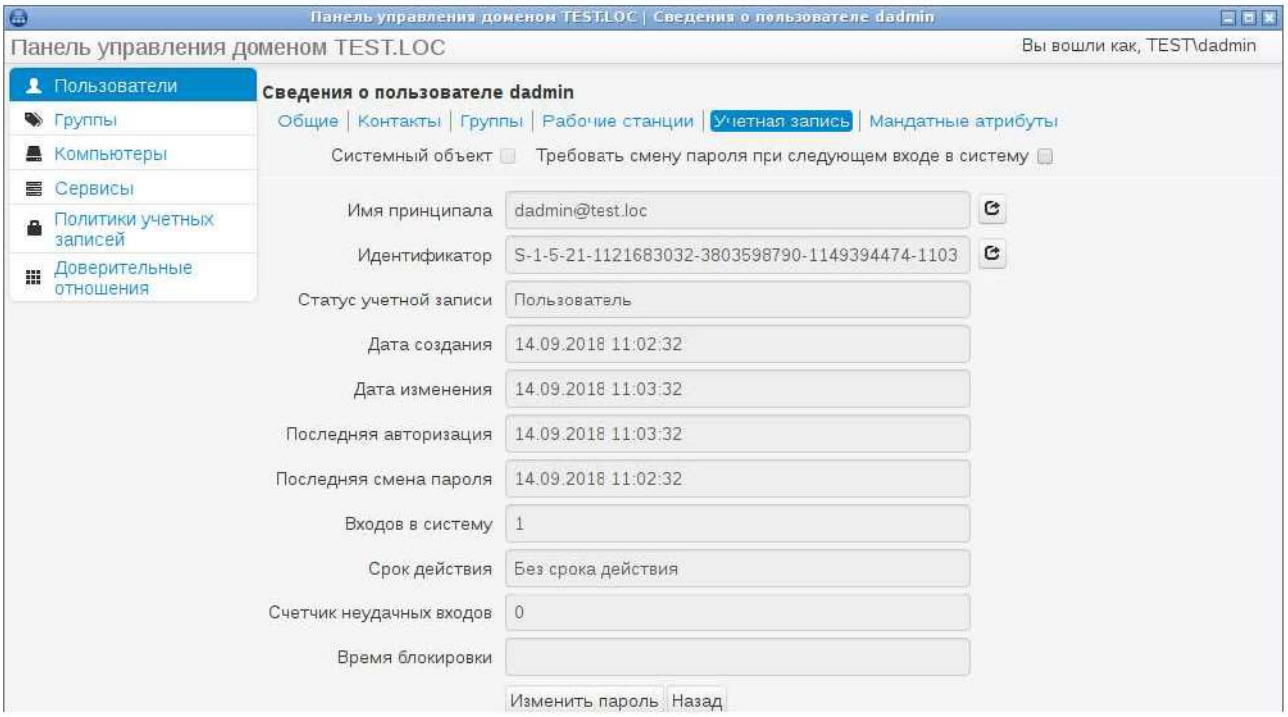

Панель управления доменом TEST.LOC | Сведения о пользователе dadmin
Панель управления доменом TEST.LOC
Вы вошли как, TEST\dadmin
Пользователи
Группы
Компьютеры
Сервисы
Политики учетных записей
Доверительные отношения
Сведения о пользователе dadmin
Общие | Контакты | Группы | Рабочие станции | Учетная запись | Мандатные атрибуты
Системный объект
Требовать смену пароля при следующем входе в систему
Имя принципала dadmin@test.loc
Идентификатор S-1-5-21-1121683032-3803598790-1149394474-1103
Статус учетной записи Пользователь
Дата создания 14.09.2018 11:02:32
Дата изменения 14.09.2018 11:03:32
Последняя авторизация 14.09.2018 11:03:32
Последняя смена пароля 14.09.2018 11:02:32
Входов в систему 1
Срок действия Без срока действия
Счетчик неудачных входов 0
Время блокировки
Изменить пароль
Назад

Рис. 75

Для изменения пароля пользователя следует выбрать пользователя, перейти на вкладку «Учетная запись» и нажать кнопку [Изменить пароль], ввести новый пароль с подтверждением и нажать кнопку [Изменить] (рис. 76).

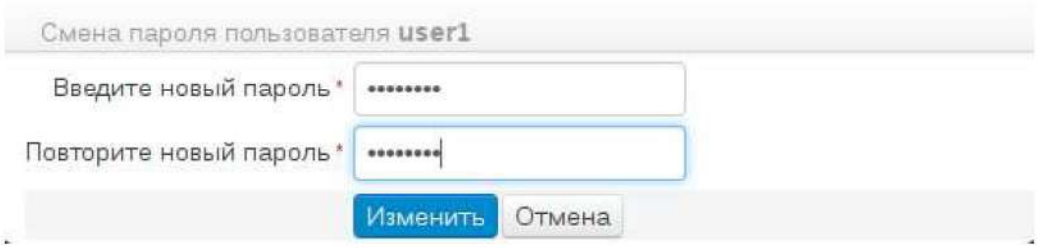


Рис. 76

Для добавления пользователя в группу нужно выбрать пользователя и на вкладке «Группы» нажать кнопку [Добавить] (рис. 77).

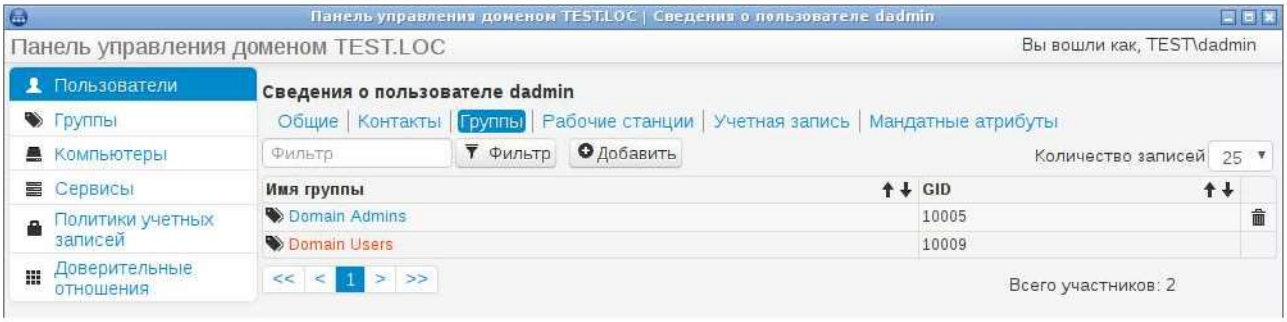
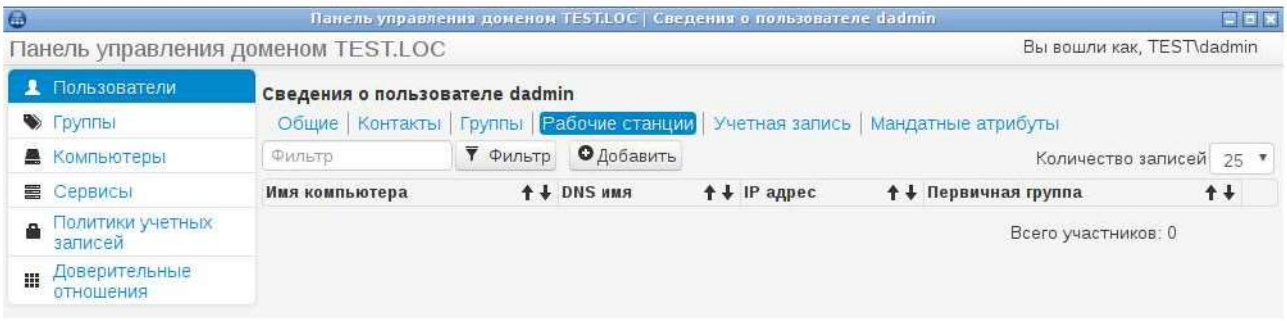


Рис. 77

Далее следует выбрать из списка нужную группу и нажать кнопку [Добавить] (для выделения нескольких групп можно использовать клавишу <Ctrl>).

Для удаления пользователя из группы нужно выбрать пользователя и на вкладке «Группы» нажать кнопку 🗑 ([Удалить пользователя из группы]).

Для добавления разрешенной рабочей станции нужно нажать на логин пользователя в столбце «Логин». На вкладке «Рабочие станции» нажать на кнопку [Добавить] (рис. 78).



Далее следует выбрать из списка нужную станцию и нажать кнопку [Добавить] (для выделения нескольких станций необходимо выбирать с зажатой клавишей <Ctrl>).

Для удаления рабочих станций пользователя нужно на вкладке «Рабочие станции» нажать на кнопку 🗑 ([Удалить рабочую станцию]).

Для управления мандатными атрибутами пользователя нужно нажать на логин пользователя в столбце «Логин». На вкладке «Мандатные атрибуты» выбрать нужные уровни и категории и нажать кнопку [Применить] (рис. 79).

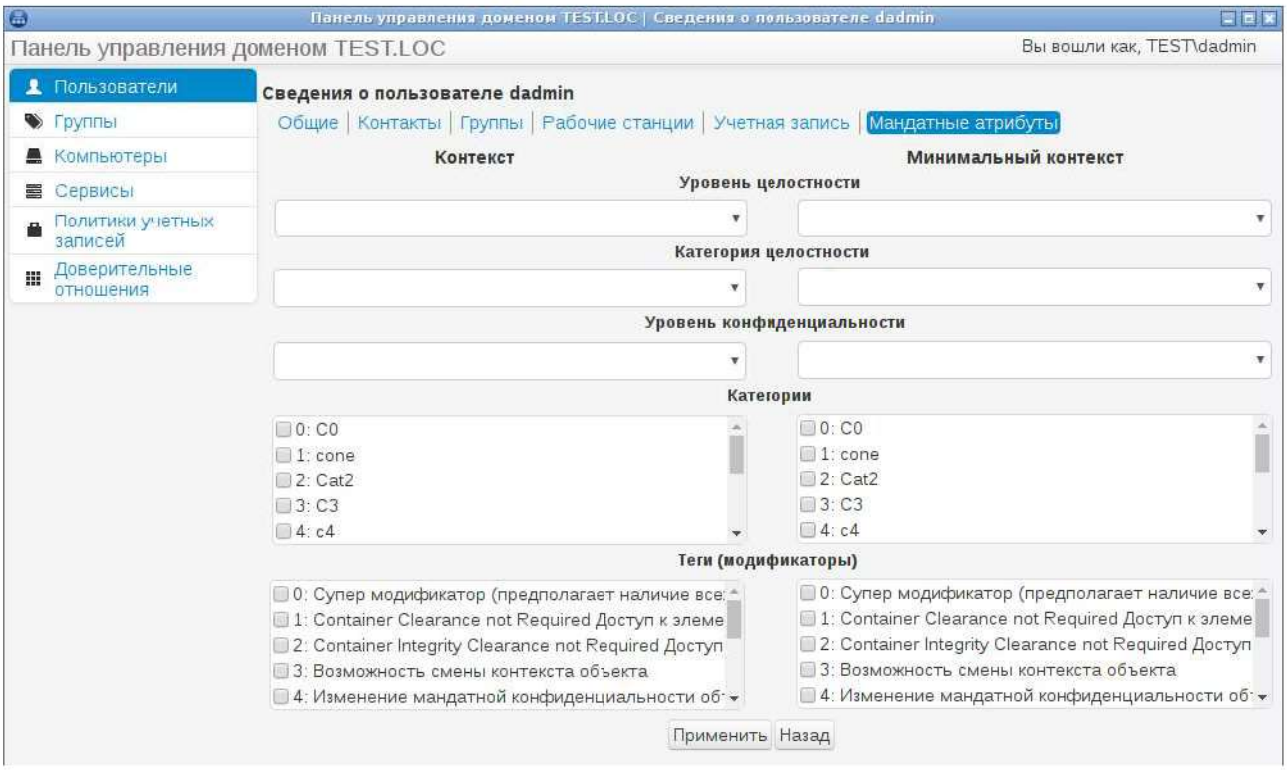


Рис. 79

3.13.3.5. Управление учетными записями групп пользователей домена

Для управления группами пользователей выбрать пункт меню «Группы» (рис. 80).

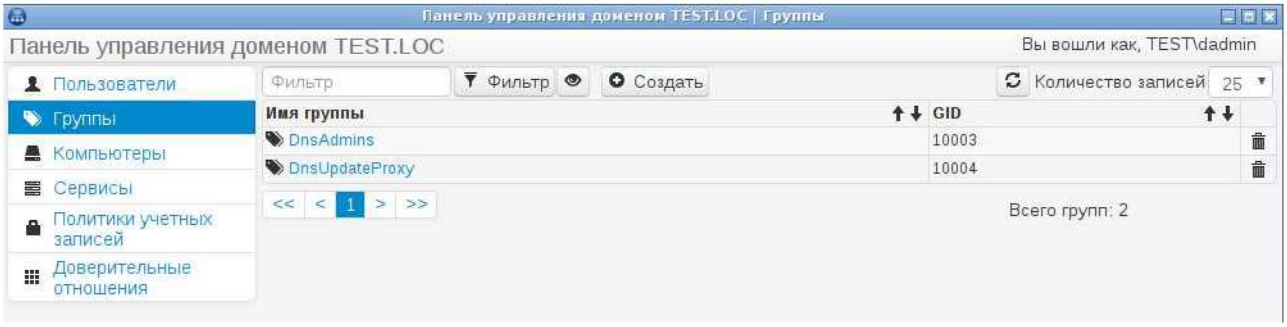


Рис. 80

Для создания группы пользователей нажать кнопку [Создать]. Заполнить форму и нажать [Создать].

Для добавления пользователей в группу необходимо выбрать группу из списка и на вкладке «Участники» нажать кнопку [Добавить участников] (рис. 81).

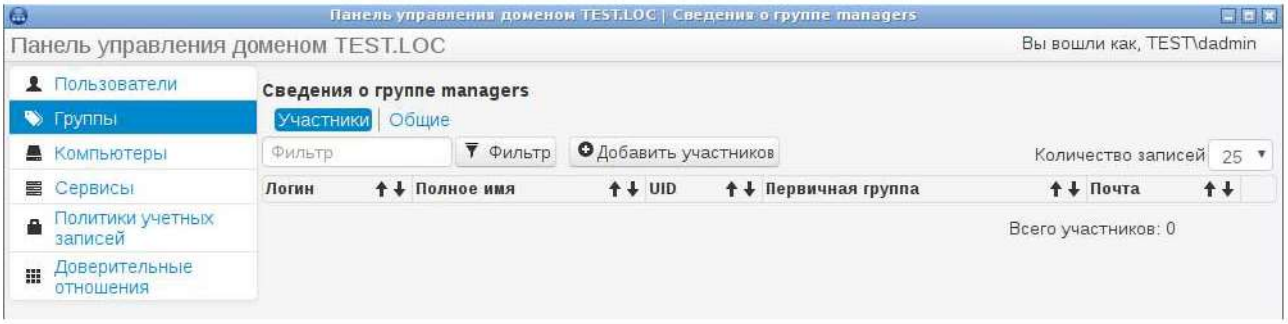


Рис. 81

Выбрать из списка нужного пользователя (для выбора нескольких пользователей зажать клавишу <Ctrl>) и нажать кнопку [Добавить] (рис. 82).

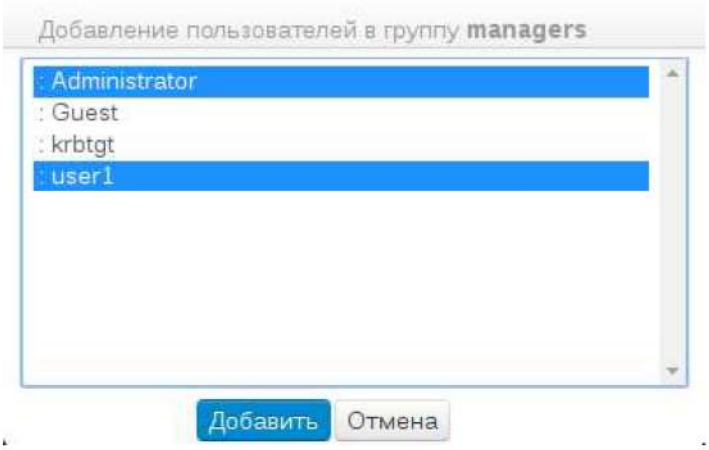


Рис. 82

Для удаления пользователей из состава группы выбрать группу из списка и на вкладке «Участники» нажать кнопку 🗑 ([Удалить пользователя из группы]).

Для просмотра состава группы в строке с соответствующей группой нажать на ссылку с соответствующим именем группы в столбце «Имя группы». Входящие в группу пользователи будут отображены на вкладке «Участники».

Для удаления группы в строке с соответствующей группой нажать кнопку 🗑 ([Удалить группу]). Во всплывающем окне подтвердить удаление, нажав кнопку [Да].

3.13.3.6. Управление учетными записями рабочих станций

Для просмотра списка рабочих станций, входящих в состав домена, перейти во вкладку «Компьютеры» (рис. 83).

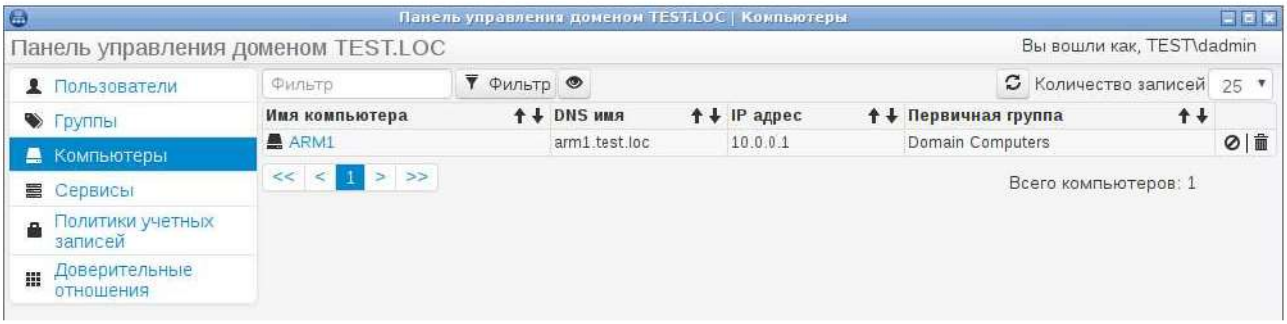


Рис. 83

Для просмотра свойств нужно в строке с соответствующей рабочей станцией нажать на ссылку с именем компьютера в столбце «Имя компьютера» (рис. 84-86).

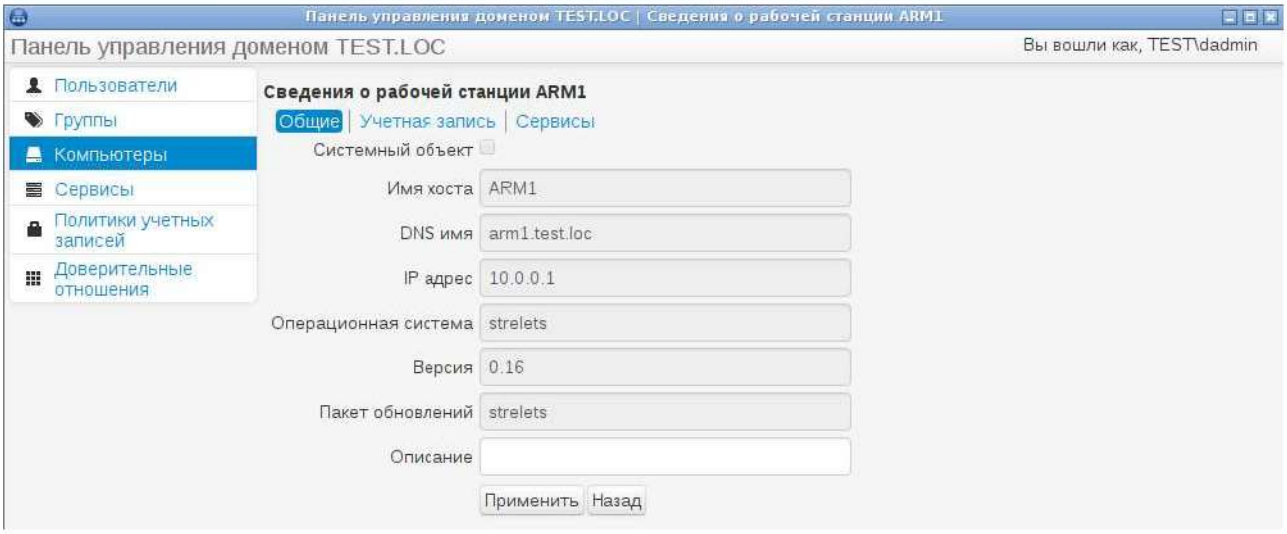


Рис. 84

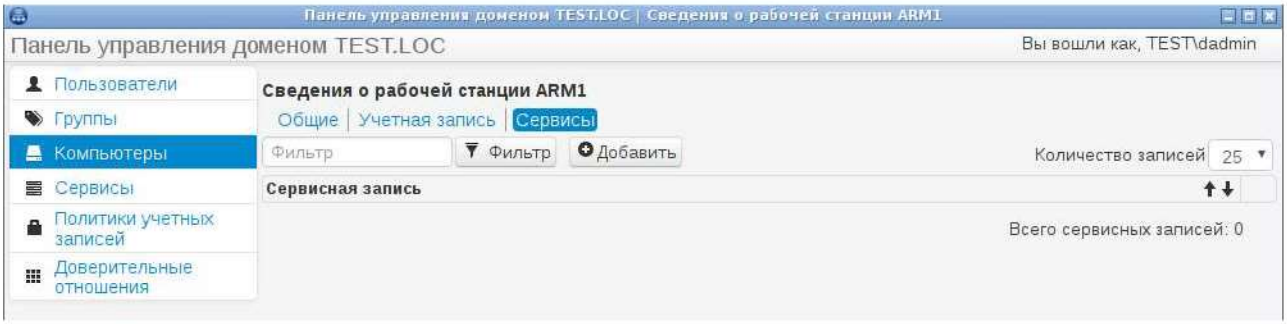


Рис. 85

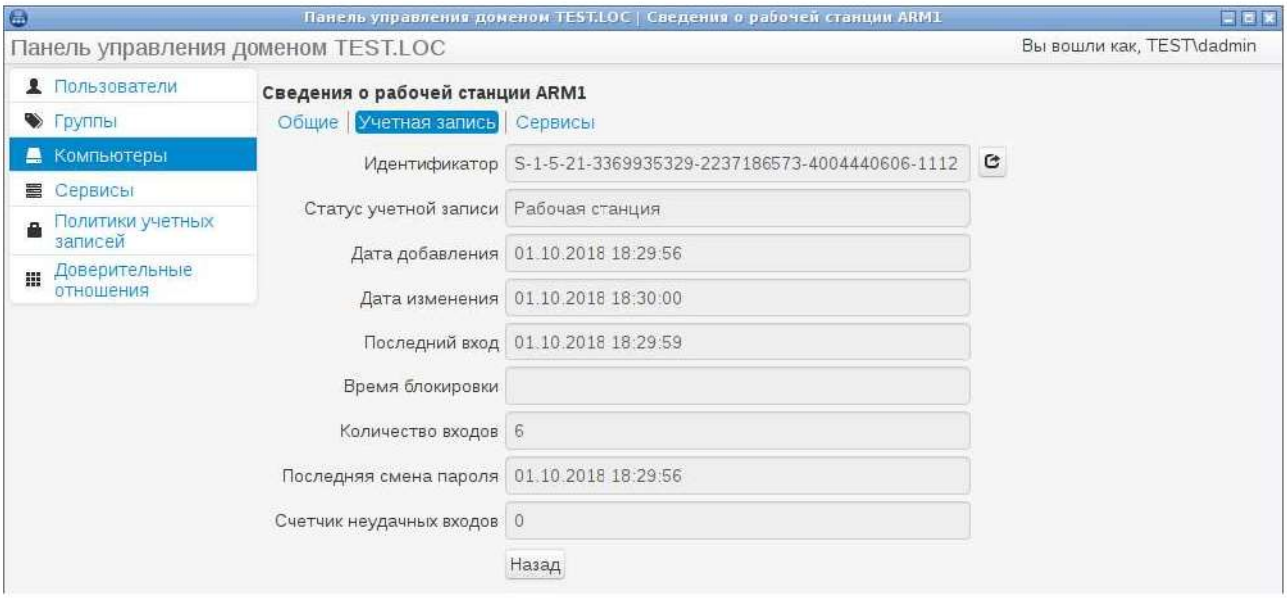


Рис. 86

Для удаления рабочей станции нужно в строке с соответствующей станцией нажать на кнопку 🗑 ([Удалить рабочую станцию]). Во всплывающем окне подтвердить удаление, нажав кнопку [Да].

3.13.3.7. Управление учетными записями служб (сетевых сервисов) домена

Для управления сервисными записями выбрать пункт меню «Сервисы» (рис. 87).

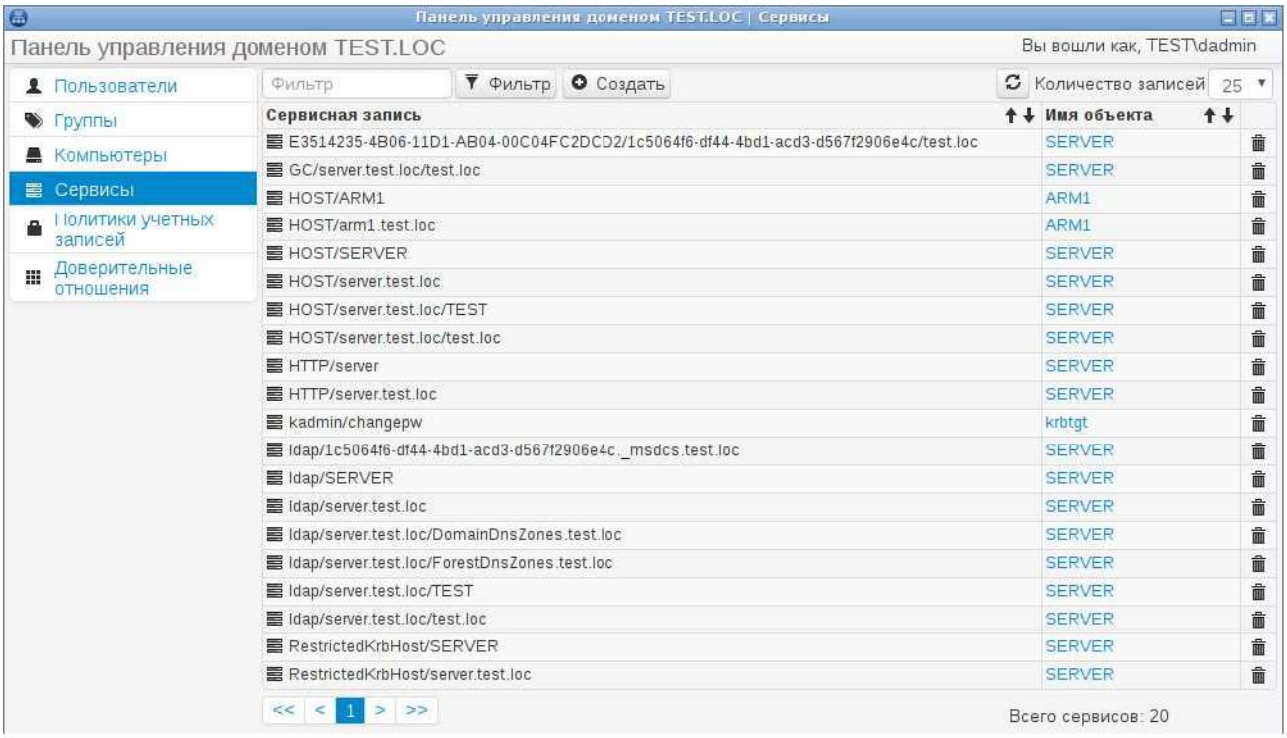


Рис. 87

Для создания сервисной записи нажать кнопку [Создать]. Заполнить поля формы

сервисную запись, и нажать кнопку [Создать].

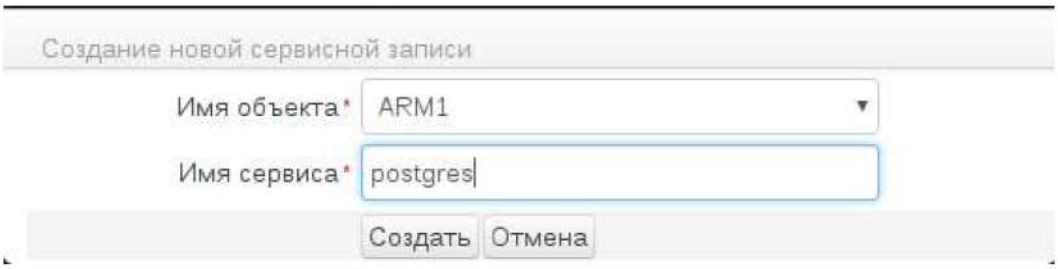


Рис. 88

Для удаления сервисной записи в строке с соответствующей записью нажать кнопку [Удалить запись]. Во всплывающем окне подтвердить удаление нажатием кнопки [Удалить].

3.13.3.8. Управление политиками учетных записей

Для управления политиками учетных записей выбрать пункт меню «Политики учетных записей» (рис. 89).

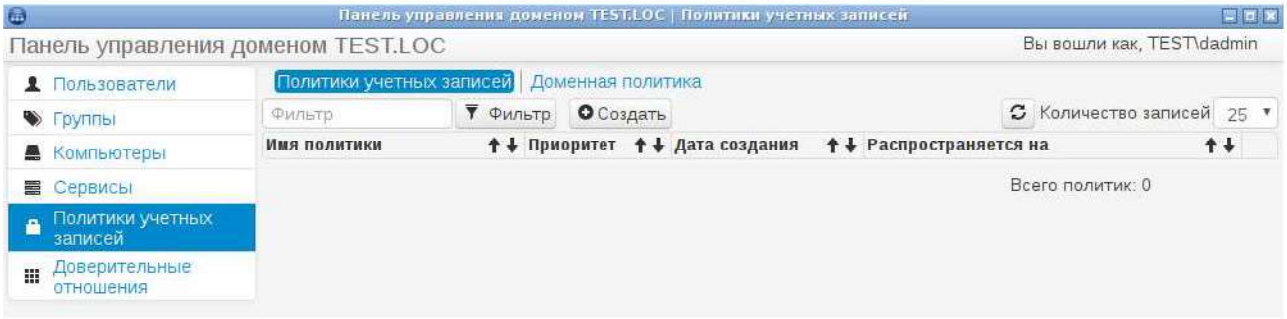


Рис. 89

Для создания гранулированной политики нажать кнопку [Создать]. Заполнить поля формы (рис. 90, 91), указав имя политики, и нажать кнопку [Создать].

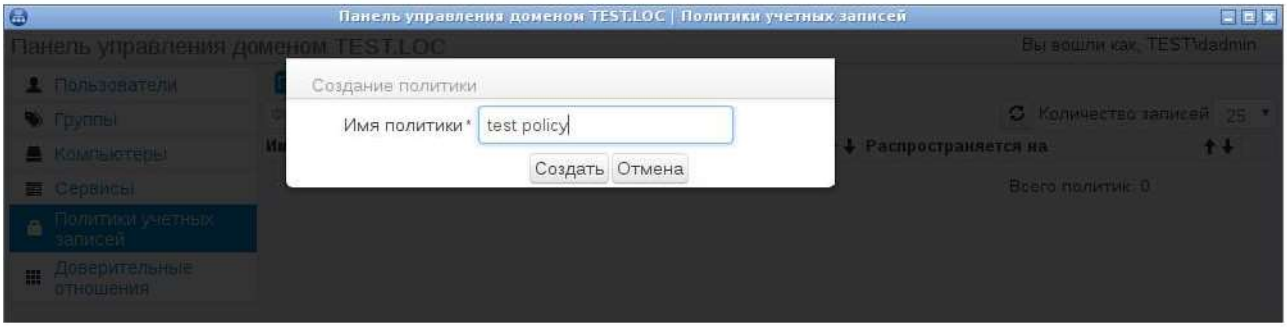


Рис. 90

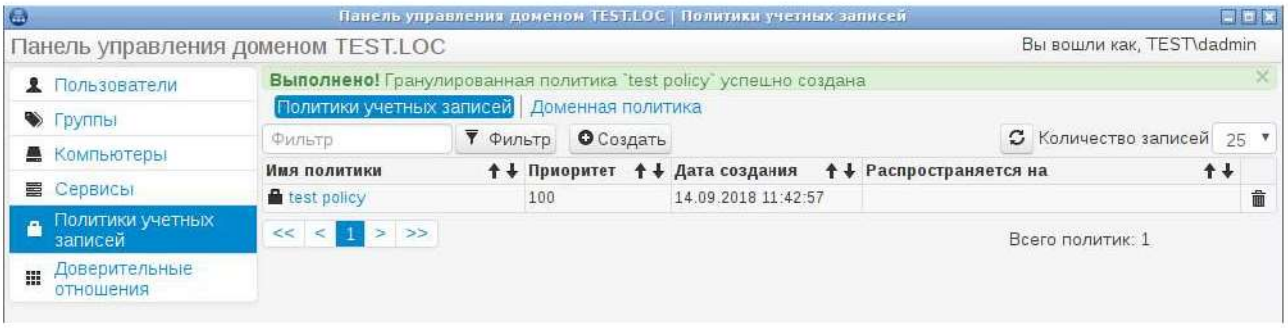


Рис. 91

Для просмотра свойств и управления гранулированной политикой нужно в строке с соответствующей политикой нажать на ссылку с именем политики в столбце «Имя политики» (рис. 92, 93).

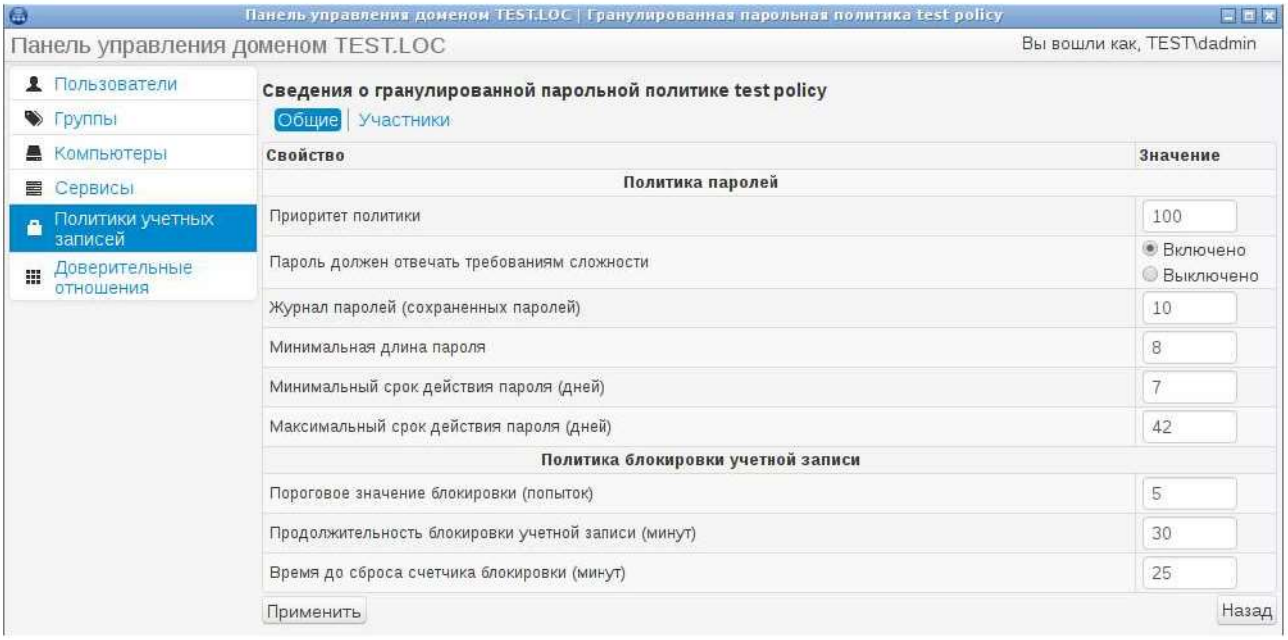


Рис. 92

Рис. 93

Для добавления участников гранулированной политики нужно на вкладке «Участники» нажать кнопку [Добавить], выбрать участника политики и нажать кнопку [Добавить] (для выделения нескольких участников можно использовать клавишу <Ctrl>)

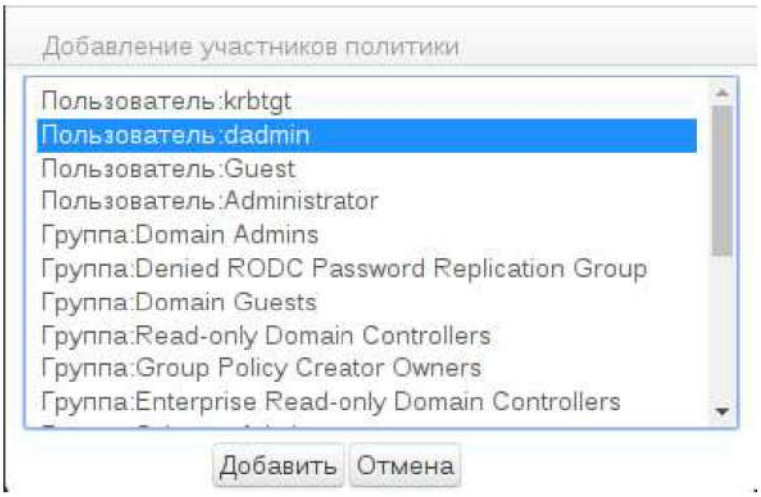


Рис. 94

Для удаления участников гранулированной политики нужно на вкладке «Участники» нажать кнопку [Удалить объект гранулированной политики] в строке с участником политики (рис. 95).

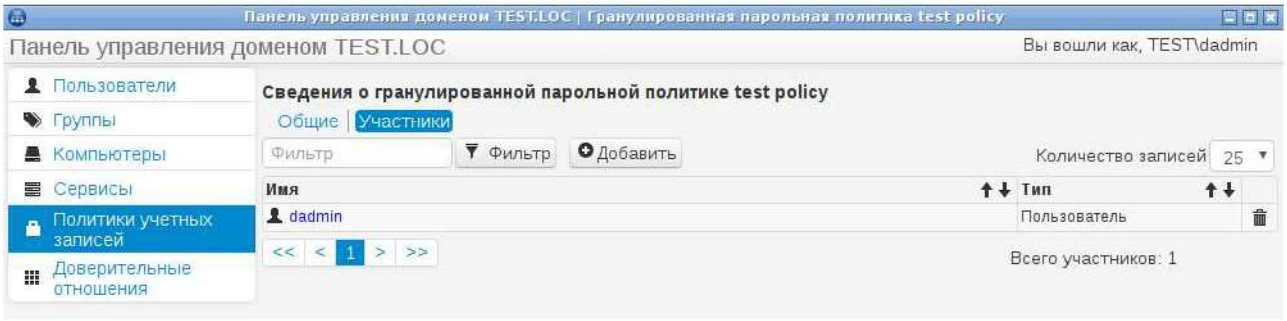


Рис. 95

Для просмотра свойств и управления глобальной доменной политикой нужно выбрать пункт меню «Политики учетных записей» и перейти на вкладку «Доменная политика» (рис. 96).

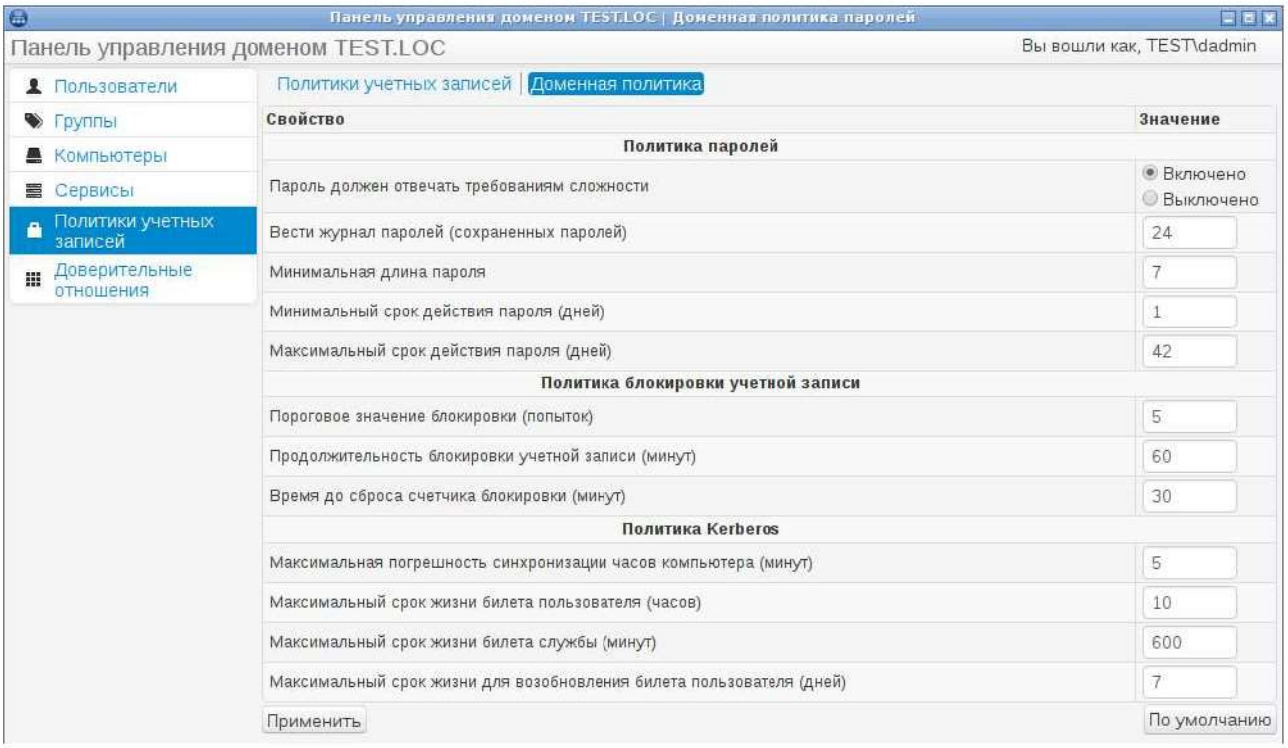


Рис. 96

3.13.3.9. Управление доверительными отношениями

Для управления доверительными отношениями нужно выбрать пункт меню «Доверительные отношения» (рис. 97).

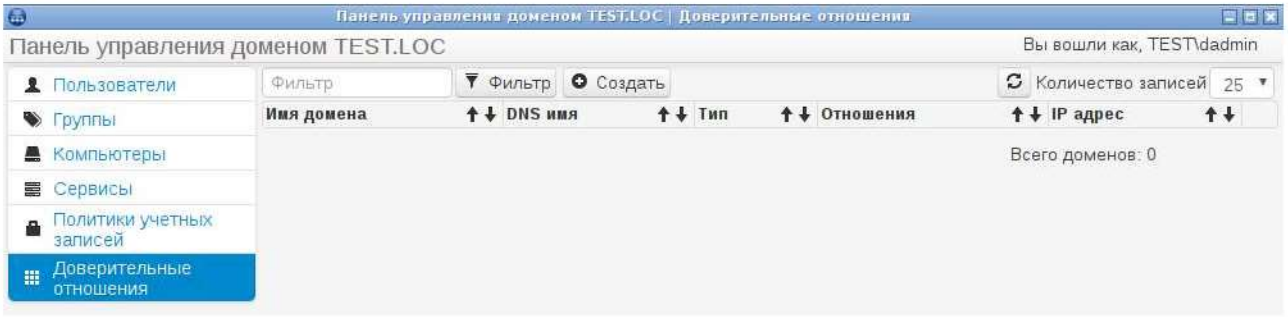


Рис. 97

Перед созданием доверительных отношений необходимо на обоих контроллерах домена добавить записи в системных файлах.

Для файла /etc/resolv.conf добавить запись вида:

nameserver <IP-адрес контроллера доверенного домена>

Пример:

nameserver 192.168.1.2

Для создания доверительных отношений нажать кнопку [Создать] и заполнить форму создания доверительных отношений (рис. 98), в поле «Домен» указать имя

доверенного домена, в поле «Пароль администратора» пароль администратора доверенного домена и нажать кнопку [Создать].

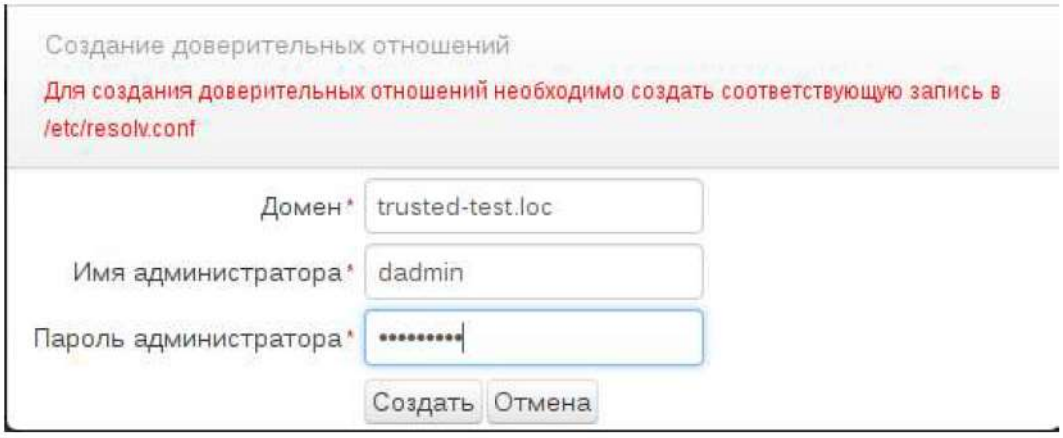


Рис. 98

Для удаления доверительных отношений нужно в строке с соответствующей записью доверенного домена нажать кнопку 🗑 ([Удалить доверительные отношения]) (рис. 99) и заполнить форму удаления доверительных отношений (рис. 100).

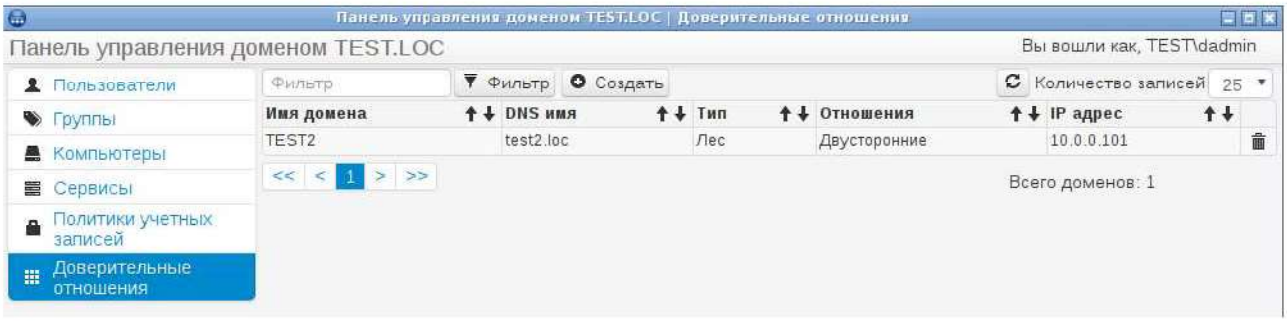


Рис. 99

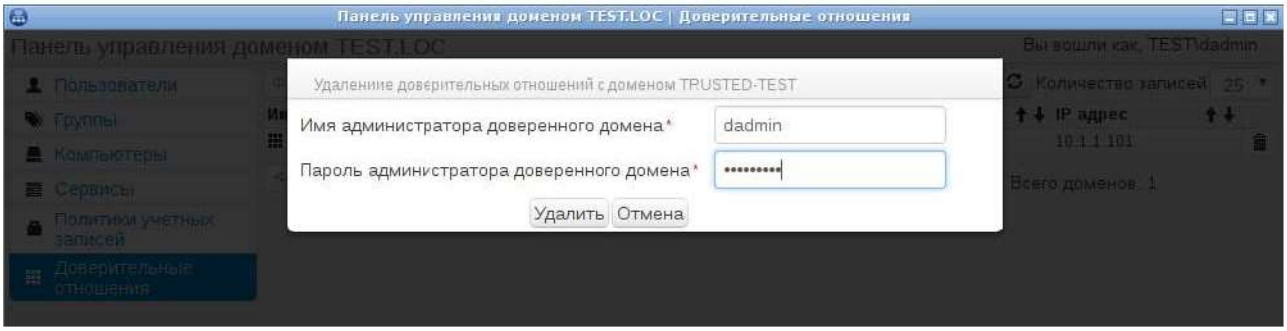


Рис. 100

3.14. Система управления базами данных

СУБД реализована на основе свободной объектно-реляционной СУБД PostgreSQL и предназначена для предоставления многопользовательского доступа к реляционным БД.

(строк) и столбцов.

СУБД PostgreSQL в составе ОС функционирует в виде серверных процессов, каждый из которых управляет одним из кластеров БД. Под кластером БД понимается логически и физически обособленный набор файлов, включающий конфигурационные файлы, расположенные в /etc/postgresql/{версия}/{кластер}, и каталоги с файлами БД кластера – в /var/lib/postgresql/{версия}/{кластер}.

3.14.1. Управление функционированием и работа с СУБД

Для управления функционированием СУБД PostgreSQL в ОС используются утилиты из состава семейства пакетов postgresql-common:

– pg_createcluster – для создания нового обособленного кластера БД (каждому кластеру создаются отдельные каталоги для конфигурационных файлов и файлов данных и выделяется порт для сетевых соединений, начиная с 5432);

– pg_dropcluster – удаление существующего кластера БД;

– pg_lscluster – просмотр статуса существующих кластеров БД;

– pg_ctlcluster – управление функционированием заданного кластера БД.

Общий вид вызова перечисленных команд содержит указание версии СУБД и имя кластера, что позволяет устанавливать и управлять кластерами разных версий СУБД:

```
$ sudo pg_ctlcluster 9.6 main restart
```

Для доступа к СУБД используются штатные утилиты из состава СУБД PostgreSQL, например интерактивный клиент psql.

Более подробно настройка и управление СУБД описаны в документации PostgreSQL и на страницах справочника man для перечисленных утилит.

3.14.2. Настройка СУБД для работы в домене

Для настройки СУБД PostgreSQL для работы в домене выполнить следующие действия (имена файлов и сетевые адресы указаны для примера):

1) создать сервисную запись для сервиса postgres;

2) выгрузить ключ сервисной записи для СУБД PostgreSQL командой:

```
$ sudo ndds-ctl createkeytab --principal postgres --output /etc/postgresql-common
```

3) назначить владельца и группу postgres на файл ключей:

```
$ sudo chown postgres.postgres /etc/postgresql-common/postgres.keytab
```

4) в конфигурационном файле /etc/postgresql/9.6/main/postgresql.conf установить:

```
listen_addresses = '*'
```

```
krb_server_keyfile = '/etc/postgresql-common/postgres.keytab'
```

5) в конфигурационном файле /etc/postgresql/9.6/main/pg_hba.conf настроить аутентификацию по методу gss:

```
host all all gss 192.168.10.0/24 gss include_realm=0
```

6) выполнить перезапуск сервера СУБД:

```
$ sudo systemctl restart postgresql
```

После выполнения настройки пользователи домена смогут обращаться к СУБД. При этом необходимо, чтобы для них существовала учетная запись в кластере БД.

3.15. Общая информация СЗИ

Функции ОС по защите информации от несанкционированного доступа (НСД) реализованы подсистемой NESS. Для управления СЗИ администратору безопасности предоставляется ряд средств управления СЗИ.

3.15.1. Состав СЗИ

В состав СЗИ входят подсистемы:

- модули подсистемы NESS, интегрированные в ядро ОС;
- модули аутентификации;
- утилиты управления безопасностью;
- средства протоколирования;
- средства контроля целостности;
- средства восстановления;
- средства разграничения доступа к устройствам ввода-вывода.

3.15.2. Контролируемые функции

СЗИ реализуют следующие функции ОС по защите от НСД:

- идентификацию и аутентификацию;
- дискреционное разграничение доступа;
- мандатное разграничение доступа;
- замкнутую программную среду;
- очистку памяти;
- регистрацию событий;
- надежное восстановление;
- контроль целостности СЗИ;
- маркировку документов;
- сопоставление пользователей с устройствами;

– защиту ввода-вывода информации на отчуждаемые носители.

Реализация функций основана на следующих положениях:

– каждому пользователю соответствует идентификатор пользователя (UID). Уникальный числовой ключ к соответствующей записи в БД пользователей. БД пользователей управляется системным администратором и содержит информацию о пользователях;

– каждый пользователь входит по меньшей мере в одну группу. Группа – это список пользователей системы, имеющий собственный идентификатор (GID). Одному UID может соответствовать несколько GID;

– каждому процессу в системе соответствует UID запустившего его пользователя. Процесс, порожденный другим процессом, наследует его UID. После идентификации и аутентификации пользователя в системе порождается первый процесс пользователя. Таким образом, все процессы пользователя буду иметь его UID. При обращении процесса к защищаемым объектам доступ осуществляется на основании дискреционных и мандатных правил разграничения доступа.

Механизмы, обеспечивающие выполнение правил разграничения доступа, реализованы в ядре ОС. Вследствие этого механизмы корректно функционируют при использовании любых компонентов ОС.

3.15.3. Механизмы реализации СЗИ

Для реализации функций СЗИ используется фреймворк Linux Security Modules (далее, LSM). LSM это встроенный механизм ядра Linux предназначенный для интеграции в ядро различных моделей безопасности, расширяющих базовую дискреционную модель доступа.

Архитектура LSM технически заключается в установленных в ядре ОС хуках. Хуки расположены во всех местах, где может понадобиться принятие решений о доступе к тем или иным объектам, и предоставляют интерфейс для внедрения собственных обработчиков. Количество хуков составляет порядка двух сотен, что позволяет создавать гибкие реализации СЗИ.

К ядру ОС применен набор патчей PaX, который модифицирует механизмы работы с памятью и затрудняет или делает невозможной реализацию эксплоитов, основанных на ошибках ПО, которые позволяют получить доступ на произвольное чтение/запись в адресном пространстве процесса.

3.15.4. Описание разграничения доступа

Для описания механизмов защиты используются следующие понятия:

- субъекты доступа: пользователи;
- объекты доступа включают в себя:
  - файлы;
  - механизмы межпроцесного взаимодействия;
  - устройства;
  - сокеты;
  - сетевые пакеты;
- типы доступа:
  - чтение;
  - запись;
  - исполнение.

3.15.5. Дискреционный механизм доступа

Для организации дискреционного доступа используются базовые механизмы Linux. Они представляют из себя комбинацию матрицы доступов и списков управления доступов (Access Control Lists)

Дискреционный контроль осуществляется для всех объектов ФС. Каждому именованному объекту при создании сопоставляется UID и GID субъекта системы, который имеет права распоряжаться объектом. Разделяются три вида доступа: чтение (read, r), запись (write, w), исполнение (execute, e). Субъекты, производящие доступ, относят к одному из трех классов: пользователя-владельца, группы-владельца и всех остальных. Объекту сопоставляется битовая маска из 9 элементов, по три для каждого класса. Если соответствующий бит выставлен в 1, то этот вид доступа для этого класса разрешен, в противном случае доступ запрещен.

При обращении процесса к объекту система проверяет соотношение UID и GID процесса с UID и GID объекта и выносит решение о доступе на основании того, к какому классу относится процесс.

Дискреционные права доступа к объекту могут быть изменены владельцем этого объекта.

Дополнительно к битовой маске существуют специальные биты: SUID, SGID, Sticky.

SUID — Set User ID бит смены идентификатора пользователя.

Когда пользователь или процесс запускает исполняемый файл с выставленным битом SUID, то порожденный процесс получает UID владельца файла. Таким образом осуществляется возможность запуска файла с правами другого пользователя.

SGID – Set Group ID бит смены идентификатора группы.

Аналогично SUID.

Sticky – определяет владельца объектов в каталоге.

Sticky-бит используется для каталогов, чтобы защитить в них файлы от удаления. Если пользователь не является владельцем каталога, то он может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

В дополнении к битовой маске, на основании расширенных атрибутов файловых систем, используются списки контроля доступа ACL (Access Control List). С их использованием можно дополнительно задавать права доступа субъектов к объекту.

ACL состоит из набора записей. Минимальный ACL включает в себя три записи для пользователя-владельца, группы-владельца и всех остальных. Дополнительно могут указываться права доступа для других именованных субъектов.

В общем случае в ACL могут присутствовать записи следующих типов:

- пользователь-владелец (текстовое представление: user::rwx);
- именованный пользователь (текстовое представление: user:user_name:rwx);
- группа-владельца (текстовое представление: group::rwx);
- именованная группа (текстовое представление: user:group_name:rwx);
- все остальные (текстовое представление: other::rwx).
- маска доступа (текстовое представление: mask::rwx).

Маска доступа используется для ограничения прав доступа именованных пользователей и групп. В случае если какой-либо вид доступа не указан в маске, то доступ будет запрещен даже если в ACL для этого пользователя указано обратное.

Субъект может выполнять проверку файла или каталога в следующем порядке:

- используются права владельца, если субъект им является;
- используются права, указанные конкретно для этого субъекта;
- используются права для группы владельца;
- проверяется разрешение действия хотя бы для одной из группы, в которую входит пользователь.

Если пользователь не входит ни в одну группу, описанную в ACL, то используются права для other.

Решение о доступе принимается на основании первой подошедшей записи ACL.

3.15.6. Мандатный механизм доступа

Реализация мандатного механизм разграничения доступа является частью

фреймворка NESS (oSnova Enchanced Security (Sub)System).

3.15.6.1. Контекст безопасности NESS

Любому объекту/субъекту доступа фреймворком NESS ставится в соответствие контекст безопасности – NESSCTX, который в общем виде имеет следующий формат:

`attrlevdec@vallevdec:attrlevtxt@vatxt1:attrcatslist@valcatsdec1, valcatsdec2,valcatsdec3,valtxt2,valtxt3:attrcatshex@0xvalhex: attrtagslist@valtagsdec1,valtagsdec2,valtagsdec3,valtxt4,valtxt5`

где:

1) `attrXXX` – атрибут контекста безопасности, порядок задания не важен:
   - `il` – мандатный уровень целостности;
   - `ic` – мандатные категории целостности:
   - `l` – мандатный уровень конфиденциальности;
   - `c` – мандатные категории конфиденциальности:
   - `t` – теги-модификаторы управления доступом;

2) `@` – операция, применяемая к атрибуту контекста безопасности:
   - `=` – присвоение;
   - `+` –увеличение для уровней, объединение для категорий и тегов;
   - `-` – уменьшение для уровней, исключение для категорий и тегов;

3) `valXXX` – значение атрибута контекста безопасности, могут быть заданы:

   а) в численном виде:

   - `vallevdecX` – численное значение мандатного уровня целостности или конфиденциальности в десятичной системе счисления в диапазоне 0-65535;
   - `valcatsdecX` – численное значение мандатных категорий в десятичной системе счисления, указывается номер бита в битовой маске категорий, начиная с 0, максимальное значение номера бита 2^32-1;
   - `0xvalhex` – численное значения мандатных категорий в виде битовой маски в шестнадцатиричной системе счисления, пример `c=0xffff1234`;
   - `valtagsdecX` – численное значение идентификаторов тегов в десятичной системе счисления в диапазоне 0-2^64-1.

   б) в текстовом виде – `valtxtX`, соответствие численных и текстовых значений задается в поддереве политики безопасности NESS: `/NESS/attr/атрибут_контекста/`, подробнее ниже.

Примеры задания контекста безопасности:

```
il=0:l=1:c=1,2,3,4,5:ic=0xf:t=0,5,21
c=cat1,cat3:l=secret:t=OMIT,CHCTX
l=1:c=5:t=OMIT:c+категория1
il=Высокий:ic=0xf:t=OMIT:ic-категория0,1
```

В случае если какой-то из атрибутов не указан, его значение берется из пустого контекста безопасности:

```
il=0:ic=0x0:l=0:ic=0x0:t=
```

3.15.6.2. Политика безопасности NESS

Возможность управления политикой безопасности системы обеспечивается включением пользователей, имеющих на это право, в группу GSP. Политика безопасности хранится в файлах /NESS/attrs/l, /NESS/attrs/il, /NESS/attrs/c, /NESS/attrs/acms.

Возможность управления метками доступа пользователей обеспечивается через включение пользователей, имеющих на это право, в группы GSPU (для изменения) и GSPUro (только для чтения). Контексты пользователей максимально доступный и минимальный хранятся в каталоге /NESS/users/<uid>/ctx. В нем находятся файлы:

– nessctxnessctx – содержит максимальный контекст безопасности, доступный пользователю;

– minnessctx.nessctx – содержит минимальный контекст безопасности, доступный пользователю.

Для просмотра и изменения доступных пользователю контекстов безопасности используется утилита nessctx-user. Подробнее в руководстве man nessctx-user.

3.15.6.3. Контексты безопасности объектов

NESS поддерживает назначение контекстов безопасности следующих типов объектов:

1) файлы, каталоги, именованные каналы (FIFO). Сохранение контекстов безопасности после перезагрузки возможно только на файловых системах, поддерживающих работу с расширенными атрибутами (xattr). Назначение выполняется посредством chcon;

2) сокеты согласно типам:

– IPv4 (PF_INET), имеются ограничения;

– доменные сокеты UNIX(PF_UNIX), назначаются любые контексты безопасности;

– остальные типы сокетов: возможно назначение только пустого контекста или контекста `t=EQU`;

3) объекты SYSV PC;

4) каналы (pipe).

Объекты, для которых не установлена метка доступа, расцениваются как объекты с нулевой меткой доступа.

3.15.6.4. Контексты безопасности субъектов

Субъектом доступа является процесс, которому ОС ставится в соответствие пользователь, от имени которого он работает.

Каждому пользователю в системе соответствует диапазон (набор) допустимых контекстов безопасности, из которых после аутентификации в системе он выбирает действующий в течении сессии.

3.15.6.5. Формальные правила

Решения о предоставлении доступа субъекта к объекту принимаются на основании контекста безопасности субъекта, объекта и типа доступа. Если в системе все объекты и субъекты имеют пустые контексты безопасности, то система функционирует аналогично стандартной ОС Linux.

Правила принятия решения описываются следующим образом. Пусть контекст безопасности субъекта содержит уровень целостности Sil, категории целостности Sic, уровень конфиденциальности Sl, категории Sc, а контекст безопасности объекта соответственно Oil, Oic, Ol, Oc.

Тогда:

– операция записи разрешена, если Sil>=Oil и Sic>=Oic и Sl==Ol и Sc==Oc;

– операции чтения и выполнения разрешены Sд>= Oд, Sl>=Ol.

При этом при наличии у субъекта или объекта тега:

– CCnR: (Container Clearance not Required). Доступ к элементам контейнера осуществляется без учета его конфиденциальности. При использовании с контейнером позволяет хранить в нем сущности с мандатными атрибутами, отличными от мандатных атрибутов контейнера, но не превосходящими их, в отношении субъектов, не являющихся контейнерами, игнорируется;

– CICnR: (Container Integrity Clearance not Required). Доступ к элементам контейнера осуществляется без учета его целостности. При использовании в отношении контейнера – позволяет хранить в нем объекты с различными уровнями целостности, но не превосходящими уровень целостности контейнеров, в отношении субъектов, не являющихся контейнерами, игнорируется;

– EQU: Пропустить проверку МРД (конфиденциальности и целостности) для

объекта. Используется с объектами, не допускающими возникновения запрещенных информационных потоков. При использовании таких объектов считается, что мандатные атрибуты контекста безопасности объекта совпадают с мандатными атрибутами контекста безопасности субъекта, который осуществляет доступ, примеры: /dev/null, /dev/full;

– OMIT: Пропустить проверку МРД (конфиденциальности и целостности) для субъекта. Используется в отношении привилегированных доверенных субъектов, не допускающих возникновения запрещенных информационных потоков, в отношении объектов игнорируется;

– OMITC: Пропустить проверку МРД в части конфиденциальности для субъекта. Используется в отношении привилегированных доверенных субъектов, не допускающих возникновения запрещенных информационных потоков, в отношении объектов игнорируется;

– OMITI: Пропустить проверку МРД в части целостности для субъекта. Используется в отношении привилегированных доверенных субъектов, не допускающих возникновения запрещенных информационных потоков, в отношении объектов игнорируется;

– CHCTX: Возможность смены контекста объекта. При использовании субъектом позволяет ему управлять контекстом безопасности объектов, в отношении объектов игнорируется;

– SETCTX: Возможность смены контекста субъекта. При использовании субъектом позволяет ему управлять собственным контекстом безопасности, в отношении объектов игнорируется;

– ADDEQUSOCK: Автоматическая установка EQU на создаваемый сокет. При использовании субъектом при создании им объекта типа «сокет» к контексту безопасности последнего будет добавлен тег EQU, в отношении объектов игнорируется;

– SETXATTR_OMITMAC: Пропустить проверку МРД в части иерархии при установке контекста безопасности объекта ФС через расширенные атрибуты. При использовании субъектом, например программой резервного копирования, позволяет ему при управлении контекстом безопасности объектов, непосредственно изменяя их расширенные атрибуты (SECURITY.NESSCTX), игнорировать МРД в части иерархии. **Применяется только** при восстановлении резервных копий, распаковки архивов и т.д., которые содержат расширенные атрибуты контекстов безопасности, и при этом невозможно обеспечить корректность этой процедуры в части иерархии согласно МРД;

– NTERPRETER: Запуск доверенных скриптов через интерпретатор посредством shebang;

– NOSEUID_FIXUP: Не модифицировать теги при смене EUID, тег **будет сброшен** после вызова seteuid(). При использовании субъектом при успешном выполнении системного вызова seteuid() теги субъекта **не будут изменены**, за исключением удаления тега NOSEUID_FIXUP. При отсутствии у субъекта тега NOSEUID_FIXUP перед вызовом seteuid() набор тегов субъекта будет сброшен. Применяется в отношении субъектов, которые осуществляют имперсонацию. В отношении объектов игнорируется;

– NOSEUID_FIXUP_PERMANENT: Не модифицировать теги при смене EUID, тег **НЕ будет сброшен** после вызова seteuid(). Аналогично NOSEUID_FIXUP, но данный тег **не сбрасывается** после успешного вызова seteuid().

3.15.7. Структура метки доступа

Метка доступа имеет следующий канонический формат:

```
il=ni:l=nl:c=nc0,nc1,nc2:acms=nacm0,nacm1,nacm2
```

где ni — числовое значение уровня целостности (0-255),

nl – числовое значение мандатного уровня конфиденциальности (0-255) ,

ncx – числовые значения мандатных категорий в десятичной системе счисления, указывается номер бита в битовой маске категорий, максимальное значение номера бита 2^32-1. Категории также могут быть заданы в виде битовой маски, а именно c=0xffff1234;

nacmx – числовые значения идентификаторов модификаторов управления доступом. Диапазон записываются в десятичном виде.

Соответствующие расшифровки числовых значений хранятся в файлах /NESS/attr/l /NESS/attr/il /NESS/attr/c /NESS/attr/acms.

Эти текстовые обозначения могут использоваться в текстовом представлении.

Варианты представления контекста:

```
il=0:l=1:c=1,2,3,4,5
c=cat1,cat3:l=secret
```

Порядок атрибутов в представлении не важен. В случае если какой-то из атрибутов не указан, его значение берется из контекста безопасности по умолчанию. Контекст по умолчанию:

```
il=низкий:l=0:c=нет:acms=нет
```

Возможность управления политикой безопасности системы обеспечивается включением пользователей, имеющих на это право, в группу GSP. Политика безопасности хранится в файлах /NESS/attrs/l /NESS/attrs/il /NESS/attrs/c. /NESS/attrs/acms.

Возможность управления метками доступа пользователей обеспечивается через включение пользователей, имеющих на это право, в группы GSPU (для изменения) и GSPUro (только для чтения). Контексты пользователей максимально доступный и

минимальный хранятся в каталоге /NESS/users/<uid>/ctx. В нем находятся файлы:

– gsctx.gsctx – содержит максимальный контекст безопасности, доступный пользователю;

– mingsctx.gsctx – содержит минимальный контекст безопасности, доступный пользователю.

Для просмотра и изменения доступных пользователю контекстов безопасности используется утилита gsctx-user. Подробнее в руководстве man gsctx-user.

3.15.8. Сетевое взаимодействие

Для сетевых соединений также необходимо осуществлять мандатный контроль доступа. С этой целью в сетевые пакеты протокола IPv4 в соответствии со стандартом RFC1108 внедряются метки доступа, соответствующие метке сокета.

В рамках стандарта RFC1108 метка снабжается классом 0xAB (Unclassified), при этом последующий битовый список (последовательность байт, в которых младший бит указывает на наличие следующего байта в потоке) опции представляет собой упакованную в соответствии со стандартом структуру мандатного контекста, где уровень занимает 8 бит, а категории – 64 бита (порядок байт – от младших к старшим). Последние (старшие) нулевые биты в соответствии со стандартом могут быть отброшены.

Пример кодирования метки для протокола IPv4:

```
IPOPT_SEC,5,0xAB,03,12  /* Битовый список: 00000011, 00001100, Контекст: уровень 1, категории 3) */
```

При этом метка сокета наследуется от процесса. Прием сетевых пакетов осуществляется согласно правилам мандатного разграничения доступа. Метка сокета может иметь тип, позволяющий принимать соединения с любыми мандатными атрибутами. Это необходимо для обеспечения работы ряда сетевых сервисов (Kerberos, LDAP, DNS) без внесения изменения в их исходный текст.

Объекты, для которых не установлена метка доступа, расцениваются как объекты с нулевой меткой доступа. Таким образом, если в системе все объекты и субъекты не используют мандатные атрибуты, они функционируют аналогично стандартной ОС Linux.

Для обеспечения функции контроля и фильтрации информационных потоков на базе мандатных атрибутов для модуля ядра netfilter реализован match-модуль NESS, который позволяет использовать в правилах iptables мандатные атрибуты.

Список доступных опций модуля приведен в таблице 33.

Таблица 33

Опция	Описание
--eq LABEL	IP пакеты с мандатными атрибутами равными LABEL

Опция	Описание
--parent LABEL	IP пакеты с мандатными атрибутами включающими в себя LABEL
--child LABEL	IP пакеты с мандатными атрибутами входящими в LABEL
--leq LEVEL	IP пакеты с мандатным уровнем равым LEVEL
--llt LEVEL	IP пакеты с мандатным уровнем меньше LEVEL
--lle LEVEL	IP пакеты с мандатным уровнем меньше или равным LEVEL
--lgt LEVEL	IP пакеты с мандатным уровнем больше LEVEL
--lge LEVEL	IP пакеты с мандатным уровнем больше или равным LEVEL
--ceq CATEGORY	IP пакеты с мандатными категориями совпадающими с CATEGORY
--csup CATEGORY	IP пакеты с мандатными категориями включающими в себя CATEGORY
--csub CATEGORY	IP пакеты с мандатными категориями входящими в множество CATEGORY
Без опций	IP пакеты с ненулевыми мандатными атрибутами

Например, разрешить пакеты с ненулевыми мандатными атрибутами:

```
$ iptables -A INPUT -p tcp -m NESS -j ACCEPT
```

Запретить пакеты с мандатным уровням больше 1:

```
$ iptables -A INPUT -p tcp -m NESS --lgt 1 -j REJECT
```

3.15.9. Ограничение доступа к страницам памяти

В состав ОС входит ядро, к которому применен набор изменений PaX, который является средством ограничения прав доступа к страницам памяти и предотвращает выполнение произвольного кода посредством контроля доступа (чтение, запись, исполнение и их комбинации) к сегментам памяти в адресном пространстве процесса на основе использования аппаратной реализации в процессоре неисполняемого бита. При этом комбинация типов доступа, запись и исполнение запрещена. Для сегментов данных невозможен доступ на исполнение, а для сегментов кода невозможен доступ на запись. Таким образом, набор изменений PaX для ядра ОС обеспечивает защиту исполняемого кода в адресном пространстве процесса, предоставляя наименьшие привилегии процессам при доступе к сегментам памяти в собственном адресном пространстве. Набор изменений PaX для ядра ОС обеспечивает:

– запрет создания сегментов памяти, доступных одновременно для исполнения и записи;

– запрет перемещения сегмента кода;

– запрет создания исполняемого стека;

– рандомизацию адресного пространства процесса;

– запрет создания исполняемых областей памяти в СПО и ПО программной платформы.

Набор изменений PaX устанавливает для сегментов данных процессов атрибуты, обеспечивающие невозможность их исполнения, а для сегментов кода программ –

атрибуты, обеспечивающие невозможность записи в них. При этом применяется механизм PAGEEXEC, который использует эмуляцию или аппаратную реализацию NX-бита. При наличии аппаратной реализации NX-бита, механизм PAGEEXEC использует ее вместо эмуляции, обеспечивая отсутствие снижения производительности. Набор изменений PaX гарантирует, что адреса с произвольным доступом не будут одновременно доступны на запись и выполнение. Гарантия реализуется использованием в функции mprotect() безопасного механизма защиты памяти MPROTECT.

Кроме того, набор изменений PaX обеспечивает случайный характер (рандомизацию) смещений сегментов кода и данных (в том числе стека и кучи) при использовании системного вызова отображения в память nmap().

Ядро ОС имеет архитектуру, обеспечивающую невозможность его перемещения в физическом адресном пространстве. Невозможность перемещения сегмента кода в адресном пространстве процесса обеспечивается при использовании PaX установкой соответствующих атрибутов доступа на сегмент кода.

При использовании ПО с открытыми исходными текстами в качестве основы при разработке ПО, обладающего заданными функциональными возможностями, возможно возникновение ситуаций, при которых ядро ОС (набор изменений PaX) останавливает исполнение ELF-файла из-за нарушения правил PaX для доступа к страницам памяти в адресном пространстве процесса. Для запускаемого исполняемого модуля формата ELF могут быть установлены специальные атрибуты PaX, разрешающие процессу выполнять определенные действия, запрещенные по умолчанию. Для этого используется утилита командной строки paxctl, которая обеспечивает отображение и установку таких атрибутов.

Синтаксис:

```
$ paxctl <опции><файлы>
```

3.16. Очистка памяти

Ядро ОС гарантирует, что обычный непривилегированный процесс не получит данные чужого процесса, если это явно не разрешено ПРД. Это означает, что средства IPC контролируются с помощью ПРД, и процесс не может получить неочищенную память (как оперативную, так и дисковую).

3.16.1. Механизм очистки ОП

Высокоуровневые библиотечные функции malloc и free, а также функции, реализующие встроенные операторы языка cpp – new и delete, опираются в своей работе на системный вызов brk(). Суть в том, что в адресном пространстве процесса непосредственно за сегментом кода (за кодом программы) находится пространство,

зарезервированное для процесса, с которым и работают malloc/free.

Системный сервис sys_brk (mm/mmap.c) несложный и, в конечном счете, вызывает do_mmap() (mm/mmap.c) с параметрами:

```
do_mmap(NULL, oldbrk, newbrk - oldbrk, PROT_READ | PROT_WRITE | PROT_EXEC, MAP_FIXED| MAP_PRIVATE, 0);
```

MAP_PRIVATE

отображение приватно для этого процесса, а MAP_FIXED означает, что отображение привязано к конкретному виртуальному адресу процесса.

do_mmap создает VMA и заполняет его. Добавляет его в список (и, если необходимо, дерево) структур vm_area_struct текущего процесса.

Это стандартное поведение ОС, т. е. память резервируется, но реально страницы будут выделяться в обработчике исключений. При первом же доступе к странице выделенной (зарезервированной) памяти происходит исключение, и обработчик исключения выделяет необходимую страницу.

Цепочка разрешения первого доступа к зарезервированной памяти:

– > handle_mm_fault (mm/memory.c);

– > handle_pte_fault (mm/memory.c);

– > do_no_page (mm/memory.c);

– > do_anonymous_page() (mm/memory.c);

– > clear_page()-> memset.

Выделение памяти осуществляется при вызове функции vmalloc_user, реализованной в файле исходных текстов mm/vmalloc.c ядра ОС. В данной функции реализовано обнуление выделяемого пользователю сегмента памяти с целью предотвращения утечки данных. Таким образом, удовлетворяется еще одно требование изолированности адресных пространств: процесс не может получить в свое распоряжение «неочищенную» память.

3.16.2. Механизм очистки внешней памяти

В ОС реализован механизм, который очищает неиспользуемые блоки ФС непосредственно при их освобождении. Работа данного механизма снижает скорость выполнения операций удаления и усечения размера файла. Данные любых удаляемых/урезаемых файлов в пределах заданной ФС предварительно очищаются предопределенной или псевдослучайной маскирующей последовательностью. Механизм является настраиваемым и позволяет обеспечить работу ФС ОС (Ext2/Ext3/Ext4) в одном из следующих режимов:

1) очистка осуществляется посредством перезаписи каждого байта в освобождаемой области посредством четырех сигнатур вида:

11111111, 01010101, 10101010, 00000000

Использование режима включается параметром secdel в конфигурационном файле /etc/fstab для раздела ФС, на котором требуется очищать блоки памяти при их освобождении (например, /dev/sda1). В список параметров монтирования добавляется параметр secdel.

Пример:

```
/dev/sda1 /home ext4 acl,defaults,secdel 0 2
```

2) очистка осуществляется посредством перезаписи каждого байта в освобождаемой области посредством четырех сигнатур вида:

11111111, 01010101, 10101010, 00000000

Количество перезаписей определяется администратором. Использование режима включается установкой значения параметра secdel в конфигурационном файле /etc/fstab для раздела ФС, на котором требуется очищать блоки памяти при их освобождении (например, /dev/sda1). При установке числа перезаписей больше четырех сигнатуры используются повторно. Например, при установке числа перезаписей, равному 6, последовательность сигнатур, используемых для перезаписи, имеет вид:

11111111, 01010101, 10101010, 00000000, 11111111, 01010101

В список параметров монтирования добавляется параметр secdel=6.

Пример:

```
/dev/sda1 /home ext4 acl,defaults,secdel=6 0 2
```

3) очистка осуществляется посредством перезаписи каждого байта в освобождаемой области посредством четырех псевдослучайных сигнатур. Использование режима включается параметром secdelrnd в конфигурационном файле /etc/fstab для раздела ФС, на котором требуется очищать блоки памяти при их освобождении (например, /dev/sda1). В список параметров монтирования добавляется параметр secdelrnd.

Пример:

```
/dev/sda1 /home ext4 acl,defaults,secdelrnd 0 2
```

4) очистка осуществляется посредством перезаписи каждого байта в освобождаемой области посредством псевдослучайных сигнатур. Количество перезаписей определяется администратором. Использование режима включается установкой значения параметра secdelrnd в конфигурационном файле /etc/fstab для раздела ФС, на котором требуется очищать блоки памяти при их освобождении

(например, /dev/sda1). Например, при установке числа перезаписей, равному 6, в список параметров монтирования добавляется параметр secdelrnd=6.

Пример:

```
/dev/sda1 /home ext4 acl,defaults,secdelrnd=6 0 2
```

5) очистка осуществляется посредством перезаписи каждого байта в освобождаемой области посредством заданной администратором. Количество перезаписей определяется администратором. Использование режима включается установкой значения параметра secdel в конфигурационном файле /etc/fstab для раздела ФС, на котором требуется очищать блоки памяти при их освобождении (например, /dev/sda1). Например, при установке числа перезаписей, равному 5, в список параметров монтирования добавляется параметр secdel=5. Маска определяется параметром secdelmask.

Пример:

```
/dev/sda1 /home ext4 acl,defaults,secdel=5,secdelmask=0xABCDEF55
0 2
```

Для включения очистки активных разделов страничного обмена установить в конфигурационном файле /etc/security/swapshred.conf для параметра ENABLE значение Y.

Пример:

```
ENABLE=Y
```

Для задания списка разделов страничного обмена, для которых не выполняется очистка, может быть использован параметр IGNORE, значение которого является списком перечисленных через пробел игнорируемых разделов страничного обмена.

Пример:

```
IGNORE="/dev/sdz10 /dev/sdz11"
```

3.17. Завершение работы и перезагрузка

Процесс сохранения данных и подготовки к выключению оборудования является важным, поскольку обеспечивает корректное завершение работы системы, позволяющее избежать потерь информации и сбоев ФС.

Для завершения работы системы могут быть использованы команды:

– `shutdown`– универсальная команда остановки, выключения и перезагрузки системы;

– `init`– изменение уровня исполнения ОС (уровень 0 – останов, уровень 6 – перезагрузка);

– `halt`– команды остановки системы;

– `poweroff`– команда выключения системы;

– `reboot`– команда перезагрузки системы.

При физическом выключении системы с помощью штатных органов управления (не резким прерыванием питания) система автоматически выполняет процедуру останова.

ВНИМАНИЕ! ВНЕЗАПНОЕ ОТКЛЮЧЕНИЕ ПИТАНИЯ МОЖЕТ ПРИВЕСТИ К ПОТЕРЕ ДАННЫХ ИЛИ ПОВРЕЖДЕНИЮ СИСТЕМНЫХ ФАЙЛОВ. НАСТОЯТЕЛЬНО РЕКОМЕНДУЕТСЯ ВЫКЛЮЧАТЬ СИСТЕМУ С ПОМОЩЬЮ СРЕДСТВ ОС И ШТАТНЫХ ОРГАНОВ УПРАВЛЕНИЯ. ВО ИЗБЕЖАНИЕ ВНЕЗАПНОГО ОТКЛЮЧЕНИЕ ВНЕШНЕГО ПИТАНИЯ ИСПОЛЬЗОВАТЬ БЕСПЕРЕБОЙНЫЕ ИСТОЧНИКИ ПИТАНИЯ (ИБП).

В отдельных случаях, когда требуется применить изменения в файлах конфигурации, которые используется только при начальной загрузке, или обновления занятых системных файлов требуется перезагрузка системы.

Примечание. Перезагрузка может быть выполнена в случае зависания системы, когда невозможно восстановить ее работоспособность иным способом.

ФЛИР.90001-01 34 01

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АС	-	автоматизированная система
БД	-	база данных
ДНК	-	дезоксирибонуклеиновая кислота
ЖД	-	жесткий диск
ИБП	-	источник бесперебойного питания
ЛКМ	-	левая кнопка «мыши»
МРД	-	мандатное разграничение доступа
МСВС	-	мобильная система Вооруженных Сил
НДВ	-	недекларируемые возможности
НСД	-	несанкционированный доступ
ОН	-	общее назначение
ОП	-	оперативная память
ОС	-	операционная система
ПРД	-	правила разграничения доступа
ПК	-	персональный компьютер
ПО	-	программное обеспечение
ПС	-	программное средство
ПЭВМ	-	персональная электронная вычислительная машина
РД	-	руководящий документ
СЗИ	-	средства защиты информации
СУБД	-	система управления базами данных
ФС	-	файловая система
ФСТЭК	-	Федеральная служба по техническому и экспортному контролю

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	изменённых	заменённых	новых	аннулированных					