

ОПЕРАЦИОННАЯ СИСТЕМА АЛЬТ СЕРВЕР 8.0

Описание функциональных характеристик

СОДЕРЖАНИЕ

1	Общие сведения об ОС Альт Сервер 8.0	7
1.1	Краткое описание возможностей	7
1.2	Структура программных средств	7
2	Загрузка операционной системы	8
2.1	Настройка загрузки	8
2.2	Получение доступа к зашифрованным разделам	9
2.3	Вход в систему	10
2.4	Рабочий стол МАТЕ	10
3	Обзор приложений	15
3.1	Веб-навигация	15
3.1.1	Mozilla Firefox	16
3.2	Обмен мгновенными сообщениями	16
3.2.1	Pidgin	17
3.3	Офисные приложения	19
3.3.1	LibreOffice	20
4	Настройка системы	21
4.1	Центр управления системой	21
4.2	Применение центра управления системой	21
4.3	Запуск Центра управления системой в графической среде	22
4.4	Использование веб-ориентированного центра управления системой	23
4.4.1	Способы запуска	23
4.5	Развёртывание офисной ИТ-инфраструктуры	25
4.5.1	Подготовка	25
4.5.2	Домен	25
4.5.3	Сервер, рабочие места и аутентификация	26

4.6	Централизованная база пользователей	27
4.6.1	Создание учётных записей пользователей	27
4.6.2	Объединение пользователей в группы.....	27
5	Организация сетевой инфраструктуры с помощью сервера.....	27
5.1	Настройка подключения к Интернету	28
5.1.1	Настройка общего подключения к сети Интернет	29
5.1.2	Автоматическое присвоение IP-адресов (DHCP-сервер).....	30
5.2	Сервер обновлений для машин локальной сети	31
5.3	Сетевая установка операционной системы на рабочие места	31
5.3.1	Подготовка сервера.....	32
5.3.2	Подготовка рабочих станций.....	32
5.4	Сервер электронной почты (SMTP, POP3/IMAP).....	33
5.4.1	Сервер электронной почты	33
5.4.2	Сервер SMTP	33
5.4.3	Сервер POP3/IMAP	33
5.5	Соединение удалённых офисов (OpenVPN-сервер)	34
5.6	Доступ к службам сервера из сети Интернет	34
5.6.1	Список блокируемых хостов	35
5.7	Статистика	35
5.7.1	Сетевой трафик	35
5.7.2	Прокси-сервер	35
5.8	Обслуживание сервера	36
5.8.1	Мониторинг состояния системы	36
5.8.2	Системные службы	36
5.8.3	Резервное копирование.....	36
5.8.4	Обновление системы	38
5.8.5	Администратор системы	38
5.8.6	Дата и время	38

5.8.7	Ограничение использования диска	38
5.8.8	Выключение и перезагрузка компьютера.....	39
5.9	Прочие возможности ЦУС	39
5.9.1	DNS-сервер	39
5.10	Samba 4 в роли контроллера домена Active Directory	39
5.10.1	Установка.....	40
5.10.2	Миграция существующего сервера	40
5.10.3	Создание нового домена.....	40
5.10.4	Запуск службы.....	42
5.10.5	Проверка работоспособности	43
5.10.6	Управление пользователями	44
5.10.7	Заведение вторичного DC	44
5.10.8	Репликация.....	45
5.11	Samba в режиме файлового сервера.....	46
5.11.1	Настройка smb.cfg.....	46
5.11.2	Монтирование ресурса Samba	47
5.12	SOGo.....	47
5.12.1	Установка.....	47
5.12.2	Подготовка среды	47
5.12.3	Включение веб-интерфейса	49
5.12.4	Настройка почты	50
5.13	FreeIPA	50
5.13.1	Установка сервера FreeIPA	50
5.13.2	Установка FreeIPA клиента и подключение к серверу	51
5.13.3	Настройка доверительных отношений с AD	52
5.14	Резервное копирование данных.....	53
5.14.1	Vacula.....	53
5.14.2	Работа в Vacula с использованием командной строки (bconsole)	64

5.15	Zabbix	64
6	Настройка сети	64
6.1	NetworkManager	64
7	Установка дополнительного программного обеспечения	65
7.1	Программа управления пакетами Synaptic	65
7.2	Добавление репозитория	66
8	Общие принципы работы ОС	66
8.1	Процессы функционирования ОС	67
8.2	Файловая система ОС	68
8.3	Организация файловой структуры	68
8.3.1	Иерархическая организация файловой системы	69
8.3.2	Имена дисков и разделов	70
8.4	Разделы, необходимые для работы ОС	71
8.5	Утилиты для работы с файловой системой	71
8.5.1	mkfs	71
8.5.2	fsck	72
8.5.3	df	77
8.5.4	du	78
8.5.5	find	80
8.5.6	which	82
8.5.7	cd	84
8.5.8	pwd	84
8.5.9	mkdir	84
8.5.10	ls	85
8.5.11	cp	90
8.5.12	mv	92
8.5.13	rm	93
8.5.14	cat	94

8.5.15	more	94
8.5.16	ln.....	96
8.5.17	file	97
8.5.18	chmod.....	101
8.5.19	chown.....	103
8.5.20	umask	103
8.5.21	chattr.....	103
8.5.22	lsattr.....	105
9	Работа с наиболее часто используемыми компонентами	106
9.1	Командные оболочки (интерпретаторы)	106
9.1.1	Командная оболочка Bash	106
9.1.2	Базовые команды оболочки Bash	107
9.2	Текстовый редактор vi.....	108
9.2.1	Открыть/создать файл	109
9.2.2	Перемещение по файлу	109
9.2.3	Редактирование файла.....	109
9.2.4	Сокращения	110
9.2.5	Запись/выход	110
9.2.6	Дополнительные возможности	111
9.3	Редактор VIM	111
9.3.1	Режимы работы	111
9.3.2	Основные возможности.....	112
9.3.3	Конфигурация.....	114
9.4	xinetd.....	114
9.5	Crontab.....	117
9.5.1	Контроль доступа к crontab	117
9.5.2	Формат записи файла crontab.....	118
9.5.3	Примеры.....	119

10	Общие правила эксплуатации.....	120
10.1	Включение компьютера	120
10.2	Выключение компьютера.....	120
11	Техническая поддержка.....	120

1 ОБЩИЕ СВЕДЕНИЯ ОБ ОС АЛЬТ СЕРВЕР 8.0

1.1 Краткое описание возможностей

Операционная система Альт Сервер 8.0 (далее – ОС Альт Сервер 8.0), представляет собой совокупность интегрированных программных продуктов, созданных на основе операционной системы Linux. ОС Альт Сервер 8.0 универсальный и многофункциональный дистрибутив.

Альт Сервер 8.0 — многофункциональный дистрибутив для серверов с возможностью использования в качестве рабочей станции разработчика комплексных систем, прежде всего, предназначен для использования в корпоративных сетях.

Альт Сервер 8.0 это комплекс серверных приложений, оснащённый удобным пользовательским интерфейсом для настройки. Управление сервером может осуществляться с любого компьютера через веб-браузер.

В зависимости от конфигурации, сервер на основе ОС Альт Сервер 8.0, может обслуживать процессы в пределах одной компьютерной системы или процессы на других машинах через каналы передачи данных или сетевые соединения. Основные преимущества:

- графическая рабочая среда МАТЕ;
- установка серверных решений и решений конечных пользователей с одного диска;
- возможность как развернуть, так и использовать только определённые службы без Alterator;
- широкий выбор различных программ для профессиональной работы в сети Интернет, с документами, со сложной графикой и анимацией, для обработки звука и видео, разработки программного обеспечения и образования.

ОС Альт Сервер 8.0 обеспечивает обработку, хранение и передачу информации в круглосуточном режиме эксплуатации.

1.2 Структура программных средств

Альт Сервер 8.0 состоит из набора компонентов предназначенных для реализации функциональных задач необходимых пользователям (должностным лицам для выполнения определённых должностными инструкциями, повседневных действий) и поставляется в виде дистрибутива и комплекта эксплуатационной документации.

В структуре Альт Сервер 8.0 можно выделить следующие функциональные элементы:

- операционная среда (далее – ОСр) изделия;
- операционная система (далее – ОС) изделия;
- ядро ОС;
- системные библиотеки;
- утилиты и драйверы;

- средства обеспечения информационной безопасности;
- системные приложения;
- средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных;
- системы мониторинга и управления;
- средства подготовки исполнимого кода;
- средства версионного контроля исходного кода;
- библиотеки подпрограмм (SDK);
- среды разработки, тестирования и отладки;
- интерактивные рабочие среды;
- программные серверы;
- web-серверы;
- системы управления базами данных;
- прочие серверные программы;
- графическая оболочка MATE;
- командные интерпретаторы;
- прочие системные приложения;
- прикладное программное обеспечение общего назначения;
- офисные приложения.

2 ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ

2.1 Настройка загрузки

Linux, установленный на жёстком диске, загружается при включении компьютера при помощи специальной программы – загрузчика. Программа-загрузчик исполняется при загрузке системы с жёсткого диска и загружает ядро ОС Linux, расположенное также на жёстком диске.

Загрузчики Linux можно также использовать для загрузки нескольких операционных систем, поскольку они позволяют выбирать при включении компьютера, какую систему нужно загрузить в этот раз. Если есть выбор из нескольких вариантов загрузки, то после некоторого времени ожидания будет загружена та система, которая выбрана по умолчанию: это не обязательно должен быть Linux, а может быть другая операционная система или специальный режим загрузки (например, восстановительный).

Например, при стандартной установке в начальном меню загрузчика ОС Альт Сервер 8.0 доступны несколько вариантов загрузки (рис.1). Дополнительно предлагается вариант загрузки в программу проверки оперативной памяти (memtest).

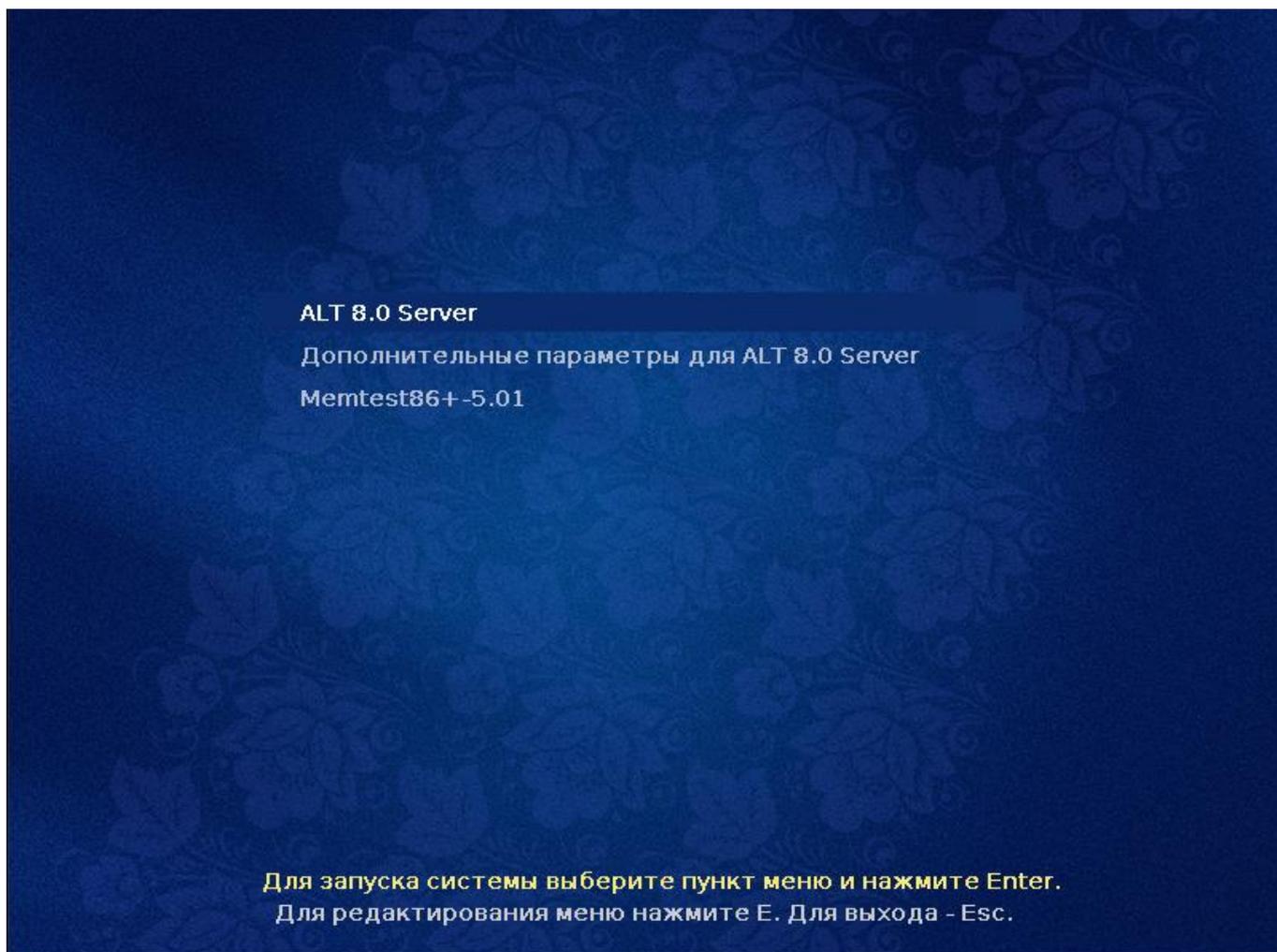


Рис.1.

Нажав «e» можно указать параметры, которые будут переданы ядру Linux при загрузке.

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

2.2 Получение доступа к зашифрованным разделам

В случае если вы создали зашифрованный раздел, вам потребуется вводить пароль при обращении к этому разделу.

Например, если вы зашифровали домашний раздел **/home**, то для того, чтобы войти в систему под своим именем пользователя, вам потребуется ввести пароль этого раздела и затем нажать **Enter**.

2.3 Вход в систему

Для регистрации в системе необходимо выбрать имя пользователя из выпадающего списка (рис.2). Далее необходимо ввести пароль, затем нажать **Enter** или щелкнуть на кнопке **Войти**. После непродолжительного времени ожидания запустится графическая оболочка операционной системы.

Окно входа в систему

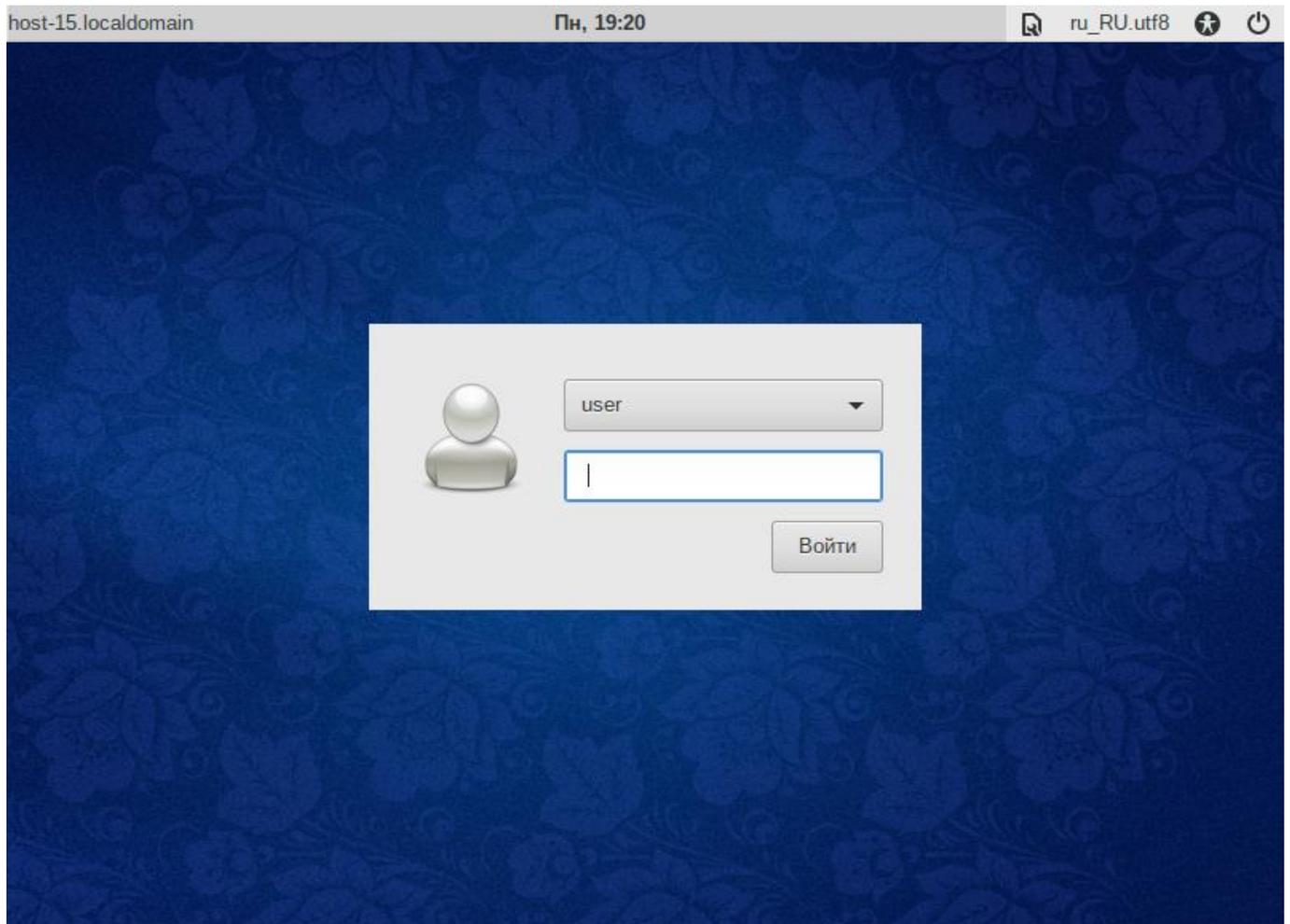


Рис.2.

Добавлять новых пользователей или удалять существующих можно после загрузки системы с помощью стандартных средств управления пользователями.

Если систему устанавливали не вы, то имя *системного пользователя* и его *пароль* вам должен сообщить системный администратор, отвечающий за настройку данного компьютера.

Поскольку работа в системе с использованием учётной записи *администратора системы* небезопасна, вход в систему в графическом режиме для суперпользователя root запрещён. Попытка зарегистрироваться в системе будет прервана сообщением об ошибке.

2.4 Рабочий стол МАТЕ

На рабочем столе МАТЕ есть три особые области. Сверху вниз (рис.3):

- верхняя панель (серая полоса вверху экрана);
- область рабочего стола (рабочая площадь в центре, занимающая большую часть экрана);
- панель со списком окон (серая полоса внизу экрана).

Рабочий стол MATE

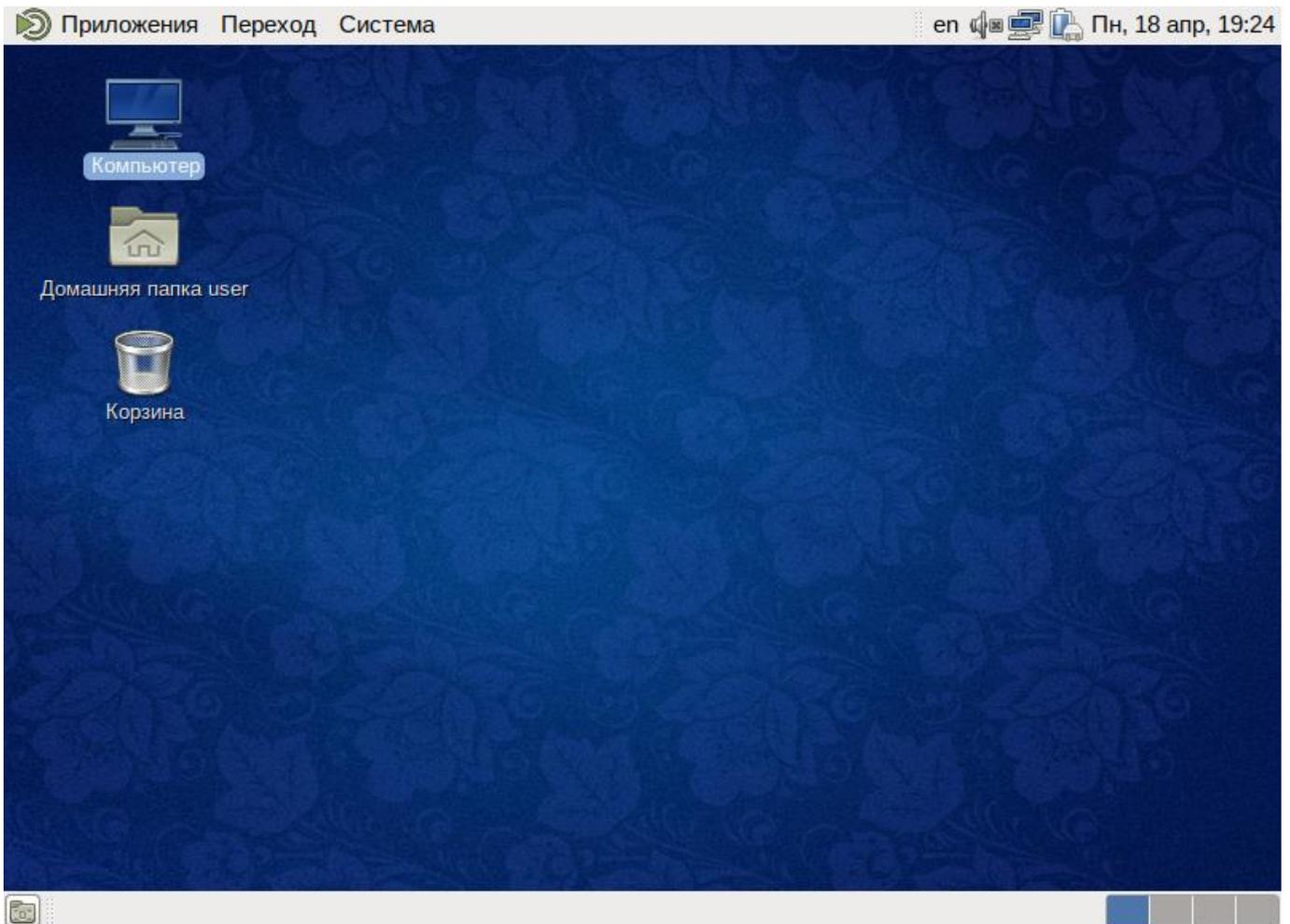


Рис.3

Верхняя панель расположена в верхней области экрана. Левая часть панели содержит:

- меню **Приложения**;
- меню **Переход**;
- меню **Система**.

Правая часть панели содержит:

- область уведомлений;
- регулятор громкости и апплет настройки звука;
- приложение «Сетевые соединения»;
- часы и календарь;
- параметры клавиатуры;

- параметры управления питанием.

Меню Приложения содержит список установленных приложений. Этот список обновляется при установке или удалении программ. При нажатии на Приложения открывается список, состоящий из следующих разделов:

- Графика;
- Инструменты;
- Мультимедиа;
- Офис;
- Сеть;
- Системные.

Другие обычные пункты, добавляемые с помощью дополнительного программного обеспечения, включают:

- Обучение;
- Другое;
- Программирование.

Меню переход разделено на четыре подраздела. Щелчок по любому пункту в меню Переход открывает файловый менеджер Саја. Для вызова руководства Саја нажмите: меню Справка →Содержание.

Первый подраздел.

- «Домашняя папка» — в ней по умолчанию хранятся ваши личные файлы.
- «Рабочий стол» — папка внутри вашей «Домашней папки», содержащая файлы и папки, отображаемые на рабочем столе.
- Дальнейшие пункты соответствуют вашим закладкам в файловом менеджере Саја.

Второй подраздел.

- «Компьютер» — этот подраздел позволяет увидеть все файлы в компьютере и файлы на подключённых внешних носителях.
- «Устройство CD/DVD» — этот подраздел позволяет получить доступ к CD/DVD дисководу.

Третий подраздел.

- «Сеть» позволяет просматривать сетевые подключения вашего компьютера. Осуществляет получение доступа к файлам и другим ресурсам, доступным в этих сетях.
- «Соединиться с сервером» позволяет вам создать подключение к публичным или локальным сетям.

Четвёртый подраздел.

- «Искать файлы» позволяет быстро найти файлы, хранящиеся на вашем компьютере.
- «Недавние документы» содержит список последних документов, с которыми вы работали. Последний пункт этого подменю позволяет очистить список.

С помощью меню Система осуществляется доступ к настройкам МАТЕ, справочной информации и функциям запуска, перезагрузки и отключения компьютера. Это меню разделено на три подраздела.

Первый подраздел.

- «Параметры» содержит доступ к различным настройкам и предоставляет доступ к инструментам администрирования системы. В меню «Параметры» входят настройки:
 - Caja-Actions Comfiguration Tool;
 - Emerald Theme Manager;
 - HP Device Manager;
 - UserPasswd позволяет изменить пароль пользователя;
 - «Адаптеры Bluetooth» позволяет настраивать Bluetooth-устройства для работы с вашим компьютером.
 - «Внешний вид» позволяет настроить внешний вид вашего рабочего стола, включая фоновую картинку.
 - «Всплывающие уведомления» позволяет настроить стиль и позицию уведомлений.
 - «Вспомогательные технологии» дают возможность выбирать программы для увеличения частей экрана или для прочтения вами содержимого экранов.
 - «Главное меню» позволяет изменять список отображаемых элементов в меню Приложений и меню Настроек.
 - «Запускаемые приложения» выбирают приложения для автоматического запуска при входе.
 - «Клавиатура» запускает диалог настройки клавиатуры. Тут же можно задать используемые в системе раскладки клавиатуры.
 - «Комбинации клавиш клавиатуры» задают сочетания клавиш для выполнения определённых заданий в окружении рабочего стола.
 - «Мышь» позволяет настроить кнопки и другие параметры вашей мыши.
 - «Обо мне» хранит ту информацию о вас, которую вы можете передать другим людям в виде электронной визитки.
 - «Окна» позволяет вам настроить параметры поведения окон.

- «Предпочтительные приложения» дают вам возможность выбрать, какие приложения вы хотите использовать для конкретных задач.
- «Программа управления пакетами Synaptic» позволяет управлять пакетами. С помощью Synaptic вы можете управлять источниками пакетов (репозиториями), получать сведения об доступных пакетах, устанавливать/удалять/обновлять пакеты, производить поиск по ключевым словам среди доступных пакетов.
- «Сетевая прокси-служба» позволяет вам настроить прокси-сервер для вашего компьютера.
- «Сетевые соединения» отображают сетевые подключения компьютера и позволяют их настраивать.
- «Управление питанием» настраивает ваш компьютер на работу с различными параметрами энергосбережения.
- «Управление файлами» влияет на предоставление вам файлов и папок.
- «Хранитель экрана» позволяет настроить заставку для рабочего стола.
- «Экраны» задаёт разрешение и другие параметры вашего монитора.
- «Администрирование» позволяет получить доступ к следующим настройкам.
 - «Настройка печати» позволяет настроить принтеры и задать параметры печати.
 - «Центр управления системой» позволяет управлять наиболее востребованными настройками системы: пользователями, сетевыми подключениями, настройками даты/времени и т. п.
- Центр управления.

Второй подраздел.

- «О среде МАТЕ» показывает информацию об установленной среде МАТЕ.

Третий подраздел.

- «Заблокировать экран» служит для запуска хранителя экрана. Для возобновления работы после блокировки необходим ввод пароля.
- «Завершить сеанс пользователя...» необходим для завершения работы пользователя без выключения компьютера.
- «Выключить...» позволяет перезагрузить либо выключить компьютер.

Область рабочего стола включает в себя три значка:

- значок «Компьютер». Если вы щелкните дважды по этому значку, то откроется окно, с доступными устройствами хранения данных.

- значок «Домашняя папка пользователя». Этот значок ведет в папку, где хранятся пользовательские файлы (например, аудиозаписи, видеозаписи, документы) по умолчанию. У каждого пользователя своя «Домашняя» директория. Каждый пользователь имеет доступ только в свою «Домашнюю» директорию.
- значок «Корзина». Обычно, когда вы удаляете файл, он не удаляется из вашей системы. Вместо этого он помещается в «Корзину». С помощью этого значка вы и можете посмотреть или восстановить «удаленные файлы». Чтобы удалить файл из вашей системы, вы должны очистить «Корзину». Чтобы очистить «Корзину», щелкните правой кнопкой мыши по значку и выберите «Очистить корзину».

У этой панели **МАТЕ** три основных компонента:

- любые открытые приложения отображаются как кнопки в средней части окна. Тут отображаются все окна с области рабочего стола вне зависимости от того, видно окно или нет. Кнопка скрытого окна будет отображаться с белым фоном. Кнопка приложения, которое выбрано в данный момент, будет с серым фоном. Что бы переключаться между приложениями с помощью мыши, кликните по желаемому приложению левой кнопкой мыши, чтобы переключиться на него;
- переключатель рабочих мест — это группа квадратов в правом нижнем углу экрана. Они позволяют вам переключать рабочие места. Каждое рабочее место предоставляет отдельный рабочий стол, на котором можно расположить приложения. По умолчанию активно 4 рабочих места. Вы можете изменить это число, нажав правой кнопкой мышки на «переключателе рабочих мест» и выбрав «Параметры»;
- свернуть все окна – кнопка позволяет свернуть (развернуть) все открытые окна на текущем рабочем месте.

3 ОБЗОР ПРИЛОЖЕНИЙ

Linux содержит огромное число приложений (программ) для выполнения всех повседневных задач. При этом важно понимать, что для выполнения одного и того же действия могут быть использованы разные приложения. Например, для написания простых текстов доступен целый ряд текстовых редакторов с разным набором возможностей.

3.1 Веб-навигация

Веб-браузеры – комплексные программы для обработки и отображения HTML-страниц по протоколу HTTP и HTTPS (открытие страниц сайтов, блогов и т.д.). Основное назначение веб-браузера – предоставление интерфейса между веб-сайтом и его посетителем. В базовые функции современных веб-браузеров входят:

- навигация и просмотр веб-ресурсов;

- показ оглавлений FTP-серверов и скачивание файлов;
- поддержка скриптовых языков.

Основные принципы работы с веб-браузером неизменны. Программа предоставляет пользователю адресную строку, в которую вносится адрес необходимого вам сайта. Эта же строка может использоваться для ввода поискового запроса. Для более быстрого доступа адреса часто посещаемых сайтов добавляются в закладки. Для перехода к предыдущей/следующей просмотренной веб-странице, как правило, предусмотрены специальные кнопки на панели инструментов.

3.1.1 Mozilla Firefox

Программа **Mozilla Firefox** – веб-браузер, поддерживающий большинство современных веб-технологий и интернет-протоколов. Браузер **Mozilla Firefox** предлагает пользователю логичный интерфейс и возможность полностью контролировать свою работу в Интернете.

Веб-браузер **Mozilla Firefox** предоставляет широкие возможности настройки: пользователь может устанавливать дополнительные темы, изменяющие внешний вид программы, и расширения, добавляющие новую функциональность.

Для того чтобы открыть интернет-страницу, введите её адрес в адресную строку браузера и нажмите **Enter**. Если вы хотите открыть ссылку на следующую страницу в новой вкладке, то нажмите на ней средней кнопкой (колесом) мыши. Возможно настроить одновременный просмотр нескольких страниц в разных вкладках одного окна.

Для быстрого доступа к часто посещаемым веб-страницам создайте ссылки на **Панели закладок**. Управление закладками и их редактирование осуществляется в рамках диалогового окна **Библиотека**.

Панель навигации помогает пользователю искать:

- интеллектуальная строка ввода адреса предоставляет окно-подсказку с историей ранее открытых сайтов;
- строка поиска предлагает пользователю функцию поиска по мере набора текста.

Веб-браузер **Mozilla Firefox** работает как полнофункциональный FTP-клиент. Процесс загрузки найденных в Интернете файлов на жёсткий диск компьютера отображается в диалоговом окне менеджера загрузок. В меню веб-браузера **Правка** → **Настройки** можно указать папку для сохранения файлов или выбрать возможность назначать папку при сохранении файлов.

3.2 Обмен мгновенными сообщениями

Для обмена сообщениями в режиме реального времени через Интернет необходима специализированная клиентская программа, передающая текстовые сообщения, а также файлы различных типов. Система мгновенного обмена сообщениями является одним из самых доступных и востребованных средств общения в Интернете. Преимущества инструментов мгновенного обмена информацией:

- скорость — мгновенные сообщения позволяют собеседникам общаться со скоростью нажатия на кнопку, без необходимости открывать письма и ждать ответа;
- удобство — программы обмена мгновенными сообщениями включают широкий набор коммуникативных и производственных функций.

Большинство современных программ мгновенного обмена сообщениями позволяют видеть, подключены ли в данный момент абоненты, занесённые в список контактов. Сообщения появляются на мониторе собеседника только после окончания редактирования и отправки. В список основных функций служб мгновенных сообщений входят:

- чат (видеочат, текстовый и голосовой);
- VoIP сервисы: звонки на компьютер, звонки на стационарные и мобильные телефоны;
- возможность отправки SMS;
- передача файлов;
- инструменты для совместной работы в режиме реального времени;
- возможность общаться в чате непосредственно на веб-странице;
- напоминания и оповещения;
- хранение истории общения по каждому контакту;
- индикация о сетевом статусе занесённых в список контактов пользователей (в сети, нет на месте и т.д.).

Существуют клиентские программы, позволяющие подключаться одновременно к нескольким сетям. Они поддерживают наиболее популярные протоколы, что избавляет вас от необходимости устанавливать отдельный IM-клиент для каждой сети.

3.2.1 Pidgin

Pidgin — мультипротокольная программа-клиент для мгновенного обмена сообщениями, позволяющая одновременно подключиться к нескольким сетям (рис.4). Поддерживает наиболее популярные протоколы: AIM, Bonjour, Facebook (XMPP), Gadu-Gadu, Google-общение, GroupWise, ICQ, IRC, MSN, Mxit, MySpaceIM, SIMPLE, Sametime, XMPP, Yahoo, Yahoo JAPAN и Zephyr.

Окно списка собеседников Pidgin

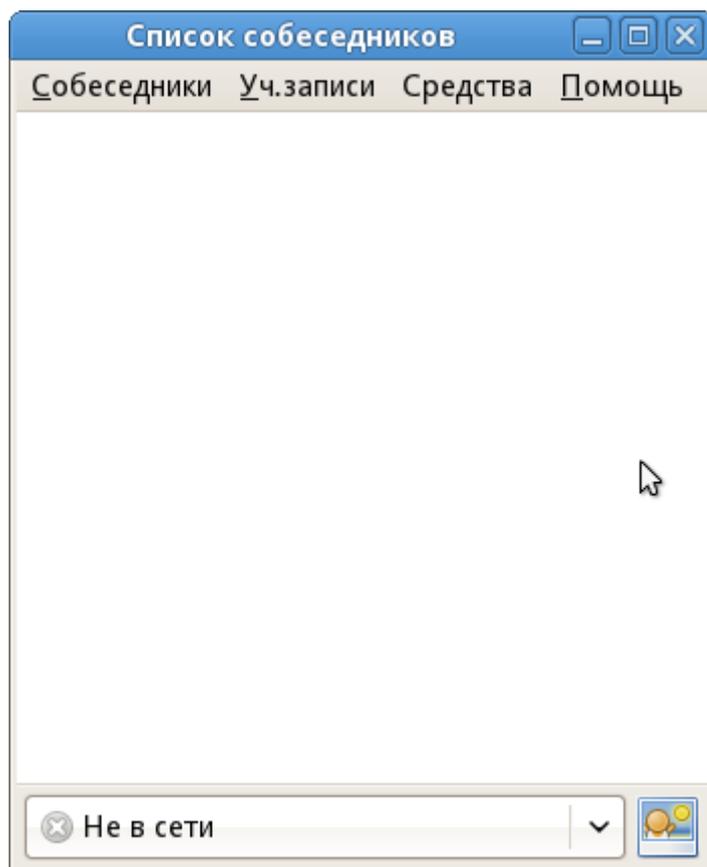


Рис.4

Функции Pidgin:

- поддержка особенностей различных сетей (статус-сообщения, значки друзей, уведомление о наборе текста...);
- зашифрованный чат;
- возможность установки модулей расширения;
- возможность «метаконтактов»;
- запись протокола событий;
- поддержка вкладок в окне разговора;
- одновременное подключение к нескольким аккаунтам;
- слежение за пользователями;
- многоязычный интерфейс.

После запуска Pidgin необходимо произвести его первоначальную настройку. При первом запуске Pidgin из меню Уч.записи → Управление учётными записями необходимо запустить диалоговое окно мастера создания учётной записи и создать учётную запись пользователя (рис.5).

Из списка поддерживаемых служб выберите ту, которую собираетесь использовать. Возможно, вы уже решили, какую службу IM будете использовать (потому что вы уже пользовались ею, либо потому что ею пользуются ваши друзья). Если вы ещё не остановили свой выбор на ка-

кой-то определённой службе IM, то выберите службу, основанную на открытых стандартах, например jabber.

Диалоговое окно мастера создания учётной записи

Добавить учётную запись

Основные Advanced Прокси Voice and Video

Параметры входа

Протокол: XMPP

Имя пользователя:

Домен:

Ресурс:

Пароль:

Запомнить пароль

Параметры пользователя

Локальный псевдоним:

Уведомления о новой почте

Использовать этот значок собеседника для этой учётной записи:

Создать эту новую учётную запись на сервере

Отменить Добавить

Рис.5

После настройки учётной записи добавьте в список контактов ваших собеседников (кнопка **Добавить собеседника...**) и, при условии, что нужный вам собеседник подключён к службе мгновенных сообщений, можете начинать общение.

За дополнительной информацией по использованию Pidgin можно обратиться к справке, вызываемой из меню Помощь → Помощь в сети.

3.3 Офисные приложения

Офисными приложениями традиционно называют пакет программ для работы с текстами, таблицами и презентациями.

3.3.1 LibreOffice

LibreOffice – пакет программ для работы с офисными документами (рис.6). Кроме стандартных для LibreOffice форматов хранения данных, вы можете успешно открывать и сохранять документы, созданные в других популярных офисных пакетах.

Пакет программ LibreOffice

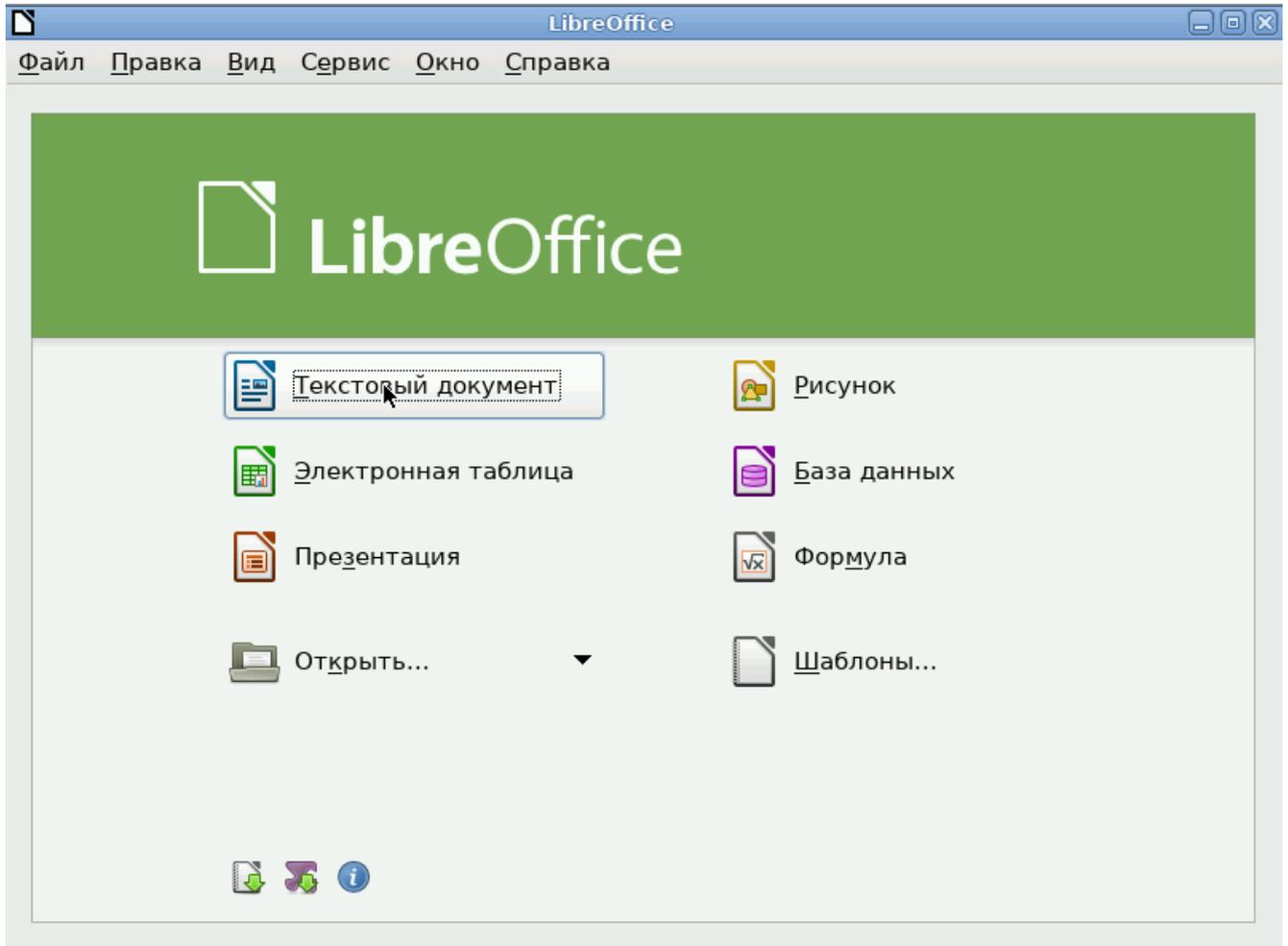


Рис.6

Текстовый процессор (LibreOffice Writer) позволяет проектировать и создавать текстовые документы, содержащие изображения, таблицы или графики. Вы можете сохранять документы в различных форматах, включая стандартизированный формат OpenDocument format (ODF), формат Microsoft Word (DOC, DOCX) или HTML. Кроме того, вы можете без труда экспортировать ваш документ в формате переносимого документа (PDF). Текстовый процессор поддерживает и другие форматы.

Электронная таблица (LibreOffice Calc) предназначена для работы с электронными таблицами. Инструментарий электронных таблиц включает мощные математические функции, позволяющие вести сложные статистические, финансовые и прочие расчёты.

Презентация (LibreOffice Impress) позволяет создавать профессиональные слайд-шоу, которые могут включать диаграммы, рисованные объекты, текст, мультимедиа и множество других элементов. При необходимости можно даже импортировать и изменять презентации Microsoft PowerPoint. Для того чтобы сделать экранные презентации более эффектными, можно использовать такие средства, как анимация, мультимедиа и переходы между слайдами.

Редактор рисунков (LibreOffice Draw) позволяет создавать рисунки различной сложности и экспортировать их с использованием нескольких общепринятых форматов изображений. Кроме того, можно вставлять в рисунки таблицы, диаграммы, формулы и другие элементы, созданные в программах LibreOffice.

Базы данных (LibreOffice Base) поддерживает некоторые обычные файловые форматы баз данных, например, BASE. Кроме того, можно использовать **LibreOffice Base** для подключения к внешним реляционным базам данных, например, к базам данных MySQL или Oracle. В базе **LibreOffice Base** невозможно изменить структуру базы данных или редактировать, вставлять и удалять записи для ниже перечисленных типов баз данных (они доступны только для чтения):

- файлы электронной таблицы;
- текстовые файлы;
- данные адресной книги.

4 НАСТРОЙКА СИСТЕМЫ

4.1 Центр управления системой

Для управления настройками установленной системы вы можете воспользоваться **Центром управления системой**. **Центр управления системой** представляет собой удобный интерфейс для выполнения наиболее востребованных административных задач: добавление и удаление пользователей, настройка сетевых подключений, просмотр информации о состоянии системы и т.п.

Центр управления системой состоит из нескольких независимых диалогов-модулей. Каждый модуль отвечает за настройку определённой функции или свойства системы.

4.2 Применение центра управления системой

Вы можете использовать центр управления системой для разных целей, например:

- настройки **Даты и времени**;
- настройки **Раскладок клавиатуры**;
- изменения **Разрешения экрана**;
- установка **Загрузчика Grub**;
- изменения пароля **Администратора системы (root)**;
- создания, удаления и редактирования учётных записей **Пользователей**.

Вы всегда можете воспользоваться кнопкой **Справка**. Все модули центра управления системой имеют справочную информацию.

4.3 Запуск Центра управления системой в графической среде

Центр управления системой можно запустить следующими способами:

- в графической среде МАТЕ: **Система** → **Администрирование** → **Центр управления системой**;
- из командной строки: командой `асс`.

При запуске необходимо ввести пароль администратора системы (root) (рис.7).

Запуск Центра управления системой

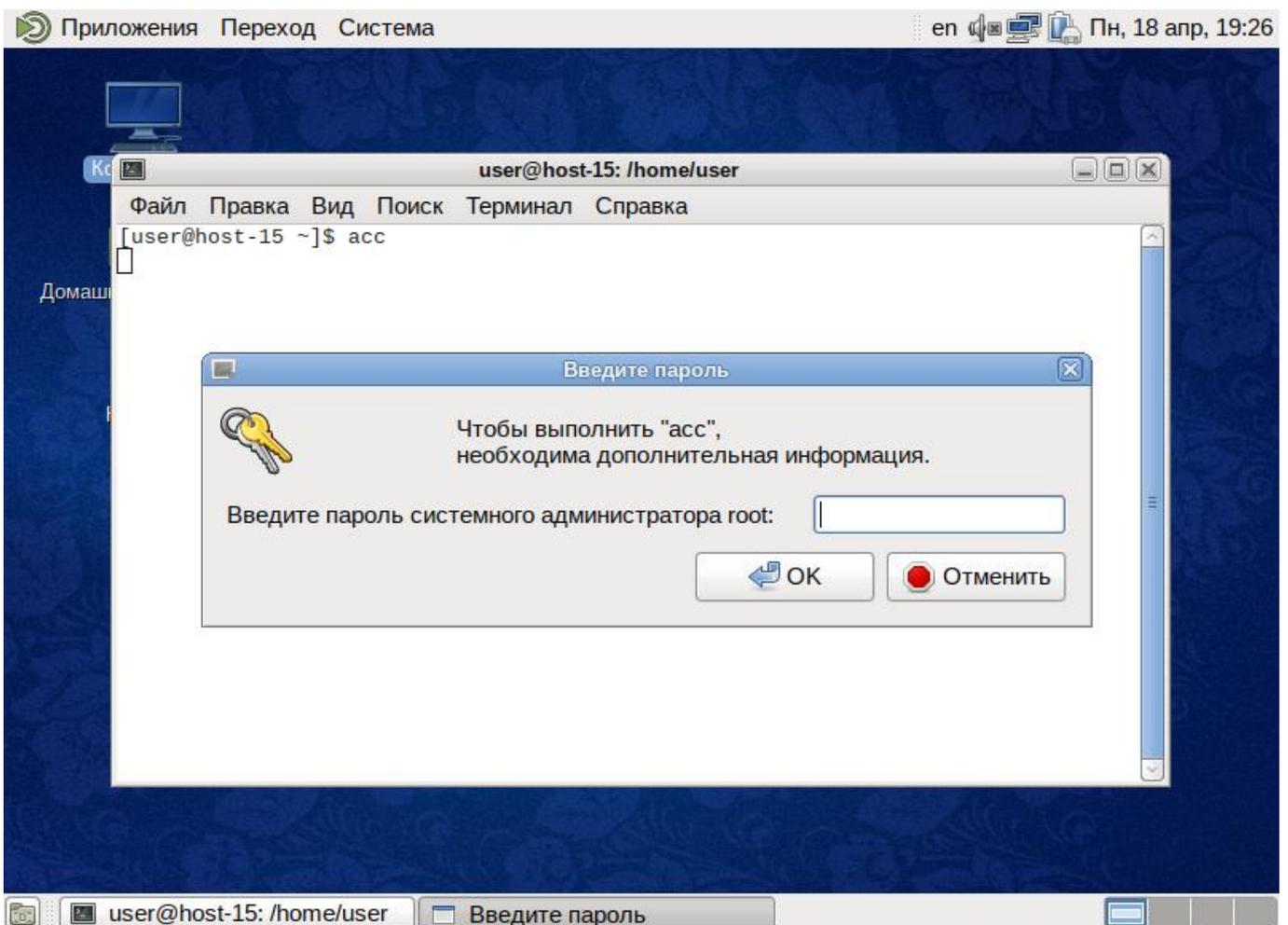


Рис.7

После успешного входа можно приступить к настройке системы (рис.8).

Центр управления системой

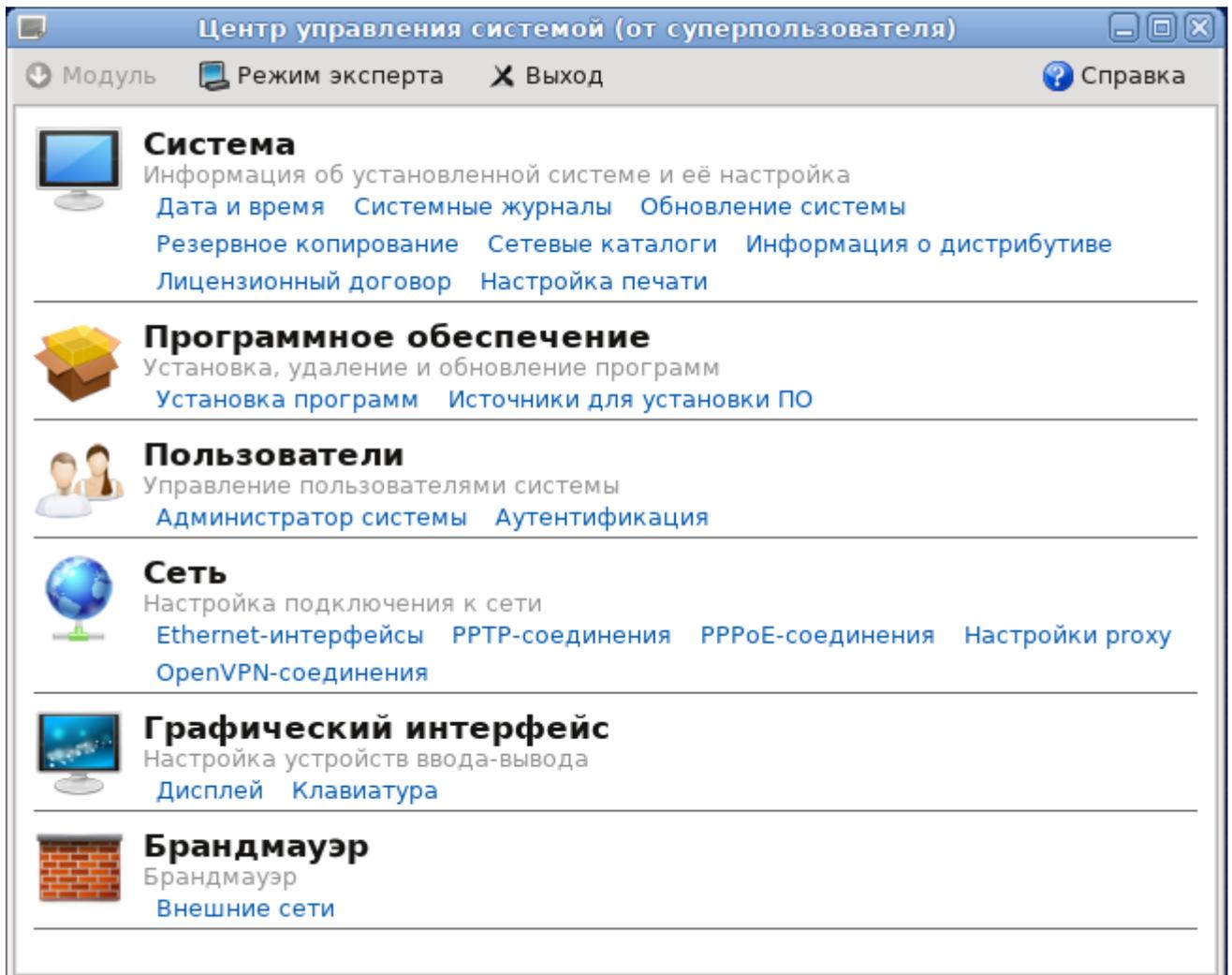


Рис.8

4.4 Использование веб-ориентированного центра управления системой

4.4.1 Способы запуска

Работа с центром управления системой происходит из любого веб-браузера. Для начала работы необходимо перейти по адресу **https://ip-адрес:8080/**.

Например, вы задали для сервера IP-адрес 192.168.0.1. В таком случае:

- интерфейс управления доступен по адресу: `https://192.168.0.1:8080/`
- документация по дистрибутиву доступна по адресу: `http://192.168.0.1/`

Так же IP-адрес сервера можно узнать, введя на сервере команду:

```
$ ip addr
```

IP-адрес будет указан после слова `inet`:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16384 qdisc noqueue state UNKNOWN
```

```
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
inet 127.0.0.1/8 scope host lo
```

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
```

```
link/ether 60:eb:69:6c:ef:47 brd ff:ff:ff:ff:ff:ff
inet 192.168.0.1/24 brd 192.168.0.255 scope global enp0s3
```

Например, тут мы видим, что на интерфейсе enp0s3 задан IP-адрес 192.168.0.1.

При запуске центра управления системой необходимо ввести в соответствующие поля имя пользователя (**root**) и пароль пользователя root (рис.9).

Запуск веб-ориентированного центра управления системой

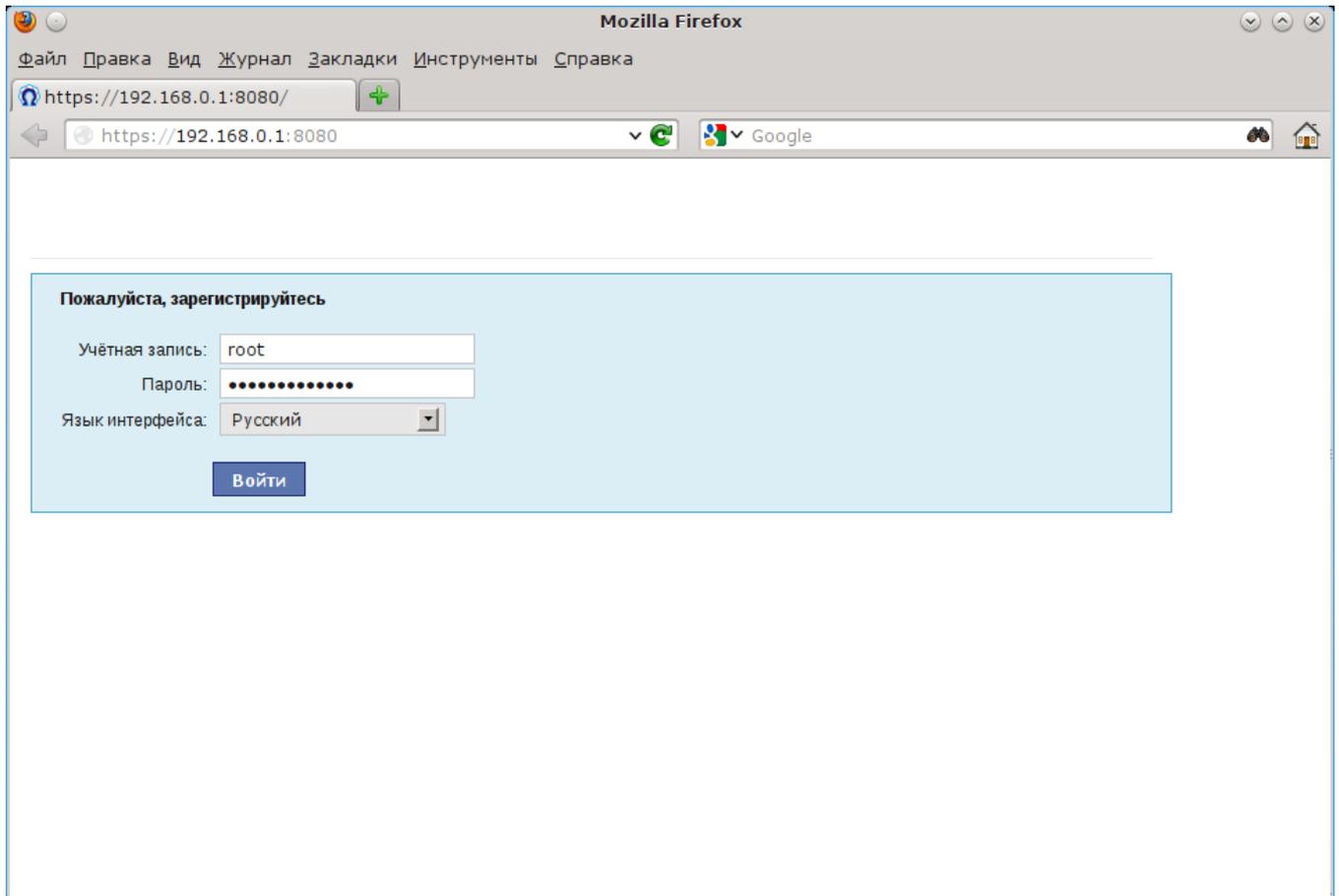


Рис.9.

Если в сети нет компьютера, который вы могли бы использовать для доступа к веб-ориентированному **Центру управления системой**, то вы можете воспользоваться браузером непосредственно на сервере.

Центр управления системой содержит справочную информацию по всем включённым в него модулям. Об использовании самого интерфейса системы управления можно прочитать, нажав, на кнопку **Справка** на начальной странице центра управления системой (рис.10).

Веб-ориентированный центр управления системой

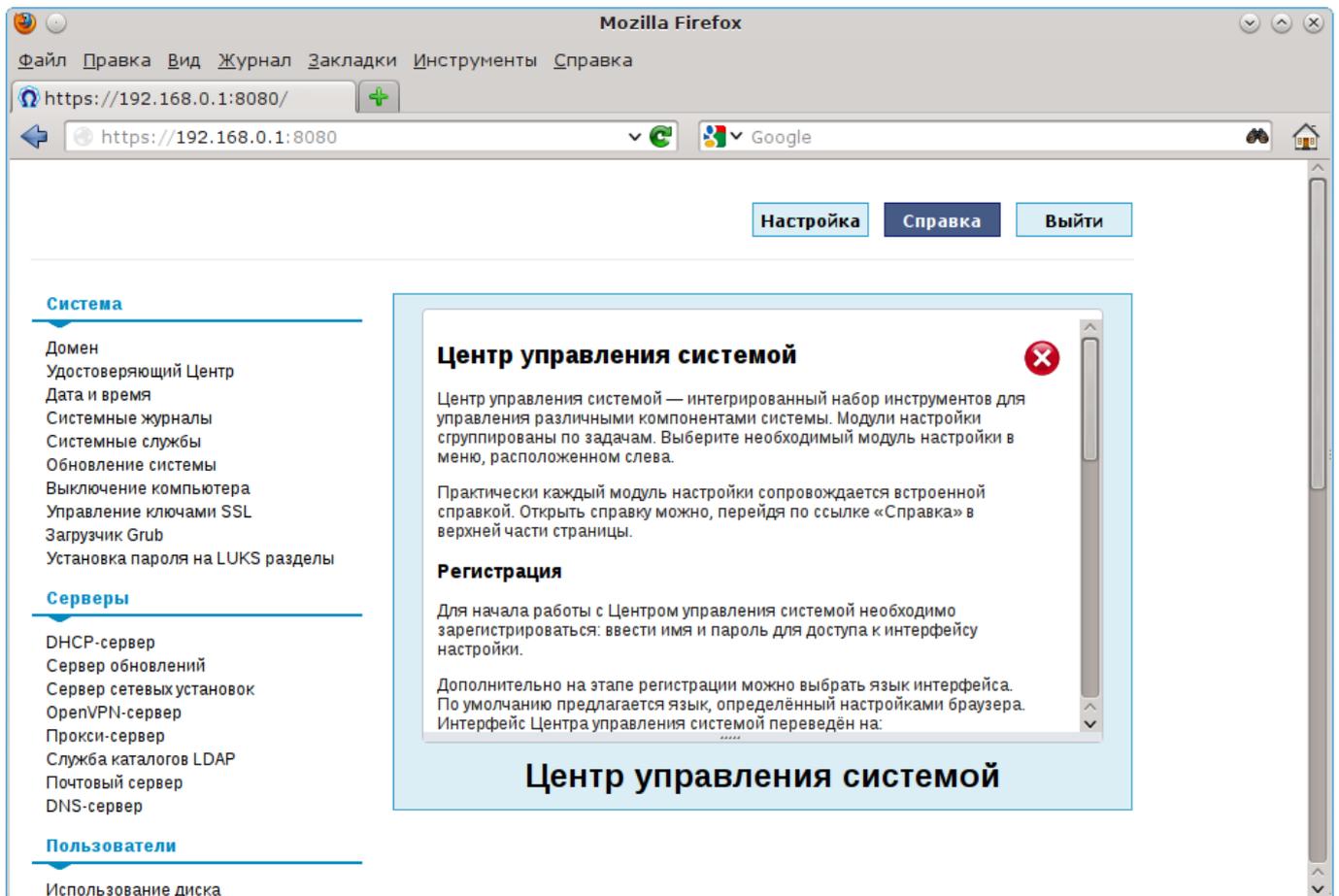


Рис.10

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

4.5 Развёртывание офисной ИТ-инфраструктуры

4.5.1 Подготовка

Перед началом развёртывания офисной ИТ-инфраструктуры необходимо провести детальное планирование. Конкретные решения в каждом случае будут продиктованы спецификой требований, предъявляемым к офисной ИТ-инфраструктуре. Как будет использоваться ОС Альт Сервер 8.0 в каждом конкретном случае решать вам. При этом важно понимать принципы взаимодействия компьютеров в сети и роль каждого конкретного компьютера: главный сервер, подчинённый сервер или компьютер-клиент (рабочее место).

Ключевым понятием для работы сети, построенной на базе ОС Альт Сервер 8.0, является **домен**.

4.5.2 Домен

Под доменом понимается группа компьютеров с разными ролями. Каждый сервер обслуживает один домен — группу компьютеров одной сети, имеющую единый центр и использующую единые базы данных для различных сетевых служб.

С помощью **Домена** вы можете:

- вести централизованную базу пользователей и групп;
- аутентифицировать пользователей и предоставлять им доступ к сетевым службам без повторного ввода пароля;
- использовать единую базу пользователей для файлового сервера, прокси-сервера, веб-приложений (например, MediaWiki);
- автоматически подключать файловые ресурсы с серверов, анонсированных по Zeroconf;
- использовать тонкие клиенты, загружаемые по сети и использующие сетевые домашние каталоги;
- аутентифицировать пользователей как на Linux, так и на Microsoft Windows.

4.5.3 Сервер, рабочие места и аутентификация

Важно понимать роль, которая будет отводиться ОС Альт Сервер 8.0 в домене. Именно сервер под управлением ОС Альт Сервер 8.0 будет являться центральным звеном сети, контролируя доступ к ресурсам сети и предоставляя различные службы для клиентских машин. Все службы, предоставляемые серверами, используются рабочими местами.

Таким образом, можно выделить:

Сервер (компьютер под управлением ОС Альт Сервер 8.0)

Сервер осуществляет контроль доступа к ресурсам сети, содержит централизованную базу данных пользователей и *удостоверяющий центр* для выдачи сертификатов службам на серверах и рабочих местах.

Рабочее место

Рабочие места — это клиентские, по отношению к серверам, компьютеры, непосредственно, использующиеся для работы пользователей.

Наибольший эффект от использования ОС Альт Сервер 8.0 достигается при использовании его вместе с рабочими местами под управлением ОС Альт Рабочая станция. Они уже содержат всё необходимое для интеграции в сеть с ОС Альт Сервер 8.0. Конечно, в качестве рабочих мест могут использоваться и другие операционные системы. Однако часть возможностей и преимуществ при этом может быть потеряна. Также возможно, на стороне компьютера-клиента потребуются дополнительная настройка.

Для доступа к ресурсам сети (например, общим файлам, расположенным на сервере, либо получения доступа в сеть Интернет) пользователю, работающему на клиентском компьютере, необходимо *авторизоваться* на сервере — ввести свои данные (имя и пароль). После проверки аутентификации главным сервером, пользователь получает определённый администратором домена объём прав доступа к ресурсам сети.

Авторизация

Типичный пример — офисное рабочее место, постоянно находящееся в локальной сети. В этом случае аутентификация в домене происходит непосредственно в момент регистрации пользователя на рабочем месте (с доменными аутентификационными данными).

Рабочие места под управлением ОС Альт Рабочая станция позволяют легко настроить такой способ аутентификации. Для этого в **Центре управления системой** (раздел **Аутентификация**) на рабочей станции, нужно выбрать домен, управляемый ОС Альт Сервер 8.0.

4.6 Централизованная база пользователей

Основной идеей домена является единая база учётных записей. При такой организации работы пользователям требуется лишь одна единственная учётная запись для доступа ко всем разрешённым администратором сети ресурсам. Наличие в сети единой централизованной базы пользователей позволяет значительно упростить работу, как самих пользователей, так и системных администраторов.

4.6.1 Создание учётных записей пользователей

Централизованная база пользователей создаётся на главном сервере. Наполнить её учётными записям можно воспользовавшись модулем ЦУС **Пользователи** (раздел **Пользователи**). После создания учётной записи пользователя не забудьте присвоить учётной записи пароль. Этот пароль и будет использоваться пользователем для регистрации в домене. После этого на рабочих местах, на которых для аутентификации установлен этот домен, можно вводить это имя пользователя и пароль.

4.6.2 Объединение пользователей в группы

Пользователи могут быть объединены в группы. Это может быть полезно для более точного распределения полномочий пользователей. Например, члены группы **wheel** могут получать полномочия администратора на локальной машине, выполнив команду `su -`. Настройка групп производится в модуле ЦУС **Группы** (раздел **Пользователи**).

За дополнительной информацией по настройке обращайтесь к встроенной справке соответствующих модулей ЦУС.

5 ОРГАНИЗАЦИЯ СЕТЕВОЙ ИНФРАСТРУКТУРЫ С ПОМОЩЬЮ СЕРВЕРА

ОС Альт Сервер 8.0 в сети организации может быть использован для решения различных задач. Он может предоставлять компьютерам сети общий доступ в Интернет, выступать в роли почтового сервера, файлового хранилища, веб-сервера и т.д. Все эти возможности обеспечиваются

соответствующими *службами*, запускаемыми на сервере. Для выбора и настройки необходимых служб воспользуйтесь **Центром управления системой**.

Дальнейшие разделы описывают некоторые возможности использования ОС Альт Сервер 8.0, настраиваемые в ЦУС.

5.1 Настройка подключения к Интернету

Помимо множества различных служб, которые ОС Альт Сервер 8.0 может предоставлять компьютерам сети, важно определить, будет ли сервер предоставлять общий доступ в Интернет для компьютеров домена или нет. В зависимости от этого сервер можно рассматривать как:

Сервер без подключения к сети Интернет

Типичный случай — это сервер с одним сетевым интерфейсом (одной сетевой картой), который и связывает его с компьютерами локальной сети. Такой сервер называется также *сервер рабочей группы*.

Шлюз

В этом случае сервер обычно имеет два сетевых интерфейса (например, две сетевые карты), одна из которых служит для подключения к локальной сети, а другая — для подключения к сети Интернет.

Как для обеспечения доступа в сеть Интернет самого ОС Альт Сервер 8.0, так и для настройки общего выхода в Интернет для компьютеров сети необходимо настроить подключение к Интернету на самом сервере. ОС Альт Сервер 8.0 поддерживает самые разные способы подключения к сети Интернет:

- Ethernet;
- PPTP;
- PPPoE;
- и т.д.

Для настройки подключения воспользуйтесь одним из разделов ЦУС **Сеть**.

Доступные разделы:

- Ethernet-интерфейсы;
- PPTP-соединения;
- PPPoE-соединения;
- OpenVPN-соединения.

Выберите раздел, соответствующий вашему типу подключения, и приступайте к настройке.

За дополнительной информацией по настройке обращайтесь к встроенной справке соответствующих модулей ЦУС.

5.1.1 Настройка общего подключения к сети Интернет

Пользователи корпоративных сетей обычно подключаются к сети Интернет через один общий канал. Для организации совместного доступа к сети Интернет стандартными средствами поддерживаются две технологии, которые могут использоваться как по отдельности, так и совместно:

- использование прокси-сервера;
- использование NAT.

Оба способа предполагают, что соединение с Интернет компьютера, через который предполагается настроить общий выход, предварительно сконфигурировано. Сделать это можно в разделе ЦУС **Сеть**.

5.1.1.1 Прокси-сервер

Отличительной особенностью использования прокси-сервера является то, что, помимо предоставления доступа к веб-сайтам, прокси-сервер кэширует загруженные страницы, а при повторном обращении к ним — отдаёт их из своего кэша. Это может существенно снизить потребление трафика.

У прокси-сервера есть два основных режима работы:

- прозрачный;
- обычный.

Для работы с прокси-сервером в прозрачном режиме не потребуется специальная настройка рабочих станций. Они лишь должны использовать сервер в качестве шлюза по умолчанию. Этого можно добиться, сделав соответствующие настройки на DHCP-сервере.

Для использования прокси-сервера в обычном режиме потребуется на каждом клиенте в настройках браузера указать данные прокси-сервера (IP-адрес и порт).

Преимуществом обычного режима работы, требующего перенастройки программ локальной сети, является возможность производить аутентификацию пользователей и контролировать их доступ во внешнюю сеть.

В различных браузерах местоположение формы настройки на прокси-сервер различное. Например, в браузере **Firefox** она доступна через меню **Правка** → **Настройки** → **раздел Дополнительно** → **вкладка Сеть+кнопка Настроить...** напротив текста «Настройка параметров соединения Firefox с Интернетом». Здесь следует выбрать **Ручная настройка сервиса прокси** и указать IP-адрес и порт прокси-сервера.

По умолчанию прокси-сервер не предоставляет доступ в Интернет никому кроме себя самого. Список сетей, обслуживаемых прокси-сервером можно изменить, нажав на кнопку «**Разрешённые сети...**» в модуле ЦУС **Прокси-сервер** (раздел **Серверы**).

Прокси-сервер позволяет вести статистику посещений страниц в Интернете. Она доступна в модуле ЦУС **Прокси-сервер** (раздел **Статистика**). Основное предназначение статистики — про-

смотреть отчёта об объёме полученных из Интернета данных в привязке к пользователям (если включена аутентификация) или к IP-адресам клиентов.

Примечание

Статистика не собирается по умолчанию. Включить её сбор следует в модуле ЦУС **Прокси-сервер** (раздел **Статистика**). Для этого отметьте **Включить сбор данных прокси-сервера** и нажмите кнопку **Применить**.

Примечание

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

5.1.1.2 NAT

NAT (Network Address Translation, преобразование сетевых адресов) — это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Таким образом, компьютеры локальной сети, имеющие IP-адреса, зарезервированные для использования исключительно в локальных сетях, могут использовать общий канал доступа к Интернет (общий внешний IP-адрес). При этом на компьютере-шлюзе, непосредственно подключённом к Интернет, выполняется преобразование адресов.

Настройка NAT осуществляется в модуле ЦУС **Внешние сети** (раздел **Брандмауэр**). Для минимальной настройки достаточно выбрать режим работы **Шлюз (NAT)**, отметить правильный внешний сетевой интерфейс и нажать на кнопку **Применить**.

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

5.1.2 Автоматическое присвоение IP-адресов (DHCP-сервер)

DHCP-сервер — это служба, позволяющая компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Это облегчает администрирование клиентских машин, избавляя администратора домена от необходимости вручную настраивать сетевые интерфейсы на компьютерах локальной сети.

Настройка DHCP-сервера осуществляется в модуле ЦУС **DHCP-сервер** (раздел **Серверы**). Здесь вы можете включить службу, задать диапазон динамических адресов, определить статические адреса и настроить прочие параметры.

Для включения DHCP-сервера установите флажок **Включить службу DHCP**, укажите начальный и конечный IP-адрес, а также шлюз по умолчанию (обычно, это IP-адрес сервера на сетевом интерфейсе, обслуживающем локальную сеть).

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

5.2 Сервер обновлений для машин локальной сети

Сервер обновлений — технология, позволяющая настроить автоматическое обновление программного обеспечения, установленного на клиентских машинах (рабочих местах).

Для операционных систем, установленных на клиентских машинах, постоянно выпускаются обновления программного обеспечения. Это могут быть как обновления, связанные с безопасностью, так и новый функционал или просто улучшение и ускорение алгоритмов. В любом случае настоятельно рекомендуется регулярно обновлять рабочие места для повышения надёжности работы.

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

Локальное зеркало репозитория

В этом режиме на сервере создаётся копия удалённого репозитория, доступная клиентским машинам по протоколу FTP. Загрузка ПО клиентскими машинами производится с локального сервера. Наличие на локальном сервере зеркала репозитория при большом количестве машин в сети позволяет существенно сэкономить трафик.

Публикация репозитория

В этом случае реального зеркалирования (загрузки пакетов) не происходит. Публикуется URL внешнего сервера, содержащего репозиторий. Такая публикация позволяет клиентским машинам автоматически настроить свои менеджеры пакетов на использование внешнего сервера. Загрузка ПО клиентскими машинами производится с внешнего сервера.

Со стороны ОС Альт Сервер 8.0 настройка производится в модуле управления **Сервер обновлений** (раздел **Система**). Помимо метода публикации репозитория, вы можете настроить и дополнительные параметры, такие как: расписание зеркалирования, выбор конкретных репозиториях для публикации и пр.

Со стороны клиентских машин на них необходимо настроить модуль **Обновление системы**, отметив в нём **Обновление системы управляемое сервером**.

За дополнительной информацией по настройке обращайтесь к встроенной справке соответствующих модулей ЦУС.

5.3 Сетевая установка операционной системы на рабочие места

Одной из удобных возможностей ОС Альт Сервер 8.0 при разворачивании инфраструктуры является сетевая установка. При помощи сетевой установки можно производить установку ОС Альт Сервер 8.0 не с DVD-диска, а загрузив инсталлятор по сети.

5.3.1 Подготовка сервера

Перед началом установки рабочих станций следует произвести предварительную настройку сервера: задать имя сервера (модуль **Ethernet-интерфейсы** в **Центре управления системой**), включить DHCP-сервер (модуль **DHCP-сервер**), задать имя домена (модуль **Домен**).

Примечание

При сетевой установке с сервера будут переняты настройки домена, и будет включена централизованная аутентификация. Если вы устанавливаете ОС Альт Сервер 8.0 с DVD-диска, то настройку домена и аутентификации надо будет производить отдельно на каждой рабочей станции.

Перед активацией сетевой установки потребуется импортировать установочный DVD-диск ОС Альт Сервер 8.0, предварительно вставив его в DVD-привод сервера, либо используя образ диска, расположенный на файловой системе на сервере. В разделе **Сервер сетевых установок**, укажите, откуда импортировать новый образ и нажмите кнопку **Добавить**. Процесс добавления занимает какое-то время. Пожалуйста, дождитесь окончания этого процесса.

После добавления образа он появится в списке **Доступные образы дисков**. Выберите из этого списка один из образов и нажмите кнопку **Выбрать**. На этом подготовка сервера к сетевой установке рабочих станций завершена.

Предупреждение

Не забудьте отключить сетевую установку по окончании процесса установки ОС на рабочих станциях. Это можно сделать выбрав в списке **Доступные образы дисков** пункт **Нет образа** и подтвердив действие нажатием кнопки **Выбрать**.

За дополнительной информацией по настройке обращайтесь к встроенной справке соответствующих модулей Центра управления системой.

5.3.2 Подготовка рабочих станций

Для сетевой установки ОС Альт Сервер 8.0 следует обеспечить возможность загрузки по сети рабочих станций, на которых будет производиться установка ОС.

Большинство современных материнских плат имеют возможность загрузки по сети, однако она по умолчанию может быть отключена в BIOS. Различные производители материнских плат дают разные названия данной возможности, например: "Boot Option ROM" или "Boot From Onboard LAN".

Примечание

Некоторые материнские платы позволяют выбрать источник загрузки во время включения компьютера. Эта возможность может называться, например "Select boot device" или "Boot menu".

Последовательность установки при установке с DVD-диска и при сетевой установке не отличаются друг от друга.

5.4 Сервер электронной почты (SMTP, POP3/IMAP)

5.4.1 Сервер электронной почты

После установки сервера и первоначальной настройки вы уже имеете преднастроенный почтовый сервер, обслуживающий почтовый домен, указанный при первоначальной настройке в поле **домен**.

ОС Альт Сервер 8.0 может служить как почтовым сервером, обслуживающим определённый домен, так и посредником (шлюзом) для пересылки почты. Почтовый сервер отвечает как за отправку писем (SMTP-сервер) исходящих от почтовых клиентов рабочих станций, так и за предоставление им входящей почты (Сервер POP3/IMAP).

Для настройки параметров работы сервера предусмотрен модуль ЦУС **Почтовый сервер** (раздел **Серверы**).

5.4.2 Сервер SMTP

Сервер SMTP отвечает за отправку сообщений и может работать в двух режимах:

Посредник

В этом режиме исходящая почта пересылается для дальнейшей отправки на указанный сервер.

Сервер

В этом режиме сервер доставляет почту самостоятельно.

5.4.3 Сервер POP3/IMAP

Сервер POP3/IMAP используется для доступа пользователей к электронной почте на сервере.

Для доступа к службам POP3 и IMAP пользователь должен включить в своём почтовом клиенте аутентификацию и указать своё имя и пароль. Выбор конкретного используемого протокола для получения почты зависит от предпочтений пользователя.

POP

При проверке почты почтовым клиентом почта передаётся на клиентскую машину, где и сохраняется. Возможность просмотра принятой/отправленной почты при этом существует даже, если клиент не имеет соединения с сервером.

IMAP

Все сообщения хранятся на сервере. Почтовый клиент может просматривать их только при наличии соединения с сервером.

Помимо включения/отключения служб, модуль ЦУС **Почтовый сервер** позволяет произвести дополнительные настройки: фильтрацию спама, настройку параметров аутентификации и т.д.

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

5.5 Соединение удалённых офисов (OpenVPN-сервер)

ОС Альт Сервер 8.0 предоставляет возможность безопасного соединения удалённых офисов, используя технологию VPN (англ. Virtual Private Network — виртуальная частная сеть), которая позволяет организовать безопасные зашифрованные соединения через публичные сети (например, Интернет) между удалёнными офисами или локальной сетью и удалёнными пользователями. Таким образом, вы можете связать два офиса организации, что, например, делает работу с документами, расположенными в сети удалённого офиса, более удобной.

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

Для организации подобного рода соединения на стороне сервера предусмотрен модуль ЦУС **OpenVPN-сервер** (раздел **Серверы**). Особое внимание при планировании и настройке подключений следует обратить на используемые сети. Они *не должны* пересекаться.

Для создания соединения установите флажок **Включить службу OpenVPN**, выберите тип **Маршрутизируемое (TUN)**, проверьте открываемую по соединению сеть (обычно это локальная сеть в виде IP-адреса и маски подсети). Помимо этого нужно подписать ключ **openvpn** в модуле **Удостоверяющий Центр** (раздел **Система**). После этого нажмите на кнопку **Применить**. Доступ к настроенной приватной сети могут получить пользователи, подписавшие свои ключи и получившие сертификат в удостоверяющем центре на том же сервере.

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

Со стороны клиента соединение настраивается в модуле ЦУС **OpenVPN-соединения**.

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

5.6 Доступ к службам сервера из сети Интернет

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

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

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

5.6.1 Список блокируемых хостов

Модуль ЦУС **Список блокируемых хостов** предназначен для блокирования любого трафика с указанными узлами. Данный модуль позволяет блокировать любой сетевой трафик с указанными в списке узлами (входящий, исходящий и пересылаемый). Блокирование трафика с указанными в списке узлами начинается после установки флажка **Использовать чёрный список**. Для добавления блокируемого узла введите IP-адрес или имя узла в поле под списком и нажмите кнопку **Добавить**. Для удаления узла из списка выберите его и нажмите кнопку **Удалить**.

5.7 Статистика

5.7.1 Сетевой трафик

Все входящие и исходящие с сервера сетевые пакеты могут подсчитываться, и выводятся по запросу для анализа. Можно оценить итоговый объём полученных и переданных данных за всё время работы сервера или за определённый период времени.

Для включения этой возможности и просмотра статистики воспользуйтесь модулем ЦУС **Сетевой трафик** (раздел **Статистика**).

Для включения сбора данных установите флажок **Включить сбор данных**, затем нажмите кнопку **Применить**.

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

Трафик на указанном интерфейсе за заданный период показывается в виде:

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

5.7.2 Прокси-сервер

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

Для включения сбора статистики и просмотра отчётов воспользуйтесь модулем ЦУС **Прокси-сервер** (раздел **Статистика**).

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

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

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

5.8 Обслуживание сервера

Для безотказной работы всего домена очень важно следить за корректной работой его центрального звена — сервера под управлением ОС Альт Сервер 8.0. Регулярный мониторинг состояния сервера, своевременное резервное копирование, обновление установленного ПО являются важной частью комплекса работ по обслуживанию сервера.

5.8.1 Мониторинг состояния системы

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

Для просмотра журналов воспользуйтесь модулем ЦУС **Системные журналы** (находится в разделе **Система**). Интерфейс позволяет просмотреть различные типы журналов с возможностью перехода к более старым или более новым записям.

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

5.8.2 Системные службы

Для изменения состояния служб вы можете использовать модуль ЦУС **Системные службы** (находится в разделе **Система**). Интерфейс позволяет изменять текущее состояние службы и, если необходимо, применить опцию запуска службы при загрузке системы.

5.8.3 Резервное копирование

Резервное копирование является важной частью работ по поддержанию работоспособности сервера и всего домена. Так как сервер является критичной частью сети, производите регулярное резервное копирование. При возникновении нештатных ситуаций, например, выхода из строя оборудования, вы сможете восстановить работоспособное состояние сервера из резервной копии.

Ниже перечислены модули, с помощью которых можно настроить «Резервное копирование».

План резервного копирования и дополнительные параметры настраиваются в модуле ЦУС **Резервное копирование**. Этот же модуль может использоваться и для восстановления данных.

Bacula — кроссплатформенное клиент-серверное программное обеспечение, позволяющее управлять резервным копированием, восстановлением, и проверкой данных по сети для компьютеров и операционных систем различных типов.

Основным диспетчером резервного копирования является Bacula Director. Дополнительно его настраивать не нужно.

Для того чтобы начать резервное копирование самого сервера или рабочей станции:

- перейдите в раздел **Сервер резервного копирования** → **Клиенты**
- укажите имя узла (для сервера это будет localhost) и операционную систему. Нажмите кнопку **Создать**;
- укажите пароль для клиента и включаемые и исключаемые каталоги;
- нажмите на кнопку **Сохранить параметры**;
- нажмите ссылку "Конфигурационный файл клиента" и сохраните файл <имя узла>-fd.bin на локальном компьютере;
- скопируйте полученный файл на рабочую станцию или сервер. Под Linux этот файл нужно сохранить под именем /etc/bacula/bacula-fd.conf;
- запустите на компьютере, где создаётся резервная копия, службу bacula-fd (пакет bacula-client).

Примечание

Для клиента под управлением ОС Linux по умолчанию создаётся резервная копия всей файловой системы, кроме каталогов с временными и служебными файлами: /dev, /.fsck, /.journal, /media, /mnt, /opt, /proc, /srv, /sys, и /tmp.

В разделе **Сервер резервного копирования** → **Расписание** указывается время проведения инкрементного резервного копирования для каждого клиента. Удостоверьтесь, что в это время на клиенте служба bacula-fd запущена. В этом же разделе можно отключить резервное копирование для выбранных клиентов.

В разделе **Сервер резервного копирования** → **Архив**:

- посмотреть общую информацию о доступном месте на диске;
- посмотреть состояние и размер архива для каждого клиента;
- принудительно запустить создание резервной копии;
- удалить резервную копию клиента;
- восстановить файл или каталог на выбранную дату.

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

5.8.4 Обновление системы

После установки системы крайне важно следить за обновлениями ПО. Обновления для ОС Альт Сервер 8.0 могут содержать как исправления, связанные с безопасностью, так и новый функционал или просто улучшение и ускорение алгоритмов. В любом случае настоятельно рекомендуется регулярно обновлять систему для повышения надёжности работы сервера.

Для автоматизации процесса установки обновлений предусмотрен модуль ЦУС **Обновление системы** (раздел **Система**). Вы можете включить автоматическое обновление через Интернет с одного из предлагаемых серверов или задать собственные настройки.

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

5.8.5 Администратор системы

В модуле **Администратор системы** (раздел **Пользователи**) можно изменить пароль суперпользователя (root), заданный при начальной настройке системы.

Также можно добавить публичную часть ключа RSA или DSA для доступа к серверу по протоколу SSH.

5.8.6 Дата и время

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

5.8.7 Ограничение использования диска

Модуль **Использование диска** (раздел **Пользователи**) позволяет ограничить использование дискового пространства пользователями, заведёнными на сервере в модуле **Пользователи**.

Примечание

Для управления квотами файловая система должна быть подключена с параметрами `usrquota`, `grpquota`.

Модуль предоставляет следующие возможности:

- задание ограничений (квот) для пользователя при использовании определённого раздела диска. Ограничить можно как суммарное количество килобайт, занятых файлами пользователя, так и количество этих файлов;
- при задании ограничений различают жёсткие и мягкие ограничения.

Мягкое ограничение: нижняя граница ограничения, которая может быть временно превышена. Временное ограничение — одна неделя.

Жёсткое ограничение: использование диска, которое не может быть превышено ни при каких условиях.

Значение 0 при задании ограничений означает отсутствие ограничений.

5.8.8 Выключение и перезагрузка компьютера

Иногда, в целях обслуживания или по организационным причинам необходимо корректно выключить или перезагрузить сервер. Для этого воспользуйтесь модулем ЦУС **Выключение компьютера** в разделе **Система**. Установите переключатель на необходимое действие:

- Продолжить работу;
- Выключить компьютер сейчас;
- Перезагрузить компьютер сейчас.

Также можно выключать компьютер каждый день в заданное время. Установите флажок **Выключать компьютер каждый день в** и укажите время выключения в поле ввода слева от этого флажка.

5.9 Прочие возможности ЦУС

Возможности ОС Альт Сервер 8.0 не ограничиваются только теми, что были описаны выше. Вы всегда можете поискать другие модули, предоставляющие прочие возможности для настройки системы в web-интерфейсе.

Установленные пакеты, которые относятся к ЦУС, вы можете посмотреть командой `rpm -qa | grep alterator*`. Прочие пакеты для ЦУС вы можете поискать командой `apt-cache search alterator*`.

5.9.1 DNS-сервер

DNS (Domain Name System — система доменных имён) — распределённая база данных, способная по запросу, содержащему доменное имя хоста (компьютера или другого сетевого устройства), сообщить IP-адрес или, наоборот, по данному IP-адресу сообщить доменное имя устройства.

Этот модуль позволяет настроить **DNS-сервер**.

5.10 Samba 4 в роли контроллера домена Active Directory

Использование Samba 4 в роли контроллера домена Active Directory позволяет вводить Windows 7/8 в домен без манипуляций с реестром.

Поддерживаются следующие базовые возможности Active Directory:

- аутентификация рабочих станций Windows и Linux и служб;
- авторизация и предоставление ресурсов;
- групповые политики (GPO);
- перемещаемые профили (Roaming Profiles);
- поддержка инструментов Microsoft для управления серверами (Remote Server Administration Tools) с компьютеров под управлением Windows;
- поддержка протоколов SMB2 и SMB3 (в том числе с поддержкой шифрования);

- репликация с другими серверами (в том числе с Windows 2012).

Samba AD DC несовместима с OpenLDAP и MIT Kerberos, поэтому службы, использующие MIT Kerberos, несовместимы с ним.

Samba AD DC функционирует на уровне контроллера доменов Windows 2008 R2. Можно ввести его в домен Windows 2012 как клиента, но не как контроллер домена.

5.10.1 Установка

Для установки Samba AD DC выполняются следующие шаги:

1. Так как Samba в режиме контроллера домена (Domain Controller, DC) использует как свой LDAP, так и свой сервер Kerberos, несовместимый с MIT Kerberos, перед установкой необходимо остановить конфликтующие службы `krb5kdc` и `slapd`, а также `bind`:

```
# for service in smb nmb krb5kdc slapd bind; do chkconfig $service off; service $service stop; done
```

2. Установить пакет `task-samba-dc`, который установит все необходимое:

```
# apt-get install task-samba-dc
```

5.10.2 Миграция существующего сервера

Для миграции существующего сервера необходимо:

1. Скопировать данные для миграции в один каталог:

```
mkdir /var/lib/samba/dbdir
cp -pv /var/lib/samba/private/* /var/lib/samba/dbdir
cp -pv /var/lib/samba/{account_policy,gencache_notrans,group_mapping}.tdb /var/lib/samba/dbdir
```

При этом должно скопироваться пять файлов `.tdb`

2. Запустить `classicupgrade` (с правами администратора):

```
# samba-tool domain classicupgrade --dbdir=/var/lib/samba/dbdir --use-xattrs=yes --realm=edu.alt /etc/samba/smb.conf
```

5.10.3 Создание нового домена

5.10.3.1 Восстановление к начальному состоянию samba

Если домен уже создавался, необходимо очистить базу и конфигурацию Samba:

```
rm -f /etc/samba/smb.conf
rm -rf /var/lib/samba
mkdir -p /var/lib/samba/sysvol
```

Перед созданием домена нужно обязательно удалить `/etc/samba/smb.conf`:

```
rm -f /etc/samba/smb.conf
```

5.10.3.2 Выбор имени домена

Имя домена для разворачиваемого DC должно состоять минимум из двух компонентов, разделённых точкой.

При этом должно быть установлено правильное имя узла и домена для сервера:

- HOSTNAME=dc.edu.alt в /etc/sysconfig/network
- hostname dc.edu.alt
- domainname edu.alt

При указании домена, имеющего суффикс .local, на сервере и подключаемых компьютерах под управлением Linux потребуется отключить службу avahi-daemon.

5.10.3.3 Создание домена

Создание контроллера домена edu.alt:

```
samba-tool domain provision --realm=edu.alt --domain edu --adminpass='Pa$$word' --
dns-backend=SAMBA_INTERNAL --server-role=dc --use-rfc2307 --use-xattrs=yes
```

где

- realm – задает область Kerberos (LDAP), и DNS имя домена;
- domain – задает имя домена (имя рабочей группы);
- adminpass – пароль основного администратора домена;
- server-role – тип серверной роли.

5.10.3.4 Интерактивное создание домена

Для интерактивного развертывания запустите `samba-tool domain provision`, это запустит утилиту развертывания, которая будет задавать различные вопросы о требованиях к установке. В примере показано создание домена edu.alt:

```
# samba-tool domain provision
Realm [EDU.ALT]:
Domain [EDU]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
DNS forwarder IP address (write 'none' to disable forwarding) [127.0.0.1]:
Administrator password:
Retype password:
Looking up IPv4 addresses
More than one IPv4 address found. Using 192.168.1.1
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
```

```

Adding DomainDN: DC=edu,DC=alt
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=edu,DC=alt
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba 4 has been generated at
/var/lib/samba/private/krb5.conf
Once the above files are installed, your Samba4 server will be ready to use
Server Role:          active directory domain controller
Hostname:             c228
NetBIOS Domain:      EDU
DNS Domain:           edu.alt
DOMAIN SID:          S-1-5-21-80639820-2350372464-3293631772

```

При запросе ввода нажимайте Enter за исключением запроса пароля администратора («Administrator password:» и «Retype password:»).

Пароль администратора должен быть не менее 7 символов и содержать символы как минимум трёх групп из четырёх возможных: латинских букв в верхнем и нижнем регистрах, чисел и других небуквенно-цифровых символов. Пароль не полностью соответствующий требованиям это одна из причин завершения развертывания домена ошибкой.

Параметры `--use-rfc2307 --use-xattrs=yes` позволяют поддерживать расширенные атрибуты типа UID и GID в схеме LDAP и ACL на файловой системе Linux.

5.10.4 Запуск службы

Установите службу по умолчанию и запустите её:

```

# chkconfig samba on
# service samba start

```

5.10.5 Проверка работоспособности

Просмотр общей информации о домене:

```
# samba-tool domain info 127.0.0.1
Forest           : edu.alt
Domain           : edu.alt
Netbios domain   : EDU
DC name          : c228.edu.alt
DC netbios name  : C228
Server site      : Default-First-Site-Name
Client site      : Default-First-Site-Name
```

Просмотр предоставляемых служб:

```
# smbclient -L localhost -Uadministrator
Enter administrator's password:
Domain=[EDU] OS=[Unix] Server=[Samba 4.0.21]

  Sharename      Type            Comment
  -----      -
  netlogon       Disk
  sysvol         Disk
  IPC$           IPC             IPC Service (Samba 4.0.21)
Domain=[EDU] OS=[Unix] Server=[Samba 4.0.21]

  Sharename      Comment
  -----      -
  Workgroup      Master
  -----      -
  EDU.ALT        C228
  WORKGROUP      HOST-15
```

Общие ресурсы netlogon и sysvol создаваемые по умолчанию нужны для функционирования сервера AD и создаются в smb.conf в процессе развертывания/модернизации.

Проверка конфигурации DNS:

1. Убедитесь в наличии nameserver 127.0.0.1 в /etc/resolv.conf:

```
host edu.alt
```

2. Проверьте имена хостов:

```
# host -t SRV _kerberos._udp.edu.alt.
_kerberos._udp.edu.alt has SRV record 0 100 88 c228.edu.alt.
# host -t SRV _ldap._tcp.edu.alt.
_ldap._tcp.edu.alt has SRV record 0 100 389 c228.edu.alt.
# host -t A c228.edu.alt.
c228.edu.alt has address 192.168.1.1
```

Если имена не находятся, проверьте выключение службы named.

Проверка Kerberos (имя домена должно быть в верхнем регистре):

```
# kinit administrator@EDU.ALT
```

Password for administrator@EDU.ALT:

Warning: Your password will expire in 41 days on Вт 14 фев 2017 08:58:30

Просмотр полученного билета:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@EDU.ALT
Valid starting      Expires            Service principal
31.01.2017 10:23:54  31.01.2017 20:23:54  krbtgt/EDU.ALT@EDU.ALT
renew until 01.02.2017 10:23:45
```

5.10.6 Управление пользователями

Создать пользователя с паролем:

```
samba-tool user create <имя пользователя>
samba-tool user setexpiry <имя пользователя>
```

Например,

```
samba-tool user create ivanov --given-name='Иван Иванов' --mail-
address='ivanov@stand.alt'
```

Просмотреть доступных пользователей:

```
samba-tool user list
```

Не забудьте разблокировать пользователя:

```
samba-tool user setexpiry <имя пользователя> --noexpiry
```

Не допускайте одинаковые имена для пользователей и компьютера, так как это может привести к коллизиям (например, такого пользователя нельзя добавить в группу).

Если компьютер с таким именем заведён, то удалить его можно командой:

```
pdbedit -x -m <имя>
```

5.10.7 Заведение вторичного DC

Присоединение дополнительного Samba DC к существующему AD отличается от инициализации первого DC в лесу AD.

В примере используется узел: dc2.edu.alt (192.168.1.106).

1. На Primary Domain Controller (PDC) выключить службу bind и, если она была включена, перезапустить службу samba.
2. Заводим адрес IP для dc2 (указание аутентифицирующей информации (имени пользователя и пароля) обязательно!):

```
# samba-tool dns add 192.168.1.1 edu.alt DC2 A 192.168.1.106 -Uadministrator
```

3. Установите следующие параметры в файле конфигурации клиента Kerberos (на dc2.edu.alt файл /etc/krb5.conf):

```
[libdefaults]
default_realm = EDU.ALT
dns_lookup_realm = true
```

```
dns_lookup_kdc = true
```

В resolvconf обязательно должен быть добавлен PDC как nameserver.

- Для проверки настройки запрашиваем билет Kerberos для администратора домена (имя домена должно быть указано в верхнем регистре):

```
# kinit administrator@EDU.ALT
Password for administrator@EDU.ALT:
Warning: Your password will expire in 37 days on Пт 17 фев 2017 14:31:40
```

- Убеждаемся, что билет получен:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@EDU.ALT
Valid starting          Expires                Service principal
07.01.2017 18:51:01     08.01.2017 04:51:01  krbtgt/EDU.ALT@EDU.ALT
renew until 08.01.2017 18:50:51
```

- Вводим в домен edu.alt в качестве контроллера домена (DC):

```
# samba-tool domain join edu.alt DC -Uadministrator --realm=edu.alt
```

Если всё нормально, в конце будет выведена информация об присоединении к домену:

```
Joined domain EDU (SID S-1-5-21-80639820-2350372464-3293631772) as a DC
```

Для получения дополнительной информации можно воспользоваться командой:

```
samba-tool domain join --help
```

- Делаем службу samba запускаемой по умолчанию:

```
# chkconfig samba on
```

Если подключались к DC под управлением Windows, запустите службу samba:

```
# service samba start
```

5.10.8 Репликация

Без успешной двунаправленной репликации в течение 14 дней DC исключается из Active Directory. Указание аутентифицирующей информации (имени пользователя и пароля) обязательно!

- Реплицируем на вторичном DC (с первичного):

```
# samba-tool drs replicate dc2.edu.alt c228.edu.alt dc=edu,dc=alt -Uadministrator
```

Сначала указывается приемник, затем источник, после этого реплицируемая ветка в LDAP.

- Реплицируем на вторичном DC (на первичный):

```
# samba-tool drs replicate c228.edu.alt dc2.edu.alt dc=edu,dc=alt -Uadministrator
```

Сначала указывается приемник, затем источник, после этого реплицируемая ветка в LDAP.

Имя домена в именах серверов можно опустить (если они одинаковые).

- Для просмотра статуса репликации на PDC, запустите на Samba DC:

```
# samba-tool drs showrepl
```

Если репликация на Windows не работает, добавьте в Active Directory Sites and Services новое соединение Active Directory. Реплицируйте на DC, подождите минут 5 и попробуйте реплицировать с Samba на Windows.

5.11 Samba в режиме файлового сервера

5.11.1 Настройка smb.cfg

Пример настройки smb.cfg для работы Samba в режиме файлового сервера с двумя расширенными ресурсами и принтером (закомментированные параметры - действуют по-умолчанию):

```
workgroup = workgroup
server string = Samba Server Version %v
map to guest = Bad User
; idmap config * : backend = tdb
guest ok = yes
cups options = raw
security = user
; encrypt passwords = yes
; guest account = nobody
[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
;    guest ok = no
;    writable = No
    printable = yes
# A publicly accessible directory, but read only, except for people in
# the "staff" group
[public]
    comment = Public Stuff
    path = /home/samba
    public = yes
    writable = yes
;    printable = no
    write list = +staff
;    browseable = yes
[Free]
    path = /mnt/win/Free
    read only = no
;    browseable = yes
    guest ok = yes
```

5.11.2 Монтирование ресурса Samba

Создать файл `/etc/samba/smbacreds` (например, командой `mkdir /etc/samba/smbacreds`), с содержимым:

```
username=имя_пользователя
password=пароль
```

Для монтирования ресурса Samba в `/etc/fstab` необходимо прописать:

```
//server/public /mnt/server_public cifs users,credentials=/etc/samba/smbacreds 0 0
```

Для защиты информации, права на файл `/etc/samba/smbacreds`, надо установить так, чтобы файл был доступен только хозяину:

```
chmod 600 /etc/samba/smbacreds
```

и принадлежать `root`:

```
chown root: /etc/samba/smbacreds
```

5.12 SOGo

SOGo – сервер групповой работы, аналогичный Microsoft Exchange, с веб-интерфейсом и доступом по MAPI для Microsoft Outlook.

SOGo обеспечивает веб-интерфейс на основе AJAX и поддерживает различные нативные клиенты с помощью стандартных протоколов. Возможности SOGo:

- общие почтовые папки, календари и адресные книги;
- веб-интерфейс, аналогичный Outlook Web Access;
- поддержка протоколов CalDAV, CardDAV, GroupDAV, Microsoft ActiveSync, IMAP и SMTP;
- доступ по MAPI для Microsoft Outlook, не требующий внешних модулей;
- делегирование, уведомления, резервирование, поддержка категорий и почтовых фильтров;
- поддержка нескольких почтовых ящиков в веб-интерфейсе;
- single sign-on с помощью CAS, WebAuth или Kerberos.

5.12.1 Установка

Установите стабильную версию SOGo и соответствующий драйвер к PostgreSQL:

```
# apt-get install task-sogo3
# apt-get install sope-gd11-postgresql
```

5.12.2 Подготовка среды

Настройка PostgreSQL:

1. Подготовить к запуску и настроить службы PostgreSQL.
2. Создать пользователя `sogo` и базу данных `sogo` (под правами `root`):

```
su - postgres -s /bin/sh -c 'createuser --no-superuser --no-createdb --no-createrole
sogo'
```

```
su - postgres -s /bin/sh -c 'createdb -O sogo sogo'
service postgresql restart
```

Настройка Samba DC:

1. Пользователи расположены в домене Active Directory, расположенном на контроллере с Samba DC.
2. Предварительно создать домен SambaDC.
3. Создать в домене пользователя sogo с паролем Pa\$\$word (при запросе дважды ввести пароль):

```
samba-tool user add sogo
```

Настройка SOGo (настраивается на домен test.alt):

1. Заполнить файл конфигурации /etc/sogo/sogo.conf:

```
{
  SOGoProfileURL = "postgresql://sogo@/sogo/sogo_user_profile";
  OCSFolderInfoURL = "postgresql://sogo@/sogo/sogo_folder_info";
  OCSSessionsFolderURL = "postgresql://sogo@/sogo/sogo_sessions_folder";
  OCSEMailAlarmsFolderURL = "postgresql://sogo@/sogo/sogo_alarms_folder";
  SOGoEnableEMailAlarms = YES;
  SOGoDraftsFolderName = Drafts;
  SOGoSentFolderName = Sent;
  SOGoTrashFolderName = Trash;
  SOGoIMAPServer = "imaps://localhost:993";
  SOGoMailingMechanism = sendmail;
  SOGoForceExternalLoginWithEmail = NO;
  NGImap4ConnectionStringSeparator = "/";
  SOGoUserSources = (
    {
      id = sambaLogin;
      displayName = "SambaLogin";
      canAuthenticate = YES;
      type = ldap;
      CNFieldName = cn;
      IDFieldName = cn;
      UIDFieldName = sAMAccountName;
      hostname = "ldaps://127.0.0.1";
      baseDN = "CN=Users,DC=test,DC=alt";
      bindDN = "CN=sogo,CN=Users,DC=test,DC=alt";
      bindPassword = "Pa$$word";
      bindFields = (sAMAccountName);
    },
    {
      id = sambaShared;
      displayName = "Shared Addressbook";
      canAuthenticate = NO;
      isAddressBook = YES;
      type = ldap;
      CNFieldName = cn;
      IDFieldName = mail;
      UIDFieldName = mail;
      hostname = "ldaps://127.0.0.1";
      baseDN = "CN=Users,DC=test,DC=alt";
      bindDN = "CN=sogo,CN=Users,DC=test,DC=alt";
      bindPassword = "Pa$$word";
      filter = "(NOT isCriticalSystemObject='TRUE') AND (mail='*') AND (NOT objectClass=contact)";
    }
  );
}
```

```

    },
    {
        id = sambaContacts;
        displayName = "Shared Contacts";
        canAuthenticate = NO;
        isAddressBook = YES;
        type = ldap;
        CNFieldName = cn;
        IDFieldName = mail;
        UIDFieldName = mail;
        hostname = "ldaps://127.0.0.1";
        baseDN = "CN=Users,DC=test,DC=alt";
        bindDN = "CN=sogo,CN=Users,DC=test,DC=alt";
        bindPassword = "Pa$$word";
        filter = "(((objectClass=person) AND (objectClass=contact) AND ((uid-
Number>=2000) OR (mail='*')))
                AND (NOT isCriticalSystemObject='TRUE') AND (NOT showInAd-
vancedViewOnly='TRUE') AND (NOT uid=Guest))
                OR (((objectClass=group) AND (gidNumber>=2000)) AND (NOT isCrite-
ticalSystemObject='TRUE') AND (NOT showInAdvancedViewOnly='TRUE'))");
        mapping = {
            displayname = ("cn");
        };
    }
);
SOGoSieveScriptsEnabled = NO;
SOGOlanguage = Russian;
SOGOtimeZone = Europe/Moscow;
SOGOfirstDayOfWeek = 1;
}

```

2. Делаем необходимые службы по умолчанию и перезапускаем их:

```
for s in samba postgresql memcached sogo httpd2;do chkconfig $s on;service $s re-
start;done
```

3. Возможные ошибки будут записаны в файл журнала /var/log/sogo/sogo.log

5.12.3 Включение веб-интерфейса

Для включения веб-интерфейса выполнить:

```

a2enmod proxy
a2enmod proxy_http
a2enmod authn_core
a2enmod authn_file
a2enmod auth_basic
a2enmod authz_user
a2enmod env
a2enmod dav
a2enmod headers
a2enmod rewrite
a2ensite SOGo
service httpd2 restart
service sogo restart

```

Войти по адресу:

`http://адрес_сервера/SOGo/`

5.12.4 Настройка почты

Аутентификация пользователей SMTP/IMAP в Postfix/Dovecot осуществляется в AD.

5.13 FreeIPA

FreeIPA – это комплексное решение по управлению безопасностью Linux-систем, 389 Directory Server, MIT Kerberos, NTP, DNS, Dogtag, состоит из веб-интерфейса и интерфейса командной строки.

FreeIPA является интегрированной системой проверки подлинности и авторизации в сетевой среде Linux, FreeIPA сервер обеспечивает централизованную проверку подлинности, авторизацию и контроль за аккаунтами пользователей сохраняя сведения о пользователе, группах, узлах и других объектах необходимых для обеспечения сетевой безопасности.

5.13.1 Установка сервера FreeIPA

Сервер FreeIPA будем устанавливать со встроенным DNS сервером и доменом EXAMPLE.TEST в локальной сети 192.168.135.0/24.

Для начала необходимо отключить `ahttpd`, работающий на порту 8080, во избежание конфликтов с разворачиваемым `tomcat`, а также отключить HTTPS в Apache2:

```
# service ahttpd stop
# a2dissite 000-default_https
# service httpd2 condreload
```

Установим необходимые пакеты:

```
# apt-get install freeipa-server freeipa-server-dns
```

Зададим имя сервера:

```
# hostnamectl set-hostname ipa.example.test
```

Запускаем скрипт настройки сервера (пароли должны быть не менее 8 символов). В пакетном режиме:

```
# ipa-server-install -U --hostname=$(hostname) -r EXAMPLE.TEST -n example.test -p
12345678 -a 12345678 --setup-dns --no-forwarders --no-reverse
```

или интерактивно:

```
# ipa-server-install
```

Обратите внимание на ответы на вопросы, не совпадающие с предложенными:

```
Do you want to configure integrated DNS (BIND)? [no]: yes
Do you want to configure DNS forwarders? [yes]: no
Do you want to search for missing reverse zones? [yes]: no
```

Остальные вопросы выбираем по умолчанию (можно просто нажать Enter). Так же при установке нужно ввести пароль администратора системы и пароль администратора каталогов.

После окончания установки настроим сервер времени, чтобы компьютеры в локальной сети могли к нему подключаться. Для этого добавим в файл `/etc/ntp.conf` следующую строчку:

```
restrict 192.168.135.0 mask 255.255.255.0 nomodify
```

Перезапустим службу:

```
# systemctl restart ntpd
```

Для возможности управлять FreeIPA сервером из командной строки необходимо получить билет Kerberos:

```
# kinit admin
```

Добавим в DNS запись о сервере времени:

```
# ipa dnsrecord-add example.test _ntp._udp --srv-priority=0 --srv-weight=100 --srv-port=123 --srv-target=ipa.example.test.
```

Веб-интерфейс доступен по адресу:

```
https://ipa.example.test/ipa/ui/
```

5.13.2 Установка FreeIPA клиента и подключение к серверу

Установим необходимые пакеты:

```
# apt-get install freeipa-client libsss_sudo
```

Зададим имя компьютера:

```
# hostnamectl set-hostname comp01.example.test
```

Добавим DNS сервер, для этого создадим файл `/etc/net/iface/ens19/resolv.conf` со следующим содержимым:

```
nameserver 192.168.135.1
```

где 192.168.135.1 - IP-адрес FreeIPA сервера.

Укажем службе `resolvconf` использовать DNS FreeIPA и наш домен для поиска. Для этого в файл `/etc/resolvconf.conf` добавим/отредактируем следующие параметры:

```
interface_order='lo lo[0-9]* lo.* ens19'
```

```
search_domains=example.test
```

где `ens19` – интерфейс, на котором доступен FreeIPA сервер, `example.test` – наш домен.

Обновим DNS адреса:

```
# resolvconf -u
```

После этого в файле `/etc/resolv.conf` должны появиться строки:

```
search example.test
```

```
nameserver 192.168.135.1
```

Запускаем скрипт настройки клиента: в пакетном режиме:

```
# ipa-client-install -U -p admin -w 12345678
```

или интерактивно:

```
# ipa-client-install
```

Если все настроено верно, скрипт должен выдать такое сообщение:

```
'''Discovery was successful!'''
```

```
Client hostname: comp02.example.test
```

```
Realm: EXAMPLE.TEST
```

```
DNS Domain: example.test
IPA Server: ipa.example.test
BaseDN: dc=example,dc=test
Continue to configure the system with these values? [no]:
```

Отвечаем yes, вводим имя пользователя, имеющего право вводить машины в домен, и его пароль.

В случае возникновения ошибки, необходимо перед повторной установкой запустить процедуру удаления:

```
# ipa-client-install -U --uninstall
```

Для работы sudo-политик для доменных пользователей на клиентской машине необходимо разрешить доступ к sudo:

```
# control sudo public
```

При первом входе пользователя будет запрошен текущий (установленный администратором) пароль и затем у пользователя запрашивается новый пароль и его подтверждение.

5.13.3 Настройка доверительных отношений с AD

FreeIPA использует Samba для интеграции в Active Directory. Для работы Samba необходим работающий стек IPv6.

Начальные данные:

- IP адрес IPA сервера: 192.168.135.130
- Имя IPA сервера: dcf
- Имя IPA домена: domf.testf
- NetBIOS имя IPA домена: DOMF
- IP адрес AD DC: 192.168.135.150
- Имя AD DC: dcc
- Имя AD домена: domc.testc
- NetBIOS имя AD домена: DOMC

Установим необходимые пакеты:

```
# apt-get install freeipa-server-trust-ad python-module-sss-murmur samba-winbind
```

Выполним предварительную настройку IPA сервера для работы с доверием:

```
# ipa-adtrust-install --netbios-name=DOMF
```

Дата и время на серверах должны совпадать.

IPA сервер в своей работе использует следующие порты:

TCP ports: 80, 88, 443, 389, 636, 88, 464, 53, 135, 138, 139, 445, 1024-1300

UDP ports: 88, 464, 53, 123, 138, 139, 389, 445

Они должны быть открыты и доступны.

Настроим Samba:

```
# net conf setparm global 'dedicated keytab file' /etc/samba/samba.keytab
# systemctl restart ipa
```

Настроим DNS на обоих серверах, чтобы они знали друг о друге. На AD сервере создадим сервер условной пересылки для зоны IPA домена:

```
C:\> dnscmd 127.0.0.1 /ZoneAdd domf.testf /Forwarder 192.168.135.130
```

На IPA сервере так же добавим зону AD домена:

```
ipa dnsforwardzone-add domc.testc --forwarder=192.168.135.150 --forward-policy=only
```

Проверка конфигурации DNS. На AD сервере:

```
C:\> nslookup
> set type=srv
> _ldap._tcp.domc.testc
> _ldap._tcp.domf.testf
> quit
```

На IPA сервере:

```
# dig SRV _ldap._tcp.domf.testf
# dig SRV _ldap._tcp.domc.testc
```

Добавление двунаправленных доверительных отношений леса (Forest Trust) с AD (необходимо ввести пароль Administrator AD):

```
# kinit admin
# ipa trust-add --type=ad domc.testc --admin Administrator --password --two-way=true
```

Далее необходимо запросить сервер AD о его доверенных доменах:

```
# ipa trust-fetch-domains domc.testc
```

При этом IPA создаст нужные id-диапазоны для доверенных доменов.

Для того чтобы увидеть список всех доверенных доменов из леса используйте следующую команду:

```
# ipa trustdomain-find domc.testc
```

5.14 Резервное копирование данных

5.14.1 Bacula

Bacula – это мощная система создания и управления резервными копиями данных, а также восстановления данных, если потребуется. Она имеет клиент-серверную структуру и легко масштабируется, позволяя делать резервные копии с десятков и сотен компьютеров по сети. Функционально Bacula состоит из компонентов (служб), каждая из которых реализует определенные функции. Взаимодействие служб показано на рис. 11.

Структура:

- Bacula Director – процесс управляющий системой в целом (управление, планирование, восстановление резервных копий).

- Storage Director – запускается на сервере, отвечающем за «физическое» хранение данных.
- File Director – сервис, запускаемый на каждом из клиентов.
- Vconsole – консоль управления.

Копирование, восстановление, верификация и административные функции оформляются в виде задания (Job). В задании задается набор файлов (FileSet), который нужно копировать, компьютер (Client), с которого надо копировать файлы, время копирования (Schedule), пул (Pool), куда копировать и дополнительные директивы.

Взаимодействие служб Bacula

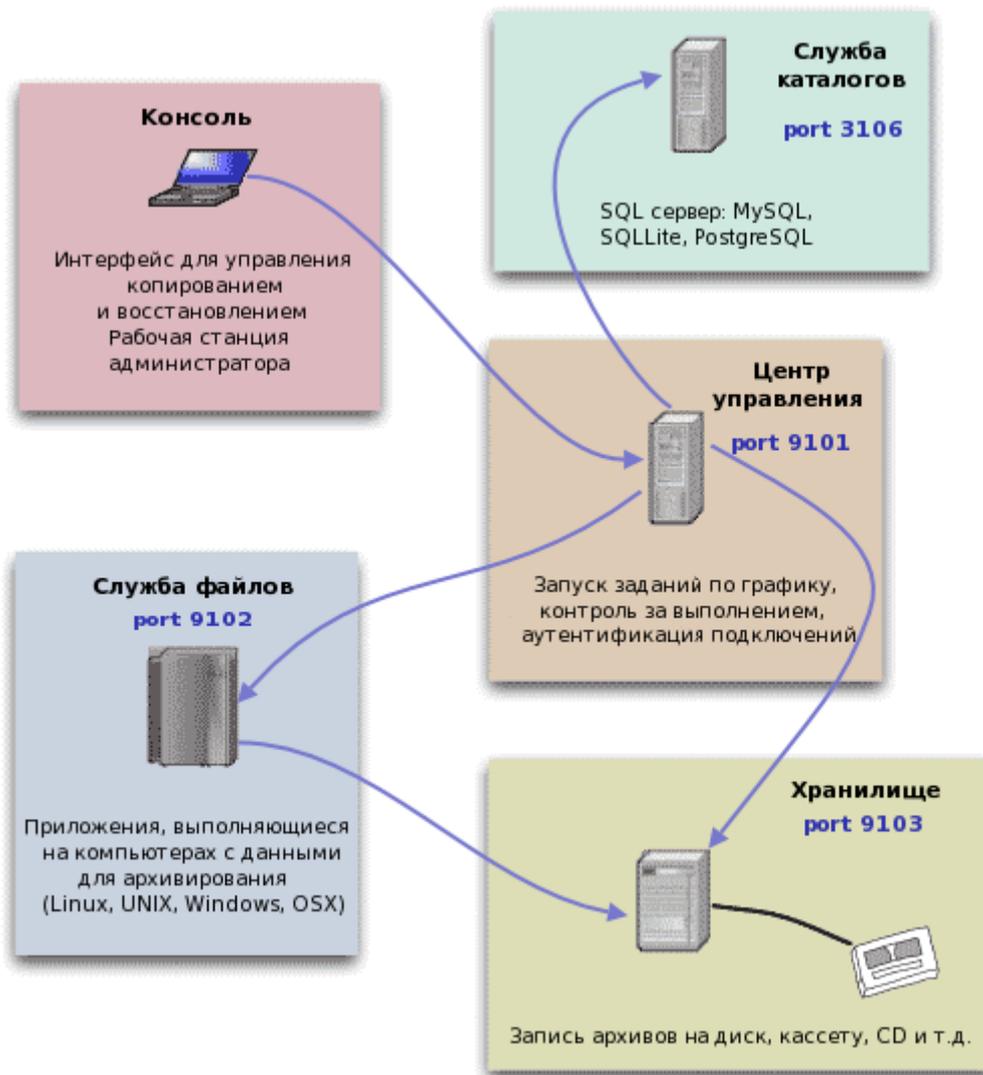


Рис.11

Задания на копирование данных определяются в конфигурационном файле Директора (Director) и там же определяется график автоматического запуска этих заданий. Директор выполняется постоянно как демон в фоновом режиме и запускает задания на копирование в соответ-

ствии с графиком. Администратор (пользователь) может также вручную запустить эти задания в любое время, используя Службу Консоль.

Файлы настройки Bacula форматированы на основе ресурсов, включающих директивы, обрамленные фигурными скобками "{}". Каждый компонент Bacula имеет индивидуальный файл в каталоге /etc/bacula.

Различные компоненты Bacula должны авторизовывать себя друг для друга. Это решается использованием директивы password. Например, пароль в ресурсе Storage файла /etc/bacula/bacula-dir.conf должен соответствовать паролю ресурса Director файла /etc/bacula/bacula-sd.conf.

В дистрибутиве установленная из пакетов Bacula уже настроена для резервного копирования конфигурации ОС.

Пример 1

Задание: наладить систему резервного копирования и восстановления данных, таким образом, что бы: на server1 в понедельник делался «полный» backup, со вторника и до воскресенья включительно дифференциальные backup.

Данные должны храниться не менее 3х недель.

Начнем с настройки Bacula Director (файл конфигурации: /etc/bacula/bacula-dir.conf). Расположение: сервер с запущенным bacula-director:

```
Director {
Name = backup-dir
Dirport = 9101
QueryFile = "/user/share/bacula/scripts/query.sql" #набор sql запросов
WorkingDirectory = "/var/lib/bacula"
PidDirectory = "/var/run/bacula"
Password = "some_password"
Messages = Daemon
DirAddress = 10.10.0.1
}
```

Отправка отчетов о проделанной работе на почту администратору:

```
Messages {
Name = Daemon
mailcommand = "/usr/lib/bacula/bsmtp -h localhost -f \"\"(Bacula\) \<%r\>\" -s
\"Bacula daemon message\" %r"
mail = admin@domain.com = all, !skipped
console = all, !skipped, !saved
append = "/var/lib/bacula/log" = all, !skipped
}
```

Определяем сервера «хранилища»:

```
Storage {
Name = stor_server1
```

```

Address = 10.10.0.2
SDPort = 9103
Password = "storage_pass"
Device = FileStorage
Media Type = File
}

```

Создаем расписания, согласно которому будем выполнять задания копирования или восстановления.

Согласно расписанию в понедельник будет создан «полная» резервная копия, в остальные дни будут создаваться дифференциальные резервные копии. Так, например, что бы восстановить данные за пятницу, необходимо развернуть задание резервного копирования, выполненное в понедельник, после чего «сверху накатить» пятничный дифференциальный backup.

```

Schedule {
Name = "WeeklyDiff"
Run = Level=Full on mon at 05:01
Run = Level=Differential on tue-sun at 02:02
}

```

Расписание, для создания «полных» backup:

```

Schedule {
Name = "WeeklyFull"
Run = Level=Full on mon-sun at 03:03
}

```

Создаем задание для резервного копирования сервера server1:

```

Job {
Name = "server1" #Имя задания
Type = Backup #Тип работы (создание backup)
Level = Differential #Уровень backup
Client=server1-fd #Клиент, на котором будет производиться backup
FileSet="server1" #Как и какие файлы будут сохраняться
Storage = stor_server1 #Куда будем копироваться backup
Pool = mainpool #Определяем с каким pool будем работать
Messages = Standard #Отчет о backup
Schedule = "WeeklyDiff" #Расписание backup
}

```

Указываем, что именно и как будем сохранять с сервера server1:

```

FileSet {
Name = "server1"
Include {
Options {
signature = MD5 #Для сверки используем MD5
Compression=GZIP #Используем GZIP компрессию
}
}
}

```

```

}
File = /etc #Что именно сохранять
File = /home/
File = /var/www

}

Exclude { #Что не сохранять, например логи
File = /home/logs
File = /var/www/logs
}
}

```

Описание параметров клиента, для server1:

```

Client {
Name = server1-fd
Address = 10.10.0.3
FDPort = 9102
Catalog = MyCatalog
Password = "fd_password"
File Retention = 28 days #Срок хранения метаданных
#о сохраненных файлах для данного клиента
Job Retention = 28 days # Срок хранения метаданных
#о заданиях для данного клиента
AutoPrune = yes #Может ли bacula очищать метаданные
}

```

В данной секции определяем параметры ротации.

Исходя из конфигурации: мы используем 2 тома, в томе храниться не более 7 заданий (недельный backup), время хранения тома 3 недели (21 день).

Таким образом, каждое задание хранится 21 день, на 22 день том с данными заданиями очищается и можно его использовать заново:

```

Pool {
Name = mainpool
Pool Type = Backup
Recycle = yes # Может ли bacula удалять задания из томов
AutoPrune = yes # Может ли bacula очищать тома
Volume Retention = 21 days # Время хранения тома
Maximum Volume Jobs = 7 # Сколько заданий хранить в каждом из томов
Maximum Volumes = 2 # максимальное количество томов
}

```

Описание задания, для восстановления данных:

```

Job {

```

```

Name = "server1-resotre"
Type = Restore
Client=server1
FileSet="server1"
Storage = stor_server1
Pool = mainpool
Messages = Standard
Where = /var/lib/bacula-restores
}

```

На сервере «хранилище» для данного задания создается 2 тома, в каждом томе хранится 7 заданий.

По заполнении всех томов, происходит очистка самого старого тома.

Описание настроек для Bacula Storage Director (файл: /etc/bacula/bacula-sd.conf). Расположение: сервер с запущенным bacula-sd (storage director):

```

Storage {
Name = stor_server1
SDPort = 9103
WorkingDirectory = "/var/lib/bacula"
Pid Directory = "/var/run/bacula"
SDAddress = 10.10.0.2
}
Director {
Name = backup-dir
Password = "storage_pass"
}

Device {
Name = FileStorage
Media Type = File
Archive Device = /var/bacula
LabelMedia = yes;
Random Access = Yes;
AutomaticMount = yes;
RemovableMedia = no;
AlwaysOpen = no;
}

Messages {
Name = Standard
director = backup-dir = all
}

```

Описание настроек для file-director на сервере server1 (файл: /etc/bacula/bacula-fd.conf). Расположение: сервер с запущенным bacula-fd (server1):

```
Director {
Name = backup-dir
Password = "server1-fd"
}

FileDaemon {
Name = server1-fd
FDport = 9102
WorkingDirectory = /var/lib/bacula
Pid Directory = /var/run/bacula
FDAddress = 10.10.0.3
}

Messages {
Name = Standard
director = server1-fd = all, !skipped, !restored
}
```

Пример 2

Задача – помимо существующего хранилища добавить новое и организовать резервное копирование на новое место. Для такой задачи необходимо выполнить 4 шага.

1. В файл конфигурации демона-хранителя bacula-sd нужно добавить секцию «Device», чтобы демон узнал о новом месте хранения.
2. В файл конфигурации демона-управляющего bacula-dir (bacula-dir.conf) нужно добавить секцию «Storage», которая опишет новое «хранилище» (организованное на новом «устройстве» демона хранения).
3. В файл bacula-dir.conf нужно добавить секцию «Pool», которая объявит новый набор томов (размещаемый на новом «хранилище»).
4. В файл bacula-dir.conf нужно добавить секцию «Job», которая опишет новое задание резервного копирования (использующее новый набор томов).

Помимо создания резервных копий нужно настроить возможность восстановления файлов. Для этого необходимо создать задание для восстановления – новую секцию «Job» с типом задания «Restore».

В случае если необходимо задать новое расписание резервного копирования, нужно будет добавить секцию «Schedule» в файл bacula-dir.conf и тогда новое расписание можно использовать в существующем или новом задании. Можно также описать несколько наборов томов, например,

для длительного хранения полных копий и кратковременного – инкрементальных архивов, после чего с этими наборами томов создать соответствующие задания.

Если же изменяются параметры набора томов (изменилось содержание какой-нибудь уже существовавшей секции «Pool»), необходимо обязательно выполнить в консоли bacula команду update. По этой команде демон bacula-dir обновит интервалы (длительность хранения) ранее созданных томов изменённого набора.

Новое место хранения (Device) в bacula-sd.conf

В конфигурации демона хранения bacula-sd добавим новое место («устройство») для хранения файлов – секцию «Device {}»:

1. Создаём файл /etc/bacula/device/backupstorage.conf – копируем имеющийся /etc/bacula/device/file.conf и меняем в нём путь для размещения файлов, имя и тип носителя. «Name» и «media type» должны быть уникальны среди описаний «Device» – они будут использоваться для выбора места хранения в задании резервного копирования:

```
# Definition of file storage device
# iSCSI volume "iqn.2012-07.com.lenovoemc:storage.ix4-300d.ix4-300d-vol13g"
# mapped to /dev/disk/by-label/backupstorage
# and mounted to /srv/backup/backupstorage
#
Device {
    Name = BackupStorage
    Media Type = File-NAS
    Archive Device = "/srv/backup/backupstorage"
    LabelMedia = yes;
    Random Access = Yes;
    AutomaticMount = yes;
    RemovableMedia = no;
    AlwaysOpen = no;
}
```

2. Добавляем этот файл в конфигурацию /etc/bacula/bacula-sd.conf рядом с другими:

```
@/etc/bacula/device/filestorage.conf
@/etc/bacula/device/tapedrives.conf
@/etc/bacula/device/backupstorage.conf
```

3. Даём демону bacula-sd команду перечитать конфигурацию:

```
service bacula-sd reload
```

или

```
/etc/init.d/bacula-sd reload
```

Новое хранилище в bacula-dir.conf

Добавим информацию о новом месте хранения в конфигурацию демона bacula-dir – вставим новую секцию «Storage {}» в файл «bacula-dir.conf»:

1. Создаём файл `/etc/bacula/storage/backupstorage.conf` – можно скопировать имеющийся `/etc/bacula/storage/file.conf` и заменить параметры «Name», «Device» и «Media Type»:

```
Storage {
    Name = NASBackupStorage
    Address = "10.0.0.2"
    SDPort = 9103
@/etc/bacula/bacula-sd-password.conf
    Device = BackupStorage
    Media Type = File-NAS
}
```

2. Вставляем `/etc/bacula/storage/backupstorage.conf` в `/etc/bacula/bacula-sd.conf` рядом с другими:

```
@/etc/bacula/client/client1.conf

@/etc/bacula/storage/file.conf
@/etc/bacula/storage/example.conf
@/etc/bacula/storage/backupstorage.conf
```

```
@/etc/bacula/messages/standart.conf
@/etc/bacula/messages/daemon.conf
```

```
@/etc/bacula/pool/default.conf
```

Добавляем пул томов в `bacula-dir.conf`

Чтобы `bacula` отдельно учитывала файлы, расположенные в NAS, создаём пул для них. Этот пул явно не привязан к хранилищу, он будет использован в задании.

Одно и то же хранилище можно привязать к множеству пулов, например, можно сделать разные пулы для еженедельных, ежемесячных и ежедневных резервных копий, в каждом разное время хранения томов.

1. Создаём файл `/etc/bacula/pool/nas.conf` с описанием секции `Pool`. Нужно изменить формат метки (чтобы отличать файлы с хранимыми данными) и имя. Также очень желательно ограничить размер тома, иначе `bacula` будет записывать всё в один том и в некий момент диск переполнится.

```
# Default pool definition for NAS Storage
Pool {
    Name = NAS
    Pool Type = Backup
    Recycle = yes
    AutoPrune = yes
    Volume Retention = 365 days
    LabelFormat = "bn"
    # Use not more Maximum Volume Bytes on disk
```

```
Maximum Volume Bytes = 4G
}
```

2. Вставляем /etc/bacula/pool/nas.conf в /etc/bacula/bacula-sd.conf рядом с другими:

```
@/etc/bacula/pool/default.conf
@/etc/bacula/pool/scratch.conf
@/etc/bacula/pool/nas.conf
```

Добавляем задания в bacula-dir.conf

Теперь добавим задание, используя существующее расписание (Schedule) «WeeklyCycle» (файл /etc/bacula/schedule/weeklycycle.conf). В существующей конфигурации есть задания «DefaultSet» (файлы сервера) с приоритетом 10 и «BackupCatalog» (база bacula) с приоритетом 11. Те же приоритеты поставим и для своих заданий.

1. Чтобы много раз не писать одинаковые параметры резервного копирования на подмонтированный том NAS, создаём файл с секцией «JobDefs» /etc/bacula/job/defaultnasjob.conf:

```
JobDefs {
  Name = "NASDefaultJob" #Имя задания
  Type = Backup #Тип работы (создание backup)
  Level = Incremental #Уровень backup
  Client = fd #Клиент на котором будет производиться backup
  FileSet = "Full Set"
  Storage = NASBackupStorage
  Messages = Standard
  Pool = NAS
  Priority = 10
}
```

2. Задание для сохранения файлов сервера на том NAS, его очередь — «10». Создаём файл задания /etc/bacula/job/baculadup.conf

```
#
# Define the duplicate of main backup job. This job stores backup into NAS.
#
Job {
  Name = "BackupFullSetDup"
  JobDefs = "NASDefaultJob"
  Schedule = "WeeklyCycle"
  Write Bootstrap = "/srv/backup/backupstorage/bacula/DUPClient1.bsr"
}
```

3. Задание для сохранения базы bacula на том NAS, которое нужно выполнять в последнюю очередь, номер очереди «11». Создаём файл задания /etc/bacula/job/backupcatalognas.conf:

```
Job {
  Name = "BackupCatalogNAS"
  JobDefs = "NASDefaultJob"
```

```

Level = Full
FileSet="Catalog"
Schedule = "WeeklyCycleAfterBackup"
RunBeforeJob = "/usr/share/bacula/scripts/make_catalog_backup"
RunAfterJob = "/usr/share/bacula/scripts/delete_catalog_backup"
Write Bootstrap = "/srv/backup/backupstorage/bacula/BackupCatalogNAS.bsr"
Priority = 11                # run after main backup
}

```

4. Вставляем новые файлы заданий в /etc/bacula/bacula-sd.conf рядом с другими:

```

@/etc/bacula/pool/default.conf
@/etc/bacula/pool/scratch.conf
@/etc/bacula/pool/nas.conf

@/etc/bacula/job/defaultjob.conf
@/etc/bacula/job/backupcatalog.conf
@/etc/bacula/job/bacula.conf

@/etc/bacula/job/defaultnasjob.conf
@/etc/bacula/job/baculadup.conf
@/etc/bacula/job/backupcatalognas.conf

```

Даём команду демону `bacula-dir` перечитать конфигурацию. Наиболее надёжный способ для этого – команда `reload` в консоли `bacula-dir`: если изменённая конфигурация содержит ошибку, демон покажет место ошибки в консоли и продолжит работу в прежней конфигурации.

Задание для восстановления файлов

Чтобы можно было быстро восстановить утерянный (удалённый или изменённый) файл, подготовим задание типа «Restore». В нём используются те же Pool, Storage, Fileset и Client, что и в задании типа «Backup», добавлен путь для восстановления («Where»), отличается тип, нет расписания и уровня.

1. Добавим файл /etc/bacula/job/restorefromnas.conf:

```

Job {
  Name = "RestoreFromNAS"
  Type = Restore
  Client=fd
  FileSet="Full Set"
  Storage = NASBackupStorage
  Pool = NAS
  Messages = Standard
  Where = /tmp/bacula-restores
}

```

2. Вставляем новый файл задания в /etc/bacula/bacula-sd.conf рядом с другими:

...

```
@/etc/bacula/job/restore.conf
```

```
@/etc/bacula/job/restorefromnas.conf
```

5.14.2 Работа в Bacula с использованием командной строки (bconsole)

Служба Console предоставляет администратору (пользователю) интерфейс (командная строка) для взаимодействия с Bacula Director. Именно через Console, в частности командную строку, bconsole можно вручную запускать задание на копирование или восстановление. Можно смотреть статус системы, исследовать содержание каталога, ставить метки, монтировать и размонтировать ленты.

5.15 Zabbix

Zabbix – система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

Для управления системой мониторинга и чтения данных используется веб-интерфейс.

Дожен быть установлен и запущен сервер PostgreSQL, с созданным пользователем zabbix и созданной базой zabbix.

Веб-интерфейс zabbix доступен по адресу:

<http://ip-сервера/zabbix>

6 НАСТРОЙКА СЕТИ

NetworkManager позволяет подключаться к различным типам сетей: проводные, беспроводные, мобильные, VPN и DSL, а также сохранять эти подключения для быстрого доступа к сети.

6.1 NetworkManager

Для управления настройками сети в ОС Альт Сервер 8.0 используется программа NetworkManager (рис.12-13).

NetworkManager

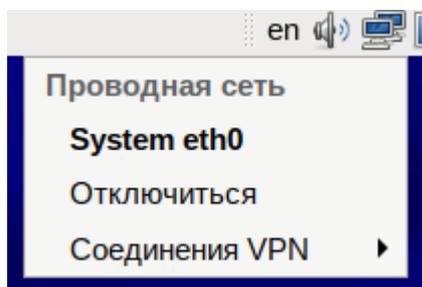


Рис.12

При нажатии левой кнопкой мыши на значок **NetworkManager**, вы увидите меню, в котором можно выбрать одну из доступных сетей и подключиться к ней. Из этого меню так же можно отключить активное Wi-Fi соединение.

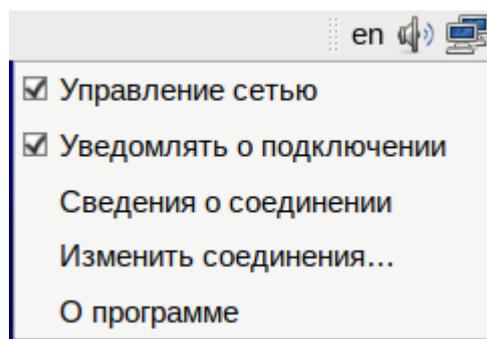
NetworkManager

Рис.13

При нажатии правой кнопкой мыши на значок **NetworkManager**, вы увидите меню, из которого можно получить доступ к изменению некоторых настроек. Вы можете узнать версию программы, получить сведения о соединении, можете изменить соединения (например, удалить Wi-Fi сеть, чтобы не подключаться к ней автоматически).

Вы можете начать работу по настройке сервера сразу после установки системы, используя для настройки **Центр управления системой** — веб-ориентированный интерфейс, позволяющий управлять сервером с любого компьютера сети.

7 УСТАНОВКА ДОПОЛНИТЕЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

После установки ОС Альт Сервер 8.0, при первом запуске, вам доступен тот или иной набор программного обеспечения. Количество предустановленных программ зависит от выбора, сделанного вами при установке системы. Если вы не обнаружили в своей системе интересующие вас программы, то вы имеете возможность доустановить их из разных источников.

Дополнительное программное обеспечение может находиться на установочном диске и/или в специальных банках программ (репозиториях), расположенных в сети Интернет и/или в локальной сети. Программы, размещённые в указанных источниках, имеют вид подготовленных для установки пакетов.

Для установки дополнительного ПО можно использовать программу управления пакетами **Synaptic**.

7.1 Программа управления пакетами Synaptic

Программа управления пакетами **Synaptic** находится в Система → Параметры → Программа управления пакетами **Synaptic**.

Для облегчения поиска доступные для установки программы разделены на группы, выводимые в левой части окна программы. Справа расположен список самих программ с указанием их текущего состояния:

- зелёная метка – пакет уже установлен;
- белая метка – пакет не установлен.

Для начала установки двойным щелчком мыши отметьте неустановленный пакет в правой половине окна и нажмите **Применить**. При необходимости, менеджер пакетов попросит вставить установочный диск.

7.2 Добавление репозиториев

Программа **Synaptic** может использоваться для выбора репозитория, совместимого с вашим дистрибутивом. Для указания конкретного репозитория в меню **Параметры** → **Репозитории** отметьте один из предлагаемых вариантов и нажмите кнопку **ОК**. Если вы сомневаетесь, то выберите строки, содержащие `ftp://ftp.altlinux.org/`. К предложенному списку вы можете самостоятельно добавить любые репозитории, нажав на кнопку **Создать** и введя необходимые данные.

После добавления репозиториев обновите информацию о них: программа управления пакетами **Synaptic**: **Правка** → **Получить сведения о пакетах**.

После выбора и добавления репозиториев необходимо получить сведения о находящихся в них пакетах. В противном случае, список доступных для установки программ будет не актуален.

8 ОБЩИЕ ПРИНЦИПЫ РАБОТЫ ОС

Работа с операционной средой заключается в вводе определенных команд (запросов) к операционной среде и получению на них ответов в виде текстового отображения.

Основой операционной среды является операционная система.

Операционная система (ОС) – совокупность программных средств, организующих согласованную работу операционной среды с аппаратными устройствами компьютера (процессор, память, устройства ввода-вывода и т. д.).

Диалог с ОС осуществляется посредством командных интерпретаторов и системных библиотек.

Каждая системная библиотека представляет собой набор программ, динамически вызываемых операционной системой.

Командные интерпретаторы – особый род специализированных программ, позволяющих осуществлять диалог с ОС посредством команд.

Для удобства пользователей при работе с командными интерпретаторами используются интерактивные рабочие среды (далее – ИРС), предоставляющие пользователю удобный интерфейс для работы с ОС.

В самом центре ОС изделия находится управляющая программа, называемая ядром. В ОС изделия используется новейшая модификация «устойчивого» ядра Linux – версия 4.1.

Ядро взаимодействует с компьютером и периферией (дисками, принтерами и т. д.), распределяет ресурсы и выполняет фоновое планирование заданий.

Другими словами, ядро ОС изолирует вас от сложностей аппаратуры компьютера, командный интерпретатор от ядра, а ИРС от командного интерпретатора.

Защита операционной среды осуществляется с помощью комплекса встроенных средств защиты информации.

ОС Альт Сервер 8.0 является многопользовательской интегрированной системой. Это значит, что она разработана в расчете на одновременную работу нескольких пользователей.

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

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

8.1 Процессы функционирования ОС

Все программы, которые выполняются в текущий момент времени, называются процессами. Процессы можно разделить на два основных класса: системные процессы и пользовательские процессы. Системные процессы – программы, решающие внутренние задачи ОС, например, организацию виртуальной памяти на диске или предоставляющие пользователям те или иные сервисы (процессы-службы).

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

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

8.2 Файловая система ОС

В ОС использована файловая система Linux, которая в отличие от файловых систем DOS и Windows(™) является единым деревом. Корень этого дерева – каталог, называемый root (рут), и обозначаемый «/». Части дерева файловой системы могут физически располагаться в разных разделах разных дисков или вообще на других компьютерах, – для пользователя это прозрачно. Процесс присоединения файловой системы раздела к дереву называется монтированием, удаление – размонтированием. Например, файловая система CD-ROM в изделии монтируется по умолчанию в каталог /media/cdrom (путь в изделии обозначается с использованием «/», а не «\», как в DOS/Windows). Текущий каталог обозначается «./».

Файловая система изделия содержит каталоги первого уровня:

- /bin (командные оболочки (shell), основные утилиты);
- /boot (содержит ядро системы);
- /dev (псевдофайлы устройств, позволяющие работать с ними напрямую);
- /etc (файлы конфигурации);
- /home (личные каталоги пользователей);
- /lib (системные библиотеки, модули ядра);
- /lib64 (64-битные системные библиотеки);
- /media (каталоги для монтирования файловых систем сменных устройств);
- /mnt (каталоги для монтирования файловых систем сменных устройств и внешних файловых систем);
- /proc (файловая система на виртуальном устройстве, ее файлы содержат информацию о текущем состоянии системы);
- /root (личный каталог администратора системы);
- /sbin (системные утилиты);
- /sys (файловая система, содержащая информацию о текущем состоянии системы);
- /usr (программы и библиотеки, доступные пользователю);
- /var (рабочие файлы программ, очереди, журналы);
- /tmp (временные файлы).

8.3 Организация файловой структуры

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

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

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

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

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

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

8.3.1 Иерархическая организация файловой системы

Каталог /:

/boot – место, где хранятся файлы необходимые для загрузки ядра системы;

/lib – здесь располагаются файлы динамических библиотек, необходимых для работы большей части приложений и подгружаемые модули ядра;

/lib64 – здесь располагаются файлы 64-битных динамических библиотек, необходимых для работы большей части приложений;

/bin – минимальный набор программ необходимых для работы в системе;

/sbin – набор программ для административной работы с системой (программы необходимые только суперпользователю);

/home – здесь располагаются домашние каталоги пользователей;

/etc – в данном каталоге обычно хранятся общесистемные конфигурационные файлы для большинства программ в системе;

/etc/rc?.d,/etc/init.d,/etc/rc.boot,/etc/rc.d – директории, где расположены командные файлы системы инициализации SysVinit;

`/etc/passwd` – база данных пользователей, в которой содержится информация об имени пользователя, его настоящем имени, личном каталоге, закодированный пароль и другие данные;

`/etc/shadow` – теневая база данных пользователей. При этом информация из файла `/etc/passwd` перемещается в `/etc/shadow`, который недоступен по чтению всем, кроме пользователя `root`. В случае использования альтернативной схемы управления теневыми паролями (ТСВ) все теневые пароли для каждого пользователя располагаются в директории `/etc/tcb/<имя пользователя>/shadow`;

`/dev` – в этом каталоге находятся файлы устройств. Файлы в `/dev` создаются сервисом `udev`;

`/usr` – обычно файловая система `/usr` достаточно большая по объему, так как все программы установлены именно здесь. Вся информация в каталоге `/usr` помещается туда во время установки системы. Отдельно устанавливаемые пакеты программ и другие файлы размещаются в каталоге `/usr/local`. Некоторые подкаталоги системы `/usr` рассмотрены ниже;

`/usr/bin` – практически все команды, хотя некоторые находятся в `/bin` или в `/usr/local/bin`;

`/usr/sbin` – команды, используемые при администрировании системы и не предназначенные для размещения в файловой системе `root`;

`/usr/local` – здесь рекомендуется размещать файлы, установленные без использования пакетных менеджеров, внутренняя организация каталогов практически такая же, как и корневого каталога;

`/usr/man` – каталог где хранятся файлы справочного руководства `man`;

`/usr/share` – каталог для размещения общедоступных файлов большей части приложений.

Каталог `/var`:

`/var/log` – место, где хранятся файлы аудита работы системы и приложений;

`/var/spool` – каталог для хранения файлов находящихся в очереди на обработку для того или иного процесса (очередь на печать, отправку почты и т. д.);

`/tmp` – временный каталог необходимый некоторым приложениям;

`/proc` – файловая система `/proc` является виртуальной и в действительности она не существует на диске. Ядро создает её в памяти компьютера. Система `/proc` предоставляет информацию о системе.

8.3.2 Имена дисков и разделов

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

`/dev/sda` – первый диск;

`/dev/sdb` – второй диск;

и т. д.

Диски обозначаются `/dev/sdX`, где `X` – `a,b,c,d,e,...` в порядке обнаружения системой.

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

8.4 Разделы, необходимые для работы ОС

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

8.5 Утилиты для работы с файловой системой

8.5.1 mkfs

Утилита `mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesystem` для файловой системы может выступать или название устройства (например, `/dev/sda1`) или точка монтирования (например, `/`, `/usr`, `/home`). Аргументом `blocks` указывается количество блоков, которые выделяются для использования этой файловой системой.

По окончании работы `mkfs` возвращает 0 в случае успеха и 1 при неудачной операции.

В общем случае, `mkfs` является простым конечным интерфейсом к доступным под Linux модулям создания файловых систем, в которых вторая часть сложных имён (`mkfs.fstype`) как раз и определяет вызываемый модуль. Поиск специфического модуля создания файловой системы осуществляется примерно в следующей последовательности каталогов: `/sbin`, `/sbin/fs`, `/sbin/fs.d`, `/etc/fs`, `/etc`. Точный список каталогов определяется во время компиляции, но как минимум содержит `/sbin` и `/sbin/fs`, а завершается каталогами, которые перечислены в переменной окружения `PATH`. Для детальной информации по созданию специфических файловых систем, пожалуйста, просмотрите соответствующие электронные справочные страницы (`man`).

Синтаксис:

```
mkfs [-V] [-t fstype] [fs-options] filesystem [blocks]
```

Параметры:

- `-t fstype` – указывает тип создаваемой файловой системы. Если этот параметр не указан, тогда, по умолчанию, принимается тип файловой системы `ext2`;
- `fs-options` – передаёт модулю создания специфической файловой системы параметры в виде списка. Следует отметить, что нет гарантии в том, что следующие перечисленные параметры будут поддерживаться большинством модулей создания файловых систем;
- `-V`, `--verbose` – производит подробный вывод, включающий все выполняемые специфические команды файловой системы. Если указать этот параметр более одного раза, то это за-

претит реальное выполнение любых специфических команд файловой системы. Использовать этот параметр целесообразно во время тестирования;

- `-V`, `--version` – выводит информацию о версии. Опция `-V` выведет информацию о версии только в том случае, когда является единственным параметром, в противном случае эта опция будет работать как `-verbose`;
- `-h` – выводит справку.

Все основные параметры должны быть указаны в начале и не должны смешиваться с параметрами, которые передаются для специфичных файловых систем. Некоторые модули создания специфичных файловых систем не поддерживают параметр `-V` (подробный вывод) или не возвращают осмысленные коды возврата. Кроме этого, некоторые модули автоматически (самостоятельно) не могут определить размер устройства и для них обязательно надо указывать параметр `size`.

Пример

Создать файловую систему типа `ext2` в разделе `/dev/sdb1` (второй жёсткий диск):

```
# mkfs -t ext2 /dev/sdb1
```

8.5.2 fsck

Утилита `fsck` используется для проверки и/или для восстановления одной, либо нескольких файловых систем. В качестве аргумента для файловой системы `filesystem` может быть именем устройства (например: `dev/hdc1`, `/dev/sdb2`), точкой монтирования (например: `/`, `/usr`, `/home`), меткой `ext2` или `UUID` (например: `LABEL=root` или `UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd`). Для уменьшения общего времени проверки `fsck` старается параллельно работать с файловыми системами на различных физических носителях.

Если при вызове `fsck` не указана ни одна файловая система, а также не указан ключ `-A`, будет произведена последовательная проверка для всех записей в `/etc/fstab`. Это эквивалентно опциям `-As`.

Код возврата `fsck` является суммой следующих состояний:

- 0 – нет ошибок;
- 1 – произошла ошибка файловой системы;
- 2 – система должна быть перезагружена;
- 4 – ошибки файловой системы не были исправлены;
- 8 – ошибка в работе;
- 16 – ошибка использования или синтаксиса;
- 32 – `fsck` был завершён по требованию пользователя;
- 128 – ошибка разделяемой библиотеки.

При проверке нескольких файловых систем код возврата `fsck` считается как побитовое ИЛИ от кодов возврата проверяемых файловых систем.

Фактически, `fsck` является оболочкой для вызова специфичной для конкретной файловой системы утилиты проверки (`fsck.fstype`), доступной в Linux. Специфичная для файловой системы утилита сначала ищется в каталоге `/sbin`, затем в `/etc/fs` и `/etc`, и, в конце, в каталогах, перечисленных в переменной окружения `PATH`. Более подробная информация доступна в руководствах к специфичным для файловых систем утилитам.

Синтаксис:

```
fsck [-lrsAVRTMNP] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]
```

Опции `fsck`:

- `-l` – заблокировать весь диск при помощи `flock`. Данная опция может быть использована только с одним устройством (то есть опции `-A` и `-l` взаимоисключающие). Это опция является рекомендованной в ситуации одновременного запуска нескольких `fsck`. Эта опция игнорируется при вызове для нескольких устройств либо для не вращающихся дисков. `fsck` не блокирует составные устройства, например, MD или DM – такая возможность ещё не реализована;
- `-g` – вывести статистику для каждого `fsck` при завершении. Эта статистика содержит код возврата, занимаемую оперативную память (в КБ), прошедшее время, и затраченное процессорное время, использованное `fsck`. Например, `/dev/sda1: status 0, rss 92828, real 4.002804, user 2.677592, sys 0.86186`;
- `-s` – выполнять `fsck` по очереди. Это хорошая идея, если вы проверяете несколько файловых систем, и проверки работают в интерактивном режиме. Обратите внимание, что `fsck` по умолчанию запускается в интерактивном режиме. Чтобы запустить его в не интерактивном режиме потребуется указать опцию `-p` или `-a` если вы хотите, чтобы ошибки были исправлены автоматически, или опцию `-n` если вы этого не хотите;
- `-C [fd]` – отображать индикатор процесса для утилиты проверки файловых систем, которые это поддерживают (в настоящее время поддерживаются только `ext2` и `ext3`). `fsck` управляет утилитами проверки файловых систем таким образом, что только одна из них будет отображать индикатор процесса в один момент времени. Графические оболочки могут указать файловый дескриптор `fd`. При этом информация об индикаторе процесса будет отправлена в этот дескриптор;
- `-M` – не проверять примонтированные файловые системы, возвращая для них код возврата 0;
- `-N` – не выполнять, а просто показать, что было бы выполнено;
- `-P` – в сочетании с опцией `-A` заставит `fsck` проверить корневую файловую систему параллельно с остальными файловыми системами. Эта опция не может считаться безопасной, так как выполнение программ (например, `e2fsck`) располагающихся на корневой файловой си-

стеме в сомнительном состоянии может вызвать ошибки. Данная опция предоставляется для тех администраторов, кто не желает разделять корневую файловую систему на более маленькие и компактные;

- **-R** – используется совместно с опцией **-A** для пропуска проверки корневой файловой системы. Это может быть полезно в том случае если корневая файловая система уже смонтирована для чтения/записи;
- **-T** – не показывать заголовков при запуске;
- **-V** – генерировать более подробный вывод, включая все специфичные для файловых систем команды;
- **-t fslist** – задаёт тип(ы) файловых систем для проверки. В сочетании опцией **-A** будут проверены только файловые системы из списка **fslist**. **fslist** это список файловых систем и опций, разделённых запятыми. Перед списком файловых систем можно указать префикс «по» или «!» (оператор отрицания), означающий, что будут проверены только файловые системы, не перечисленные в списке. Если никакая из файловых систем в списке не имеет такого префикса, то проверяться будут файловые системы из списка. В **fslist** могут быть включены опции файловой системы. Они должны быть указаны в формате **opts=fs-option**.

Если указаны опции файловой системы, то будут проверены только те файловые системы, для которых в **/etc/fstab** указаны эти опции. Если опции файловой системы указаны с оператором отрицания, то будут проверены только те файловые системы, для которых не указаны эти опции в **/etc/fstab**.

Например, если в **fslist** есть **opts=ro**, то будут проверены файловые системы, для которых указана опция **ro** в **/etc/fstab**.

Если в **fslist** содержится файловая система типа **loop**, то такая ситуация обрабатывается как будто для опции **-t** указан аргумент **opts=loop**.

Обычно, тип файловой системы определяется поиском по **/etc/fstab**. Если же таким способом определить файловую систему не удаётся и задана одна файловая система с опцией **-t**, то **fsck** будет использовать указанный тип файловой системы. Если этот тип недоступен, то будет использован тип файловой системы по умолчанию (в данный момент это **ext2**);

- **-A** – попытаться проверить сразу все файловые системы, указанные в **/etc/fstab**. Обычно эта опция используется системой при загрузке системы (вместо запуска множества отдельных команд для проверки каждой файловой системы).

Если не указана опция **-P**, то корневая файловая система будет проверена в первую очередь. После этого файловые системы будут проверяться в порядке, соответствующем значению **fs_passno** в **/etc/fstab** (шестое поле). Файловые системы с **fs_passno** равным 0 не проверяются. Файловые системы с **fs_passno** более нуля проверяются в порядке возрастания **fs_passno**.

Если есть несколько файловых систем с одинаковым `fs_passno`, то `fsck` попытается проверить их параллельно, при этом стараясь не запускать несколько проверок для файловых систем, расположенных на одном физическом диске.

`fsck` не производит параллельную проверку составных устройств (RAID, `dm-crypt`, ...) с другими устройствами. Для проверки зависимостей между устройствами используется файловая система `/sys`.

Таким образом, очень часто в `/etc/fstab` встречается, что для корневой файловой системы `fs_passno` равен 1, а для остальных – 2. Это позволяет `fsck` запускать параллельную проверку файловых систем, в случае если это будет полезно. Системный администратор может принять решение не использовать настройки по умолчанию, например, если в компьютере очень мало оперативной памяти и её может быть недостаточно для параллельной проверки. `fsck` не проверяет факт существования устройства перед вызовом специфичной для файловой системы утилиты. Если утилита проверки файловой системы возвращает критическую ошибку при попытке проверить несуществующую файловую систему, то это приведёт к входу в восстановительный режим при загрузке операционной системы. Чтобы этого не происходило можно указать опцию монтирования «`nofail`» в `/etc/fstab`. Также `fsck` пропустит несуществующие устройства, для которых задан специальный тип файловой системы «`auto`»;

- `fs-specific-options` – опции, не обрабатываемые `fsck` передаются в специфичную для файловой систем утилиту. Эти опции не должны содержать аргументов, так как `fsck` не может правильно решить, какие опции принимают аргументы, а какие не принимают.

Опции и аргументы следующие за «`--`» расцениваются как специфичные для файловой системы и передаются в специфичную для файловой системы утилиту.

Обратите внимание, что `fsck` не рассчитан на передачу произвольных сложной опций в специфичные для файловой системы утилиты. Если вы делаете что-то сложное, пожалуйста, пользуйтесь непосредственно утилитой для конкретной файловой системы. Если вы пытаетесь вызвать `fsck` со сложными аргументами, и это не работает, то не следует сообщать об этом как об ошибке – скорее всего вы делаете что-то, чего не следует делать при помощи `fsck`.

Опции для различных файловых систем не стандартизированы. Если вы сомневаетесь, пожалуйста, обратитесь к руководству по специфичной для файловой системы утилите. Мы не гарантируем, что следующие опции поддерживаются всем утилитами:

- `-a` – автоматически восстановить файловую систему, не задавая каких-либо вопросов (используйте эту опцию с осторожностью). Обратите внимание, что `e2fsck` содержит поддержку опции `-a` только для обратной совместимости. Она соответствует опции `e2fsck -p`, которая более безопасна;

- `-n` – для некоторых утилит эта опция означает не пытаться исправить какие-либо проблемы, а просто сообщить о них. Однако не все утилиты поддерживают данную опцию. Например, `fsck.reiserfs` не сообщает о каких-либо ошибках, а `fsck.minix` вовсе не поддерживает `-n`;
- `-r` – интерактивное восстановление файловой системы (с запросом подтверждений). Обратите внимание, что использование этой опции не является хорошей идеей при параллельной работе нескольких утилит проверки. Также имейте в виду, что такое поведение является поведением по умолчанию для утилиты `e2fsck`, а опция `-r` в ней сохранена для обратной совместимости;
- `-u` – для некоторых утилит опция `-u` означает автоматически пытаться исправить все повреждения файловой системы. В некоторых случаях эксперт может сделать это лучше вручную. Обратите внимание, что не все утилиты содержат реализацию данной опции. Например, `fsck.minix` и `fsck.cramfs` не поддерживают `-u`.

Переменные окружения:

- `FSCK_FORCE_ALL_PARALLEL` – если задана эта переменная окружения, то `fsck` будет пытаться параллельно проверять все заданные файловые системы вне зависимости от того находятся они на одном устройстве или нет. Это может быть полезно для RAID-систем или высокопроизводительных систем хранения от таких компаний как IBM и EMC. Обратите внимание, что `fs_passno` продолжает действовать;
- `FSCK_MAX_INST` – при помощи данной переменной окружения можно ограничить количество запусков специфичных для файловых систем утилит, работающих одновременно. Для систем с большим количеством дисков это предотвратит запуск большого числа проверок, что могло бы привести к перегрузке процессора или памяти. Если значение равно нулю, то количество запусков не ограничивается. Это является поведением по умолчанию, но будущие версии `fsck` могут пытаться автоматически определить, сколько проверок может быть запущено на основании данных операционной системы;
- `PATH` – используется для поиска специфичных для файловых систем утилит. Сначала поиск осуществляется в каталогах `/sbin`, `/sbin/fs.d`, `/sbin/fs`, `/etc/fs` и `/etc`. Затем, в каталогах, указанных в переменной `PATH`;
- `FSTAB_FILE` – позволяет администратору указать альтернативное положение файла `/etc/fstab`. Эта переменная окружения также может быть полезна разработчикам при тестировании `fsck`;
- `LIBBLKID_DEBUG=0xffff` – включает вывод отладочной информации;
- `LIBMOUNT_DEBUG=0xffff` – включает вывод отладочной информации.

8.5.3 df

Утилита `df` показывает количество доступного дискового пространства в файловой системе, в которой содержится файл, переданный как аргумент. Если ни один файл не указан, показывается доступное место на всех смонтированных файловых системах. Размеры указаны в блоках по 1КБ по умолчанию, за исключением заданной переменной окружения `POSIXLY_CORRECT` (в этом случае используются 512-байтовые блоки).

Если файл указан как абсолютный путь к файлу устройства, содержащего смонтированную файловую систему, `df` показывает доступное место для этой файловой системы, а не для файловой системы, содержащей этот файл устройства (которая всегда будет корневой файловой системой). Эта версия `df` не показывает доступное место на несмонтированных файловых системах, потому что на большинстве систем это требует глубокого знания структуры файловой системы.

По умолчанию показывает информация обо всех смонтированных файловых системах.

Синтаксис:

```
df [опция]... [файл]...
```

Опции:

- `-a, --all` – включать виртуальные файловые системы;
- `-B, --block-size=РАЗМЕР` – использовать блоки указанного РАЗМЕРА (в байтах). Например, укажите `-BM` для вывода блоков размером в 1048576 байт;
- `--total` – подсчитать общий объём в конце;
- `-h, --human-readable` – печатать размеры в удобочитаемом формате (например, 1К 234М 2G);
- `-H, --si` – то же, но использовать степени 1000, а не 1024;
- `-i, --inodes` – вывести информацию об индексных дескрипторах, а не об использовании блоков;
- `-k` – аналог `--block-size=1К`;
- `-l, --local` – ограничиться выводом локальных файловых систем;
- `--no-sync` – не вызывать `sync` перед получением информации (по умолчанию);
- `--output[=СПИСОК_ПОЛЕЙ]` – использовать формат, определённый в СПИСОК_ПОЛЕЙ, или печатать все поля, если СПИСОК_ПОЛЕЙ не задан;
- `-P, --portability` – выводить в формате POSIX;
- `--sync` – вызывать `sync` перед получением информации;
- `-t, --type=ТИП` – перечислить только файловые системы указанного ТИПА;
- `-T, --print-type` – печатать тип файловой системы;
- `-x, --exclude-type=ТИП` – исключить файловые системы указанного ТИПА;
- `-v` – игнорируется;

- --help – вывод справки и выход;
- --version – вывод информации о версии и выход.

Отображаемые значения будут в единицах, описанных в --block-size либо в переменных окружения DF_BLOCK_SIZE, BLOCK_SIZE и BLOCKSIZE. Значение по умолчанию 1024 байтов (512 байтов, если задана переменная окружения POSIXLY_CORRECT).

РАЗМЕР может быть таким (или это может быть целое число, за которым может следовать одно из перечисленного): kB 1000, K 1024, MB 1000*1000, M 1024*1024 и аналогично для G, T, P, E, Z, Y.

СПИСОК_ПОЛЕЙ это разделённый запятыми список колонок для вывода на экран. Доступны для использования следующие поля: 'source', 'fstype', 'itotal', 'iused', 'iavail', 'ipcent', 'size', 'used', 'avail', 'pcent' и 'target'.

8.5.4 du

Утилита du подсчитывает использование диска каждым файлом, для каталогов подсчёт происходит рекурсивно.

Синтаксис:

```
du [опции] [файл...]
```

Опции:

- -0, --null – вместо символа новой строки выводить нулевой байт;
- -a, --all – выводить общую сумму для каждого заданного файла, а не только для каталогов;
- --apparent-size – печатать видимый размер вместо реального использования диска. Обычно видимый размер меньше реального. Однако он может быть и меньше из-за "дырок" в разряженных файлах, внутреннего состояния флагов, непрямых блоков и других причин;
- -B, --block-size=РАЗМЕР – перед печатью преобразовать размеры к РАЗМЕР. Например, используйте -BM для печати объёма в единицах равных 1048576 байтов;
- -b, --bytes – тоже что и '--apparent-size --block-size=1';
- -c, --total – подсчитать общий объём в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;
- -D, --dereference-args – раскрывать только те символичные ссылки, которые указаны в командной строке (полезно для поиска использованного дискового пространства в таких каталогах, как /usr/tmp, которые часто являются символическими ссылками);
- -d, --max-depth=N – выводить объём для каталога (или файлов, если указано --all) только если она на N или менее уровней ниже аргументов командной строки. --max-depth=0 эквивалентно --summarize;

- `--files0-from=F` – вывести отчет об использовании диска теми файлами, чьи имена (завершённые нулем) перечислены в файле F. Если F равно -, то имена читаются со стандартного ввода;
- `-H, --si` – выводить размеры в удобном для чтения человеком формате, но использовать степени 1000, а не 1024;
- `-h, --human-readable` – выводить размеры в удобном для чтения человеком формате (например, 1K 234M 2G);
- `-k` – тоже что и `--block-size=1K`;
- `-L, --dereference` – раскрывать символические ссылки (показывать дисковое пространство, используемое файлом или каталогом, на которые указывает ссылка, вместо пространства, используемого самой ссылкой);
- `-l, --count-links` – учитывать объём несколько раз, если есть несколько жёстких ссылок;
- `-m` – тоже что и `--block-size=1M`;
- `-P, --no-dereference` – не следовать по символически ссылкам (по умолчанию);
- `-S, --separate-dirs` – выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;
- `-s, --summarize` – отобразить только сумму для каждого аргумента;
- `-t, --threshold=РАЗМЕР` – исключить элементы, которые меньше РАЗМЕРА, если это значение положительно, или значение которых больше, если значение отрицательно;
- `--time` – показать время последнего изменений для всех файлов в каталоге или во всех его подкаталогах;
- `--time=СЛОВО` – показывать указанное СЛОВОМ время, а не время изменения: `atime`, `access`, `use`, `ctime` или `status`;
- `--time-style=СТИЛЬ` – отображать время используя СТИЛЬ. Доступны следующие стили: `full-iso`, `long-iso`, `iso`, `+FORMAT`. `FORMAT` обрабатывается так же, как и в команде 'date';
- `-X, --exclude-from=ФАЙЛ` – исключить файлы, соответствующие любому образцу из файла ФАЙЛ;
- `--exclude=ШАБЛОН` – исключить файлы, соответствующие образцу ШАБЛОН;
- `-x, --one-file-system` – пропустить каталоги на других файловых системах;
- `--help` – напечатать справку и выйти;
- `--version` – напечатать информации о версии и выйти.

Обязательные аргументы в длинных опциях обязательны также и для коротких опций.

Отображаемые значения будут в единицах, описанных в `--block-size` либо в переменных окружения `DF_BLOCK_SIZE`, `BLOCK_SIZE` и `BLOCKSIZE`. Значение по умолчанию 1024 байтов (512 байтов, если задана переменная окружения `POSIXLY_CORRECT`).

РАЗМЕР – это целое число, за которым опционально следует единица изменения. Например, 10М соответствует $10 \cdot 1024 \cdot 1024$. Доступны следующие единицы измерения: К, М, G, Т, P, E, Z, Y (возведение в степень числа 1024), KB, MB и т.д. (возведение в степень числа 1000).

ШАБЛОН – это образец командной строки (не регулярное выражение). Знак `?` соответствует любому символу. Знак `*` соответствует любой строке (ни одного, один или несколько символов). Например, `*.o` будет соответствовать всем файлам, чьи имена заканчиваются на `.o`. Таким образом, команда `du --exclude='*.o'` пропустит все файлы и подкаталоги, заканчивающиеся на `.o` (в том числе и файл `.o`).

8.5.5 find

Утилита `find` используется для поиска файлов.

Синтаксис:

```
find [-H] [-L] [-P] [-0уровень] [-D help|tree|search|stat|rates|opt|exec] [путь...]
[выражение]
find [путь] [опции] [критерии поиска] [действия над файлами]
```

В качестве пути для поиска можно использовать как абсолютные, так и относительные пути, а также список путей, разделенных пробелом. Путём по умолчанию является текущий подкаталог. Выражение по умолчанию `-print`.

Основные опции

- `-d, --depth` – поиск в подкаталогах перед поиском в самом каталоге;
- `-L` – при поиске следовать по символическим ссылкам;
- `-P` – никогда не следовать по символическим ссылкам;
- `-maxdepth N` – при поиске проверять не более чем N вложенных уровней каталогов;
- `-mindepth N` – не проверять вложенные каталоги уровня N и меньше;
- `-mount` – не искать в каталогах других файловых систем.

У команды `find` может быть несколько критериев поиска (`tests`). Каждый критерий представляет собой определенное условие проверки, которое возвращает либо `true` либо `false`. В процессе обработки очередного файла команда `find` по очереди проверяет каждый критерий, и если очередной критерий возвращает `false`, тогда команда `find` переходит к следующему файлу.

Основные критерии поиска:

- `-name шаблон` – имя файла (шаблон имени) без указания пути. Рекомендуется всегда заключать шаблон в кавычки;

- `-atime N` – последний доступ к файлу производился N дней назад. (`-atime +1` найдёт файлы, доступ к которым осуществлялся как минимум два дня назад);
- `-mtime N` – последнее изменение файла было N дней назад;
- `-ctime N` – статус файла последний раз изменялся N дней назад;
- `-newer другой_файл` – файл был модифицирован позднее, чем другой_файл;
- `-size [±]N[cwbkMG]` – размер файла равен N блокам, если указано $+N$, тогда размер файла больше N , $-N$ – меньше. Символ после N означает размер блока (b – 512 байт, c – байт, w – 2 байта, k – килобайт, M – мегабайт, G – гигабайт);
- `-type C` – файл имеет тип c , где c есть b (блочный специальный файл), c (символьный специальный файл), d (каталог), p (именованный канал), f (обычный файл), l (символьная ссылка) или s (сокет);
- `[-perm] [-]восьмеричное_число` – режим доступа к текущему файлу в точности равен восьмеричному_числу. Если перед восьмеричным_числом указан знак -, то для сравнения из режима файла берутся только биты, соответствующие битам восьмеричного_числа, равным единице;
- `-links n` – на файл имеется n ссылок;
- `-user имя_пользователя` – файл принадлежит пользователю с данным именем. Разрешены цифровые идентификаторы пользователя;
- `-group имя_группы` – файл принадлежит группе с данным именем. Разрешены цифровые идентификаторы группы.

Критерии можно объединять, используя операторы. Ниже приведены операторы в порядке убывания их приоритета:

- унарная операция отрицания, обозначается ! (! критерий);
- логическое И, обозначается пробелом (критерий1 критерий2);
- логическое ИЛИ, обозначается -o (критерий1-o критерий2).

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

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

- `-print` – вывод имени файла на экран.

Примеры

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

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

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

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

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

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

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

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

8.5.6 which

Утилита `which` – отображает полный путь к указанным командам или сценариям.

`Which` принимает один или более аргументов `имя_программы`. Для каждого из них она выводит тот полный путь к исполняемому файлу, который будет использован командной оболочкой, если `имя_программы` ввести в качестве команды в командной строке. Эта утилита выполняет поиск исполняемых файлов или сценариев в каталогах, перечисленных в переменной окружения `PATH`, используя тот же алгоритм что и `bash`.

Синтаксис:

```
which [опции] [--] имя_программы [...]
```

Опции:

- `--all`, `-a` – выводит все совпавшие исполняемые файлы по содержимому в переменной окружения `PATH`, а не только первый из них;
- `--read-alias`, `-i` – считывает псевдонимы, поступающие из стандартного ввода и направляет на стандартный вывод информацию по совпавшим. Эта опция полезна в сочетании с использованием псевдонима для самой команды `which`. Например: `which='alias | which -i'`;
- `--skip-alias` – игнорирует опцию «`--read-alias`». Эта опция полезна для точного поиска обычных двоичных файлов, даже если опция «`--read-alias`» задана в псевдониме или в функции для `which`;
- `--read-functions` – считывает функции, определённые в командной оболочке и поступающее со стандартного ввода, а затем направляет на стандартный вывод информацию по совпавшим. Эта опция полезна в сочетании с функциями командной оболочки для самой команды `which`. Например: `which() { declare -f | which --read-functions $@ } export -f which`;

- `--skip-functions` – игнорирует опцию «`--read-functions`», если таковая имеется. Это опция полезна для точного поиска обычных двоичных файлов, даже если опция «`--read-functions`» задана в псевдониме или в функции для `which`;
- `--skip-dot` – пропускает все каталоги из переменной окружения `PATH`, которые начинаются с точки;
- `--skip-tilde` – пропускает все каталоги из переменной окружения `PATH`, имена которых начинаются с символа тильды (`~`), а также все исполняемые файлы, которые расположены в каталоге, указанном в переменной `HOME`;
- `--show-dot` – если имя каталога из переменной окружения `PATH` начинается с точки и соответствующий исполняемый файл был найден в этом пути, то вывести относительный путь вместо полного;
- `--show-tilde` – выводит тильду, в том случае если исполняемый файл хранится в каталоге, указанном в переменной окружения `HOME` (т.е. с домашним каталогом). Данная опция игнорируется при вызове `which` от пользователя `root`;
- `--tty-only` – не обрабатывает опции, которые находятся правее этой, если они поступают не с терминала (`tty`);
- `--version`, `-v`, `-V` – выводит информацию о версии на стандартный вывод и завершает её работу;
- `--help` – выводит краткое описание опций программы на стандартный вывод и завершает её работу.

Домашний каталог определяется из переменной окружения `HOME`, поэтому если эта переменная не определена, то это вызовет прерывание выполнения данной программы. Программа `which` будет считать два эквивалентных каталога различными, если один из них содержит путь с символической ссылкой.

Обычно утилиту `which` используют в различных псевдонимах (`C shell`) или сценариях (`Bourne shell`) командных оболочек. Например, в командной оболочке `bash`:

```
which () {
(alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot $@
}
export -f which
```

В следующем примере показано два возможных варианта вывода одной и той же команды `which`, в зависимости от контекста ее применения. В первом случае, вместо полного пути выводится `~/` и `./` (для исполняемого файла, расположенного в домашнем каталоге пользователя), когда команда `which` непосредственно запущена пользователем из командной строки. Во втором – команда `which` запущена из скрипта. Здесь будет выведен полный путь к указанной программе:

```
$ which q2
~/bin/q2
```

```
$echo which q2
/nome/test/bin/q2
```

8.5.7 cd

Команда `cd` применяется для того, чтобы сделать заданный каталог текущим. Если каталог не указан, используется значение переменной окружения `HOME` (домашний каталог пользователя). Если каталог задан полным маршрутным именем, он становится текущим. Если маршрутное имя не полное, команда `cd` пытается найти каталог по одному из маршрутов, заданных переменной окружения `CDPATH`. Способ задания и семантика этой переменной такие же, как у `PATH`. По отношению к новому каталогу нужно иметь право на выполнение, которое в данном случае трактуется как разрешение на поиск.

Синтаксис:

```
cd [-L|-P] [каталог]
```

Опция `-L` заставляет следовать по символическим ссылкам.

Поскольку для выполнения каждой команды создается отдельный процесс, `cd` не может быть обычной командой; она распознается и выполняется командной оболочкой

Если в качестве аргумента задано `-`, то это эквивалентно `$OLDPWD`.

Если переход был осуществлён по переменной окружения `CDPATH` или в качестве аргумента был задан `-` и смена каталога была успешной, то абсолютный путь нового рабочего каталога будет выведен на стандартный вывод.

8.5.8 pwd

Утилита `pwd` выводит полный путь текущего (рабочего) каталога.

Синтаксис:

```
pwd [-LP]
```

Опции:

- `-P` – вывод не будет содержать символических ссылок;
- `-L` – вывод может содержать символические ссылки.

Возвращается значение 0, за исключением случаев, когда не произошла ошибка во время чтения имени текущего каталога или, когда передан некорректный аргумент.

8.5.9 mkdir

По команде `mkdir` создается один или несколько каталогов, если они ранее не существовали.

Синтаксис:

```
mkdir [опции]... каталог...
```

Опции:

- `-m, --mode=РЕЖИМ` – установить права доступа для создаваемых каталогов;

- `-p, --parents` – перед созданием нового каталога предварительно создаются все несуществующие вышележащие каталоги. В случае существования каталога не будет выведена ошибка;
- `-v, --verbose` – выводить сообщение для каждого созданного каталога;
- `-Z, --context[=CTX]` – задать контекст SELinux для каждого создаваемого каталога. Если CTX не задан, то контекст будет равным типу по умолчанию.

Пример

Создать поддерево каталогов `tmpdir/temp/dir`:

```
mkdir -p tmpdir/temp/dir
```

8.5.10 ls

`ls` – выдача информации о файлах или каталогах. Команда `ls` для каждого имени каталога распечатывает список входящих в этот каталог файлов; для файлов – повторяется имя файла и выводится дополнительная информация в соответствии с указанными флагами. По умолчанию имена файлов выводятся в алфавитном порядке (если не заданы опции `-cftuvSUX` или `--sort`). Если имена не заданы, выдается содержимое текущего каталога.

Синтаксис:

```
ls [опции]... [файл]...
```

Существует три основных формата выдачи. По умолчанию выдается по одному файлу в строке; флаги `-C` и `-x` позволяют выдавать информацию в несколько колонок, а флаг `-m` задает свободный формат. Для определения формата вывода при указании флагов `-C`, `-x` и `-m` используется переменная окружения `COLUMNS`, значение которой равно количеству символов в выходной строке.

Опции:

- `-a, --all` – вывести список всех файлов (обычно не выводятся файлы, имена которых начинаются с точки);
- `-A, --almost-all` – вывести все файлы, кроме «.» и «..»;
- `-b, --escape` – заменять неграфические символы в именах файлов, используя алфавитно-цифровые последовательности, а также последовательности вида `-Q`, за исключением того, что имена файлов не заключаются в двойные кавычки;
- `--block-size=РАЗМЕР` – выдает размеры в блоках по РАЗМЕР байт. Например, `--block-size=M` для вывода объема в единицах равных 1048576 байтов;
- `-B, --ignore-backups` – не показывать файлы, заканчивающиеся на «~», если они не заданы в командной строке;
- `-c, --time=ctime, --time=status` – сортировать содержимое каталога в соответствии с временем изменения состояния файла. Если с помощью опции `-l` задан этот формат, то выдавать вре-

- мя изменения файла вместо времени его модификации (т.е. время создания файла, изменения режима доступа к нему и т.п.). С опцией `-t` показать время последней модификации описания файла и сортировать по имени;
- `-C, --format=vertical` – вывод в несколько колонок с сортировкой по вертикали;
 - `--color[=КОГДА]` – использовать цвета в выводе. КОГДА по умолчанию «always». Также можно использовать «never» и «auto».
 - `-d, --directory` – если аргумент является каталогом, то выводить только его имя, а не содержимое. Часто используется с флагом `-l` для получения сведений о состоянии каталога;
 - `-D, --dired` – генерировать вывод как в Emacs в режиме dired;
 - `-f` – не сортировать содержимое каталога. Выдавать файлы в том порядке, в котором они записаны на диск. Эта опция разрешает `-aU`, запрещает `-ls --color`;
 - `-F, --classify` – добавлять к названию индикатор (один из `*/=>@|`);
 - `--file-type` – тоже что и `--classify`, но без индикатора «*»;
 - `--format=СЛОВО` – тоже что и `-l --time-style=full-iso`;
 - `-g` – тоже что и `-l`, но не выводить владельца;
 - `--group-directories-first` – сгруппировать каталоги перед файлами. Сочетается с опцией `--sort`, но в сочетании с `--sort=none (-U)` группировка отключается;
 - `-G, --no-group` – тоже что и `-l`, но не показывать имя группы.
 - `-h, --human-readable` – в сочетании с `-l` показывает размеры в удобочитаемом формате (например, 1K 234M 2G);
 - `--si` – тоже что и `-h`, но используя возведение в степень 1000, а не 1024;
 - `-H, --dereference-command-line` – следовать по символьным ссылкам, переданным в командной строке;
 - `--dereference-command-line-symlink-to-dir` – следовать по всем символьным ссылкам, переданным в командной строке, если они указывают на каталог;
 - `--hide=ШАБЛОН` – не показывать записи, соответствующие ШАБЛОНУ командного интерпретатора (отменяется ключами `-A` и `-a`);
 - `--indicator-style=СТИЛЬ` – добавлять к именам элементов индикатор с указанным СТИЛЕМ: none (по умолчанию), slash (`-p`), file-type (`--file-type`) или classify (`-F`);
 - `-i, --inode` – показывать индекс каждого файла;
 - `-I, --ignore= ШАБЛОН` – не показывать записи, соответствующие ШАБЛОНУ командного интерпретатора;
 - `-k, --kibibytes` – использовать блоки по 1024 байта;
 - `-l` – вывод в длинном формате;

- -L, --dereference – при отображении информации для символической ссылки, отображать информацию для файла, на который она ссылается, а не о самой ссылке;
- -m – показать записи в список шириной в размер терминала, имена файлов разделяются запятыми;
- -n, --numeric-uid-gid – то же, что и -l, но идентификаторы владельца и группы выводятся в виде чисел, а не в виде имен;
- -N, --literal – печатать имена записей как есть (не обрабатываются, например, управляющие последовательности);
- -o – то же, что и -l, но идентификатор группы не выводится;
- -p, --indicator-style=slash – если файл является каталогом, то выдавать после его имени символ /;
- -q, --hide-control-chars – выдавать непечатные символы, входящие в имя файла, в виде символа ?;
- --show-control-chars – показывать непечатаемые символы как есть (по умолчанию для программы ls и для вывода на терминал);
- -Q, --quote-name – заключить имена в двойные кавычки;
- --quoting-style= СТИЛЬ – задать стиль кавычек: literal, locale, shell, shell-always, с или escape;
- -r, --reverse – изменить порядок сортировки на обратный;
- -R, --recursive – рекурсивно обойти встретившиеся подкаталоги;
- -s, --size – выдавать размер файлов в блоках;
- -S – отсортировать по размеру файлов, большие сначала;
- --sort=СЛОВО – сортировать по СЛОВУ, а не по имени: none (без сортировки) -U, extension (расширение) -X, size (размер) -S, time (время) -t или version (версия) -v;
- -t – файлы сортируются по времени последнего изменения (сначала идут самые новые файлы);
- -T, --tabsize=РАЗМЕР – использовать табуляцию указанного РАЗМЕРА, а не 8;
- -u – с опциями -lt: вместо времени последнего изменения использовать время последнего доступа для сортировки. С опцией -l показать время последнего доступа и сортировать по имени. В остальных случаях – сортировать по времени последнего доступа;
- -U – не сортировать, отображать записи в обычном порядке;
- -v – сортировать по номерам (версии) в текстовом представлении;
- -w, --width=ЧИСЛО – задать ширину экрана (в символах);
- -x – вывод в несколько колонок с сортировкой по строкам;

- `-Z, --context` – напечатать контекст SELinux для каждого файла;
- `-l` – отображать по одному файлу в строке;
- `--help` – показать справку и выйти;
- `--version` – показать информацию о версии и выйти.

РАЗМЕР – это целое число, за которым опционально следует единица изменения. Например, 10М соответствует $10 \cdot 1024 \cdot 1024$. Доступны следующие единицы измерения: К, М, G, Т, P, E, Z, Y (возведение в степень числа 1024), KB, MB и т.д. (возведение в степень числа 1000).

Выделение цветом для различных типов файлов отключено по умолчанию, а также отключается опцией `--color=never`. С `--color=auto` ls будет выводить коды цветов, только если стандартный выход осуществляется на терминал. Переменная окружения `LS_COLORS` может изменять настройки. Для её установки используйте команду `dircolors`.

Возвращаемое значение:

- 0 – всё хорошо;
- 1 – незначительные проблемы (например, не удалось получить доступ к подкаталогу);
- 2 – серьёзные проблемы (например, не удалось получить доступ к аргументу команды).

Режим доступа к файлу при указании флага `-l` выводится в виде 10 символов. Первый символ означает:

- `d` – файл является каталогом;
- `b` – файл является специальным блочным файлом;
- `c` – файл является специальным символьным файлом;
- `p` – файл является именованным каналом;
- `-` – обычный файл.

Остальные 9 символов делятся на три группы по три символа: права доступа владельца, других пользователей из его группы, всех прочих пользователей. Внутри каждой группы используются три символа, обозначающие права на чтение, запись и выполнение файла соответственно. Для каталога под правом на выполнение подразумевается право на просмотр в поисках требуемого файла.

Пример:

```
ls -l /util/by
-rwxr-xr-x 1 root sys 50 Jun 22 10:42 /util/by
```

Содержимое файла `/util/by` последний раз изменялось в 10 часов 42 минуты 22 января. Размер файла 50 байт. Владелец этого файла принадлежит группе `sys`, к тому же он является суперпользователем (входное имя - `root`). Следующее число, в данном случае 1, обозначает количество ссылок на файл `/util/by`. Последовательность минусов и букв указывает, что владелец, члены груп-

пы и прочие пользователи могут читать и выполнять файл, а владелец (и только он) имеет право писать в файл.

Права обозначаются следующим образом:

- r – право на чтение;
- w – право на запись;
- x – право на выполнение (поиск в каталоге);
- - – данное право доступа отсутствует;
- l – учёт блокировки доступа (бит переустановки идентификатора группы равен 1, бит права на выполнение членами группы равен 0). Располагается на месте права на выполнение для членов группы;
- s – право переустанавливать идентификатор группы или идентификатор владельца и право выполнения файла для членов группы или владельца;
- S – неопределенная комбинация бит: право переустанавливать идентификатор владельца есть, а право выполнения файла для владельца отсутствует;
- t – установлен бит навязчивости у файла, который могут выполнять прочие пользователи. Располагается на месте права на выполнение для прочих пользователей;
- T – бит навязчивости установлен, а права на выполнение у прочих пользователей нет. Располагается на месте права на выполнение для прочих пользователей.

Примеры

Если файл доступен владельцу для чтения, записи и выполнения, а членам группы и прочим пользователям только для чтения, он имеет режим:

```
-rwxr--r-
```

Файл доступен владельцу для чтения, записи и выполнения, а членам группы и прочим пользователям только для чтения и выполнения. Разрешена переустановка при выполнении идентификатора пользователя на идентификатор владельца файла:

```
-rwsr-xr-x
```

Файл доступен для чтения и записи только владельцу и членам группы; может быть блокирован при доступе:

```
-rw-rwl--
```

Вывести имена всех файлов в текущем каталоге, включая и те, которые начинаются с точки и обычно не выдаются:

```
ls -a
```

Вывести разнообразную информацию: список всех файлов, включая те, которые обычно не выводятся (a); номера описателей файлов будут выведены в левой колонке (l); размеры файлов (в блоках) выводятся во второй колонке (s); наконец, будут выданы числовые идентификаторы владельцев и групп (n):

```
ls -alsn
```

8.5.11 `cp`

Утилита `cp` предназначена для копирования файлов и каталогов.

Синтаксис:

```
cp [ОПЦИЯ]... [-T] ИСТОЧНИК НАЗНАЧЕНИЕ
cp [ОПЦИЯ]... ИСТОЧНИК... КАТАЛОГ
cp [ОПЦИЯ]... -t КАТАЛОГ ИСТОЧНИК...
```

Копирует **ИСТОЧНИК** в **НАЗНАЧЕНИЕ** или несколько **ИСТОЧНИКОВ** в **КАТАЛОГ**.

Опции:

- `-a`, `--archive` – тоже что и `-dR --preserve=all`, по возможности сохраняет структуру и атрибуты исходных файлов при копировании (но не сохраняет структуру каталогов);
- `--attributes-only` – не копировать данные файла, только атрибуты;
- `--backup[=CONTROL]` – сделать резервную копию каждого целевого файла;
- `-b` – тоже что и `--backup`, но не принимает аргументы;
- `--copy-contents` – копировать содержимое специальных файлов при рекурсивном копировании;
- `-d` – тоже что и `--no-dereference --preserve=links`, копирует символичные ссылки как ссылки, а не файлы на которые они указывают, и сохраняет жесткие ссылки между исходными файлами;
- `-f`, `--force` – если невозможно открыть существующий файл, то удалить его и попробовать ещё раз (данная опция игнорируется, если используется совместно с `-n`);
- `-i`, `--interactive` – спросить перед перезаписью (отменяет ранее указанный ключ `-n`);
- `-H` – следовать символьным ссылкам в **ИСТОЧНИКЕ**;
- `-l`, `--link` – создавать жёсткие ссылки вместо копирования;
- `-L`, `--dereference` – всегда следовать символьным ссылкам в **ИСТОЧНИКЕ**;
- `-n`, `--no-clobber` – не перезаписывать существующие файлы (отменяет стоящую перед ней опцию `-i`);
- `-P`, `--no-dereference` – не следовать символическим ссылкам в **ИСТОЧНИКЕ**;
- `-p` – тоже что и `--preserve=mode,ownership,timestamps`;
- `--preserve[=СПИСОК_АТРИБУТОВ]` – сохранять указанные атрибуты: `mode` (права доступа), `ownership` (владелец и группа), `timestamps` (временные штампы). Доступны следующие дополнительные атрибуты: `context`, `links`, `xattr` и `all`;
- `--no-preserve=СПИСОК_АТРИБУТОВ` – не сохранять указанные атрибуты;
- `--parents` – формирует имя каждого копируемого файла путём добавления к имени каталога, в который осуществляется копирование, символа косой черты (`/`) и указанного полного имени исходного файла;
- `-R`, `-r`, `--recursive` – копировать каталоги рекурсивно;

- --reflink[=КОГДА] – контролировать клонирование и CoW-копирование;
- --remove-destination – удалить каждый существующий файл назначения перед попыткой копирования;
- --sparse=КОГДА – контролировать создание разреженных файлов;
- --strip-trailing-slashes – удалить все завершающие слэши в каждом ИСТОЧНИКЕ;
- -s, --symbolic-link – создать символическую ссылку вместо копирования;
- -S, --suffix=СУФФИКС – использовать для резервных копий заданный СУФФИКС;
- -t, --target-directory=КАТАЛОГ – копировать все аргументы ИСТОЧНИК в КАТАЛОГ;
- -T, --no-target-directory – обрабатывать НАЗНАЧЕНИЕ как обычный файл;
- -u, --update – копировать, только если файл ИСТОЧНИК новее, чем файл назначения или если файл назначения отсутствует;
- -v, --verbose – выводить имя каждого файла перед копированием;
- -x, --one-file-system – оставаться в пределах одной файловой системы;
- --help – показать справку и выйти;
- --version – показать информацию о версии и выйти.

По умолчанию разреженные файлы ИСТОЧНИКА распознаются при помощи грубой эвристической процедуры и соответственно создаются разреженные НАЗНАЧЕНИЯ. Такое же поведение задаётся опцией --sparse=auto. С ключом --sparse=always всегда создаётся разреженный файл НАЗНАЧЕНИЯ, вне зависимости от того, содержит ли ИСТОЧНИК достаточно длинные последовательности нулевых байтов. Использование --sparse=never запретит создание разреженных файлов.

Если указано --reflink[=always], выполняется облучённое копирование, при котором копируются только изменённые блоки данных. Если это невозможно, или если указано --reflink=auto, то копирование прерывается с ошибкой, а то копирование переключится обратно в обычный режим.

По умолчанию суффикс для резервных копий «~». Его можно переопределить при помощи опции --suffix или переменной окружения SIMPLE_BACKUP_SUFFIX. Способ контроля версий может быть задан через опцию --backup или через переменную окружения VERSION_CONTROL. Допустимые значения:

- none, off – никогда не создавать резервные копии (даже если задана опция --backup);
- numbered, t – создать нумерованные резервные копии;
- existing, nil – если существуют нумерованные резервные копии, то создавать нумерованные резервные копии, если нет, то создавать простые;
- simple, never – всегда создавать простые резервные копии.

Как исключение, `sr` создаёт резервную копию ИСТОЧНИКА, если заданы ключи `-f` и `-b`, а если ИСТОЧНИК совпадает с НАЗНАЧЕНИЕМ, `sr` создаёт резервную копию НАЗНАЧЕНИЯ.

8.5.12 mv

Утилита `mv` – перемещение (переименование) файлов.

Синтаксис:

```
mv [ОПЦИЯ]... [-T] ИСТОЧНИК НАЗНАЧЕНИЕ
mv [ОПЦИЯ]... ИСТОЧНИК... КАТАЛОГ
mv [ОПЦИЯ]... -t КАТАЛОГ ИСТОЧНИК...
```

Переименовать ИСТОЧНИК в НАЗНАЧЕНИЕ или переместить ИСТОЧНИК(и) в КАТАЛОГ.

Опции:

- `--backup[=КОНТРОЛЬ]` – создавать резервные копии файлов, которые будут перезаписаны или удалены;
- `-b` – тоже что и `--backup`, но не принимает аргументы;
- `-f`, `--force` – удаляет файлы назначения, если они существуют, не спрашивая об этом пользователя;
- `-i`, `--interactive` – просит подтверждения на замену существующего файла;
- `-n`, `--no-clobber` – не переписывать существующий файл. Если указано несколько опций `-i`, `-f` и `-n`, то действовать будет только последняя;
- `--strip-trailing-slashes` – удалить все завершающие слэши в каждом ИСТОЧНИКЕ;
- `-S`, `--suffix=СУФФИКС` – переопределить обычный суффикс для резервных копий;
- `-t`, `--target-directory=КАТАЛОГ` – переместить все аргументы ИСТОЧНИК в КАТАЛОГ;
- `-T`, `--no-target-directory` – обрабатывать НАЗНАЧЕНИЕ как обычный файл;
- `-u`, `--update` – перемещать только, если файл ИСТОЧНИК новее, чем файл назначения или если файл назначения отсутствует;
- `-v`, `--verbose` – выдавать имя каждого файла перед его переносом;
- `--help` – показать справку и выйти;
- `--version` – показать информацию о версии и выйти.

По умолчанию суффикс для резервных копий «~». Его можно переопределить при помощи опции `--suffix` или переменной окружения `SIMPLE_BACKUP_SUFFIX`. Способ контроля версий может быть задан через опцию `--backup` или через переменную окружения `VERSION_CONTROL`.

Допустимые значения:

- `none`, `off` – никогда не создавать резервные копии (даже если задана опция `--backup`);
- `numbered`, `t` – создать нумерованные резервные копии;
- `existing`, `nil` – если существуют нумерованные резервные копии, то создавать нумерованные резервные копии, если нет, то создавать простые;

- `simple, never` – всегда создавать простые резервные копии.

8.5.13 `rm`

Утилита `rm` предназначена для удаления файлов или каталогов. Удаляет каждый указанный файл. По умолчанию не удаляет каталоги.

Синтаксис:

```
rm [ОПЦИЯ]... ФАЙЛ...
```

Если заданы опции `-I` или `--interactive=once`, а также заданы более трёх файлов или опции `-r`, `-R` или `--recursive`, то `rm` запросит подтверждение для выполнения операции. Если не будет получен подтверждающий ответ, то не будет выполнена вся операция.

Если же файл не доступен для записки, стандартный ввод соответствует терминалу и не заданы опции `-f` или `--force`, или заданы опции `-i` или `--interactive=always`, то `rm` запросит подтверждение для удаления файла. Если ответ не подтверждающий, то файл будет пропущен.

Обратите внимание, что при достаточном уровне знаний и времени можно восстановить содержимое удалённого файла. Для того чтобы иметь гарантию невозможности данных следует использовать `shred`.

Опции:

- `-f, --force` – игнорировать несуществующие файлы и аргументы. Никогда не запрашивать подтверждение;
- `-i` – запрашивать подтверждение перед каждым удалением;
- `-I` – запрашивать подтверждение один раз перед удалением более трёх файлов или перед рекурсивным удалением. Данная опция менее назойлива по сравнению с опцией `-i`, но тем не менее даёт защиту от большинства ошибок;
- `--interactive[=КОГДА]` – запрашивать подтверждение в соответствии с КОГДА: `never`, `once` (`-I`) или `always` (`-i`). Если КОГДА пропущено, то подразумевается `always`;
- `--one-file-system` – при рекурсивном удалении пропускать все каталоги, находящиеся на других по отношению к аргументам командной строки файловых системах;
- `--no-preserve-root` – позволяет удалить `«/»`;
- `--preserve-root` – не удалять `«/»` (по умолчанию);
- `-r, -R, --recursive` – рекурсивно удалять каталоги и их содержимое;
- `-d, --dir` – удалять пустые каталоги;
- `-v, --verbose` – пояснять производимые действия;
- `--help` – показать справку и выйти;
- `--version` – показать информацию о версии и выйти.

По умолчанию `rm` не удаляет каталоги. Используйте опцию `--recursive` (`-r` или `-R`) для того чтобы удалить все перечисленные каталоги вместе с их содержимым.

Для удаления файла, начинающегося на «-» (например: «-foo») используйте одну из следующих команд:

```
rm -- -foo
rm ./-foo
```

8.5.14 cat

cat – сцепляет содержимое файлов и печатает его на стандартный вывод.

Синтаксис:

```
cat [ОПЦИЯ]... [ФАЙЛ]...
```

Опции:

- -A, --show-all – тоже что и -vET;
- -b, --number-nonblank – нумеровать непустые выводимые строки (отменяет -n);
- -e – тоже что и -vE;
- -E, --show-ends – отображать символ «\$» в конце каждой строки;
- -n, --number – нумеровать выводимые строки;
- -s, --squeeze-blank – скрывать повторяющиеся пустые строки в выводе;
- -t – тоже что и -vT;
- -T, --show-tabs – отображать символ табуляции как ^I;
- -u – (игнорируется);
- -v, --show-nonprinting – использовать ^- и M-нотацию для всех непечатаемых символов кроме LFD (перевод строки и табуляция) и табуляции;
- --help – показать справку и выйти;
- --version – показать информацию о версии и выйти.

Если ФАЙЛ не задан или задан как «-», то читать из стандартного ввода.

Примеры

Вывести содержимое файла f, затем со стандартного ввода, затем – содержимое файла g:

```
cat f - g
```

Скопировать стандартный ввод на стандартный вывод:

```
cat
```

8.5.15 more

Утилита more – осуществляет постраничное отображение текста файла.

Синтаксис:

```
more [ОПЦИИ] ФАЙЛ [...]
```

Опции:

- -номер – задаёт число строк на экране;
- -d – печатать в конце каждого заполненного экрана сообщение "[Press space to continue, 'q' to quit.]" ("Нажмите пробел для продолжения, 'q' – для выхода"). В случае если нажата не-

корректная клавиша будет напечатано сообщение "[Press 'h' for instructions.]" ("Нажмите 'h' для отображения инструкций");

- -l – по умолчанию more обрабатывает ^L (прогон страницы) как специальный символ и останавливается после каждой строки с таким символом. Опция -l отключает это поведение;
- -f – считать логические строки, а не экранные строки (как будто длинные строки не были перенесены);
- -p – не прокручивать текст на экране. При этом экран будет очищен и на нём будет выведен новый текст;
- -c – не прокручивать текст на экране. При этом текст будет построчно заменён на новый;
- -s – сжать несколько пустых линий до одной;
- -u – скрыть подчёркивание;
- +/- – указать строку для поиска в файле перед его отображением;
- +номер – начать с заданного номера строки.

Опции more также они принимаются из переменной окружения MORE. Опции, указанные в командной строке, имеют больший приоритет.

Команды:

- h или ? – отобразить справку по интерактивным командам;
- ПРОБЕЛ – показать следующие k строк текста. По умолчанию – размер экрана;
- z – показать следующие k строк текста. По умолчанию – размер экрана. Аргумент становится новым значением по умолчанию;
- ВВОД – показать следующие k строк текста. По умолчанию – 1. Аргумент становится новым значением по умолчанию;
- d или ^D – прокрутить k строк. По умолчанию – 11. Аргумент становится новым значением по умолчанию;
- q или Q или ПРЕРЫВАНИЕ – выход;
- s – пропустить вперёд k строк. По умолчанию – 1;
- f – пропустить вперёд k экранов текста. По умолчанию – 1;
- b или ^B – прокрутить назад k экранов текста. По умолчанию – 1. Работает только с файлами, но не с каналами;
- ' – перейти в место, в котором был начал предыдущий поиск;
- = – показать текущий номер строки;
- /шаблон – искать k-тое вхождение регулярного выражения. По умолчанию – 1;
- n – искать k-тое вхождение последнего регулярного выражения. По умолчанию – 1;

- !команда или :!команда – выполнить команду оболочки;
- v – открыть текущий файл в редакторе (с текущей позиции). Редактор ищется в переменных окружения VISUAL и EDITOR (в этом порядке). Если обе переменные не заданы, то будет использован редактор vi;
- ^L – перерисовать экран;
- :n – перейти к k-тому следующему файлу. По умолчанию – 1;
- :P – перейти к k-тому предыдущему файлу. По умолчанию – 1;
- :f – показать имя файла и номер строки;
- . – повторить предыдущую команду.

Утилита more использует следующие переменные окружения:

- MORE – позволяет задать опции;
- SHELL – текущий интерпретатор команд (обычно задаётся самим интерпретатором при входе в систему);
- TERM – указывает тип терминала. Используется для определения символов управления экраном;
- VISUAL – указывает предпочитаемый редактор;
- EDITOR – указывает редактор, в случае если не задана переменная окружения VISUAL.

8.5.16 ln

ln – создание ссылок между файлами.

Синтаксис:

```
ln [ОПЦИЯ]... [-T] НАЗНАЧЕНИЕ ИМЯ_ССЫЛКИ (1ая форма)
ln [ОПЦИЯ]... НАЗНАЧЕНИЕ (2ая форма)
ln [ОПЦИЯ]... НАЗНАЧЕНИЕ... КАТАЛОГ (3ья форма)
ln [ОПЦИЯ]... -t КАТАЛОГ НАЗНАЧЕНИЕ... (4ая форма)
```

В первой форме создаёт ссылку на НАЗНАЧЕНИЕ с именем ИМЯ_ССЫЛКИ. Во второй форме создаёт ссылка на НАЗНАЧЕНИЕ в текущем каталоге. В третьей и четвёртой формах создаёт ссылки для каждого НАЗНАЧЕНИЯ в заданном КАТАЛОГЕ. По умолчанию создаются жёсткие ссылки, а с ключом --symbolic – символьные. Также по умолчанию предполагается, что файл назначения не существует. При создании жёстких ссылок НАЗНАЧЕНИЕ должно существовать. Символьные ссылки могут содержать произвольный текст. Относительные ссылки интерпретируются относительно их родительского каталога.

Опции:

- --backup[=КОНТРОЛЬ] – создавать резервные копии файлов, которые будут перезаписаны или удалены;
- -b – тоже что и --backup, но не принимает аргументы;

- -d, -F, --directory – позволить суперпользователю создавать жёсткие ссылки на каталоги (вероятно, это не получится из-а системных ограничений, даже для суперпользователя);
- -f, --force – удаляет файлы назначения, если они существуют, не спрашивая об этом пользователя;
- -i, --interactive – просит подтверждения на замену существующего файла;
- -L, --logical – разыменовывать НАЗНАЧЕНИЯ, если это символьные ссылки;
- -n, --no-dereference – рассматривать ИМЯ_ССЫЛКИ, как обычный файл, если это символьная ссылка на каталог;
- -P, --physical – создавать жёсткие ссылки непосредственно на символьные ссылки;
- -r, --relative – создавать символьные ссылки относительно положения ссылки;
- -s, --symbolic – создавать символьные ссылки, вместо жёстких ссылок;
- -S, --suffix=СУФФИКС – переопределить обычный суффикс для резервных копий;
- -t, --target-directory=КАТАЛОГ – задать каталог, в котором создавать ссылки;
- -T, --no-target-directory – всегда обрабатывать ИМЯ_ССЫЛКИ как обычный файл;
- -v, --verbose – печатать имя каждого файла при создании ссылки;
- --help – показать справку и выйти;
- --version – показать информацию о версии и выйти.

По умолчанию суффикс для резервных копий «~». Его можно переопределить при помощи опции --suffix или переменной окружения SIMPLE_BACKUP_SUFFIX. Способ контроля версий может быть задан через опцию --backup или через переменную окружения VERSION_CONTROL. Допустимые значения:

- none, off – никогда не создавать резервные копии (даже если задана опция --backup);
- numbered, t – создать нумерованные резервные копии;
- existing, nil – если существуют нумерованные резервные копии, то создавать нумерованные резервные копии, если нет, то создавать простые;
- simple, never – всегда создавать простые резервные копии.

При использовании опции -s будут проигнорированы опции -L и -P. В противном случае будет принята во внимание только указанная последней опция командной строки.

8.5.17 file

file – определение типа файла.

Синтаксис:

```
file [-bchikLnNprsvz] [--mime-type] [--mime-encoding] [-f имя_файла] [-F разделитель]
[-m magicfiles] файл
file -C [-m magic_файл]
file [--help]
```

Опции:

- -b, --brief – не выводить имя файла в начале строки (краткий режим);
- -c, --checking-printout – выводит детальную распечатку разобранного magic-файла. Обычно используется в сочетании с опцией -m для целей отладки нового magic-файла;
- -C, --compile – создать файл magic.mgc содержащий обработанную версию содержимого magic-файла или каталога;
- -e, --exclude имя_теста – не выполнять именованный тест. Доступны следующие имена тестов:
 - arpture – проверки приложений типа EMX (только для EMX);
 - ascii – проверка различных типов ascii-файлов;
 - compress – не проверять сжатые файлы;
 - elf – не печатать подробности elf;
 - fortran – не проверять наличие специфичных для fortran признаков в ascii-файлах;
 - soft – не использовать magic-файлы;
 - tar – не проверять tar-файлы;
 - token – не искать известные символы в ascii-файлах;
 - troff – не искать специфичные для troff последовательности в ascii-файлах.
- -f, --files-from имя_файла – прочитать имена проверяемых файлов из файла имя_файла перед списком аргументов (по одному имени в строке). Должен быть указан как минимум один файл для проверки (либо в имя_файла, либо в аргументах команды). Список файлов может быть прочитан со стандартного ввода. Для этого следует указать «-» в качестве имени_файла;
- -F, --separator разделитель – задаёт разделитель между именем файла и результатом для этого файла. По умолчанию «:»;
- -h, --no-dereference – не разыменовывать символические ссылки (на тех файловых системах, которые поддерживают символические ссылки). Эта опция включается автоматически, если не задана переменная окружения POSIXLY_CORRECT;
- -i, --mime – выводить строку mime-типа вместо более традиционного и удобочитаемого представления. Например, вместо «ASCII text» будет выведено «text/plain charset=us-ascii». Для того чтобы этого добиться file меняет способ распознавания файлов для большинства текстовых файлов, каталогов и других типов файлов. При этом используется альтернативный magic-файл. Для работы этой опции file меняет способ распознавания файлов – он использует альтернативный magic-файл;
- --mime-type, --mime-encoding – тоже что и -i, но печатает только заданные элементы;
- -k, --keep-going – не прекращать поиск при первом совпадении. Для отделения последующих совпадений будет использоваться символ «\012»;

- `-L, --dereference` – разыменовывать символические ссылки (на системах, поддерживающих символические ссылки). Это поведение по умолчанию, если задана переменная окружения `POSIXLY_CORRECT`;
- `-m, --magic-file list` – задать альтернативное положение `magic`-файлов. Аргумент может быть списком с двоеточием в качестве разделителя элементов списка. Если в указанном каталоге будет найден скомпилированный `magic`-файл, то будет использован именно он;
- `-n, --no-buffer` – выводить на стандартный вывод после проверки каждого файла. Это опция имеет смысл, только если проверяется несколько файлов. Она может быть полезна в сочетании с программами, ожидающими типы файлов в канале;
- `-N, --no-pad` – не делать отступ рядом с именами файлов таким образом, чтобы они были выровнены на выводе;
- `-p, --preserve-date` – на системах, поддерживающих `utime` или `utimes` попытаться сохранить неизменным время доступа анализируемых файлов;
- `-r, --raw` – не переводить непечатаемые символы в вид `\ooo`. По умолчанию `file` переводит непечатаемые символы в их восьмеричное представление;
- `-s, --special-files` – обычно `file` пытается определить тип файла только для тех файлов, для которых `stat` сообщил, что это нормальный файл. Это предотвращает возникновение проблем при чтении специальных файлов. Задание данной опции заставит `file` также, читать специальные блочные и символьные файлы. Это может быть полезно для определения типа файловой системы на разделах (доступ к разделам предоставляется как к специальному блочному файлу). Также данная опция говорит `file` игнорировать размер файла, сообщённый `stat`, так как на некоторых системах для разделов он сообщается как ноль;
- `-v, --version` – напечатать версию и выйти;
- `-z, --uncompress` – попытаться проверить содержимое сжатых файлов;
- `-0, --print0` – выводить нулевой символ `«\0»` в конце имени файла. Может быть, удобно при использовании в сочетании с `cut`. Данная опция не затрагивает обычный разделитель, который также будет напечатан;
- `--help` – напечатать помощь и выйти.

Утилита `file` проверяет каждый аргумент, пытаясь определить тип файла. Существует три набора тестов, выполняемых в следующем порядке: проверки файловой системы, специальные проверки типа файла (`magic test`) и проверки языка. Первый удачно выполнившийся тест выведет тип файла.

Выводимый тип файла обычно содержит одно из следующих слов: `text`, `executable` или `data`. `text` – содержит печатаемые символы и некоторые управляющие символы. Скорее всего, данный файл безопасно просматривать в ASCII-терминале. `executable` – файл содержит результаты компи-

лирования программы в форме понимаемой ядром UNIX. data – остальное (данные обычно двоичные или непечатаемые). Исключением являются файлы определённого формата (например, архивы или core-файлы), для них содержание двоичных данных считается нормальным.

Проверки файловой системы основаны на проверке кода возврата системного вызова stat. Программа проверяет, не пустой ли файл и не является ли он специальным файлом (примером специального файла может быть сокет, символическая ссылка или именованный канал). Специальные файлы могут быть определены, если они описаны в системном заголовочном файле `<sys/stat.h>`.

Специальные проверки типа файла используются для определения наличия данных специального формата в файлах. Каноническим примером является двоичный выполняемый (скомпилированная программа) файл `a.out`. Его формат определён в `<elf.h>`, `<a.out.h>` и, возможно, в `<exec.h>`. Такие файлы содержат в определённом месте в начале файла «магический» идентификатор, сообщающий операционной системе, что данный файл является бинарным исполняемым. Такие магические идентификаторы являются дополнением к файлам с данными. Любой файл с некоторым идентификатором на малом расстоянии от начала файла может быть определён при помощи утилиты `file`. Информация для определения типа файла может быть прочитана из скомпилированного файла `/usr/share/file/magic.mgc` или из файлов в каталоге `/usr/share/file/magic` (если нет скомпилированного файла). Также, если `$HOME/.magic.mgc` или `$HOME/.magic` существует, то для магических проверок будет использоваться их содержимое. Если файл не совпадает ни с одной записью в `magic`-файлах, то будет осуществлена проверка на то, не является ли файл текстовым. Кодировки ASCII, ISO-8859-x, non-ISO 8-bit extended-ASCII, UTF-8-encoded Unicode, UTF-16-encoded Unicode и EBCDIC могут быть определены по различным диапазонам и последовательностям байтов, из которых состоит текст в каждой из кодировок.

Если файл проходит какие-то другие тесты, то его кодировка будет сообщена. Файлы ASCII, ISO-8859-x, UTF-8 и extended-ASCII определяются как «text» так как они могут быть нормально прочитаны почти в любом терминале. UTF-16 и EBCDIC определяются как «character data» так как, не смотря на то, что файлы содержат текст, текст требует перевода, перед тем как его можно будет прочитать. Кроме кодировки `file` пытается определить другие характеристики текстовых файлов. Если строки оканчиваются на «CR», «CRLF» или «NEL» (вместо стандартного для Linux «LF»), то это будет также сообщено. Также отдельно будет сообщено, если файл содержит встроенные `escape`-последовательности или содержит какие-то другие особенности.

После того как была определена кодировка файла `file` пытается также определить язык на котором он написан. Языковые проверки ищут специфичные строки в нескольких первых блоках файла. Например, ключевое слово «`br`» говорит о том, что, скорее всего это выходной файл `troff`, а

ключевое слово «struct» является индикатором программы на C. Такие тесты менее надёжны по сравнению с двумя предыдущими группами проверок, и они проводятся в последнюю очередь.

Если для файла не удалось определить кодировку, в которой он написан, то он будет обозначен как «data».

Файлы:

- /usr/share/file/magic.mgc – используемый по умолчанию скомпилированный magic-файл;
- /usr/share/file/magic – используемая по умолчанию каталог с magic-файлами.

В переменной окружения MAGIC можно указать используемый по умолчанию magic-файл. Если эта переменная задана, то file не будет пытаться открыть \$HOME/.magic. При необходимости file добавит «.mgc» к значению этой переменной. Переменная окружения POSIXLY_CORRECT определяет (если система поддерживает символические ссылки), будет ли file разыменовывать символичные ссылки. Если она задана, то file будет разыменовывать символичные ссылки, если не задана – не будет. Также это поведение контролируется опциями -L и -h.

8.5.18 chmod

chmod – изменяет права доступа к файлу.

Синтаксис:

```
chmod [ОПЦИЯ]... РЕЖИМ[,РЕЖИМ]... ФАЙЛ...
chmod [ОПЦИЯ]... ВОСЬМЕРИЧНЫЙ-РЕЖИМ ФАЙЛ...
chmod [ОПЦИЯ]... --reference=ИФАЙЛ ФАЙЛ...
```

Опции

- -c, --changes – тоже что и --verbose, но сообщено будет только о выполненных изменениях;
- -f, --silent, --quiet – не показывать большинство сообщений об ошибках;
- -v, --verbose – выводить диагностическую информацию для каждого файла;
- --no-preserve-root – не обрабатывать «/» специальным образом (по умолчанию);
- --preserve-root – не выполнять рекурсивные операции с «/»;
- --reference=ИФАЙЛ – использовать режим файла ИФАЙЛ;
- -R, --recursive – рекурсивно изменять файлы и каталоги;
- --help – показать справку и выйти;
- --version – показать информацию о версии и выйти.

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

Формат символьного режима следующий:

```
[ugoа...][[+=][разрешения...]]...
```

Здесь разрешения – это ноль или более букв из набора «rwxXst» или одна из букв из набора «ugo».

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

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

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

Числовой режим состоит из не более четырёх восьмиричных цифр (от нуля до семи), которые складываются из битовых масок 4, 2 и 1. Любые пропущенные разряды дополняются лидирующими нулями. Первая цифра выбирает установку идентификатора пользователя (setuid) (4) или идентификатора группы (setgid) (2) или sticky-бита (1). Вторая цифра задаёт права доступа для пользователя, владеющего данным файлом: чтение (4), запись (2) и выполнение (1). Третья – права доступа для группы (с теми же значениями, что и вторая цифра). Четвёртая – права доступа для остальных пользователей (с теми же значениями, что и вторая цифра).

chmod никогда не меняет права доступа на символьные ссылки, так как этого не умеет делать системный вызов chmod. Это не является проблемой, так как права доступа символьных ссылок никогда не используются. Однако, для каждой символьной ссылке, заданной в командной строке, chmod произведёт изменение прав доступа для файлов, на которые они ссылаются. При этом chmod игнорирует символьные ссылки, встречающиеся при рекурсивном обходе каталогов.

Примеры

Чтобы установить права, позволяющие владельцу читать и писать в файл, а членам группы и прочим пользователям только читать, надо сложить 0400, 0200, 0040 и 0004. Таким образом, команду можно записать двумя способами:

```
chmod 644 f1
chmod u=rw,go=r f1
```

Позволить всем выполнять файл f2:

```
chmod +x f2
```

Запретить удаление файла f3:

```
chmod +t f3
```

Дать всем права на чтение запись и выполнение, а также на переустановку идентификатора группы при выполнении файла f4:

```
chmod =rwx,g+s f4
```

```
chmod 2777 f4
```

8.5.19 chown

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

Синтаксис:

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

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

Примеры

Поменять владельца /u на пользователя test:

```
chown test /u
```

Поменять владельца и группу /u:

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

Поменять владельца /u и вложенных файлов на test:

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

8.5.20 umask

Утилита `umask` задает маску режима создания файла в текущей среде командного интерпретатора равной значению, задаваемому операндом режим. Эта маска влияет на начальное значение битов прав доступа всех создаваемых далее файлов.

Синтаксис:

```
umask [-p] [-S] [режим]
```

Пользовательской маске режима создания файлов присваивается указанное восьмеричное значение. Три восьмеричные цифры соответствуют правам на чтение/запись/выполнение для владельца, членов группы и прочих пользователей соответственно. Значение каждой заданной в маске цифры вычитается из соответствующей «цифры», определенной системой при создании файла. Например, `umask 022` удаляет права на запись для членов группы и прочих пользователей (у файлов, создававшихся с режимом `777`, он оказывается равным `755`; а режим `666` преобразуется в `644`).

Если маска не указана, выдается ее текущее значение.

Команда `umask` распознается и выполняется командным интерпретатором `bash`.

Команду `umask` целесообразно включить в пользовательский профайл; тогда она будет автоматически вызываться при входе в систему и установит нужный режим доступа к создаваемым файлам и каталогам.

8.5.21 chattr

`chattr` – изменяет атрибуты файлов файловой системы `ext2fs`.

Синтаксис:

`chattr [-Rvf] [+=aAcCdDeijsSTtu] [-v версия] файлы...`

Оператор '+' означает добавление выбранных атрибутов к существующим атрибутам; '-' означает их снятие; '=' означает определение только этих указанных атрибутов для файлов.

Символы 'ASacDdijsTtu' указывают на новые атрибуты для файлов: не обновлять время последнего доступа (*atime*) к файлу (*A*), синхронное обновление (*S*), только добавление к файлу (*a*), сжатый (*c*), синхронное обновление каталогов (*D*), не архивировать (*d*), неизменяемый (*i*), журналирование данных (*j*), безопасное удаление (*s*), вершина иерархии каталогов (*T*), нет tail-merging (*t*), неудаляемый (*u*).

Опции:

- **-R** – рекурсивно изменять атрибуты каталогов и их содержимого. Символические ссылки игнорируются;
- **-V** – выводит расширенную информацию и версию программы;
- **-f** – подавлять сообщения об ошибках;
- **-v версия** – установить номер версии/генерации файла.

Когда изменяется файл с атрибутом (*A*) время последнего доступа к нему не изменяется. Это позволяет избежать некоторого количества дисковых операций ввода/вывода на ноутбуках.

Файл с атрибутом (*a*) можно открыть для записи только в режиме добавления. Только суперпользователь или процесс с возможностью `CAP_LINUX_IMMUTABLE` может устанавливать и снимать этот атрибут.

Файл с атрибутом (*c*) автоматически сжимается на диске ядром. Чтение из такого файла возвращает несжатые данные. При записи в такой файл данные перед записью на диск сжимаются.

Когда изменяется каталог с атрибутом (*D*) изменения синхронно записываются на диск. Это эквивалентно опции монтирования `'dirsync'` примененной к подмножеству файлов.

Файл с атрибутом (*d*) не является кандидатом на архивирование при использовании команды `dump`.

Атрибут (*E*) используется экспериментальными сжимающими патчами для того, чтобы показать, что сжатый файл содержит ошибки сжатия. Он не может быть установлен или сброшен с помощью `chattr`, хотя его можно просмотреть с помощью `lsattr`.

Атрибут (*I*) используется кодом `htree` для того, чтобы показать, что каталог индексируется с использованием хэши-рующих деревьев. Он не может быть установлен или сброшен с помощью `chattr`, хотя его можно просмотреть с помощью `lsattr`.

Файл с атрибутом (*i*) не может быть изменен: он не может быть удален или переименован, к этому файлу не могут быть созданы ссылки, и никакие данные не могут быть записаны в этот файл. Только суперпользователь или процесс с возможностью `CAP_LINUX_IMMUTABLE` может устанавливать и снимать этот атрибут.

При записи в файл с атрибутом (j) все данные записываемые в такой файл записываются в журнале ext3, прежде чем они будут записаны непосредственно в файл, если файловая система смонтирована с опциями "data=ordered" или "data=writeback". Если файловая система смонтирована с опцией "data=journalled", то все данные журналируются, и этот атрибут не дает никакого эффекта. Только суперпользователь или процесс с возможностью CAP_SYS_RESOURCE может устанавливать или снимать этот атрибут.

Когда удаляется файл с атрибутом (s), все его блоки заполняются нулями.

Когда изменяется файл с атрибутом (S), все изменения синхронно записываются на диск; это эквивалентно опции монтирования 'sync' примененной к подмножеству файлов.

Каталог с атрибутом (T) будет поднята на вершину иерархии каталогов для целей Orlov block allocator.

У файла с атрибутом (t) в конце не будет partial block fragment соединенного с другими файлами (для тех файловых систем, которые поддерживают tail-merging). Это необходимо для приложений, таких как LILO, которые читают файловую систему напрямую, и которые не понимают tail-merged файлы.

Когда удаляется файл с атрибутом (u) его содержимое сохраняется. Это позволяет пользователю восстановить файл.

Атрибут (X) используется экспериментальными сжимающими патчами, чтобы показать, что исходное содержимое сжатых файлов доступно напрямую. В данное время он не может быть установлен или переустановлен с помощью chatter(1), но может быть показан с помощью lsattr.

Атрибут (Z) используется экспериментальными сжимающими патчами, чтобы показать, что сжатый файл не сохранен. Он не может быть установлен или переустановлен с помощью chatter, но может быть показан с помощью lsattr.

8.5.22 lsattr

lsattr – выдает список атрибутов файлов на Linux ext2fs.

Синтаксис:

```
lsattr [ -RVadv ] [ файлы... ]
```

Опции:

- -R – рекурсивно изменять атрибуты каталогов и их содержимого. Символические ссылки игнорируются;
- -V – выводит расширенную информацию и версию программы;
- -a – просматривает все файлы, в каталоге включая те, имена которых начинаются с '!';
- -d – отображает каталоги также как и файлы вместо того, чтобы просматривать их содержимое.
- v – просматривает номера версий/генераций файлов.

9 РАБОТА С НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫМИ КОМПОНЕНТАМИ

9.1 Командные оболочки (интерпретаторы)

Для управления ОС используются командные интерпретаторы (shell).

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

`bash` – самая распространённая оболочка под linux. Она ведёт историю команд и предоставляет возможность их редактирования.

`pdksh` – клон `korn shell`, хорошо известной оболочки в UNIX(™) системах.

Оболочкой по умолчанию является «Bash» (Bourne Again Shell) Чтобы проверить, какую оболочку Вы используете, наберите команду:

```
echo $SHELL
```

У каждой оболочки свой синтаксис. Мы рекомендуем Вам использовать Bash. Все примеры в дальнейшем построены с использованием этой оболочки.

9.1.1 Командная оболочка Bash

В `bash` имеется несколько приемов для работы со строкой команд. Например, используя клавиатуру, можно:

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

Можно использовать «;» для того, чтобы ввести несколько команд одной строкой. Клавиши «вверх» и «вниз», позволяют перемещаться по истории команд. Для того, чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, необходимо набрать:

```
Ctrl + R
```

Команды, присутствующие в истории, отображаются в списке пронумерованными. Для того чтобы запустить конкретную команду нужно набрать:

```
!номер команды
```

Для запуска последней из набранных команд:

```
!!
```

Иногда, имена программ и команд слишком длинны. Bash сам может завершать имена. Нажав клавишу [TAB], можно завершить имя команды, программы или каталога. Например, пред-

положим, что необходимо использовать программу декомпрессии `bunzip2`. Для этого нужно набрать:

```
bu
```

затем нажать [TAB]. Если ничего не происходит, то вероятно существует несколько возможных вариантов завершения команды.

Если нажать клавишу [TAB] еще раз, то выведется список имен, начинающихся с «bu».

Например:

```
$ bu buildhash builtin bunzip2
```

Для продолжения ввода наберите:

```
n
```

(`bunzip` – это единственное имя, третьей буквой которого является «n»), а затем нажмите клавишу табуляции. Оболочка дополнит имя и остается лишь нажать «Enter», чтобы запустить команду!

Заметим, что программу, вызываемую из командной строки, Bash ищет в каталогах, определяемых в системной переменной `PATH`. По умолчанию, в этот перечень каталогов не входит текущий каталог, обозначаемый «./» (точка слэш) (если только не был выбран один из двух самых слабых уровней защиты, см. об этом ниже). Поэтому, для запуска программы `prog` из текущего каталога, надо дать команду: `./prog`

9.1.2 Базовые команды оболочки Bash

Все команды, приведенные ниже, могут быть запущены в режиме консоли.

Для получения более подробной информации используйте команду `man`. Пример:

```
man ls
```

Команда `su` позволяет получить права администратора. Если набрать `su`, оболочка запросит пароль суперпользователя (`root`). После ввода пароля и нажатия `Enter`: Вы получите привилегии суперпользователя. Чтобы вернуться к правам Вашего пользовательского бюджета, наберите `exit`.

Команда `cd` позволяет сменить каталог. Она работает как с абсолютными, так и с относительными путями. Если находясь в своем домашнем каталоге Вам необходимо перейти в его подкаталог `docs/`, то можно ввести относительный путь:

```
cd docs/
```

Чтобы перейти в каталог `/usr/bin`, наберите (абсолютный путь):

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

Сделать текущим родительский каталог (обратите внимание на пробел между `cd` и `..`):

```
cd ..
```

Вернуться в предыдущий каталог:

```
cd -
```

Команда `cd`, без параметров переводит вас в ваш домашний каталог.

Команда `ls` (`list`) выдает список файлов в текущем каталоге. Две основные опции: `-a` – просмотр всех файлов, включая скрытые, `-l` – отображение более подробной информации.

Команда `rm` используется для удаления файлов.

Внимание: удалив файл, вы не сможете его восстановить!

Синтаксис:

```
rm имя_файла
```

У данной программы существует ряд параметров. Самые часто используемые: `-i` – запрос на удаление файла, `-r` – рекурсивное удаление (т.е. удаление, включая подкаталоги и скрытые файлы).

Пример:

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

Удаляет все файлы `html`, в вашем каталоге `html`.

Команда `mkdir` позволяет создать каталог, команда `rmdir` удаляет каталог, при условии, что он пуст.

Синтаксис:

```
mkdir имя_каталога
```

```
rmdir имя_каталога
```

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

Команда `less` позволяет постранично просматривать текст:

```
less имя_файла
```

Крайне полезно просмотреть файл, перед тем как его редактировать. Для выхода нажмите `q`.

Команда `grep` имеет много опций и предоставляет возможности поиска символьной строки в файле:

```
grep шаблон_поиска файл
```

Команда `ps` отображает список текущих процессов. Колонка команд указывает имя процесса, колонка `PID` (идентификатор процесса) – номер процесса (этот номер используется, для операций с процессом, например, чтобы «убить» его командой `kill`):

```
ps аргументы
```

Аргумент `u` предоставляет больше информации, а `x` позволяет просмотреть те процессы, которые не принадлежат вам (такие как те, что были запущены во время процесса загрузки).

Команда `kill` используется, чтобы завершить программу, которая перестала отвечать или зависла:

```
kill PID_номер
```

Иногда, необходимо будет использовать `kill -9 PID_номер` (когда обычная команда `kill` не дает желательного эффекта). Номер `PID` выясняется при помощи команды `ps`.

9.2 Текстовый редактор `vi`

Текстовый редактор `vi` имеет модальный интерфейс – одни и те же клавиши в разных режимах работы выполняют разные действия. В редакторе `Vi` есть несколько режимов:

- «Командный режим» – перемещение по файлу, удаление текста и другие редактирующие функции. По умолчанию, работа начинается в командном режиме. Перейти в него из любого другого режима <ESC>, иногда 2 раза;
- «Режим ввода» – ввод текста (удаление и ввод текста происходит в двух разных режимах). Переход в режим ввода из командного режима осуществляется командой “i”(от слова insert);
- «Режим строчного редактора ED» – это специальный режим, в котором редактору даются сложные команды. При вводе этих команд они отображаются в последней строке экрана. Например команда ”wq” позволяет записать файл и покинуть редактор Vi, а команда “q!” – выйти из редактора Vi без сохранения изменений. В этом режиме обычно вводятся команды, название которых состоит из нескольких символов. Переход в него из командного режима осуществляется командой ":".

Опишем операции, которые можно произвести с файлом в командном режиме.

9.2.1 Открыть/создать файл

Команда `vi filename.txt` – открывает один файл.

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

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

`:e filename`

Перед этим нужно сохранить предыдущий файл:

`:w` – сохраняет файл с существующим именем или

`:sav filename` – «Сохранить как».

9.2.2 Перемещение по файлу

Перемещение по файлу происходит с помощью стрелок (стрелки работают при правильном описании терминала, если проблема описания у вас еще не решена – используйте h,j,k,l). Также можно использовать быстрые клавиши перехода:

- ^ или 0 – в начало текущей строки;
- \$ – в конец текущей строки;
- w – на слово вправо;
- b – на слово влево.

9.2.3 Редактирование файла

Для редактирования текста необходимо перейти в режим ввода.

Основные команды редактирования:

- R,i – переход в режим ввода – замена текста под курсором;

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

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

9.2.4 Сокращения

<ESC> – нажатие клавишу Escape (или Ctrl-[]).

<CR> – ... Enter.

<SHIFT> – ... Shift.

^x – ... Ctrl-x.

9.2.5 Запись/выход

<ESC>:w<CR> – записать файл.

<ESC>:w!<CR> – записать файл.

Эта команда может помочь, если файл заблокирован другим пользователем, либо отсутствуют такие привилегии. При попытке записи без "!" будет выдано соответствующее предупреждение.

<ESC>:w new_file<CR>.

Создать новый файл "new_file" и записать в него текущее содержимое. Если файл существует, будет показано предупреждение. Далее вы продолжаете работать со старым файлом – filename.txt

<ECS>:q<CR> – выйти из редактора.

Если файл был изменен, то выйти из редактора не получится. В таких случаях необходимо добавлять после команды "!":

<ECS>:q!<CR> – выйти из редактора не сохраняя изменения.

<ESC>:wq<CR> или <ESC>ZZ<CR> – записать файл и выйти.

9.2.6 Дополнительные возможности

Есть несколько команд, с которыми редактирование становится проще:

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

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

- :set fileformat=dos;
- :set fileformat=unix.

Задать размер табуляции в 4 пробела:

:set ts=4.

9.3 Редактор ViM

9.3.1 Режимы работы

В ViM существует 3 режима работы:

- основной – предназначен, для просмотра файла, ввода команд и перехода из него в другие режимы. Из любого режима в командный можно попасть по нажатию (иногда два раза) <ESC>. При нажатии на ":" становится доступна командная строка ViM в которой можно вводить команды. Например: команда выхода quit (или q); команда сохранения write (или w), параметром которой может быть имя файла; вызов справки help (или h). На остальные клавиши (и их последовательности) можно назначить любые команды, либо использовать значения по умолчанию;
- визуальный – предназначен в первую очередь для выделения блоков текста. Есть 3 варианта перехода в этот режим – v для посимвольного выбора, <Shift>+v для построчного и <Ctrl>+v для блочного. В нормальном режиме (при переходе по "v") можно оперировать следующими сущностями: слово ("w"), предложение ("s"), параграф ("p") и блок ("b"). Вы-

деление при этом начинается с позиции курсора ("a"), или же с начала блока ("i"). Например, выделение текущего блока (участка, ограниченного парными элементами) можно произвести следующим образом <Esc>vib. Копирование в буфер выделенного текста осуществляется по "y", вырезание по "d" а вставка соответственно "p";

режим редактирования – переход на него осуществляется, к примеру, нажатием <Ins>.

9.3.2 Основные возможности

Все возможности и команды редактора ViM перечислить весьма затруднительно (ибо HTML-документация по нему занимает около пяти мегабайт), но на группе наиболее полезных остановиться необходимо. Перечисленные ниже команды вводятся в основном режиме (если нет специального уточнения). Все они имеют команднорочные аналоги и могут быть легко переопределены.

9.3.2.1 Переходы

Команда G используется для перехода на строку с номером n. Так, для перехода к началу текста необходимо набрать 1G, для сотой строки 100G, а для конца - \$G. Для перехода на n символов в нужную сторону можно использовать клавиши со стрелками. То есть для перехода на 1000 символов вниз нужно набрать 1000 и нажать стрелку вниз.

Для перемещения по тексту можно использовать следующие команды: "(, ")" для перемещения по предложениям, "{,}" для параграфов, "[[,"]]" для функций, "% – переход к парной скобке, "" – к предыдущему положению, а "<CTRL>-O, <CTRL>-I" – соответственно назад и вперед по истории переходов.

9.3.2.2 Метки

Метки используются для отметки позиции (метка, где меткой является любая буква) и быстрого к ней перехода (^метка). Метки нижнего регистра действительны в пределах данного файла, метки же верхнего регистра действуют во всех открытых файлах. Список всех меток можно получить командой marks.

9.3.2.3 Регистры

Доступно множество именованных регистров (хранилищ данных, буферов). Регистр отмечается ""буква". К нему применимы все стандартные действия – копирование в него ("меткау), вырезание ("меткаd), и вставка из него ("меткаp, можете вместо p использовать [r,]r для вставки соответственно перед, или после курсора). В режиме редактирования вставка из регистра осуществляется по "<Ctrl>+R метка". Для добавления данных в регистр необходимо использовать заглавную метку.

Также можно писать в регистр, воспользовавшись командой "qметка" и завершая запись по "q". Таким образом, сохраняется макрос, выполнить который можно по "@метка".

Регистры с метками "*" и "+" совпадают с X-Window clipboards, "%" – соответствует редактируемому файлу. Для просмотра содержимого всех регистров можно воспользоваться командой :registers, либо :reg метка1метка2... для просмотра некоторых.

9.3.2.4 Фолды

Фолды предназначены для сокрытия не нужных в данный момент данных, дабы те не отвлекали внимания. Например, кода подпрограммы с которой вы в данный момент не работаете. По умолчанию фолды активированы в режиме их ручной расстановки. Если нужно их авто активировать по отношению к табуляции, то необходимо добавить в конфиг строку set foldmethod=indent. Все команды для работы с фолдами начинаются с "z". Открытие фолда производится, например, по zo (или стрелке вправо) на нем, закрытие кода в фолд – по zs.

9.3.2.5 Сессии

При ведении группы проектов нередко желательно сохранить текущие состояния и настройки редактора, чтобы в дальнейшем продолжить работу с того же места. Для этого и предназначены сессии, которые создаются командой :mksession /path/to/Session.vim, а читаются командой :so /path/to/Session.vim. Гораздо чаще возникает потребность в сохранении не всей сессии, а только текущего контекста (во что входит, например, положение курсора в коде, текущая расстановка фолдов и много другое). Это действие выполняет команда :mkview, чтение – :loadview. Очень удобно сделать сохранение и чтение контекста автоматическим при начале и окончании редактирования файла. Это может быть реализовано следующим кодом (применяется для всех файлов, имеющих точку в имени):

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

9.3.2.6 Поиск и замена

Поиск осуществляется командами "/" для поиска (по регулярному выражению) вперед, и "?/" в обратном направлении. Для продолжения поиска необходимо использовать "n", а для прошлого варианта "N". Для поиска слова под курсором используются соответственно "#/" и "*#".

Для поиска с заменой используется :%s/что/на что/gic, где "%" означает работу со всем текстом (а не с текущей строкой), "g" – глобальная замена (а не первое совпадение), "i" – игнорирование регистра, а "c" – подтверждение каждого действия.

9.3.2.7 Автодополнение, Отмена, Смена регистра, Повтор

Автодополнение производится по содержимому данного файла, а также указанных в переменной dictionary по нажатию клавиш "<Ctrl>+n" и "<Ctrl>+p".

Клавиша "u" используется отмены изменения.

Клавиша "~" используется для смены регистра выделенного участка (или буквы под курсором) на противоположный регистр (строчные в прописные и прописные в строчные), "U" – принудительно установит верхний регистр, а "u" соответственно нижний.

Клавиша ”.” используется для повтора последней команды, включая ввод текста.

9.3.3 Конфигурация

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

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

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

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

В Vim присутствует подробная документация по настройкам -:options.

9.4 xinetd

Xinetd запускает процессы, которые предоставляют различные сервисы интернет. В отличие от сервисов, которые стартуют во время инициализации системы и пребывают в бездействии в ожидании запросов, xinetd представляет собой только один процесс, слушающий на всех портах сервисов, перечисленных в файле конфигурации xinetd.conf. Когда приходит запрос xinetd запускает соответствующий сервер. По причине такой работы xinetd называют еще суперсервером.

Сервисы, перечисленные в конфигурационном файле xinetd можно разделить на две группы. Сервисы из первой группы называются multi-threaded (многопоточными), и на каждый новый запрос запускается новый серверный процесс. Для таких сервисов xinetd продолжает слушать сеть, на соответствующем порту ожидая новых запросов и готов породить новый процесс. В другую группу включаются сервисы службы, которые в состоянии обрабатывать новые соединения. Такие сервисы называются single-threaded (однопоточными), для таких сервисов xinetd прекращает обработку новых запросов до тех пор, пока серверный процесс не завершит свою работу. Сервисы в этой группе обычно основаны на передаче данных через датаграммы (datagram-based, UDP).

Таким образом, причиной существования суперсервера является факт сохранения системных ресурсов за счет не запуска множества серверных процессов, которые возможно будут без-

действовать большую часть своей жизни. Полностью соответствуя назначению запускать требуемые сервисы, xinetd осуществляет так же функции контроля доступа и регистрации событий. Кроме того, xinetd не ограничен сервисами, перечисленными в файле /etc/services. Можно использовать xinetd для запуска сервисов специального назначения.

Синтаксис:

```
xinetd [опции]
```

Параметры:

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

- `-logprocs limit` – устанавливает ограничение на количество одновременно запущенных серверов на один идентификатор удаленного пользователя;
- `-shutdownprocs limit` – устанавливает ограничение на количество одновременно запущенных серверов для завершения работы службы;
- `-version` – вывести информацию о версии `xinetd`;
- `-ss interval` – `xinetd` будет выполнять периодические проверки своего внутреннего состояния каждые `interval` секунд.

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

Файлы `xinetd`:

- `/etc/xinetd.conf` – стандартный конфигурационный файл;
- `/var/run/xinetd.dump` – стандартный файл дампа.

Управление `xinetd`

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

Сигналы:

- `SIGUSR2` – заставляет выполнить жесткую перенастройку, которая означает, что `xinetd` перечитает файл с настройками и завершит работу серверов для тех служб, которые больше не доступны. Управление доступом выполняется снова на уже запущенные сервера через проверку удаленных подключений, времени доступа и копий серверов. Если количество копий серверов уменьшается, то некоторые произвольно выбранные сервера будут убиты, чтобы соблюсти ограничение; это случится после завершения работы тех серверов, которые попадают под ограничение доступа с удаленных адресов или ограничение времени доступа. Также, если флаг `INTERCEPT` был сброшен и происходит его установка, то будет завершена работа любых запущенных серверов для служб с этим флагом. Цель такого поведения – убедиться, что после жесткой перенастройки не будет запущено серверов, которые могут принимать пакеты с тех адресов, которые не соответствуют критериями управления доступом;
- `SIGQUIT` – приводит к завершению работы;
- `SIGTERM` – завершает работу всех запущенных серверов перед завершением работы `xinetd`;

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

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

9.5 Crontab

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

Синтаксис:

```
crontab [имя_файла]
crontab [ -e|l|r ] имя_пользователя
```

Опции:

-e – редактирует копию файла `crontab` текущего пользователя или создает пустой файл для редактирования, если соответствующего файла `crontab` не существует. Когда редактирование завершается, файл устанавливается в качестве пользовательского файла `crontab`. Переменная среды `EDITOR` задает редактор, вызываемый при указании опции -e. Все задания в файле `crontab` должны создаваться с помощью утилиты `crontab`;

-l – отображает текущий файл `crontab` на стандартный вывод;

-r – удаляет текущий файл `crontab`.

9.5.1 Контроль доступа к crontab

Доступ пользователя к `crontab` разрешен, если:

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

Доступ пользователя к `crontab` не разрешен, если:

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

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

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

9.5.2 Формат записи файла crontab

Для редактирования crontab вашего пользователя, используйте команду:

```
crontab -e
```

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

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

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

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

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

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

- @hourly – выполнять команду раз в час, "0 * * * *".

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

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

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

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

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

9.5.3 Примеры

Пример 1

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

```
crontab: installing new crontab
```

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

Пример 2

```
# использовать для запуска команд /bin/sh
# не обращая внимание на то, что написано в /etc/passwd
SHELL=/bin/sh
# отправлять вывод выполнения команд по электронной почте пользователю `paul`
# не обращая внимания на то, чей это crontab
MAILTO=paul
#
```

```
# запускать пять минут пополуночи, каждый день
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# запускать в 14:15 первого числа каждого месяца
15 14 1 * * $HOME/bin/monthly
# запускать в 22.00 каждый рабочий день
0 22 * * 1-5 mail -s "Уже 10 вечера"
23 0-23/2 * * * echo "запуск в 00:23, 2:23, 4:23 ..., каждый день"
5 4 * * sun echo "запуск в 4:05 каждое воскресенье"
```

10 ОБЩИЕ ПРАВИЛА ЭКСПЛУАТАЦИИ

10.1 Включение компьютера

Для включения компьютера необходимо:

- включить стабилизатор напряжения, если компьютер подключен через стабилизатор напряжения;
- включить принтер, если он нужен;
- включить монитор компьютера, если он не подключен к системному блоку кабелем питания;
- включить компьютер (переключателем на корпусе компьютера либо клавишей с клавиатуры).

После этого на экране компьютера появятся сообщения о ходе работы программ проверки и начальной загрузки компьютера.

10.2 Выключение компьютера

Для выключения компьютера надо:

- закончить работающие программы;
- выбрать функцию завершения работы и выключения компьютера, после чего ОС самостоятельно выключит компьютер, имеющий системный блок формата АТХ;
- выключить компьютер (переключателем на корпусе АТ системного блока);
- выключить принтер;
- выключить монитор компьютера (если питание монитора не от системного блока);
- выключить стабилизатор, если компьютер подключен через стабилизатор напряжения.

11 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Способы обращения в службу технической поддержки (доступность зависит от приобретенного вами уровня техподдержки) указаны на странице <https://www.basealt.ru/sales/support/>.

По вопросу приобретения технической поддержки на дистрибутив обращайтесь по электронной почте: sales@basealt.ru