ОПЕРАЦИОННАЯ СИСТЕМА БАЗАЛЬТ РАБОЧАЯ СТАНЦИЯ 8.0

Описание функциональных характеристик

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ ОБ ОС БАЗАЛЬТ РАБОЧАЯ СТАНЦИЯ 8.0

1.1 Краткое описание возможностей

Операционная система Базальт Рабочая станция 8.0 (далее – ОС Базальт Рабочая станция 8.0), представляет собой совокупность интегрированных программныха продуктов, созданных на основе операционной системы Linux. ОС Базальт Рабочая станция 8.0 универсальный и многофункциональный дистрибутив.

Дистрибутив предоставляет интегрированную операционную систему на единой оптимизированной пакетной базе с поддержкой различных аппаратных платформ, с возможностью установки графического окружения.

Базальт Рабочая станция 8.0 – это комплекс необходимых программ для эффективного выполнения типовых задач: электронная почта, работа с документами и презентациями, прослушивание аудиофайлов и просмотр видео, работа в сети Интернет и многое другое.

Основные преимущества:

- графическая рабочая среда MATE;
- выбор разворачиваемых решений (например, виртуализация, мультимедиа приложения) на этапе установки;
- возможность как развернуть, так и использовать только определённые службы без Alterator;
- широкий выбор различных программ для профессиональной и домашней работы в сети Интернет, с документами, со сложной графикой и анимацией, для обработки звука и видео, разработки программного обеспечения и образования.

ОС Базальт Рабочая станция 8.0 обеспечивает обработку, хранение и передачу информации в круглосуточном режиме эксплуатации.

1.2 Структура программных средств

Базальт Рабочая станция 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 Базальт Рабочая станция 8.0 Кентавр 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;
- 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 Базальт Рабочая станция 8.0 доступны несколько вариантов загрузки. Среди которых есть, например, вариант с recovery mode – загрузка с минимальным количеством драйверов, что может оказаться необходимым в случае неполадок. Дополнительно предлагается вариант загрузки в программу проверки оперативной памяти (memtest).

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 быть 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 шифрованный раздел, вам потребуется вводить пароль при обращении к этому разделу (рис.1).

Получение доступа к зашифрованным разделам

Рис.1.

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

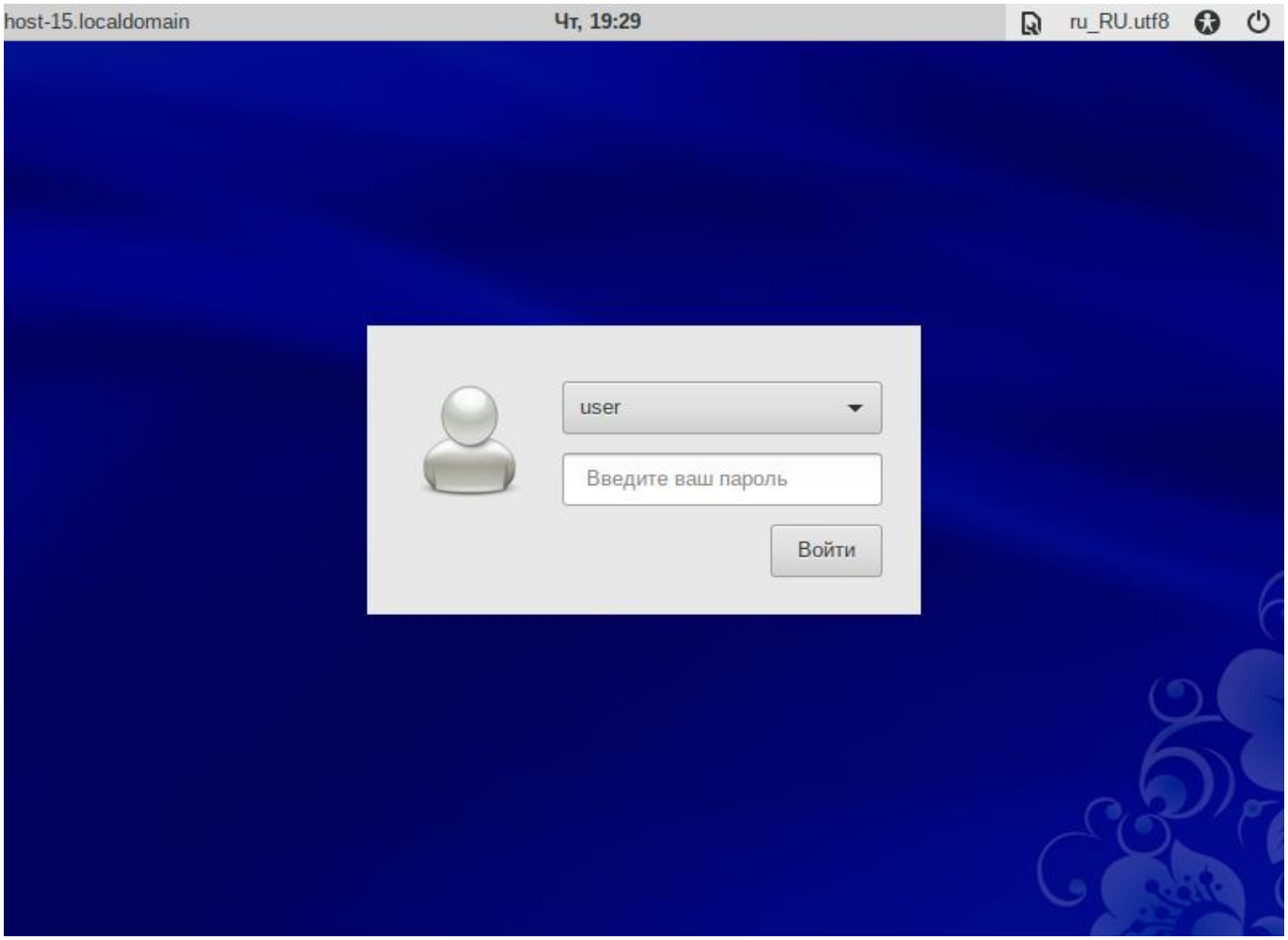


Рис.2.

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 MATE

На рабочем столе MATE 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

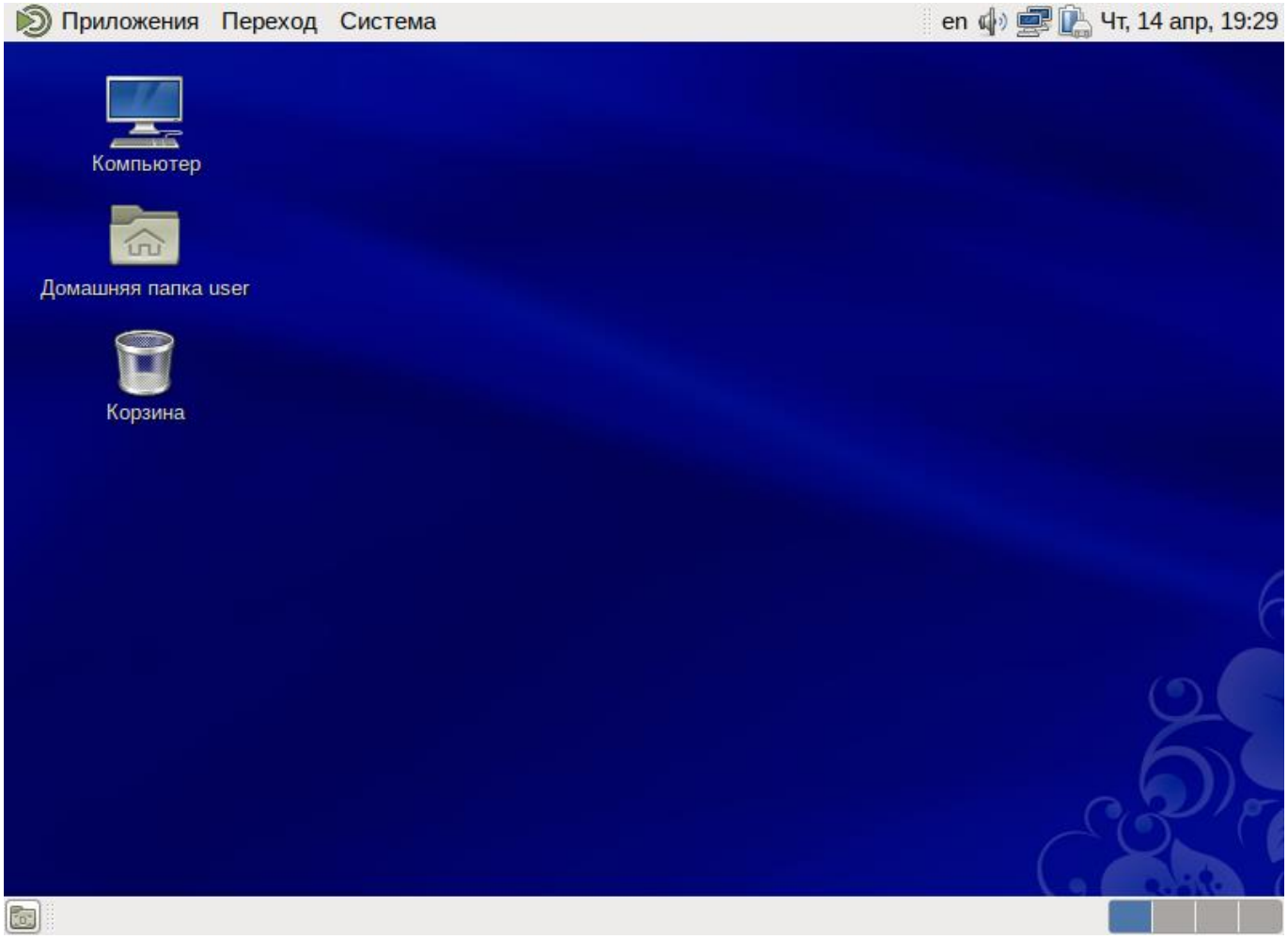


Рис.3

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 Caja. Для вызова руководства Caja 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 Caja.

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 MATE, 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 MATE» показывает информацию об установленной среде MATE.

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 **MATE** 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 ДЛЯ РАБОЧЕЙ СТАНЦИИ

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:

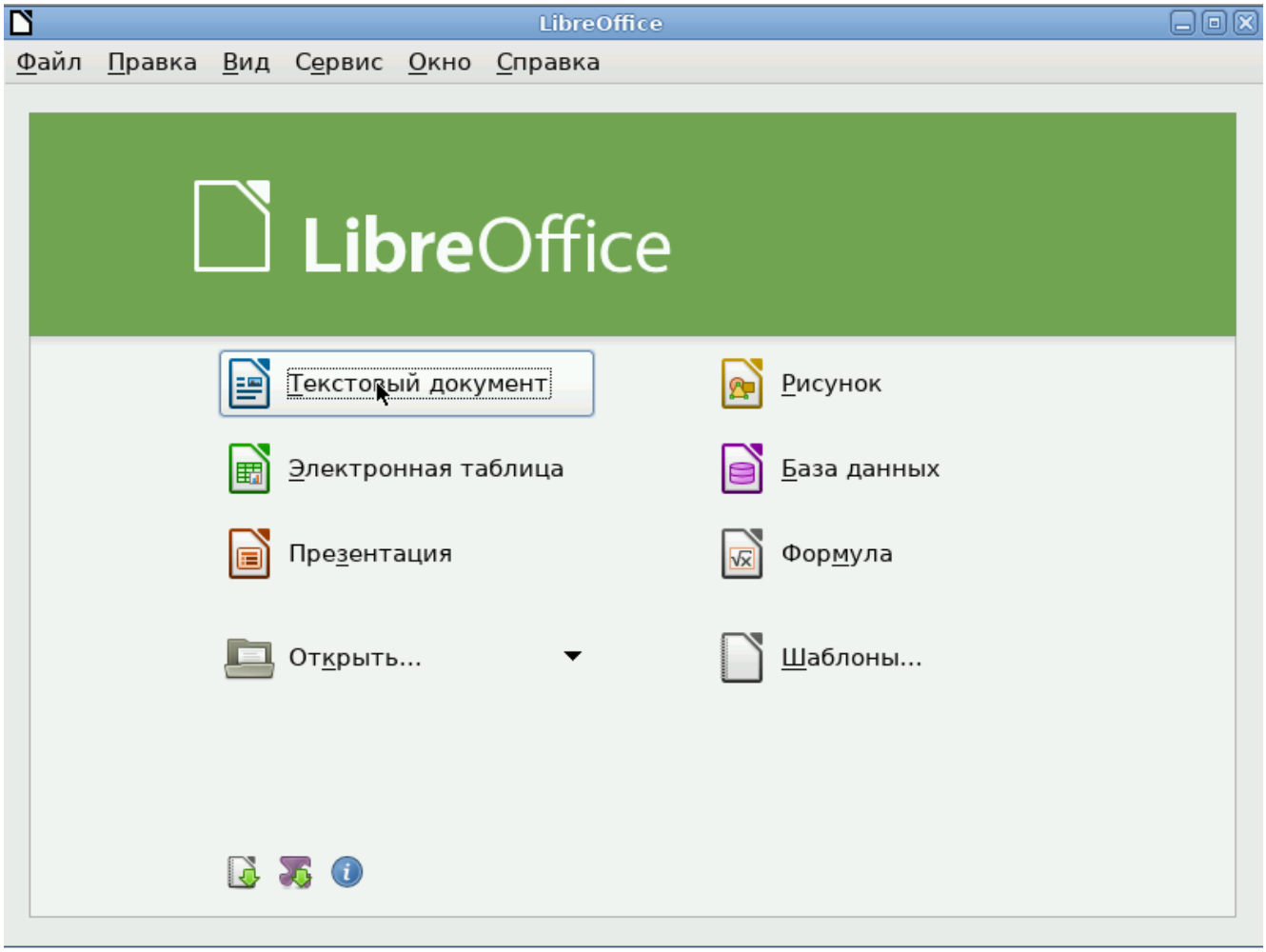
- интеллектуальная строка ввода адреса предоставляет окно-подсказку с историей ранее открытых сайтов;
- строка поиска предлагает пользователю функцию поиска по мере набора текста.

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 Офисные приложения

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

3.2.1 LibreOffice



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

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 презентацииMicrosoft

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.**

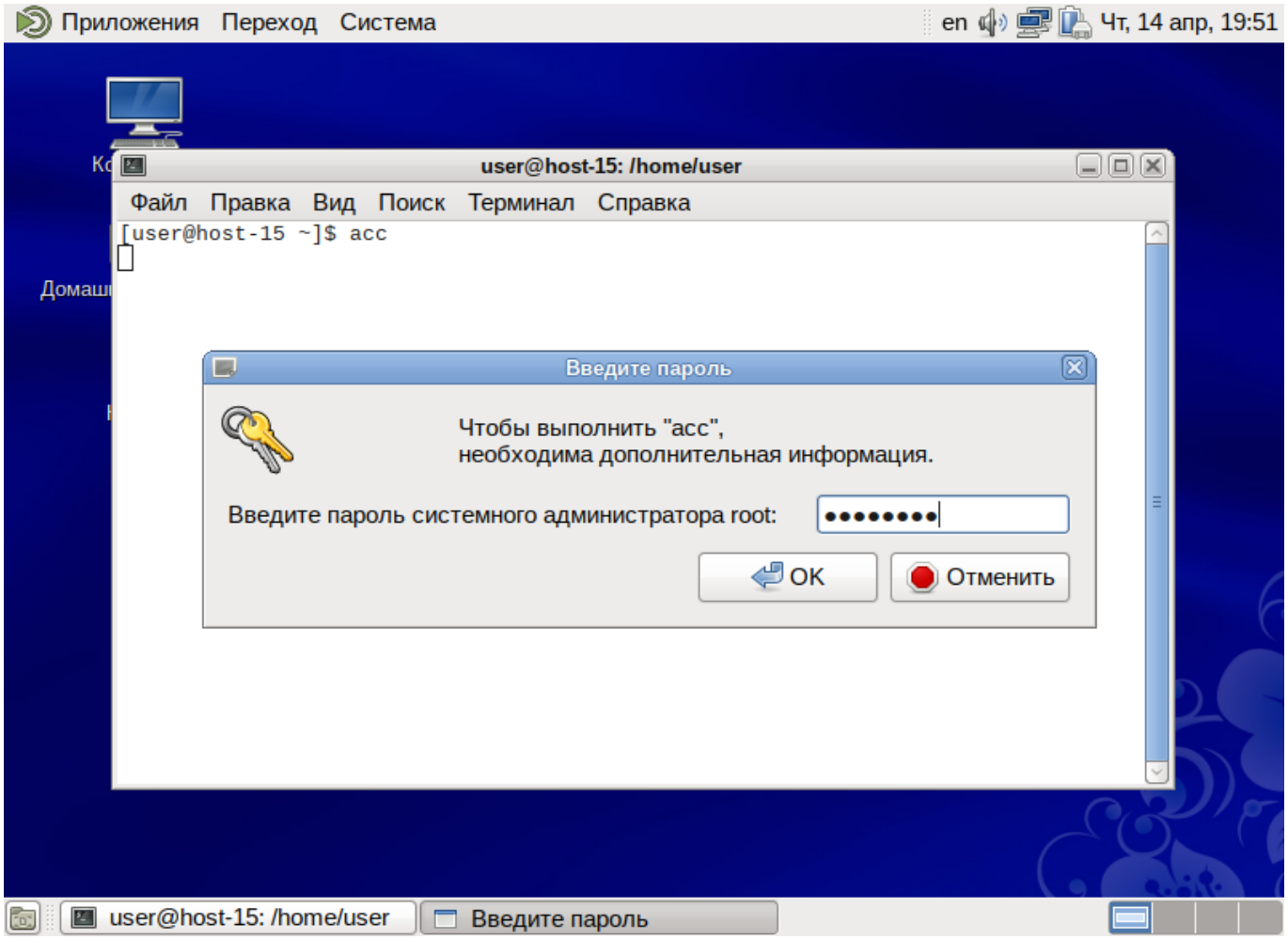
Вы всегда можете воспользоваться кнопкой **Справка**. Все модули центра управления системой имеют справочную информацию.

4.3 Запуск Центра управления системой в графической среде

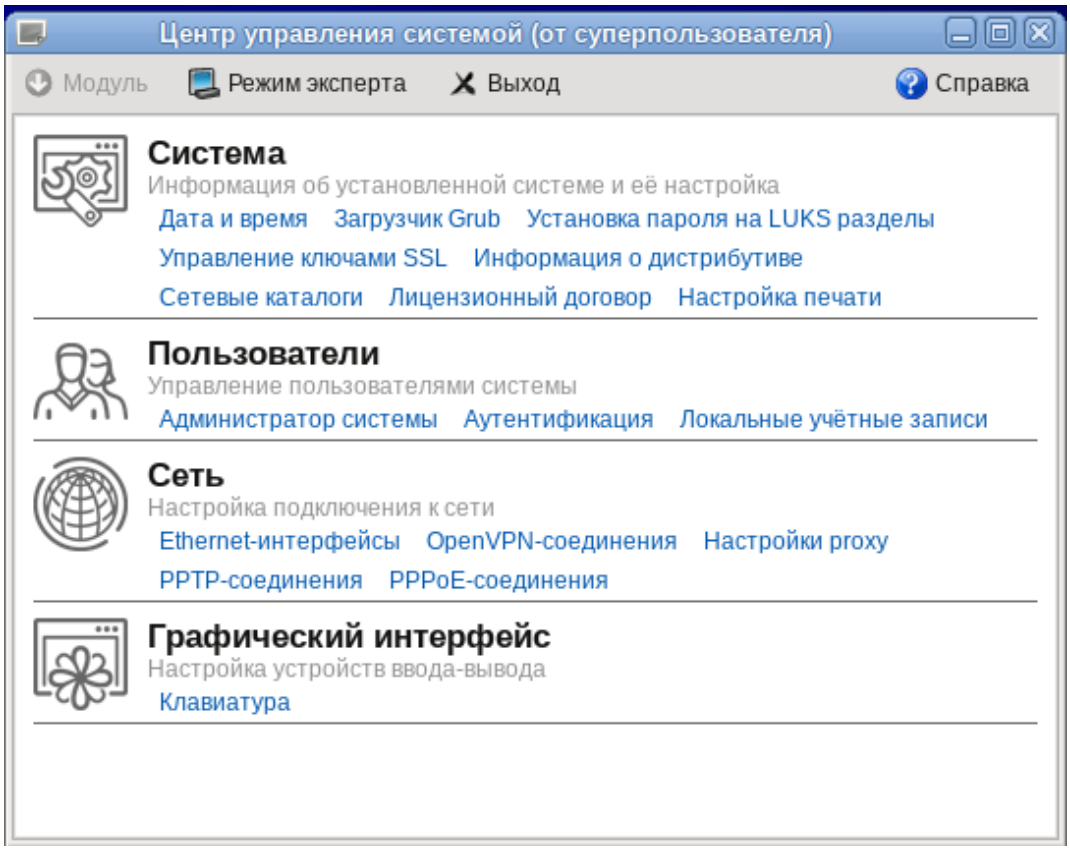
Центр управления системой можно запустить следующими способами:

- в графической среде MATE: **Система → Администрирование → Центр управления системой**;
- из командной строки: командой acc.

При запуске необходимо ввести пароль администратора системы (root).



После успешного входа можно приступать к настройке системы.

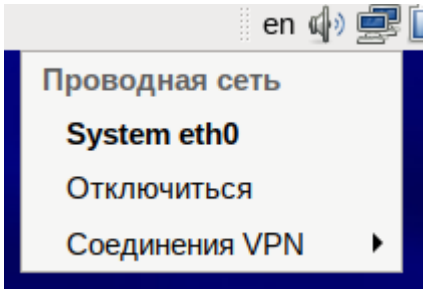


5 НАСТРОЙКА СЕТИ

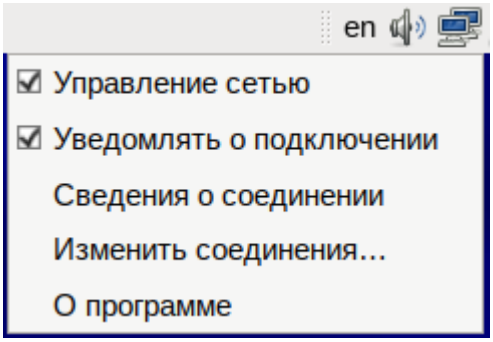
NetworkManager позволяет подключаться к различным типам сетей: проводные, беспроводные, мобильные, VPN и DSL, а также сохранять эти подключения для быстрого доступа к сети. Например, если вы подключались к сети в каком-либо интернет-кафе, то можно сохранить настройки этого подключения и в следующее посещение этого кафе подключиться автоматически.

5.1 NetworkManager

Для управления настройками сети в ОС Базальт Рабочая станция 8.0 используется программа NetworkManager.



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.



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).

6 УСТАНОВКА ДОПОЛНИТЕЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА РАБОЧУЮ СТАНЦИЮ

После установки ОС Базальт Рабочая станция 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**.

6.1 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.

6.2 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 **OK**. Если вы сомневаетесь, то выбирайте 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.

Непосредственная установка пакетов из добавленных репозиториев ничем не отличается от описанной выше в главе «Установка дополнительного ПО с установочного диска».

7 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.

7.1 Процессы и файлы

ОС Базальт Рабочая станция 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.

7.1.1 Процессы функционирования ОС

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

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

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

7.1.2 Файловая система ОС

В ОС использована файловая система Linux, которая в отличие от файловых систем DOS и Windows(™) является единым деревом. Корень этого дерева – каталог, называемый root (рут), и обозначаемый «/». Части дерева файловой системы могут физически располагаться в разных разделах разных дисков или вообще на других компьютерах, – для пользователя это прозрачно. Процесс присоединения файловой системы раздела к дереву называется монтированием, удаление –

размонтированием. Например, файловая система CD-ROM в 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).

7.1.3 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 символам:

- 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.

7.1.4 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 (TCB) 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 находящих 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.

7.1.5 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.

7.1.6 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.

7.1.7 Утилиты для работы с файловой системой

7.1.7.1 mkfs

mkfs – создаёт новую файловую систему Linux.

Синтаксис:

mkfs [-V] [-t fstype] [fs-options] filesys [blocks]

Описание:

mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys 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).

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 – производит подробный вывод, включающий все выполняемые специфические команды файловой системы. Если указать этот параметр более одного раза, то это запретит реальное выполнение любых специфических команд файловой системы. Использовать этот параметр целесообразно во время тестирования.

-V – 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.

Примеры:

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

mkfs -t ext2 /dev/sdb1

Ошибки:

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.

7.1.7.2 fsck

fsck – проверяет и восстанавливает файловые системы Linux.

Синтаксис:

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

Описание:

fsck используется для проверки и (опционально) для восстановления одной, либо нескольких файловых систем. filesys 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 директории /sbin, затем в /etc/fs и /etc, и, в конце, в директориях, перечисленных в переменной окружения PATH. Более подробная информация доступна в руководствах к специфичным для файловых систем утилитам.

Опции fsck:

-l – заблокировать весь диск при помощи flock(2). 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.

-r – 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.

-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 «no» или «!» (оператор отрицания), означающий что проверены будут 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 (для совместимости с дистрибутивами Mandrake, чьи стартовые скрипты зависят от неавторизованных изменений через пользова-

тельский интерфейс) 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 (см. далее). 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. См. ниже FSCK_FORCE_ALL_PARALLEL. 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».

-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(8)) располагающихся на корневой файловой системе 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 заголовок при запуске.

-V – генерировать более подробный вывод включая все специфичные для файловых систем команды.

Опции внешних утилит:

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 сложный 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(8) 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(8) не сообщает о каких-либо ошибках, а fsck.minix(8) 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.

-y – для некоторых утилит опция -y 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(8) и fsck.cramfs(8) не поддерживают -y.

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 директориях /sbin, /sbin/fs.d, /sbin/fs, /etc/fs и /etc. Затем, в директориях, 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.

7.1.7.3 df

df – сведения о числе свободных блоков и описателей файлов.

Синтаксис:

df [опция]... [файл]...

Описание:

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 показывается доступное место для этой файловой системе, 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.

Опции:

По умолчанию показывает информация обо всех смонтированных файловых системах

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

-a, --all – включать виртуальные файловые системы.

-B, --block-size=РАЗМЕР – использовать блоки указанного РАЗМЕРА (в байтах). Например, -BM для печати объёма в единицах равных 1048576 байтов.

--total – подсчитать общий объём в конце.

-h, --human-readable – печатать размеры в удобочитаемом формате (например, 1K 234M 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=1K.

-l, --local – ограничиться выводом локальных файловых систем.

--no-sync – не вызывать sync перед получением информации (по умолчанию).

-P, ---output[=СПИСОК_ПОЛЕЙ] – использовать формат, определённый в СПИСОК_ПОЛЕЙ, или печатать все поля если СПИСОК_ПОЛЕЙ не задан.

-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 использоваться следующие поля: 'source', 'fstype', 'itotal', 'iused', 'iavail', 'ipcent', 'size', 'used', 'avail', 'pcent' и 'target'.

7.1.7.4 du

du – подсчёт использованного дискового пространства.

Синтаксис:

du [опция]... [файл]...

du [опция]... --files0-from=F

Описание:

Подсчитывает использование диска каждым файлом, для директорий подсчёт происходит рекурсивно.

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

-0, --null – вместо символа новой строки выводить нулевой байт.

-a – выводить общую сумму для каждого заданного файла, а не только для директорий.

--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, BM для печати объёма в единицах равных 1048576 байтов.

-b, --bytes – тожечтои '--apparent-size --block-size=1'.

-c, --total – подсчитать общий объём в конце.

-D, --dereference-args – разыменовывать ссылки, указанные в командной строке.

-d, --max-depth=N – выводить объём для директории (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 – подсчитать использование диска для нуль-терминированных названий файлов, указанных в F. Если F равно -, то имена читаются со стандартного ввода.

-H – тожечтои --dereference-args (-D).

-h, --human-readable – выводить размеры в удобном для чтения человеком формате (например, 1K 234M 2G).

-k – тоже что и --block-size=1K.

-L, --dereference – разыменовывать все символические ссылки.

-l, --count-links – учитывать объём несколько раз для жёстких ссылок.

-m – тоже что и --block-size=1M.

-P, --no-dereference – не следовать по символически ссылкам (по умолчанию).

-S, --separate-dirs – не включать размер поддиректорий.

--si – тоже что и -h, но используя возведение в степень числа 1000, а не 1024.

-s, --summarize – отобразить только сумму для каждого аргумента.

-t, --threshold=РАЗМЕР – если это возможно, исключить записи менее чем РАЗМЕР. В противном случае – исключить записи более чем РАЗМЕР.

--time – показать время последнего изменений для всех файлов в директории или для её поддиректорий.

--time=WORD – вместо времени последнего изменений показать время как: atime, access либо use – время последнего доступа, ctime либо status – время создания.

--time-style=STYLE – отображать время используя стиль. Доступны следующие стили: full-iso, long-iso, iso, +FORMAT. FORMAT обрабатывается также, как и в команде 'date'.

-X, --exclude-from=FILE – исключить файлы, соответствующие любому образцу из файла FILE.

--exclude=PATTERN – исключить файлы, соответствующие образцу PATTERN.

-x, --one-file-system – пропустить директории на других файловых системах.

--help – напечатать справку и выйти.

--version – напечатать информации о версии и выйти.

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).

Unitsare K, M, G, T, P, E, Z, Y (powersof 1024) or KB, MB, ... (powers of 1000).

РАЗМЕР – это целое число, за которым опционально следует единица изменения. Например, 10M соответствует 10*1024*1024. Доступны следующие единицы измерения: K, M, G, T, P, E, Z, Y (возведение в степень числа 1024), KB, MB и т.д. (возведение в степень числа 1000).

ОБРАЗЕЦ — 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 файлу и поддиректории, заканчивающие на .o (в том числе и файл .o).

7.1.7.5 find

find - поиск файлов.

Синтаксис:

find список_поиска выражение

Описание:

Команда find рекурсивно просматривает каждый из каталогов, перечисленных в списке_поиска, отыскивая файлы, удовлетворяющие логическому выражению, построенному с помощью описанных ниже средств. В список_поиска могут входить и обычные файлы. Далее n обозначает целое десятичное число, на месте которого могут также указываться комбинации +n, что означает "больше, чем n", и -n, что означает "меньше, чем n". Элементарные логические выражения и их результаты:

-name шаблон_файлов

Истина, если текущий файл удовлетворяет шаблону_файлов. Символы шаблона, имеющие для командной оболочки специальный смысл, должны быть экранированы.

[-perm] [-]восьмеричное_число

Истина, если 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.

-type c

Истина, если 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).

-links n

Истина, если на файл имеется n ссылок.

-user имя_пользователя

Истина, если файл принадлежит пользователю с данным именем. Разрешены цифровые идентификаторы пользователя.

-group имя_группы

Истина, если файл принадлежит группе с данным именем. Разрешены цифровые идентификаторы группы.

-size n[c]

Истина, если файл занимает n блоков (по 512 байт). Если указана буква c, то размер файла задается в символах. Напомним, что с помощью комбинаций +n и -n можно проверять размер (и три указанные ниже характеристики) не только на равенство, но и на неравенство.

-atime n

Истина, если последний доступ к файлу производился n*24 часов назад. При поиске игнорируются любые дробные ча-сти. То есть -atime +1 найдёт файлы, доступ к которым осуществлялся как минимум два дня назад.

-mtime n

Истина, если файл последний раз модифицировался n*24 часов назад.

-ctime n

Истина, если статус файла последний раз изменялись n*24 часов назад.

-exec команда;

Истина, если после выполнения команды возвращается нулевой код завершения. Запись команды должна заканчиваться экранированной точкой с запятой. Строка «{}» заменяется текущим маршрутным именем файла.

-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

Всегда истина; вызывает выдачу маршрутного имени текущего файла.

-newer файл

Истина, если текущий файл был модифицирован позднее указанного файла.

-depth

Всегда истина; изменяет способ просмотра каталогов: сначала просматриваются входящие в каталог файлы, а затем уже сам каталог.

-mount

Всегда истина; ограничивает поиск файловой системой, содержащей указанный в списке_поиска каталог. Если в списке_поиска не указан ни один каталог, используется текущий.

(выражение)

Истина, если истинно заключенное в скобки выражение (скобки должны быть экранированы от интерпретации командной оболочкой).

Элементарные логические выражения могут комбинироваться с помощью следующих операций (в порядке уменьшения приоритета):

1. Унарная операция отрицания, обозначается !.

2. Логическое И, обозначается пробелом.

3. Логическое ИЛИ, обозначается -o.

Пример:

1. 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 {} \;

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

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

Файлы:

/etc/passwd

/etc/group

7.1.7.6 which

which – ищет программный файл.

Синтаксис:

which [опции] [--] программа [...]

Описание:

Which принимает один или более аргументов. Для каждого аргумента он печатает на стандартный вывод полный путь выполняемого файла, который был бы выполнен при запуске программы из командной строки. При этом осуществляется поиск выполняемого файла или скрипта в директориях, перечисленных в переменной окружения PATH, используя тот же алгоритм что и bash(1).

Which возвращает количество не найденных аргументов или 255 если не задан аргумент программа.

Рекомендованный способ использования утилиты which это использование её в функции командной оболочки bash:

which () {

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

}

export -f which

Опции:

--all, -a – печатать все выполняемые файлы в PATH, а не только первый.

--read-alias, -i – прочитать псевдонимы из стандартного ввода сообщая о совпадениях на стандартный вывод. Эта опция может быть полезна, если which является псевдонимом, например, таким:

which='alias | which -i'.

--skip-alias – игнорировать опцию «--read-alias». Это может быть полезно для явного поиска только обычных исполняемых файлов, даже если опция «--read-alias» задана в псевдониме или в функции для which.

--read-functions – прочитать функцию командной оболочки со стандартного ввода сообщая о совпадениях на стандартный вывод. Эта опция может быть полезна если which является функцией, например, такой: which() { declare -f | which --read-functions $@ } export -f which.

--skip-functions – игнорироватьопцию «--read-functions». Это может быть полезно для явного поиска только обычных исполняемых файлов, даже если опция «--read-functions» задана в псевдониме или в функции для which.

--skip-dot – пропускать директории в PATH в начале которых есть точка.

--skip-tilde – пропускать директории в PATH в начале которых есть тильда (~) и выполняемые файлы, находящиеся в домашней директории.

--show-dot – если директория в PATH начинается с точки и подходящий исполняемый файл нашёлся, то вывести относительный путь вместо полного.

--show-tilde – вывести с тильдой в том случае если исполняемый файл хранится в домашней директории. Данная опция игнорируется при вызове which от пользователя root.

--tty-only – не обрабатывать опции правее этой если which выполняется не на tty.

--version, -v, -V – напечатать информацию о версии на стандартный вывод и выйти.

--help – напечатать информацию об использовании на стандартный вывод и выйти.

Ошибки:

Домашняя директория определяется по содержимому переменной окружения HOME, и не может быть определена, если переменная окружения HOME не задана. Если в PATH содержатся две эквивалентные директории, различающиеся потому, что какая-либо из них содержит символическую ссылку в пути, то which будет их рассматривать как отдельные директории.

7.1.7.7 cd

cd – смена текущего каталога.

Синтаксис:

cd [-L|-] [каталог]

Описание:

Команда cd применяется для того, чтобы сделать заданный каталог текущим. Если каталог не указан, используется значение переменной окружения HOME (обычно это каталог, в который

Вы попадаете сразу после входа в систему). 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.

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

Опция -L заставляет следовать по символическим ссылкам.

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 директории была успешной, то абсолютный путь новой рабочей директории будет напечатан на стандартный вывод.

Возвращаемое значение 0 если рабочая директория успешно изменена и 1 если нет.

7.1.7.8 pwd

pwd–вывод имени текущего каталога.

Синтаксис:

pwd [-LP]

Описание:

Команда pwd выдает имя текущего (рабочего) каталога.

Если задана опция -P, то напечатанный путь не будет содержать символических ссылок.

Если задана опция -L, то напечатанный путь может содержать символические ссылки.

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

7.1.7.9 mkdir

mkdir – создание каталога.

Синтаксис:

mkdir [опция]... директория...

Описание:

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

Опции:

-m, --mode=режим_доступа – явное задание режима_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 – печатать сообщение для каждой созданной директории.

-Z, --context=CTX – задать контекст SELinux для каждой создаваемой директории.

Примеры:

Чтобы создать поддерево каталогов tmpdir/temp/dir, надо выполнить команду:

mkdir -p tmpdir/temp/dir

7.1.7.10 ls

ls – выдача информации о файлах или каталогах.

Синтаксис:

ls [опция]... [файл]...

Описание:

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.

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.

Командой ls обрабатываются следующие флаги:

-a, --all – вывести список всех файлов (обычно не выводятся файлы, имена которых начинаются с точки).

-A, --almos-all – не показывать подразумеваемые «.» и «..».

-b, --escape – выдавать непечатные символы, входящие в имя файла, в восьмеричном виде (\ddd).

--block-size=РАЗМЕР – перед печатью преобразовать размеры к РАЗМЕР. Например, --block-size=M для печати объёма в единицах равных 1048576 байтов.

-B, --ignore-backups – не показывать записи, заканчивающиеся на «~».

-c – с опциями -lt: вместо времени последнего изменения использовать время последней модификации описателя файла (т. е. время создания файла, изменения режима доступа к нему и т. п.) для сортировки. С опций -t показать время последней модификации описания файла и сортировать по имени. В остальных случаях – сортировать по времени последней модификации описания файла, новые сначала.

-C – вывод в несколько колонок с сортировкой по колонкам.

--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. Включает -aU, выключает -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 показывать владельца.

--group-directories-first – сгруппировать директории 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 самое, но используя возведение в степень 1000, а не 1024.

-H, --dereference-command-line – разыменовывать символические ссылки, переданные в командной строке.

--dereference-command-line-symlink-to-dir – разыменовывать символические ссылки, переданные в командной строке если они указывают на директории.

--hide=ОБРАЗЕЦ – не показывать записи, соответствующие ОБРАЗЦУ командного интерпретатора (не работает совместно с -A и -a).

--indicator-style=СЛОВО – добавить индикатор со стилем СЛОВО к имени: none (по умолчанию), slash (-p), file-type (--file-type) или classify (-F).

-i, --inode – выдавать в первой колонке номера описателей файлов.

-I, --ignore=СЛОВО – не показывать записи, соответствующие ОБРАЗЦУ 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=СЛОВО – задатьстилькавычек: literal, locale, shell, shell-always, c или escape.

-r, --reverse – изменить порядок сортировки на обратный алфавитный.

-R – рекурсивно обойти встретившиеся подкаталоги.

-s, --size – выдавать размер файлов в блоках.

-S – отсортировать по размеру фалов.

--sort=СЛОВО – сортировать не по имени, а по критерию: none -U, extension -X, size -S, time -t или version -v.

--time=СЛОВО – с -l отображать вместо времени последней модификации: atime -u, access -u, use -u, ctime -c или status -c; использовать для сортировки в сочетании с опцией —sort=time.

--time-style=СТИЛЬ – с -l отображать время используя стиль: full-iso, long-iso, iso, locale, +FORMAT. FORMAT обрабаты-вается так же как и в команде 'date'. Если ФОРМАТ задан как ФОРМАТ1<newline>ФОРМАТ2, то ФОРМАТ1 применяется к старым файлам, а ФОРМАТ2 – к новым. Если СТИЛЬ содержит префикс 'posix-', то стиль будет применяться только вне локали POSIX.

-t – файлы сортируются по времени последнего изменения (сначала идут самые свежие файлы).

-T, --tabsize=ЧИСЛО – задать tab stop для колонок.

-u – с опциями -lt: вместо времени последнего изменения использовать время последнего доступа для сортировки. С опций -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 файлы с учётом цифр в названиях.

-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.

-1 – 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, 10M соответствует 10*1024*1024. Доступны следующие единицы измерения: K, M, G, T, 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 стандартный выход осуществляется на терминал. Переменная окружения LS_COLORS может изменять настройки. Используйте dircolors чтобы задать её.

Возвращаемое значение:

0 – всё хорошо;

1 – незначительные проблемы (например, не удалось получить доступ к поддиректории);

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.

При использовании команды

ls -l /util/by

результат выглядит примерно так:

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

Читая справа налево можно увидеть, что 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. Наконец, последовательность ми-

нусов и букв указывает, что владелец, члены группы 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:

1. Если файл имеет режим

-rwxr--r-

он доступен владельцу для чтения, записи и выполнения, а членам группы и прочим пользователям только для чтения.

2. Режим

-rwsr-xr-x

свидетельствует о том, что 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.

3. В случае режима

-rw-rwl--

файл доступен для чтения и записи только владельцу и членам группы; может быть блокирован при доступе.

4. По команде

ls -a

будут выведены имена всех файлов в текущем каталоге, включая и те, которые начинаются с точки и обычно не выдаются.

5. По команде

ls -aisn

будет выдана разнообразная информация: список всех файлов, включая те, которые обычно не выводятся (a); номера описателей файлов будут выведены в левой колонке (i); размеры файлов (в блоках) выводятся во второй колонке (s); наконец, будут выданы числовые идентификаторы владельцев и групп (n).

Файлы:

/etc/passwd

Идентификаторы пользователей (для ls -l и ls -o).

/etc/group

Идентификаторы групп для (ls -l и ls -o).

/usr/lib/terminfo/?/*

Сведения о терминале.

7.1.7.11 cp

cp – копирование файлов и директорий.

Синтаксис:

cp [ОПЦИЯ]... [-T] ИСТОЧНИК НАЗНАЧЕНИЕ

cp [ОПЦИЯ]... ИСТОЧНИК... ДИРЕКТОРИЯ

cp [ОПЦИЯ]... -t ДИРЕКТОРИЯ ИСТОЧНИК...

Копирует ИСТОЧНИК в НАЗНАЧЕНИЕ или несколько ИСТОЧНИКОВ в ДИРЕКТОРИЮ.

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

Опции:

-a, --archive – тожечтои -dR --preserve=all.

--attributes-only – не копировать данные, только атрибуты.

--backup[=CONTROL] – сделать резервную копию каждого существующего файла.

-b – тоже что и --backup, но не принимает аргументы.

--copy-contents – копировать содержимое специальных файлов при рекурсивном копировании.

-d – тожечтои --no-dereference --preserve=links

-f, --force – если существующий файл назначения не может быть открыт, то удалить его и попробовать ещё раз (данная опция игнорируется если используется совместно с -n).

-i, --interactive – спросить перед перезаписью (данная опция игнорируется если используется совместно с -n).

-H – следовать символическим ссылкам из командной строки в ИСТОЧНИКЕ.

-l, --link – сделать жёсткую ссылку вместо копирования.

-L, --dereference – всегда следовать символическим ссылкам в ИСТОЧНИКЕ.

-n, --no-clobber – не перезаписывать существующие файлы (отменяет стоящую перед ней опцию -i).

-P, --no-dereference – никогда не следовать символическим ссылкам в ИСТОЧНИКЕ.

-p – тожечтои --preserve=mode,ownership,timestamps.

--preserve[=СПИСОК_АТРИБУТОВ] – сохранятьуказанныеатрибуты (поумолчанию: mode,ownership,time-stamps). Доступны следующие дополнительные атрибуты: context, links, xattr и all.

--no-preserve=СПИСОК_АТРИБУТОВ – не сохранять указанные атрибуты.

--parents – использовать полное имя ИСТОЧНИК в ДИРЕКТОРИЯ.

-R, -r, --recursive – копировать директории рекурсивно.

--reflink[=КОГДА] – контролировать клонирование и CoW-копирование. См. ниже.

--remove-destination – удалить каждый существующий файл назначения перед попыткой копирования (см. также --force).

--sparse=КОГДА – контролировать создание разряжённых файлов. См. ниже.

--strip-trailing-slashes – удалить все завершающие слэши в каждом ИСТОЧНИКЕ.

-s, --symbolic-link – создать символическую ссылку вместо копирования.

-S, --suffix=СУФФИКС – переопределить обычный суффикс для резервных копий.

-t, --target-directory=ДИРЕКТОРИЯ – копировать все аргументы ИСТОЧНИК в ДИРЕКТОРИЮ.

-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 – показать, что было выполнено.

-x, --one-file-system – оставаться в одной файловой системе.

--help – показать справку и выйти.

--version – показать информацию о версии и выйти.

По умолчанию разряжённые файлы из ИСТОЧНИК определяются эвристически, и соответствующие файлы НАЗНАЧЕНИЕ также создаются разряжёнными. Это же поведение задаётся опцией --sparse=auto. Использование --sparse=always позволит создавать разряжённый файл НАЗНАЧЕНИЕ в том случае если ИСТОЧНИК содержит достаточно большие последовательности нулевых байтов. Использование --sparse=never запретит создание разряжённых файлов.

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

Суффикс для резервных копий «~». Его можно переопределить при помощи опции --suffix или переменной окружения SIMPLE_BACKUP_SUFFIX. Метод контроля версий может быть задан через опцию --backup или через переменную окружения VERSION_CONTROL. Доступные значения:

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

Есть особый случай, когда cp создаёт резервную копию ИСТОЧНИКА. При этом заданы опции --force и --backup, и ИСТОЧНИК совпадает с НАЗНАЧЕНИЕМ и является обычным файлом.

7.1.7.12 mv

mv – перемещение (переименование) файлов.

Синтаксис:

mv [ОПЦИЯ]... [-T] ИСТОЧНИК НАЗНАЧЕНИЕ

mv [ОПЦИЯ]... ИСТОЧНИК... ДИРЕКТОРИЯ

mv [ОПЦИЯ]... -t ДИРЕКТОРИЯ ИСТОЧНИК...

Описание:

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

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

--backup[=CONTROL] – сделать резервную копию существующего файла назначения.

-b – тоже что и --backup, но не принимает аргументы.

-f, --force – не спрашивать перед перезаписью.

-i, --interactive – спросить перед перезаписью.

-n, --no-clobber – не перезаписывать существующий файл. Если вы укажете 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=ДИРЕКТОРИЯ – переместить все аргументы ИСТОЧНИК в ДИРЕКТОРИЮ.

-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 – показать, что было выполнено.

--help – показать справку и выйти.

--version – показать информацию о версии и выйти.

Суффикс для резервных копий «~». Его можно переопределить при помощи опции --suffix или переменной окружения SIMPLE_BACKUP_SUFFIX. Метод контроля версий может быть задан через опцию --backup или через переменную окружения VERSION_CONTROL. Доступные значения:

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

7.1.7.13 rm

rm – удаление файлов или каталогов.

Синтаксис:

rm [ОПЦИЯ]... ФАЙЛ...

Описание:

Удаляет каждый указанный ФАЙЛ. По умолчанию не удаляет директории.

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 невосстановимости данных следует использовать shred.

Опции:

-f, --force – игнорировать несуществующие файлы и аргументы. Никогда не запрашивать подтверждение.

-i – запросить подтверждение перед каждым удалением.

-I – запросить 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] – спрашивать в соответствии с КОГДА: never, once (-I) или always (-i). Если КОГДА пропущено, то подразумевается always.

--one-file-system – при рекурсивном удалении пропускать любые директории, находящиеся на других по отношения к аргументам командной строки файловых системах.

--no-preserve-root – позволяетудалить «/».

--preserve-root – не удалять «/» (по умолчанию).

-r, -R, --recursive – удалить директории и их содержимое рекурсивно.

-d, --dir – удалять пустые директории.

-v, --verbose – показать, что было выполнено.

--help – показать справку и выйти.

--version – показать информацию о версии и выйти.

По умолчанию rm не удаляет директории. Используйте опцию --recursive (-r или -R) для того чтобы удалить каждую заданную директорию и всё их содержимое.

Для того чтобы удалить содержащий в начале имени «-» (например: «-foo») используйте один из вариантов команды:

rm -- -foo

rm ./-foo

Примеры:

Опция -i часто используется совместно с -r. По команде:

rm -ir dirname

запрашивается подтверждение:

rm: удалить каталог «dirname/»?

7.1.7.14 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:

cat f - g

Выводит содержимое файла f, затем со стандартного ввода, и в конце – содержимое файла g.

cat

Копирует стандартный ввод на стандартный вывод.

7.1.7.15 more

more– осуществляет поэкранное отображение текста файла.

Синтаксис:

more [ОПЦИИ] ФАЙЛ [...]

Опции:

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

-номер – задаёт число строк в 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 некорректная клавиша в конце экрана 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 этому 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.

Команды:

Интерактивные команды more во много похожи на команды vi.

h или ? – отобразить справку по интерактивным командам.

ПРОБЕЛ – показать следующие k линий текста. По умолчанию – размер экрана.

z – показать следующие k линий текста. По умолчанию – размер экрана. Аргумент становится новым значением по умолчанию.

ВВОД – показать следующие k линий текста. По умолчанию – 1. Аргумент становится новым значением по умолчанию.

d или ^D – прокрутить k линий. 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.

Переменные окружения:

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 том случае если не задана переменная окружения VISUAL.

7.1.7.16 ln

ln – создание ссылок между файлами.

Синтаксис:

ln [ОПЦИЯ]... [-T] НАЗНАЧЕНИЕ ИМЯ_ССЫЛКИ (1ая форма)

ln [ОПЦИЯ]... НАЗНАЧЕНИЕ (2ая форма)

ln [ОПЦИЯ]... НАЗНАЧЕНИЕ... ДИРЕКТОРИЯ (3ья форма)

ln [ОПЦИЯ]... -t ДИРЕКТОРИЯ НАЗНАЧЕНИЕ... (4ая форма)

Описание:

В первой форме создаёт ссылку на НАЗНАЧЕНИЕ с именем ИМЯ_ССЫЛКИ. Во второй форме создаёт ссылка на НАЗНАЧЕНИЕ в текущей директории. В третьей и четвёртой формах создаёт ссылки для каждого НАЗНАЧЕНИЯ в заданной ДИРЕКТОРИИ. По умолчанию создаются жёсткие ссылки. Также по умолчанию предполагается, что файл назначения не существует. При создании жёстких ссылок НАЗНАЧЕНИЕ не должно существовать. Символические ссылки могут содержать произвольный текст. Относительные ссылки интерпретируются относительно их родительской директории.

Опции:

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

--backup[=CONTROL] – сделать резервную копию каждого существующего файла назначения.

-b – тоже что и --backup, но не принимает аргументы.

-d, -F, --directory – позволяет супер-пользователю попытаться создать жёсткую ссылку для директории (обратите внимание: скорее всего это будет неудачным из-за ограничений файловой системы).

-f, --force – удалять существующие файлы назначения.

-i, --interactive – спросить перед удалением назначения.

-L, --logical – разыменовывать НАЗНАЧЕНИЯ если это символические ссылки.

-n, --no-dereference – обрабатывать ИМЯ_ССЫЛКИ как обычный файл вне зависимости от того является ли он символической ссылкой или директорией.

-P, --physical – создавать жёсткие ссылке непосредственно на символические ссылки.

-r, --relative – создавать символические ссылки относительно положения ссылки.

-s, --symbolic – делать символические, а не жёсткие ссылки.

-S, --suffix=СУФФИКС – переопределить обычный суффикс для резервных копий.

-t, --target-directory=ДИРЕКТОРИЯ – задать директорию для создания ссылок.

-T, --no-target-directory – всегда обрабатывать ИМЯ_ССЫЛКИ как обычный файл.

-v, --verbose – напечатать имя каждого файла для которого была создана ссылка.

--help – показать справку и выйти.

--version – показать информацию о версии и выйти.

Суффикс для резервных копий «~». Его можно переопределить при помощи опции --suffix или переменной окружения SIMPLE_BACKUP_SUFFIX. Метод контроля версий может быть задан через опцию --backup или через переменную окружения VERSION_CONTROL. Доступные значения:

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

7.1.7.17 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]

Описание:

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 напечатает 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 компилирования программы в форме понимаемой ядром UNIX. data – остальное (данные обычно двоич-

ные или непечатаемые). Исключением являются файлы определённого формата (например, архивы или core-файлы), для них содержание двоичных данных считается нормальным. При добавлении локального определения в файл /etc/magic убедитесь, что описанные выше слова используются в вашем определении. Пожалуйста, не делайте то, что было сделано в Berkeley: они, например, изменили «shell commands text» на «shell script».

Проверки файловой системы основаны на проверке кода возврата системного вызова stat(2). 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 директории /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(1), 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».

Опции:

-b, --brief – не печатать 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 директории.

-e, --exclude имя_теста – не выполнять именованный тест. Доступны следующие имена тестов:

- apptype – 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» будетвыведено «text/plain charset=us-ascii». Для того чтобы этого добиться file меняет способ распознавания файлов для большинства текстовых

файлов, директорий и других типов файлов. При этом используется альтернативный magic-файл. Для работы этой опции file меняет способ распознавания файлов – он использует альтернативный magic-файл (см. далее под заголовком Файлы).

--mime-type, --mime-encoding – тоже что и -i, но печатает только заданные элементы.

-k, --keep-going – не прекращать поиск при перовом совпадении. Для отделения последующих совпадений будет использоваться символ «\012» (если вы хотите использовать для этого перевод строки, то смотрите опцию -r).

-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 указанной директории 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(2) или utimes(2) попытаться сохранить неизменным время доступа анализируемых файлов.

-r, --raw – не переводить непечатаемые символы в виде \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(2) сообщил, что это нормальный файл. Это предотвращает возникновение проблем при чтении специальных файлов. Задание данной опции заставит file читать также читать специальный 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(2), 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(1). Данная опция не затрагивает обычный разделитель, который также будет напечатан.

--help – напечатать помощь и выйти.

Файлы:

/usr/share/file/magic.mgc – используемый по умолчанию скомпилированный magic-файл.

/usr/share/file/magic – используемая по умолчанию директория с magic-файлами.

Переменные окружения:

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 переменой. Переменная окружения POSIXLY_CORRECT определяет (если система поддерживает символические ссылки), будет ли file разыменовывать символические ссылки. Если она задана, то file будет разыменовывать символические ссылки, если не задана – не будет. Также это поведение контролируется опциями -L и -h.

7.1.7.18 chmod

chmod – изменяет биты режима ФАЙЛА на РЕЖИМ. С опцией --reference изменяет режим ФАЙЛА на режим ИФАЙ-ЛА.

Синтаксис:

chmod [ОПЦИЯ]... РЕЖИМ[,РЕЖИМ]... ФАЙЛ...

chmod [ОПЦИЯ]... ВОСЬМЕРИЧНЫЙ-РЕЖИМ ФАЙЛ...

chmod [ОПЦИЯ]... --reference=ИФАЙЛФАЙЛ...

Описание:

chmod изменяет биты режима для каждого заданного файла согласно режима, который может быть текстовым представлением производимых изменений или восьмеричным представлением нового набора битов режима.

Формат текстового представления следующий: [ugoa...][[+-=][разрешения...]...]. Здесь разрешения – это ноль или более букв из набора «rwxXst» или одна из букв из набора «ugo». Несколько символических режимов могут быть разделены запятыми.

Буквы «ugoa» определяют для каких пользователей будет произведено изменение режима: «u» означает изменение для владельца файла, «g» – для пользователей, входящих в группу к которой принадлежит файл, «o» – остальные пользователи, «a» означает изменения сразу всех перечисленных выше. Если ни одна из этих буков не указана, то chmod произведёт изменения как будто была указана «a», но при этом не будут затронуты биты, заданные в umask.

Оператор «+» означает добавление указанных битов режима к битам режима каждого файла. «-» означает удаление ука-занных битов режима. «=» означает добавление указанных битов и удаление неуказанных за исключением директорий, для которых неуказанные биты для пользователя и группы не будут затронуты.

Буквы «rwxXst» задают биты режима для пользователей: «r» – чтение, «w» – запись, «x» – выполнение (или поиск для директорий), «X» – выполнение/поиск только если это директория или

же файл с уже установленным битом выполнения, «s» – задать ID пользователя и группы при выполнении, «t» – запрет удаления.

Числовой режим представляет из себя от одной до четырёх цифр от нуля до семи, полученных в результате сложения цифр 4, 2 и 1. Неуказанные цифры подразумеваются нулями. Первая цифра задаёт ID пользователя (4), ID группы (2) и ограничение на удаление (1). Вторая цифра задаёт права доступа для пользователя: 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 для символических ссылок так как системный вызов не может изменить права для символических ссылок. Это не является проблемой так как права доступа символических ссылок никогда не используются. Те не менее, для указанных в командной строке символических ссылок chmod произведёт изменение прав доступа для файлов, на которые они ссылаются. А вот при рекурсивном обходе директорий chmod будет игнорировать символические ссылки.

Биты задания группы и владельца

Если группа файла не соответствует эффективному идентификатору группу или одной из групп, в которые входи пользователь, то chmod удалить бит SGID у обычного файла.

chmod ну будет задавать SUID и SGID биты пока вы явно не укажите это. Вы можете задать либо снять биты в символьном режиме, например при помощи «u+s» и «u-s». В символьном режиме вы можете задать (но не снять) эти биты.

Запрет удаления

За запрет удаления отвечает один бит, и его интерпретация зависит от типа файла. Для директорий он предотвращает удаление и переименование файлов непривилегированными пользователями. Обычно он используется в публично доступных на запись местах, таких как /tmp.

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 директории.

--help – показать справку и выйти.

--version – показать информацию о версии и выйти.

Примеры:

1. 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

2. Позволить всем выполнять файл f2:

chmod +x f2

3. Запретить удаление файла f3:

chmod +t f3

4. 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

7.1.7.19 chown

chown – смена владельца файла.

Синтаксис:

chown владелец файл ...

Описание:

Команда chown дает файлу нового владельца, который может быть задан либо десятичным идентификатором пользователя, либо входным именем из файла /etc/passwd.

Изменить владельца может только владелец файла или суперпользователь.

Файлы:

/etc/passwd

/etc/group

7.1.7.20 umask

umask – установка маски режима создания файлов.

Синтаксис:

umask [маска]

Описание:

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.

7.1.7.21 chattr

chattr – изменяет атрибуты файлов файловой системы ext2fs.

Синтаксис:

chattr [-Rv] [-v версия] [атрибуты] файлы...

Описание:

chattr изменяет атрибуты файлов файловой системы ext2fs.

Формат символьного режима: +-=[ASacDdIijsTtu].

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 директорий (D), не архивировать (d),неизменяемый (i), журналирование данных (j), безопасное удаление (s), вершина иерархии директорий (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.

-v версия

Установить номер версии/генерации файла.

Атрибуты:

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

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 директория 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 директория 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.

Директория с атрибутом (T) будет поднята на вершину иерархии каталогов для целей. Orlov block allocator (который используется на системах с Linux 2.5.46 и выше).

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 chattr(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 chattr, но может быть показан с помощью lsattr.

7.1.7.22 lsattr

lsattr – выдает список атрибутов файлов на Linux ext2fs.

Синтаксис:

lsattr [-RVadv] [файлы...]

Описание:

lsattr выдает список атрибутов файлов на ext2fs. В chattr(1) описаны все атрибуты и их назначение.

Опции:

-R

Рекурсивно выдает список атрибутов каталогов и их содержимого.

-V

Выводит 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.

8 РАБОТА С НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫМИ КОМПОНЕНТАМИ

8.1 Командные оболочки (интерпретаторы)

Для управления ОС используется 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 ведет 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.

8.1.1 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

После ввода:

!!

запустится последняя, 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, предположим, 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, товыведется 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

8.1.2 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/

Некоторые варианты команды:

cd ..

позволяет сделать текущим родительский каталог (обратите внимание на пробел между cd и ..).

cd -

позволяет вернуться в предыдущий каталог. 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.

Синтаксис:

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.

Синтаксис:

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).

Синтаксис:

ps аргументы

Аргумент u предоставляет вам 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.

Синтаксис:

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.

8.2 Текстовый редактор Vi

Текстовый редактор Viимеетмодальныйинтерфей – 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.Напримеркоманда ”wq” позволяет записать файл и покинуть редакторVi, а команда “q!” – выйти из редактораViбез 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.

8.2.1 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».

8.2.2 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.

8.2.3 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 нажатьShift+:.

8.2.4 Сокращения

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

<CR>– ... Enter.

<SHIFT>– ... Shift.

^x – ... Ctrl-x.

8.2.5 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. Дальше 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.

8.2.6 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.

:setnu[mber] – отобразить слева нумерацию строк(:setnonu[mber] – спрятать нумерацию).

:setwrap – переносить длинные строки(:setnowrap– не переносить).

:colorscheme<name> – задать цветовую тему (где <name> имя темы, TAB работает как автодополнение).

/мама – поиск текста "мама" в файле.

n – повторить поиск.

:h или :help – список возможной помощи(:viusage, :exusage).

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

:set fileformat=dos.

:setfileformat=unix.

Задать размер табуляции в 4 пробела:

:setts=4.

8.3 Редактор VIM

8.3.1 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 нажатии (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"). Выделение при этом начина-

ется 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>.

8.3.2 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 команднострочные аналоги и могут быть легко переопределены.

8.3.2.1 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 нажатьстрелку 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.

8.3.2.2 Метки

Метки используются для отметки позиции (mметка, 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.

8.3.2.3 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 ("меткар, можете вместо p использовать [p,]p для вставки соответственно перед, или после курсора). В режиме редактирования вставка из регистра осуществляется по ”<Ctrl>+R метка”. Для добавления данных в регистр 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.

8.3.2.4 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 zc.

8.3.2.5 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
```

8.3.2.6 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.

8.3.2.7 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.

8.3.3 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 Vimприсутствует подробная документация по настройкам –:options.

8.4 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 супер-сервером.

Сервисы, перечисленные в конфигурационном файле xinetd можно разделить на две группы. Сервисы из первой группы называются multi-threaded и на каждый новый запрос запускается новый серверный процесс. Для таких сервисов xinetd продолжает слушать сеть на соответствующем порту ожидая новых запросов и готовай породить новый процесс. В другую группу включаются сервисы службы, которых в состоянии обрабатывать новые соединения. Такие сервисы называются single-threaded и xinetd прекращает обработку новых запросов до тех пор, пока серверный процесс не завершит свою работу. Сервисы в этой группе обычно datagram-based.

Итак, причиной существования супер-сервера является факт сохранения системных ресурсов за счет не запуска множества серверных процессов, которые возможно будут бездействовать 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.

8.4.1 Параметры

-d

Активирует режим отладки. Указание этой опции приводит к большому количеству отладочных сообщений, которые делают возможным использование отладчика на xinetd.

-syslog syslog_facility

Данная опция разрешает протоколирование создаваемых xinetd сообщений через syslog с заданным syslog facility. Поддерживаются следующие имена facility: daemon, auth, user, local[0-7] (посмотрите syslog.conf(5) 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 неэффективна в режиме отладки, так как все необходимые сообщения отправляются на терминал.

-f файл_настроек

Задает файл, который xinetd использует для настройки. По умолчанию это /etc/xinetd.conf.

-pidfile pid_файл

В этот файл записывается идентификатор процесса. Данная опция неэффективна в режиме отладки.

-dontfork

Говорит xinetd оставаться в интерактивном режиме, вместо отключения от терминала, для поддержки запуска из init или daemontools. Эта опция автоматически устанавливает -stayalive (см. ниже).

-stayalive

Говорит xinetd оставаться запущенным, даже если не задано никаких служб.

-remlock

Говорит xinetd удалить файл блокировки (по умолчанию /var/lock/subsys/xinetd) при выходе.

-limit proc_limit

Данная опция устанавливает ограничение на количество одновременно запущенных процессов, которые может запустить xinetd. Ее назначение предотвращать переполнение таблицы процессов.

-logprocs limit

Данная опция устанавливает ограничение на количество одновременно запущенных серверов на один идентификатор удаленного пользователя.

-version

Эта опция говорит xinetd напечатать информацию о своей версии.

-inetd_compat

Эта опция заставляет xinetd считывать /etc/inetd.conf в дополнение к стандартным конфигурационным файлам. /etc/inetd.conf будет прочитан после стандартных конфигурационных файлов xinetd.

-cc interval

Данная опция говорит xinetd выполнять периодические проверки своего внутреннего состояния каждые interval секунд.

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

8.4.2 Управление xinetd

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

SIGHUP

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.

SIGUSR1

Приводит к снятию дампа внутреннего состояния (по умолчанию файл дампа это /var/run/xinetd.dump; чтобы изменить данное имя файла нужна правка config.h и перекомпиляция).

SIGIOT

Производит внутреннюю проверку того, что структуры данных, используемые программой не повреждены. Когда проверка завершится xinetd сгенерирует сообщение, которое скажет успешно прошла проверка или нет.

При реконфигурации лог-файлы закрываются и вновь открываются. Это позволяет удалять старые логи.

8.4.3 Файлы

/etc/xinetd.conf

стандартный конфигурационный файл.

/var/run/xinetd.dump

стандартный файл дампа.

8.5 Crontab

crontab – таблицы, управляющие работой службы cron. Файл содержит инструкции службы cron(8) в общей форме: запускать указанную команду в заданное время и в заданные дни. На компьютере обычно имеются общесистемный файл (/etc/crontab).

Хотя по сути cron файл является обыкновенным текстовым файлом, он не должен редактироваться обычными средствами. Для создания, изменения и удаления следует использоваться специальную утилиту, crontab(1).

Для редактирования crontab вашего пользователя, используйте команду

crontab -e

Пустые строки, ведущие пробелы и символы табуляции игнорируются. Строки, начинающиеся с символа ('#') считаются комментариями и игнорируются. Заметьте, что комментарии не допускаются в тех же строках, где расположены команды cron(8), 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(8).

Задание переменной среды:

Можно определять среду (набор переменных среды), в которой будет выполняться команда. Задание переменной среды осуществляется в следующей форме:

имя_переменной = значение

где пробелы вокруг знака равенства (`=') необязательны, и любые пробелы после Fa значения будут использованы как часть значения переменной Fa имя переменной . Строка значение может быть заключена в кавычки (одинарные или двойные) для возможности сохранения пробелов в начале и конце.

Несколько переменных среды устанавливаются автоматически службой cron(8). SHELL устанавливается в /bin/sh a LOGNAME и HOME определяются по файлу /etc/passwd (в соответствии с владельцем crontab). Значения переменных HOME и SHELL можно переопределить директивами crontab.

В системах семейства BSD переменная LOGNAME может называться USER.

В дополнение к LOGNAME , HOME и SHELL cron(8) может использовать переменную MAILTO в случаях если в дан-ном crontab была указана отправка почты. Если MAILTO определена (и не пуста), электронная почта отправляется указанному в переменной пользователю. Если MAILTO определена, но пустая, (MAILTO = `,') электронная почта отправляться не будет. В противном случае, почта посылается владельцу crontab. Эта переменная полезна при запуске команд от псевдо-пользователей, для которых не определены почтовые адреса в системе.

Команды cron:

Формат команд cron(8) аналогичен стандарту V7 и является совместимым с ним. Каждая стока в системном состоит из шести полей и команды:

минута час число месяц день_недели пользователь команда

Каждая стока в пользовательском состоит из пяти полей и команды:

минута час число месяц день_недели команда

Поля отделяются друг от друга пробелами или символами табуляции. Команда может состоять из нескольких полей.

В таблице 1 приведены допустимые значиения полей команд cron.

Таблица 1 - Допустимые значения полей

Поле	Допустимые значения
минута	* или 0-59
час	* или 0-23
число	* или 1-31
месяц	*, 1-12 или имя месяца (см. ниже)
день-недели	*, 0-7 или имя дня (воскресенье - это 0 и 7)
пользователь	имя существующего пользователя
команда	строка

Допустимо указание нескольких значений (и диапазонов черезтире) через запятую. Примеры: "1, 2, 5, 9" "0-4, 8-12".

Диапазон указывается как два числа, разделенных дефисом. Указываемые числа включаются в диапазон. Например, значение поля час 8-11 приведёт к выполнению команды в 8, 9, 10 и 11 часов.

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 (по стандарту V7 пришлось бы указывать "0,2,4,6,8,10,12,14,16,18,20,22"). Шаг можно указывать также после звёздочки: "каждые два часа" соответствует значению "*/2".

Звёздочка ('*') без шага соответствует полному диапазону значений.

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.

Поле команда (остаток строки) определяет запускаемую по расписанию команду. Вся оставшаяся часть строки до символа перевода строки или символа %, будет выполнен вызов /bin/sh или другой оболочки, определенной в переменной SHELL в crontab Знак процента ('%') в команде (если он не экранирован обратной косой чертой ('\')) будет соответствовать символу перевода строки и все данные после первого '%' будут посланы для команды на стандартный ввод.

Служба cron(8) запускает команды, когда значения полей минута,час, месяц и хотя бы одно из полей Fa число и Fa день_недели, совпадают с текущим временем (см. замечание ниже). Служба cron(8) сверяет директивы с текущим временем раз в минуту.

Замечание: день выполнения команды может быть задан в двух полях – число и день_недели. Если оба поля определены (т.е не равны *), то команда будет запущена, когда любое поле совпадёт с текущим временем. Например, запись:

30 4 1,15 * 5

приведёт к выполнению команды в 4:30 по полуночи первого и пятнадцатого числа каждого месяца, плюс в каждую пятницу.

Вместо первых пяти полей допустимо указание одного из восьми специальных триггеров, перечисленных в таблице 2.

Таблица 2 - Триггеры

Строка	Значение
@reboot	Выполнить команду один раз, при запуске cron(8).
@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 * * * *".

8.5.1 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 мая если это пятница
7 18 13 5 5 /home/www/myscript.pl
# Выполнять задание по воскресеньям в 10 час 30 минут
30 10 * * 0 /home/www/myscript.pl
```

crontab: installingnewcrontab

Вывод «crontab: installingnewcrontab» означает, чтоновыйcrontabуспешноустановлен.

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, назло Джо
0 22 * * 1-5 mail -s "Уже 10 вечера" joe%Joe,%%Где твои дети?%
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"
```

8.5.2 Нестандартные возможности

При задании дня недели, и 0 и 7 соответствуют воскресенью. BSD и ATT не поддерживают такое поведение.

Можно указывать одновременно и списки, и диапазоны в одном и том же поле. "1-3,7-9" не будет принято cron систем ATT и BSD они допускают только либо "1-3", либо "7,8,9".

Диапазоны можно указывать с пропусками, например, "1-9/2" соответствует "1,3,5,7,9".

Допустимо указание месяцев или дней недели по имени.

В crontab можно задавать переменные среды. В BSD и ATT, среда для дочерних процессов определяется файлом /etc/rc.

Вывод команд отсылается почтой владельцу файла crontab (в BSD это невозможно), а также может отправляться кому-либо другому (это невозможно в SysV), либо отправка может быть вообще отключена (это также невозможно в SysV).

Любая из команд с префиксом '@' может заменять первые пять полей файла.

9 ОБЩИЕ ПРАВИЛА ЭКСПЛУАТАЦИИ

9.1 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.

9.2 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 ATX;
- выключить компьютер (переключателем на корпусе AT системного блока);
- выключить принтер;
- выключить монитор компьютера (если питание монитора не от системного блока);
- выключить стабилизатор, если компьютер подключен через стабилизатор напряжения.