

Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
Московский технический университет связи и информатики

Факультет Сети и системы связи
Кафедра Сети связи и системы коммутации

М.Г. Канищева, Е.Е. Маликова

Сборник практических и лабораторных работ по курсу «Технологии связи»
«Основы работы Linux-подобных операционных систем»

Москва 2021

В сборнике изложены основы работы с операционной системой Linux, приведены лабораторные работы и практические задания, способствующие выработке у обучающихся практических навыков по администрированию ОС Linux. Подробно изложена последовательность выполнения работ, сопровождающаяся пояснениями и иллюстрациями.

Для выполнения работ на компьютере обучающихся должны быть установлены программное обеспечение: Oracle virtualbox либо VMware, ubuntu 18.04.5 LTS. Базовая операционная система: предпочтительно Windows. Для нормальной работы на виртуальной машине нужно: ОЗУ - 4 ГБ, процессор - 2.2 ГГц, место на жестком диске - 40 ГБ, доступ в сеть Интернет.

Содержание

ВВЕДЕНИЕ	4
Практическая работа №1. Теоретические основы работы linux-подобных операционных систем. Технология виртуализации	5
Лабораторная работа №1. Установка Ubuntu 18.04.5 LTS в виртуальную машину	17
Практическая работа №2. Файловая система.	34
Практическая работа №3. Основные команды для администрирования ОС Linux	44
Лабораторная работа №2. Изучение основ работы с каталогами	52
Лабораторная работа №3. Операции с пользователями в системе	56
Лабораторная работа №4. Утилиты ОС Linux	61
Лабораторная работа №5. Изучение основ работы с файлами	67
Лабораторная работа №6. Изучение флагов в командах	71
Практическая работа №4. Сетевая модель	74
Практическая работа №5. Изучение команд для настройки сети	78
Лабораторная работа №7. Изучение команд для настройки сети	80
Лабораторная работа №8 Изучение работы пользователей в системе Linux с различными правами доступа	83
Лабораторная работа №9 Изучение инструмента для перенаправления результатов работы команды	88
Практическая работа №6 Изучение файла настроек Shell	91
Лабораторная работа №10 Изучение файла настроек Shell (.bashrc) и команды alias	93
Лабораторная работа №11 Изучение переменных окружения	97
Лабораторная работа №12 Установка и удаление утилит	102
Лабораторная работа №13 Запуск собственных скриптов	105
Практическая работа №7 Удаленное управление ОС	109
Лабораторная работа №14 Подключение по SSH	112
Лабораторная работа №15 Изменение пароля пользователя	114
Лабораторная работа №16 Планирование выполнения команд	116
Практическая работа №8 Использование CRON	120
Лабораторная работа №17 Анализ сетевого трафика	121
Список литературы	124

ВВЕДЕНИЕ

В настоящее время инфокоммуникационные сети достигли высокого уровня развития, что требует от персонала, занимающегося их обслуживанием, соответствующих знаний и практических навыков.

Linux-подобные операционные системы наиболее востребованы в инфокоммуникационной сфере, особенно при построении программно-конфигурируемых сетей (SDN), OpenStack-решений и использовании технологии виртуализации сетевых функций (NFV). Представленный сборник позволяет обучающимся в интерактивной форме усвоить основы администрирования ОС Linux и современные технологии виртуализации.

Практическая работа №1. Теоретические основы работы linux-подобных операционных систем. Технология виртуализации

Краткая теория

Теоретические основы работы linux-подобных операционных систем

Краткая история Linux

Операционная система Linux является современной, свободно распространяемой системой, основанной на стандартах UNIX. Впервые Linux была разработана как небольшое, но самодостаточное ядро ОС в 1991 году Линусом Торвалдсом (Linus Torvalds), с основной целью добиться совместимости с UNIX.

История Linux – это история многолетнего (удаленного) взаимодействия пользователей всего мира, которое осуществляется почти исключительно через Интернет. Система была спроектирована с целью эффективного и надежного использования на распространенных персональных компьютерах, но она также используется и на многих других аппаратных платформах (например, серверах).

Основная часть ОС Linux – полностью оригинальна, но на ней может также исполняться значительная часть свободно распространяемого программного обеспечения для UNIX, и в результате имеется оригинальная свободно распространяемая совместимая с UNIX система, в которой нет коммерческого (proprietary) кода.

Ядро ОС Linux

Основой операционной системы Linux является ядро. Ядро отвечает за поддержку основных концепций (абстракций) ОС. Код ядра исполняется в привилегированном режиме, и ему полностью доступны все аппаратные ресурсы компьютера. Весь код и структуры данных ядра хранятся и исполняются в едином адресном пространстве [3].

Версия ядра 0.01 (май 1991 года) не содержала сетевых средств, выполнялась только на 80386-совместимых Intel – процессорах, имела очень ограниченный набор драйверов устройств и поддерживала только файловую систему Minix.

Версия ядра ОС Linux 1.0 (март 1994 года) включала следующие новые возможности:

- поддержку стандартных для UNIX сетевых протоколов TCP/IP;
- BSD-совместимый интерфейс сокетов для сетевого программирования;

- поддержку драйверов устройств для использования IP в сетях типа Ethernet;
- расширенную файловую систему;
- поддержку большого диапазона SCSI – контроллеров для высокопроизводительного доступа к дискам.

Версия ядра 1.2 (март 1995 года) была последней версией ядра Linux только для персональных компьютеров (ПК).

Дистрибутивы ОС Linux

Стандартный предварительно откомпилированный набор пакетов называется дистрибутивом. Он включает базовую систему Linux, утилиты для инсталляции системы и управления системой, а также готовые к инсталляции пакеты инструментов для UNIX.

Самые первые дистрибутивы это SLS и Slackware. RedHat и Debian – популярные дистрибутивы, соответственно, основанные на коммерческих и некоммерческих исходных текстах. Они появились в 1993 году. Дистрибутив Ubuntu появился 20 октября 2004 года, он основан на операционной системе Debian. Единый формат файла пакета -RPM обеспечивает совместимость между различными дистрибутивами ОС Linux [1].

Лицензирование ОС Linux

Ядро Linux распространяется на условиях лицензии GNU General Public License (GPL), которые установлены организацией Free Software Foundation. Лицензия GNU GPL выполняет следующие условия [4]:

- 1) программное обеспечение (ПО) поставляется с исходным кодом;
- 2) авторские права принадлежат разработчикам;
- 3) можно свободно и без оплаты запускать программы, изучать и изменять код, распространять ПО бесплатно или за плату, улучшать ПО и распространять улучшения;
- 4) программы распространяются без гарантий;
- 5) за настройку и сопровождение ПО можно брать плату.

Программист, использующий ОС Linux, либо создающий свои собственные системы на базе Linux, не имеет права превращать свой продукт в коммерческий, программное обеспечение, распространяемое на основе лицензии GPL, не может распространяться только в виде двоичного кода.

Принципы проектирования ОС Linux

Linux – многопользовательская и многозадачная ОС с полным набором UNIX-совместимых инструментов. Ее файловая система соответствует традиционной семантике UNIX. Она полностью реализует стандартную сетевую модель UNIX. Основные цели проектирования ОС Linux – высокая скорость, эффективность и стандартизация работы.

Важно отметить, что ОС Linux является чувствительным к регистру.

Компоненты системы Linux

В целом, считается, что ОС Linux состоит из ядра и утилит, однако при детальном рассмотрении компонентов системы важно отметить, что система состоит из:

- 1) загрузчика (GRUB2 – GRand Unified Bootloader), отвечающего за загрузку ядра в память и передачу ему параметров, указанных пользователем;
- 2) ядра;
- 3) модулей ядра, по сути – драйверов, обеспечивающих поддержку оборудования;
- 4) сервисов (служб), которые выполняются в фоновом режиме (например, веб-серверы, FTP-сервер и т.д.);
- 5) системы инициализации, управляющей запуском служб (в современных дистрибутивах systemd, в более старых init, init-ng и upstart);
- 6) программ, распространяемых в виде пакетов (архивов) с расширениями .rpm (RedHat, CentOS, Fedora) или .deb (Debian, Ubuntu);
- 7) менеджера пакетов (установка, удаление, обновление и т.д.), например, для Ubuntu – apt.

Как большинство реализаций UNIX, ОС Linux состоит из трех основных групп кода – ядро, системных библиотек и системных утилиты. Наиболее важно понимать различие между ядром и всеми остальными компонентами.

Системные библиотеки определяют стандартный набор функций, с помощью которых приложения взаимодействуют с ядром и которые реализуют основную часть функциональности ОС, не требующую исполнения в привилегированном режиме. Системные утилиты выполняют индивидуальные специфические задачи.

Модули ядра

Части (секции) кода ядра могут компилироваться, загружаться и выгружаться, независимо от остальной части ядра. Модуль ядра может реализовывать драйвер устройства, файловую систему или сетевой протокол. Модульный интерфейс позволяет третьим сторонам разрабатывать и распространять на своих собственных условиях драйверы или файловые системы, которые не могут распространяться на основе лицензии GPL. Модули ядра позволяют установить ОС Linux в виде стандартного, минимального ядра, без использования каких-либо встроенных устройств.

Три компоненты модуля Linux поддерживают:

- управление модулем;
- регистрацию драйвера;
- разрешение конфликтов.

Управление модулем

Управление модулем управляет загрузкой модуля в память и его взаимодействием с остальной частью ядра. Управление модулем разбито на две части:

- 1) управление частями кода модуля в памяти ядра;
- 2) управление символами, на которые модуль разрешает ссылаться.

Module requestor – управляет загрузкой запрошенных, но еще не загруженных модулей, он также регулярно опрашивает ядро, чтобы убедиться, что модуль до сих пор используется, и выгрузит его, если модуль долгое время активно не использовался.

Регистрация драйверов

Регистрация драйвера предоставляет модулю возможность сообщить ядру, что новый драйвер доступен. Ядро поддерживает динамическую таблицу всех известных драйверов и обеспечивает набор подпрограмм для добавления драйверов в эти таблицы или удаления драйверов из них в любое время.

Разрешение конфликтов

Разрешение конфликтов – механизм, который позволяет различным драйверам устройств резервировать аппаратные ресурсы и защищать эти ресурсы

от случайного использования другими драйверами. Целями модуля разрешения конфликтов являются: предотвращение конфликтов, связанных с использованием аппаратуры, предотвращение автопроверки (autoprobes) от пересечения с уже существующими драйверами устройств и разрешение конфликтов различных драйверов, пытающихся иметь доступ к одной и той же аппаратуре.

Программирование в ОС Linux

Широкое распространение при программировании в ОС Linux получил компилятор GNU Compiler Collection (GCC). Он является стандартным семейством компиляторов для большинства Linux-систем. Кроме того, GCC обеспечивает frontend для C, C++, Java. Большинство дистрибутивов включают в себя установленные интерпретаторы Perl, Python и других сценарных языков. Существует ряд сред для разработки (IDE): KDevelop, Eclipse, NetBeans, Lazarus. Также доступны и традиционные текстовые редакторы, такие как Emacs и Vim. Двумя распространёнными библиотеками визуальных элементов для создания графических интерфейсов пользователя являются Qt и GTK+.

Процессы в ОС Linux

Для каждой запускаемой программы в ОС Linux создается отдельный процесс. В рамках процесса программе выделяется процессорное время, оперативная память и другие системные ресурсы. У каждого процесса есть свой идентификатор, Process ID или просто PID, по нему, чаще всего и определяются процессы Linux. PID определяется неслучайно, программа инициализации получает PID 1, а каждая следующая запущенная программа - на единицу больше.

Файловая система

Файловая система (ФС) - предоставляет пользователям (и процессам) ресурсы долговременной памяти компьютера. Операционная система Windows может быть установлена на файловую систему NTFS, поэтому обычно у пользователей не возникает вопросов какую ФС лучше использовать. ОС Linux имеет значительные отличия. Так, в ядро системы встроены и могут использоваться несколько файловых систем, каждая из которых оптимизирована для решения определенных задач.

Чтобы на каждом разделе можно было работать с файлами и каталогами, необходима файловая система. Кроме записи содержимого файлов на диск нужно еще хранить данные о папках, имена файлов, их размер, адрес на жестком диске, атрибуты доступа. Всем этим занимается файловая система. От файловой системы зависит очень многое: скорость работы с файлами, скорость записи и даже размер файлов. Также от стабильности файловой системы будет зависеть сохранность файлов.

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы. Далее будут рассмотрены типы файловых систем Linux, включая специальные файловые системы [2].

Основные виды виртуализации

Развитие рынка технологий виртуализации за последние несколько лет происходило во многом благодаря увеличению мощностей аппаратного обеспечения, позволившего создавать эффективные платформы виртуализации, как для серверных систем, так и для настольных компьютеров. Технологии виртуализации позволяют запускать на одном физическом компьютере (хосте) несколько виртуальных экземпляров операционных систем (гостевых ОС) в целях обеспечения их независимости от аппаратной платформы и сосредоточения нескольких виртуальных машин на одной физической. Виртуализация предоставляет множество преимуществ, как для инфраструктуры предприятий, так и для конечных пользователей. За счет виртуализации обеспечивается существенная экономия на аппаратном обеспечении, обслуживании, повышается гибкость ИТ-инфраструктуры, упрощается процедура резервного копирования и восстановления после сбоев. Виртуальные машины, являясь независимыми от конкретного оборудования единицами, могут распространяться в качестве предустановленных шаблонов, которые могут быть запущены на любой аппаратной платформе поддерживаемой архитектуры.

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированного от аппаратной реализации и обеспечивающего при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе. Примером использования виртуализации является возможность запуска нескольких операционных систем на

одном компьютере: каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор. Операционные системы могут быть запущены в виртуальных машинах. Также могут быть подвергнуты виртуализации сети передачи данных, сети хранения данных, платформенное и прикладное программное обеспечение.

Виртуальная машина (ВМ) — это окружение, которое представляется для гостевой операционной системы, как аппаратное. Однако на самом деле это программное окружение, которое эмулируется программным обеспечением хостовой системы. Эта эмуляция должна быть достаточно надёжной, чтобы драйверы гостевой системы могли стабильно работать. Можно выделить несколько примеров применения ВМ:

- Тестовые лаборатории и обучение: тестированию в виртуальных машинах удобно подвергать приложения, влияющие на настройки операционных систем, например инсталляционные приложения. За счёт простоты в развёртывании виртуальных машин, они часто используются для обучения новым продуктам и технологиям.

- Распространение предустановленного программного обеспечения: многие разработчики программных продуктов создают готовые образы виртуальных машин с предустановленными продуктами и предоставляют их на бесплатной или коммерческой основе. Такие услуги предоставляют VMware VMTN или Parallels PTN.

На настоящий момент можно выделить следующие основные виды виртуализации: программная и полная (аппаратная) виртуализация, виртуализация уровня операционной системы (ОС) и виртуализация приложений.

Программная и полная (аппаратная) виртуализация

Как было рассмотрено выше, гипервизор занимается созданием виртуальных машин и их управлением. Гипервизор обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между запущенными ОС. В зависимости от типа используемой виртуализации,

гипервизор может работать как напрямую с железом без хост-системы, так и через основную операционную систему, установленную на хост-машину. В первом случае используется аппаратная виртуализация (рисунок 1), во втором — программная виртуализация (рисунок 2). Программную виртуализацию используют ПО Oracle VM VirtualBox и VMware Workstation. Одним из самых популярных аппаратных гипервизоров является ESXi. Полная (аппаратная) виртуализация – виртуализация с поддержкой специальной процессорной архитектуры. С помощью данной техники возможно использование изолированных гостевых систем, управляемых гипервизором напрямую. Осуществляется полная эмуляция компьютера со всеми аппаратными средствами. Гипервизор распределяет аппаратные средства между виртуальными машинами, в которые устанавливаются операционные системы. В результате получаем практически полноценные физические компьютеры. Виртуальные машины полностью изолированы друг от друга.

Аппаратная виртуализация должна поддерживаться процессором оборудования, на котором она осуществляется. Необходимо включить в программу BIOS эту функцию, для разных процессоров она имеет название либо Intel VT-x, либо AMD-V.

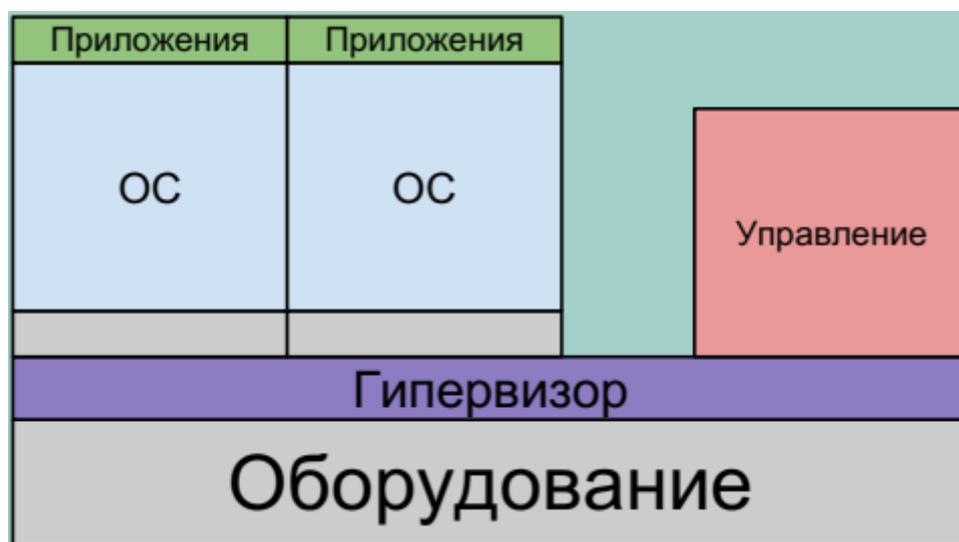


Рисунок 1. Полная (аппаратная) виртуализация

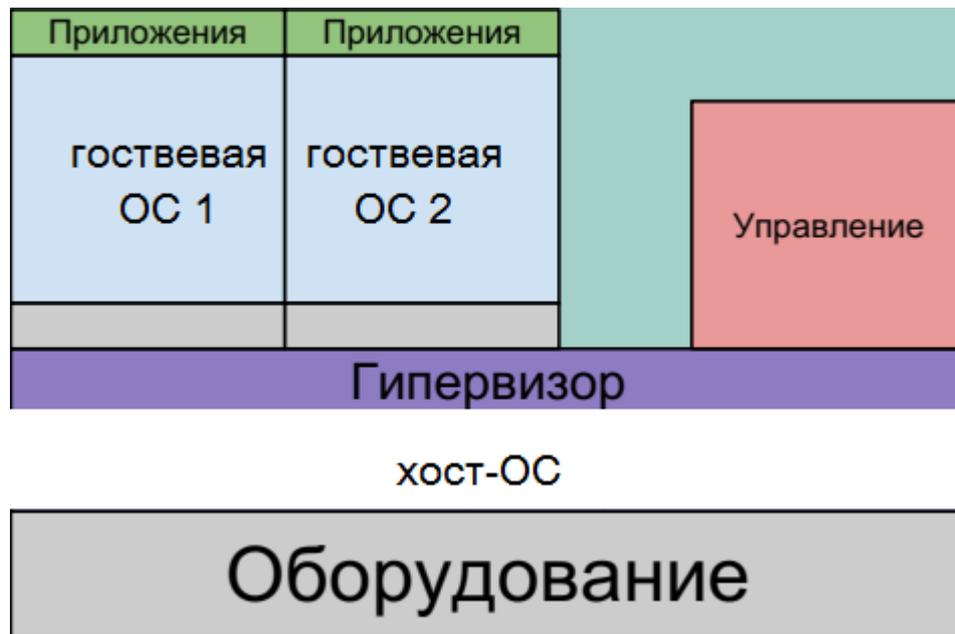


Рисунок 2. Программная виртуализация

Аппаратная виртуализация имеет ряд важных достоинств по сравнению с программной виртуализацией.

Во-первых, упрощается разработка программных платформ виртуализации за счет предоставления аппаратных интерфейсов управления и поддержки виртуальных гостевых систем, что уменьшает трудоемкость и время на разработку систем виртуализации.

Во-вторых, появляется возможность увеличения быстродействия платформ виртуализации, т.к. управление виртуальными гостевыми системами осуществляет напрямую небольшой промежуточный слой программного обеспечения, гипервизор, что дает увеличение быстродействия.

В-третьих, улучшается защищённость, появляется возможность переключения между несколькими запущенными независимыми платформами виртуализации на аппаратном уровне. Каждая из виртуальных машин может работать независимо в своем пространстве аппаратных ресурсов полностью изолированно от других ВМ. Это позволяет устранить потери быстродействия на поддержание хостовой платформы и увеличить защищенность.

В-четвертых, гостевая система становится не привязана к архитектуре хостовой платформы и к реализации платформы виртуализации. Технология аппаратной виртуализации делает возможным запуск 64-битных гостевых систем на 32-битных хостовых системах (с 32-битными средами виртуализации на хостах).

Виртуализация уровня операционной системы

Виртуализация на уровне операционной системы (рисунок 3) позволяет запускать изолированные и безопасные виртуальные машины на одном физическом узле, но не позволяет запускать операционные системы с ядрами, отличными от типа ядра базовой операционной системы.



Рисунок 3. Виртуализация уровня ОС

При виртуализации на уровне операционной системы не существует отдельного слоя гипервизора. Вместо этого сама хостовая операционная система отвечает за разделение аппаратных ресурсов между несколькими виртуальными машинами и поддержку их независимости друг от друга. Такой тип виртуализации применяют модифицированное ядро Linux OpenVZ, системf виртуализации LXC и др. Не используется эмуляция аппаратных средств. Создается что-то вроде копии большинства процессов исходной операционной системы. Такие процессы изолируются в отдельный контейнер. Иногда рассматриваемый тип виртуализации называют контейнерным. При этом в контейнере получают почти полноценную операционную систему, куда могут быть установлены приложения. Такой подход дает выигрыш в производительности по сравнению с полной виртуализацией, т.к. нет необходимости эмулировать аппаратные средства.

Виртуализация приложений

Виртуализация приложений (рисунок 4) — это технология, дающая пользователям возможность осуществлять доступ к приложению и использовать его не с того компьютера, на котором установлено приложение, а с другого.

Используя ПО виртуализации приложений, ИТ-администраторы могут устанавливать удаленные приложения на сервере, а затем доставлять их на компьютер конечного пользователя. Для пользователя работа с виртуализированным приложением не отличается от работы с приложением, установленным на физической машине.



Рисунок 4. Виртуализация приложений

Другие основные виды виртуализации

Также кроме рассмотренных выше основных видов виртуализации необходимо упомянуть виртуализацию сети, виртуализацию сетевых функций и виртуализацию систем хранения данных.

Виртуализация сети (Network Virtualization) — процесс объединения аппаратных и программных сетевых ресурсов в единую виртуальную сеть. Можно выделить 2 вида виртуализации сети:

- 1) Внешняя виртуализация, соединяющая множество сетей в одну виртуальную;
- 2) Внутренняя виртуализация, создающая виртуальную сеть между программными контейнерами внутри одной системы;

Одним из видов виртуализации сети можно считать виртуальные частные сети (Virtual Private Network, VPN) — обеспечение одного или нескольких сетевых соединений поверх другой сети.

Виртуализация сетевых функций (Network Functions Virtualization - NFV) — концепция сетевой архитектуры, предлагающая использование технологии виртуализации для виртуализации целых классов функций сетевых узлов в виде

составных элементов, которые могут быть соединены вместе или связаны в цепочку для создания телекоммуникационных услуг (сервисов) [7]. Технология NFV отличается от традиционных способов виртуализации, используемых в информационных технологиях уровня предприятия. Виртуализируемая сетевая функция (Virtualized Network Function, VNF) может включать одну или несколько виртуальных машин, использующих разное программное обеспечение и процессы, серверы, коммутаторы и хранилища большого объема или даже инфраструктуру облачных вычислений вместо отдельных аппаратных решений для каждой сетевой функции. Например, виртуализированный граничный контроллер сессий может быть развернут для защиты сети без обычных затрат и сложности получения и установки физических устройств. Другие примеры NFV включают виртуализированные балансировщики нагрузки и брандмауэры.

Виртуализация систем хранения данных – представление набора физических носителей в виде единого физического носителя, позволяет сделать незаметным для пользователя распределенное хранение информации и резервное копирование в случае отказа основного хранилища данных.

В настоящем учебном пособии предполагается изучение основ работы с Linux-подобными операционными системами на примере ОС Linux Ubuntu 18.04.5 LTS. Для удобства работы необходимо осуществить установку Linux Ubuntu 18.04.5 LTS на виртуальную машину, что позволит безопасно проводить обучение без риска нарушения работы основной операционной системы компьютера. Используется гипервизор Oracle VM VirtualBox версии 5.1.22.

Контрольные вопросы

1. Какие компоненты Linux системы вы знаете?
2. Что такое ядро ОС Linux?
3. Опишите назначение модулей ядра Linux.
4. Что такое дистрибутив?
5. Поясните понятие процессов в ОС Linux.
6. Что такое файловая система? Зачем она нужна? Какие функции выполняет?
7. Перечислите основные файловые системы, используемые для ОС Linux.
8. Для каких целей используются виртуальные файловые системы?
9. Перечислите основные каталоги ОС Linux.

10. Зачем нужны файлы устройств?
11. Какие права имеют обычные пользователи в ОС Linux?
12. Какая команда позволяет установить утилиту, используя менеджер пакетов **apt**?
13. Как полностью удалить утилиту из ОС Linux?
14. Что такое виртуализация?
15. Какие основные виды виртуализации вы знаете?
16. В чем отличия между программной и аппаратной виртуализацией?

Лабораторная работа №1. Установка Ubuntu 18.04.5 LTS в виртуальную машину

Цель работы

Освоить установку операционной системы Linux на виртуальную машину.

Задание

1. Установить на компьютер Oracle VirtualBox;
2. Создать виртуальную машину и установить ПО Ubuntu 18.04.5 LTS;
3. Сформировать отчет;
4. Ответить на вопросы.

Краткая теория

При осуществлении установки Linux Ubuntu 18.04.5 LTS в Oracle VM VirtualBox необходимо правильно настроить параметры виртуальной машины. Для операционной системы Linux Ubuntu 18.04.5 LTS (64-битная) рекомендуются следующие характеристики оборудования (таблица 1):

Таблица 1. - Системные требования Linux Ubuntu 18.04.5 LTS (64-битная)

Параметр	Требования
Оперативная память	2ГБ и более - Unity, Gnome, KDE; 1ГБ и более - XFCE, Mate, LXDE, Openbox, Fluxbox, IceWM, Enlightenment
Процессор	Двухъядерный процессор с частотой 1,6ГГц и мощнее - Unity, Gnome, KDE Одноядерный процессор с частотой 1,3ГГц и мощнее - XFCE, Mate, LXDE, Openbox, Fluxbox, IceWM, Enlightenment
Видеокарта	Любая, для которой есть драйвер
Жесткий диск	5ГБ и более - Unity, Gnome, KDE, XFCE, Mate, LXDE; 4ГБ и более - Openbox, Fluxbox, IceWM, Enlightenment

Порядок выполнения работы

Скачивание и установка виртуальной машины

Необходимо для скачивания ПО открыть следующую ссылку:

<https://www.virtualbox.org/wiki/Downloads>.

Перейдя по ссылке (рисунок 1), нужно выбрать ПО под нашу операционную систему. В нашем случае – Windows. Нажимаем «Windows hosts» и скачиваем.



Рисунок 1. Внешний вид веб-страницы, с которой необходимо скачать VirtualBox

Запускаем установщик и проходим стандартную процедуру установки. После этого переходим к следующему пункту.

Настройка виртуальной машины для Linux Ubuntu 18.04.5 LTS

Изначально надо в окне Oracle VM VirtualBox выбрать вкладку «Создать» (рисунок 2).



Рисунок 2. Начальное окно Oracle VM VirtualBox

В первом окне введите название новой виртуальной машины, выберите тип (Linux) и дистрибутив, например, Ubuntu (64-bit) (рисунок 3):

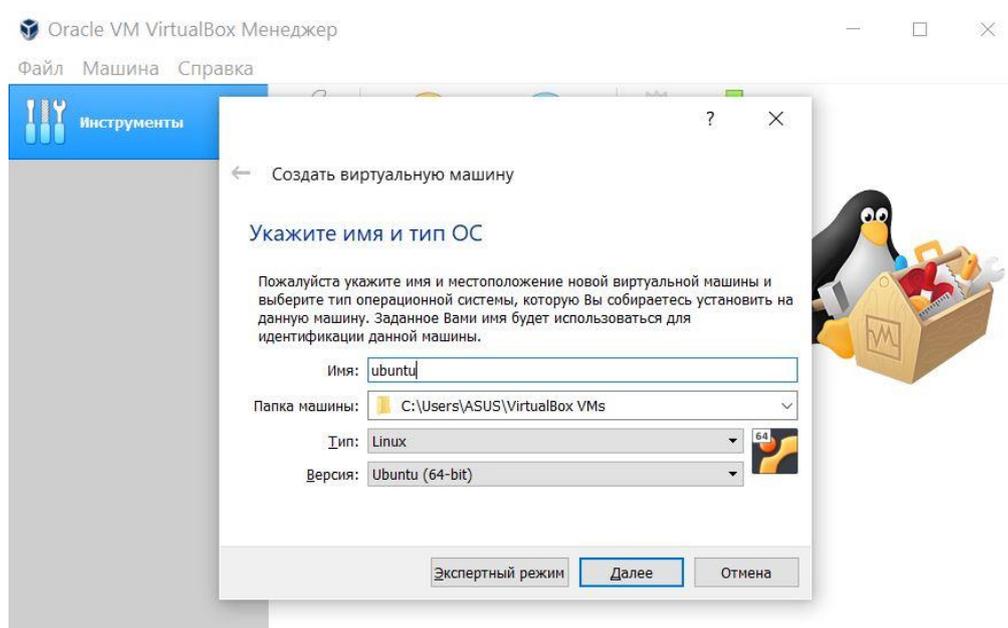


Рисунок 3. Создание виртуальной машины, выбор имени и типа ОС для Linux Ubuntu

Далее выберите необходимый объем оперативной памяти (рисунок 4):

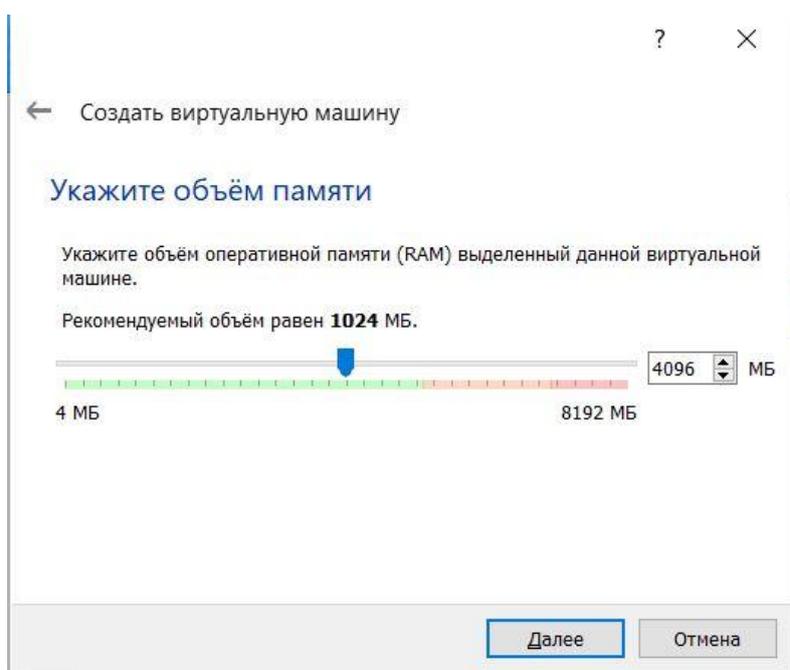


Рисунок 4. Выбор объема оперативной памяти для Linux Ubuntu

После этого необходимо создать жесткий диск для установки системы. Объем диска зависит от операционной системы, но поскольку вы можете создавать динамический диск, который будет изменять размер реального файла в файловой системе по мере наполнения, то можно указать 30 или даже 50 ГБ, чтобы было достаточно. Сначала выберите "Создать новый виртуальный жесткий диск" (рисунок 5), затем тип диска (рисунок 6):

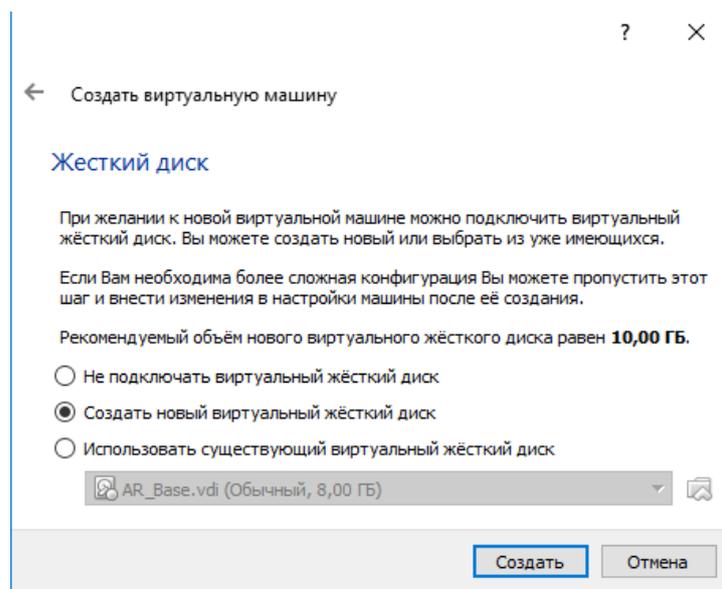


Рисунок 5. Создание жесткого диска для Linux Ubuntu

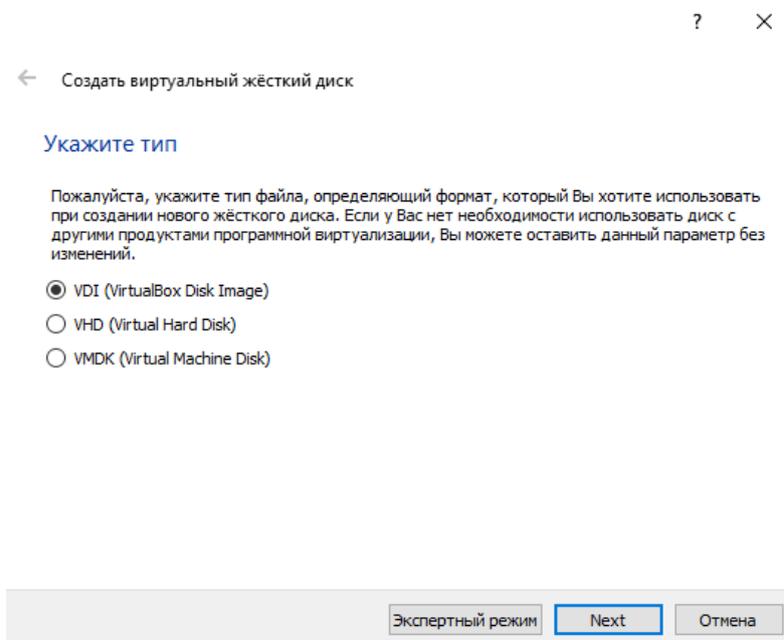


Рисунок 6. Выбор типа жесткого диска для Linux Ubuntu

Наиболее часто для виртуальных машин используется диск типа VDI (представляет собой образ CD-диска, созданный с помощью Virtuo CD Manager, и содержит точную копию диска. При монтировании в виртуальный привод образ позволяет работать с данными, как будто они записаны на физический носитель. Файлы образов в формате VDI могут быть открыты при помощи совместимых программ для эмуляции). Затем выберите размер диска и имя для его файла. Существует возможность использовать место на жестком диске по мере необходимости. Тем не менее, лучше выделить этот объем на диске сразу, что увеличит производительность до 10% (рисунок 7):

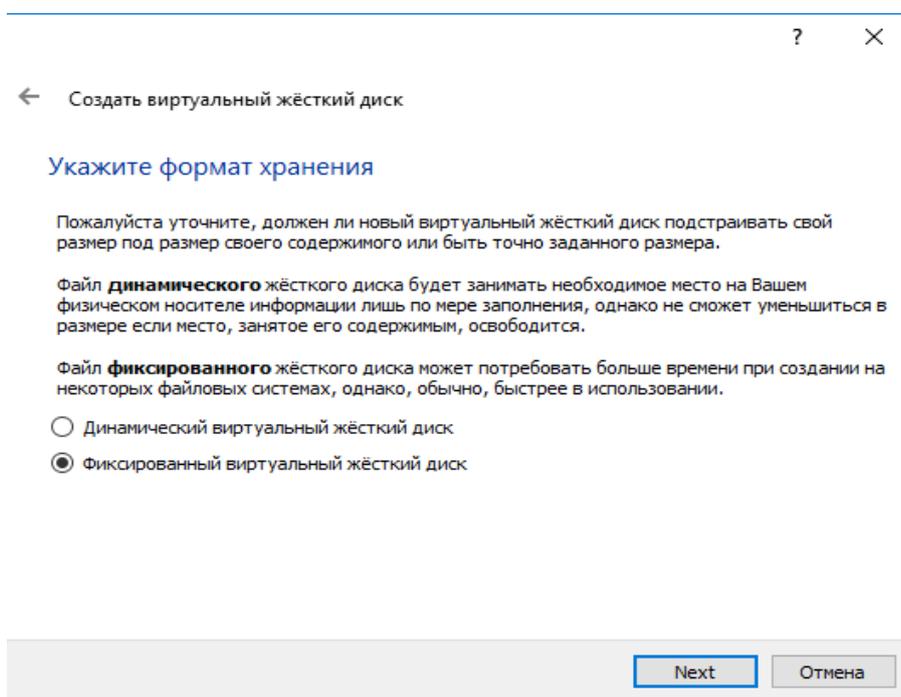


Рисунок 7. Выбор формата хранения для Linux Ubuntu

Выберите размер создаваемого жесткого диска (рисунок 8):

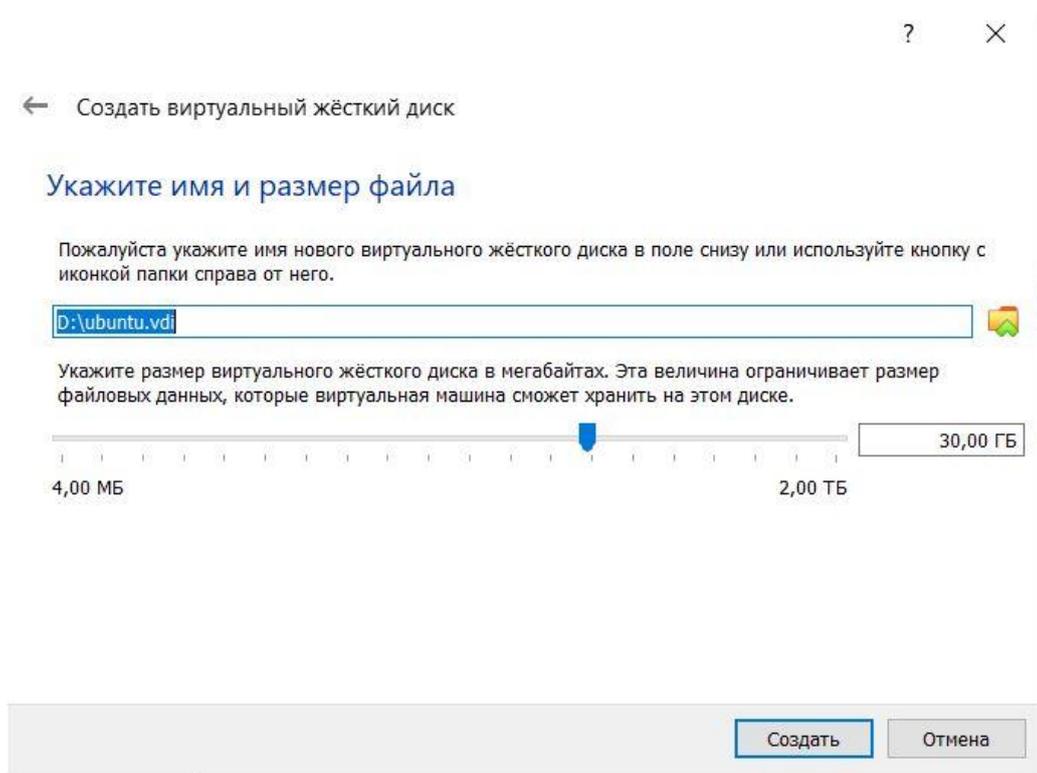


Рисунок 8. Выбор имени и размера жесткого диска для Linux Ubuntu

По умолчанию Oracle VM VirtualBox выделяет 16 МБ видеопамати для виртуальных машин, но современным дистрибутивам этого недостаточно, поэтому нажмите "Настройки", а затем на вкладке "Дисплей" установите значение видеопамати в 128 МБ. Кроме того, отметьте галочку 3D ускорения (рисунок 9):

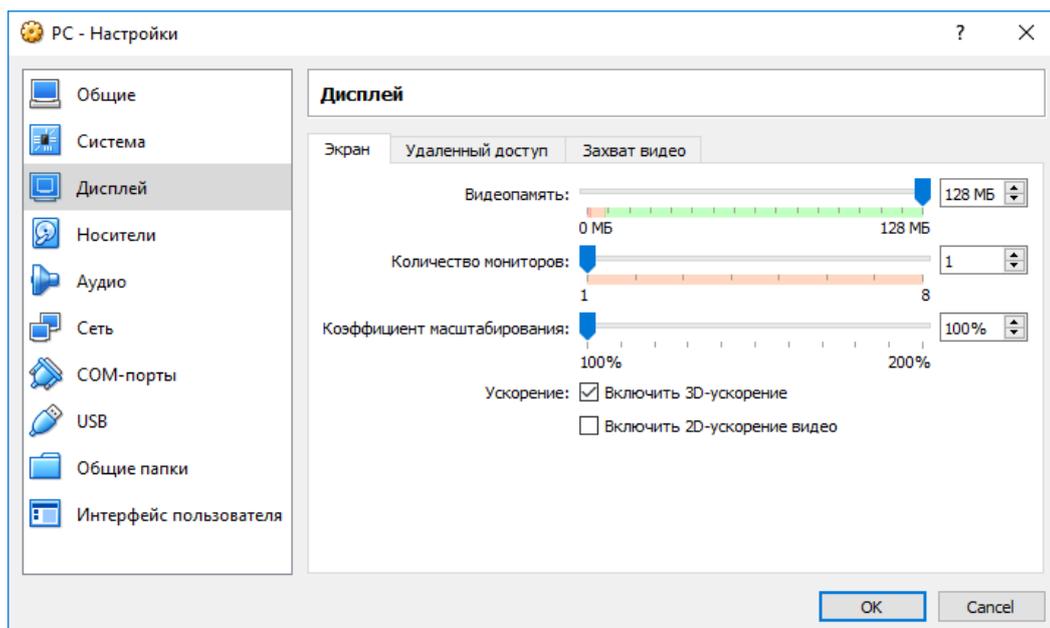


Рисунок 9. Настройки видеокарты виртуальной машины для Linux Ubuntu

В разделе "Система", во вкладке "Материнская плата", нужно выбрать чипсет "ICH9", Linux работает с ним лучше, также отметьте галочки часов в UTC (рисунок 10):

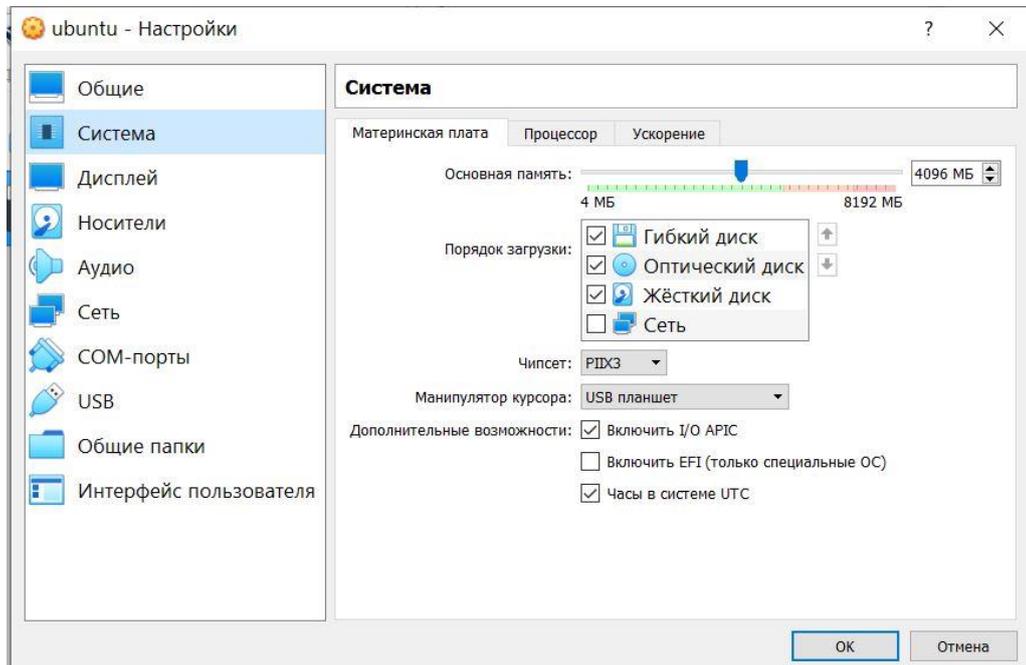


Рисунок 10. Настройки материнской платы виртуальной машины для Linux Ubuntu

На вкладке "Процессор" нужно отметить использовать "PAE/NX", выберите количество процессоров (рисунок 11). В разделе "Ускорение" отметьте все галочки и выберите гипервизор, в Linux лучше всего использовать KVM (рисунок 12):

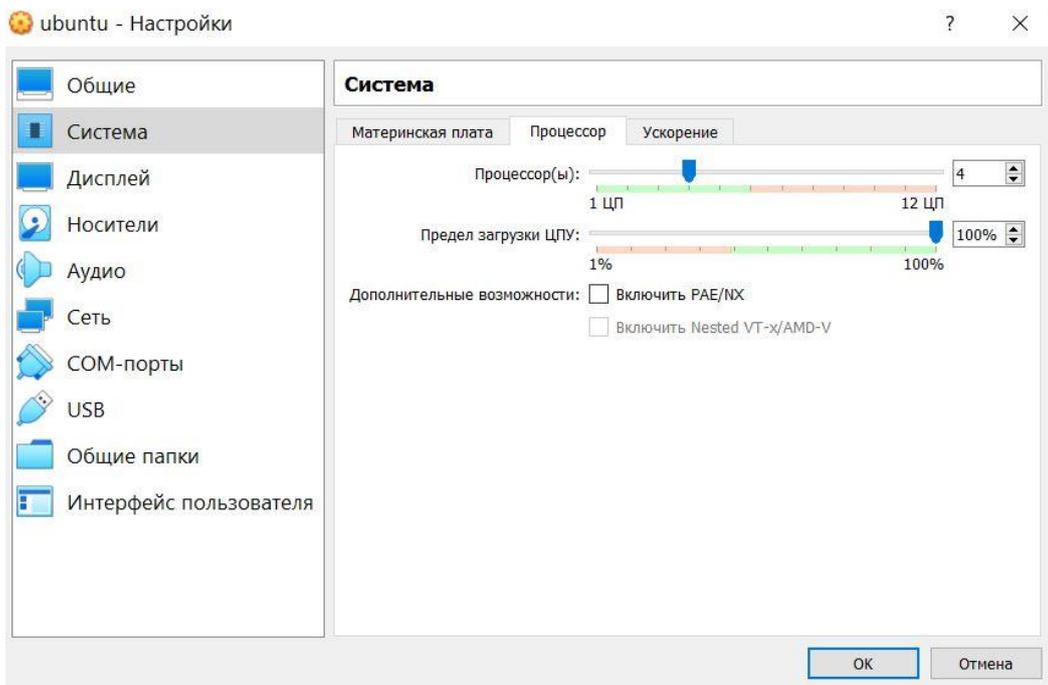


Рисунок 11. Настройка процессора виртуальной машины для Linux Ubuntu

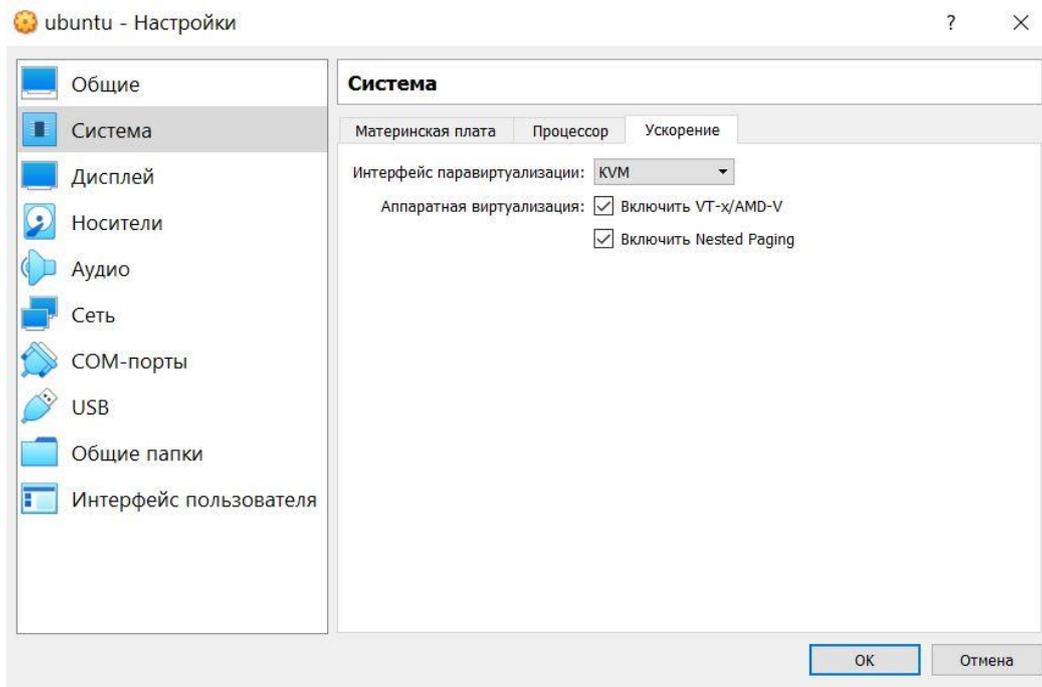


Рисунок 12. Настройка ускорения виртуальной машины для Linux Ubuntu

Теперь в разделе "Носители" сначала нажмите на пункт "Контроллер IDE", затем в атрибутах выберите "Оптический привод" и нажмите на значок диска. Выберите необходимый образ. Вы должны получить изображение как на рисунке 13. Нажмите "ОК".

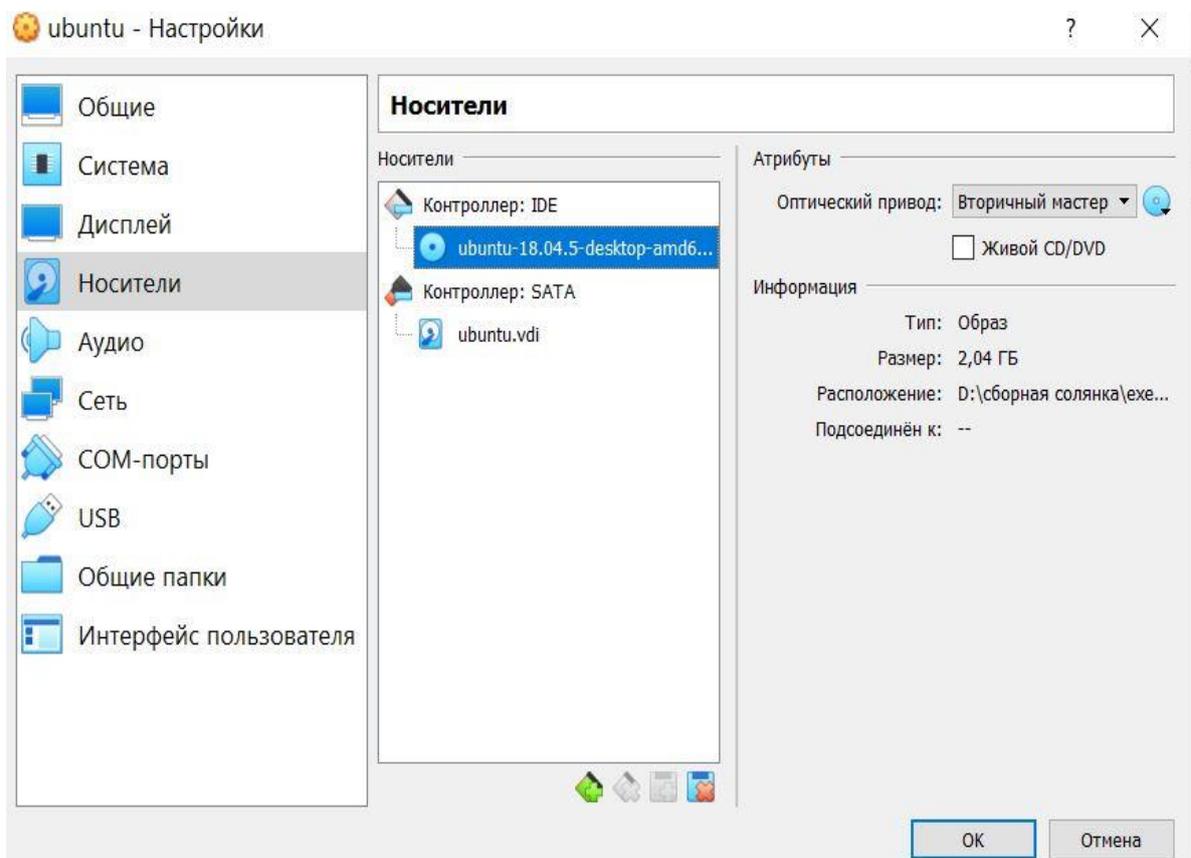


Рисунок 13. Выбор образа Linux Ubuntu 18.04.5 LTS

Переходим к следующему этапу.

Установка Linux Ubuntu 18.04.5 LTS

После включения виртуальной машины вы увидите экран как на рисунке 13, нажмите любую клавишу, чтобы запустить меню выбора языка (рисунок 14):

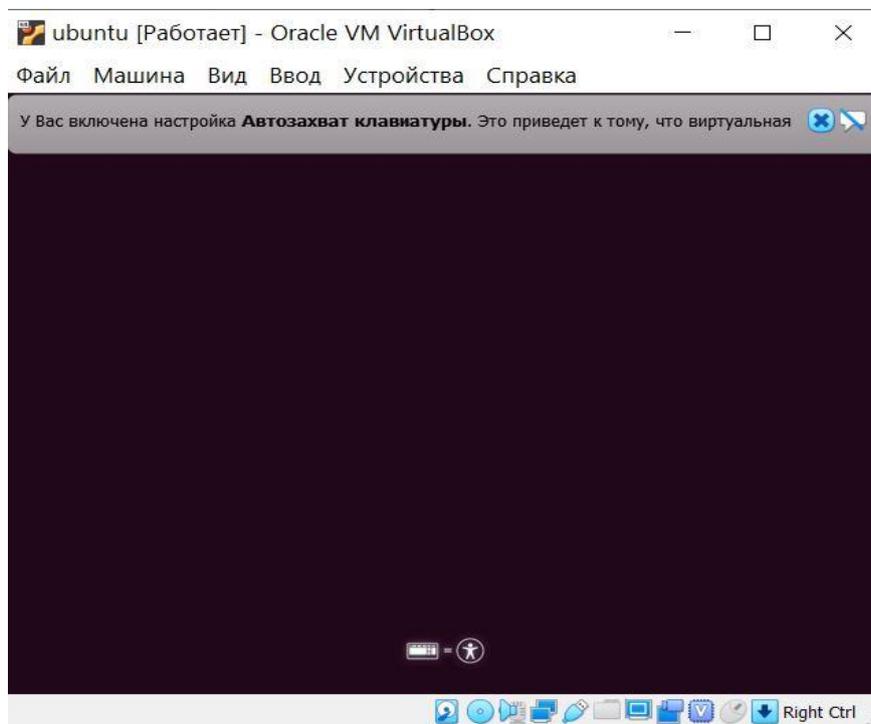


Рисунок 14. Начало установки Linux Ubuntu

Выберите язык, на котором хотите работать с системой (рисунок 15):

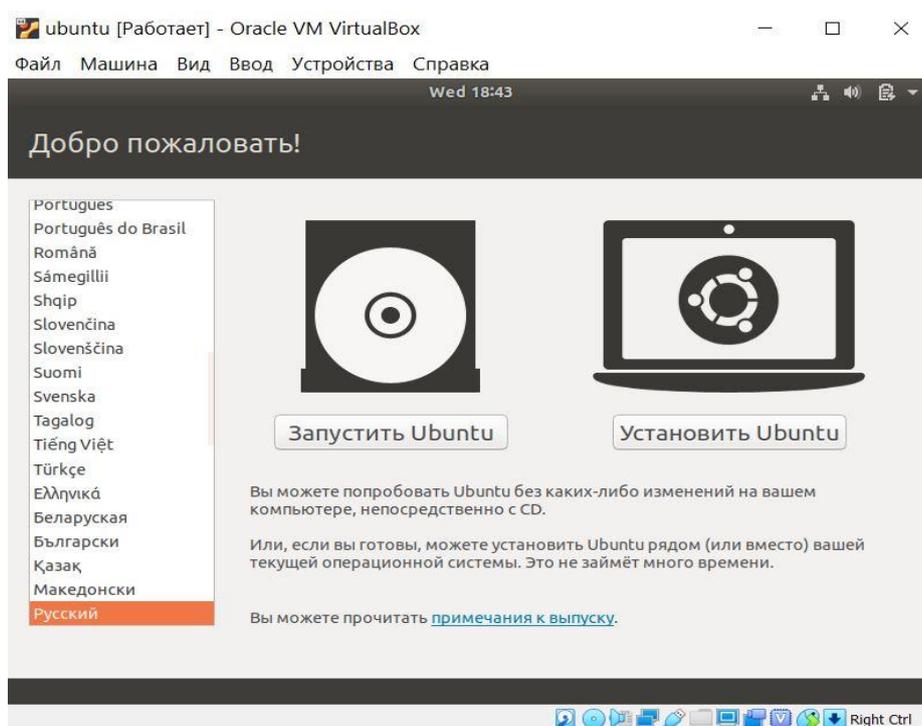


Рисунок 15. Выбор языка системы

Далее выбираем **Запустить Ubuntu**.

В таком режиме вы сможете ознакомиться с системой и даже пользоваться Интернетом. На данном этапе вы можете посмотреть, какие программы установлены, как работает система, и т.д. Когда завершите, нажмите на ярлык **Установить Ubuntu 18.04.5 LTS** (рисунок 16):

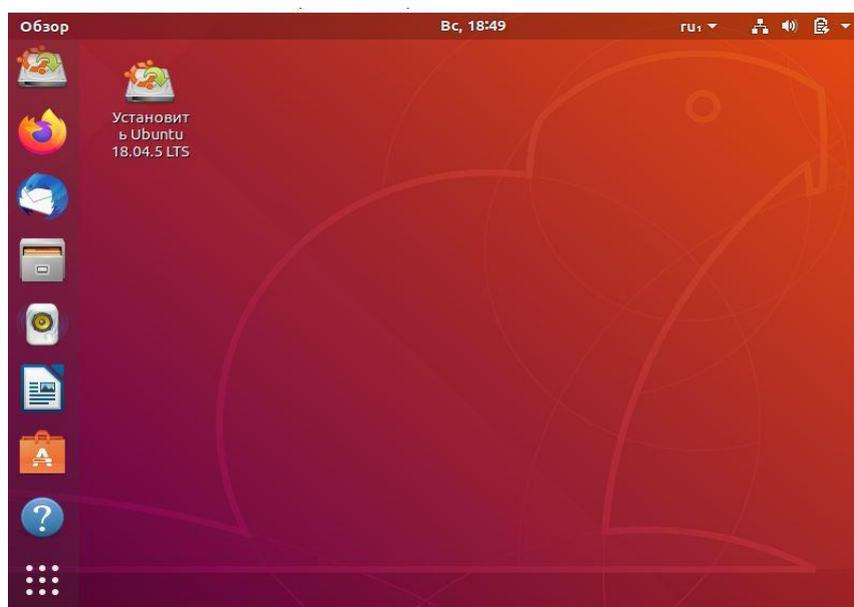


Рисунок 16. Запуск мастера установки

На первом шаге мастера нужно выбрать язык, на котором будет выполняться установка Ubuntu (рисунок 17):

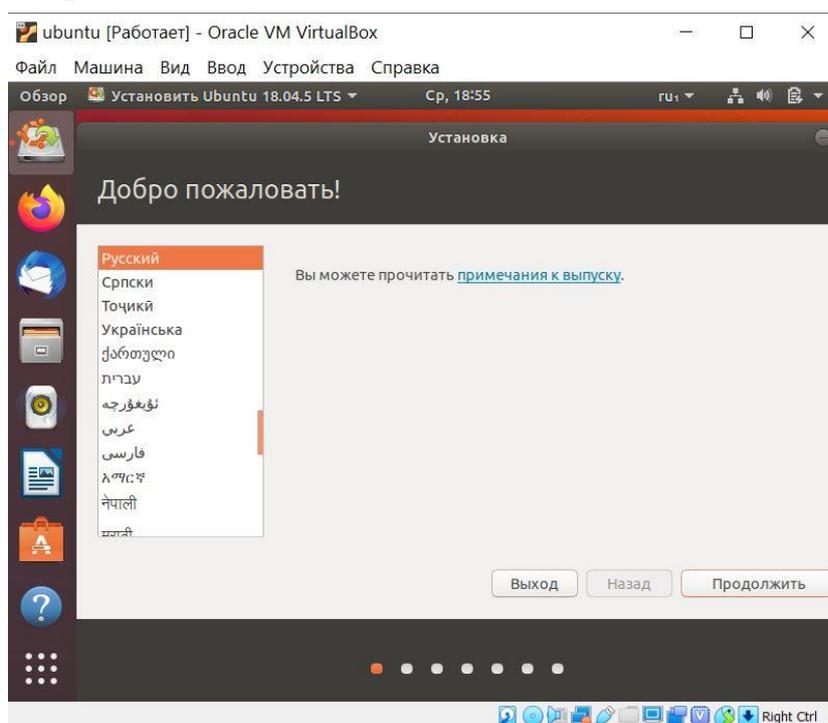


Рисунок 17. Выбор языка установки

Выберите раскладку клавиатуры по умолчанию (рисунок 18):

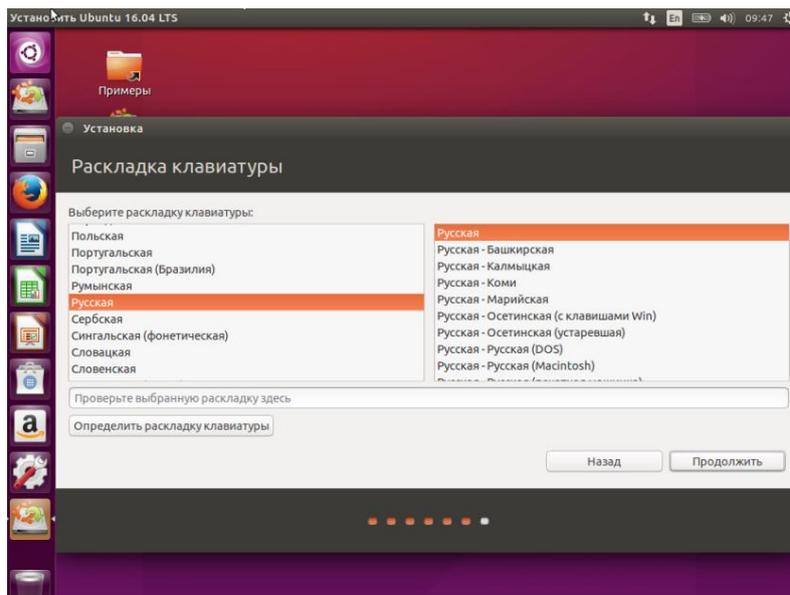


Рисунок 18. Выбор раскладки клавиатуры

Нажмите «Продолжить». Отметьте обе галочки, чтобы установить обновления и дополнительное ПО (рисунок 18):

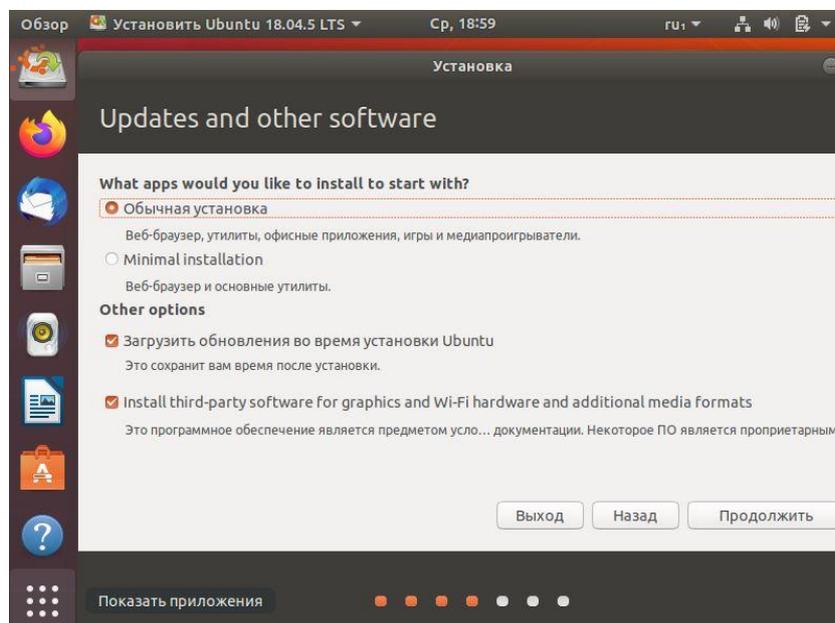


Рисунок 18. Подготовка к установке

Далее при выборе типа установки можно просто отформатировать весь жесткий диск и установить Ubuntu 18.04.5 на него, но рассмотрим ручной вариант разбивки диска (рисунок 19):

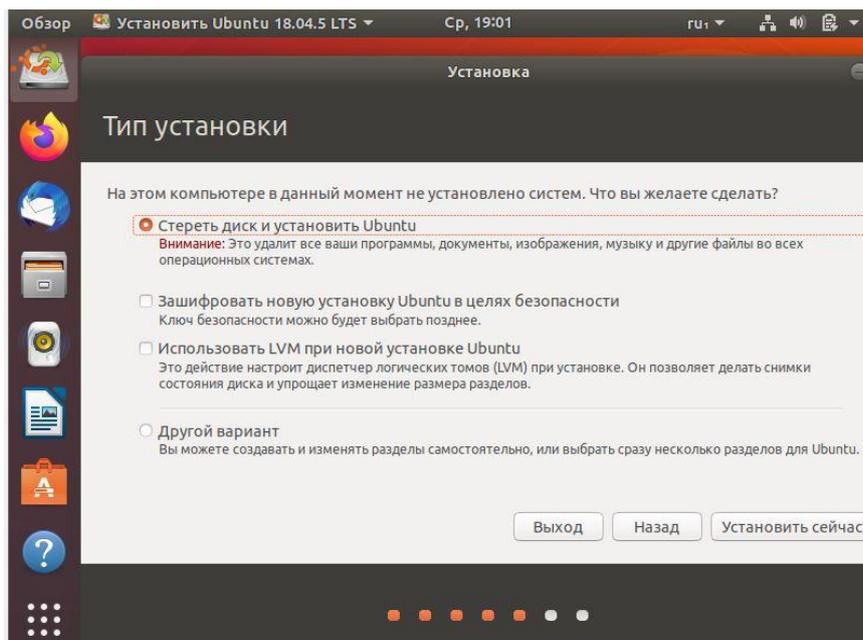


Рисунок 19. Определение типа установки

Установите переключатель в положение «Другой вариант» и нажмите «Далее». В открывшемся окне, если у вас ещё неразмеченный жесткий диск, нужно создать таблицу разделов, для этого нажмите кнопку «новая таблица разделов»(рисунок 20):

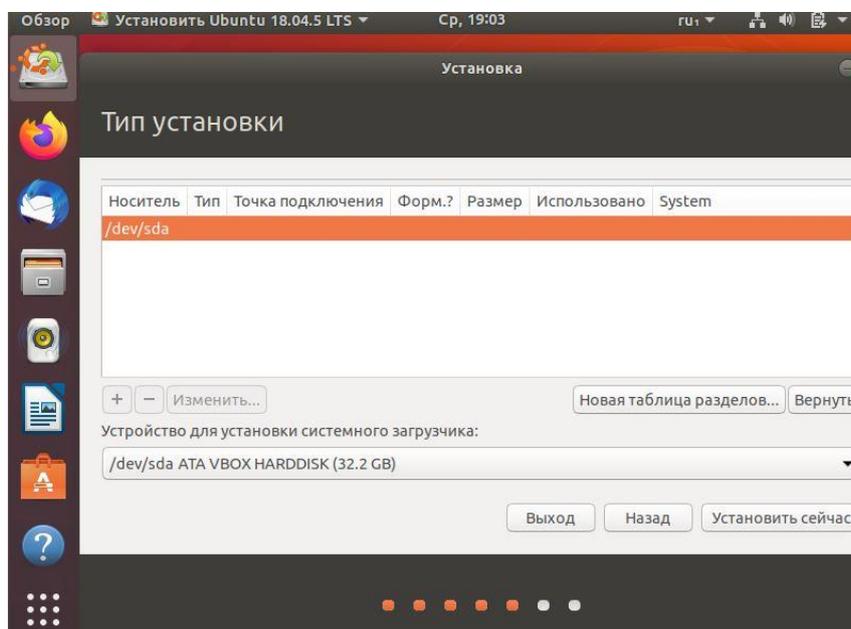


Рисунок 20. Создание таблицы разделов

В Linux рекомендуется использовать четыре раздела:

- / - ext4, размер 10-50 Гб, для установки системы;
- /boot - ext2, размер 500 Мб, для файлов загрузчика;
- swap - swar, размер равняется ОЗУ, для подкачки;

- **/home** - ext4, всё оставшееся пространство.

Чтобы создать новый раздел, нажмите кнопку + (рисунок 21):

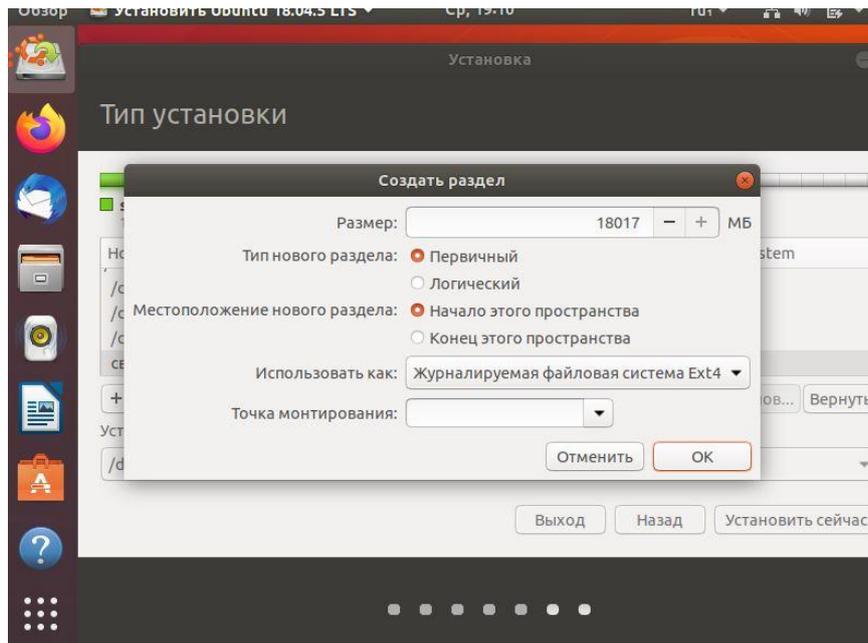


Рисунок 21. Создание раздела

Здесь нужно указать точку монтирования, например, / или **/home**, размер, файловую систему и можно задать метку (рисунок 22).

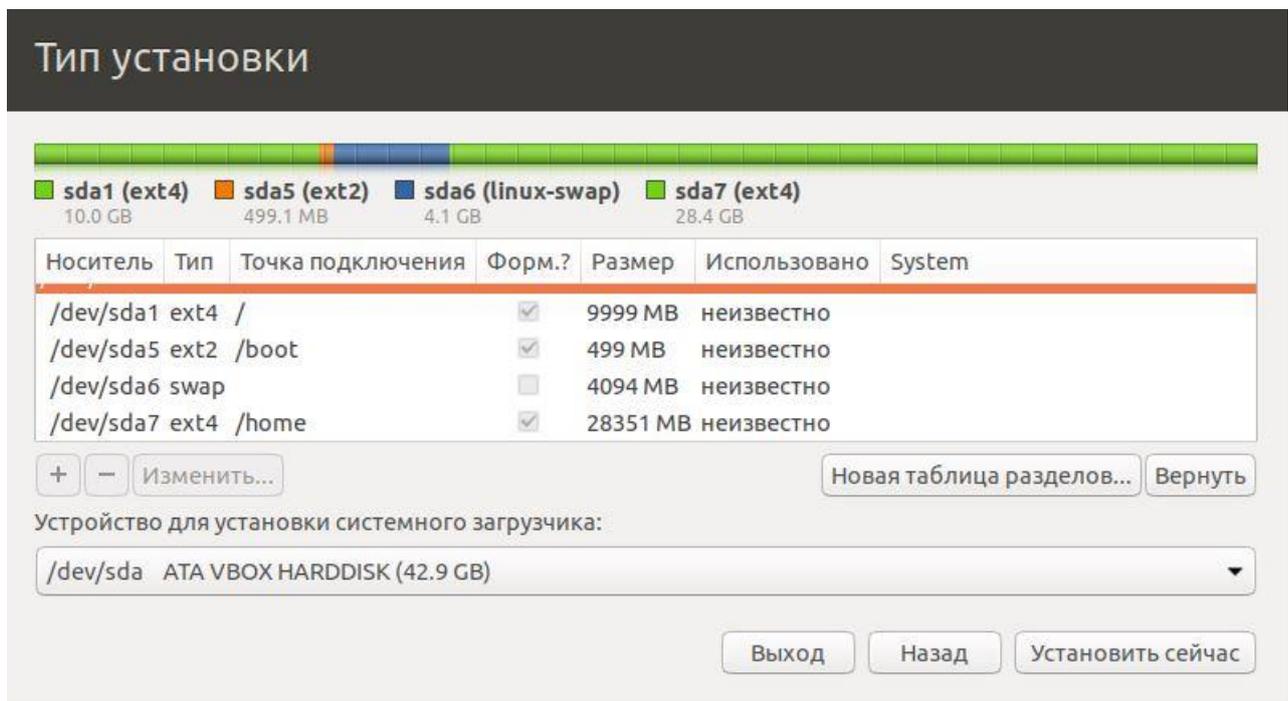


Рисунок 22. Размеченный жесткий диск

Потом нажмите «Далее». Проверьте, всё ли правильно и нажмите «Продолжить» (рисунок 23).

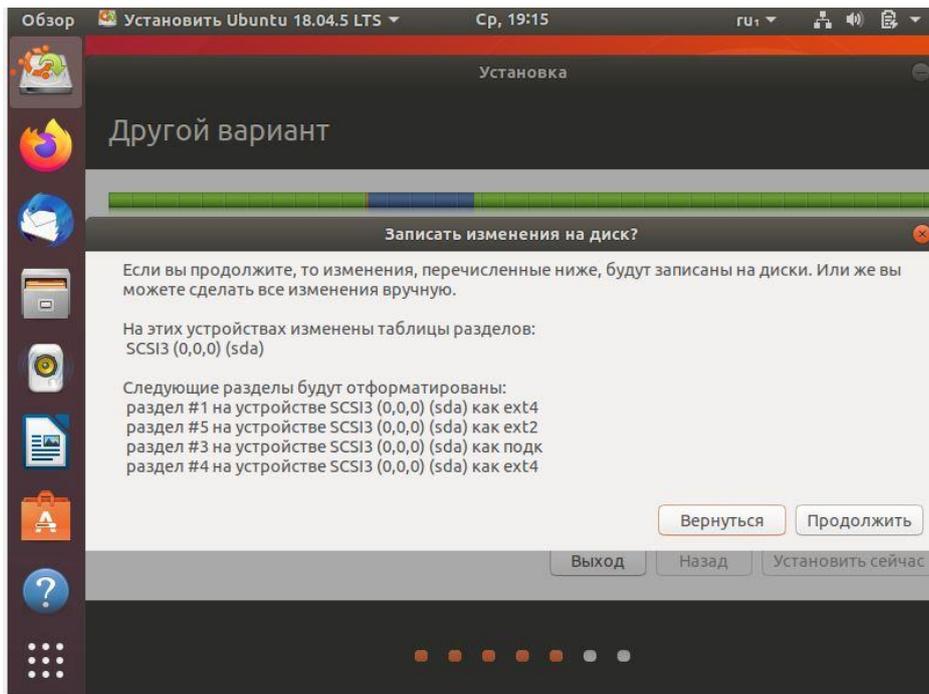


Рисунок 23. Проверка правильности настроек

Выберите свой часовой пояс (рисунок 24):

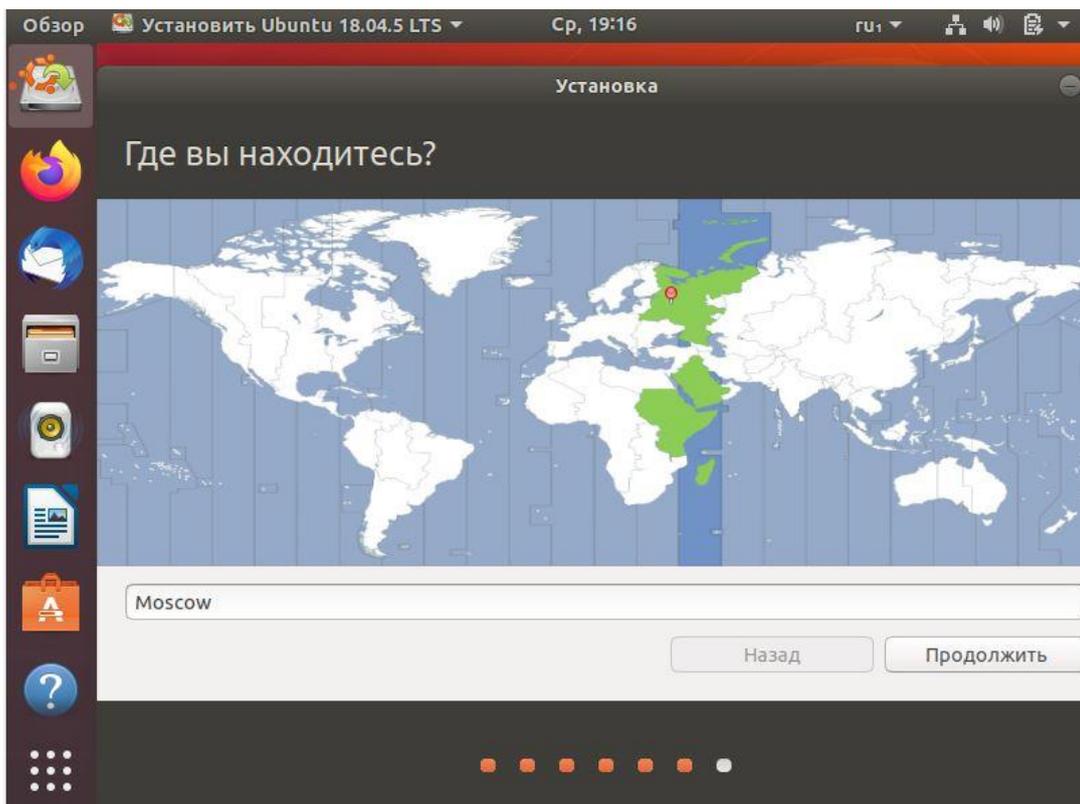


Рисунок 24. Выбор часового пояса

Введите имя пользователя, имя компьютера и пароль (рисунок 25):

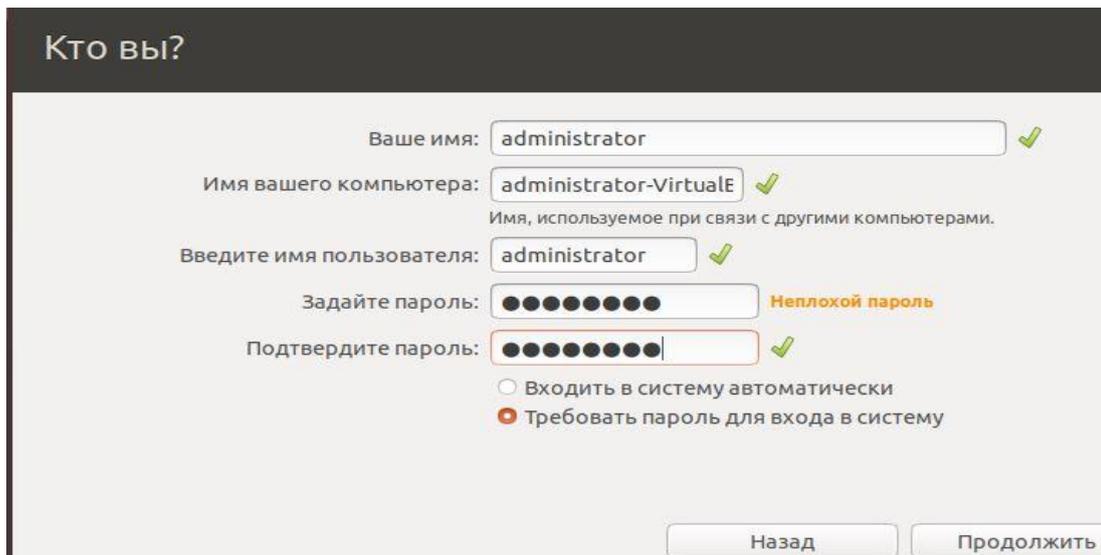


Рисунок 25. Ввод данных пользователя

Также здесь можно указать, нужно ли спрашивать пароль при входе в систему, и зашифровать домашнюю папку. Запишите введенные данные, они вам потребуются для дальнейшей работы с системой. Нажмите «Продолжить». Дождитесь завершения установки (рисунок 26):

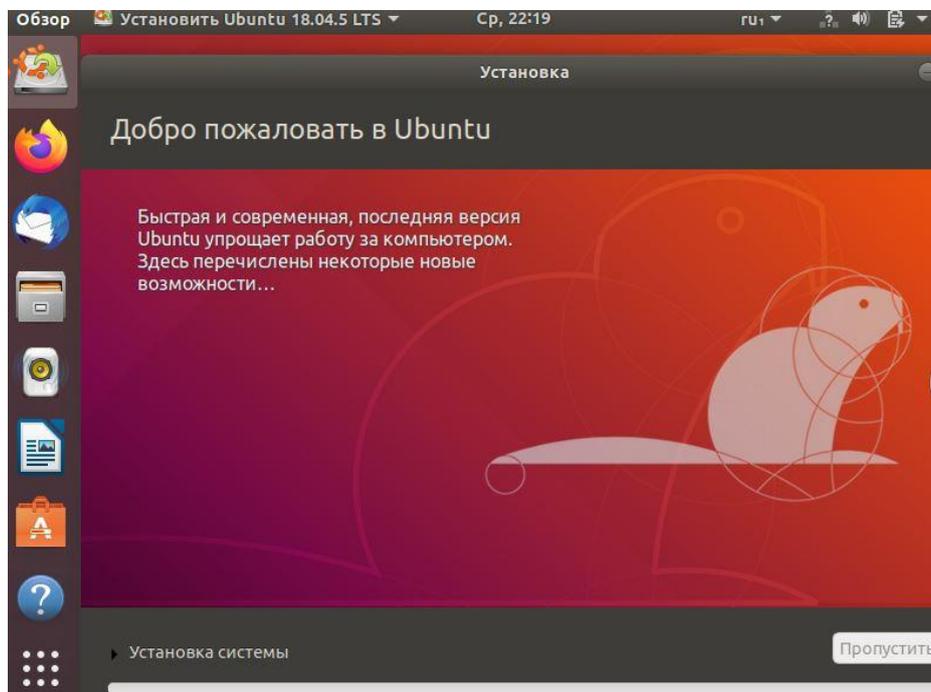


Рисунок 26. Установка Linux Ubuntu 18.04.5 LTS

Перезагрузите компьютер (рисунок 27) и извлеките установочный носитель нажав «Enter» во время загрузки ОС:

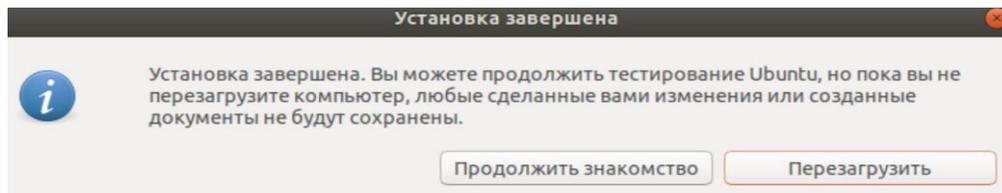


Рисунок 27. Завершение установки

Дождитесь завершения загрузки системы. Введите логин и пароль, указанный во время установки (рисунок 28):

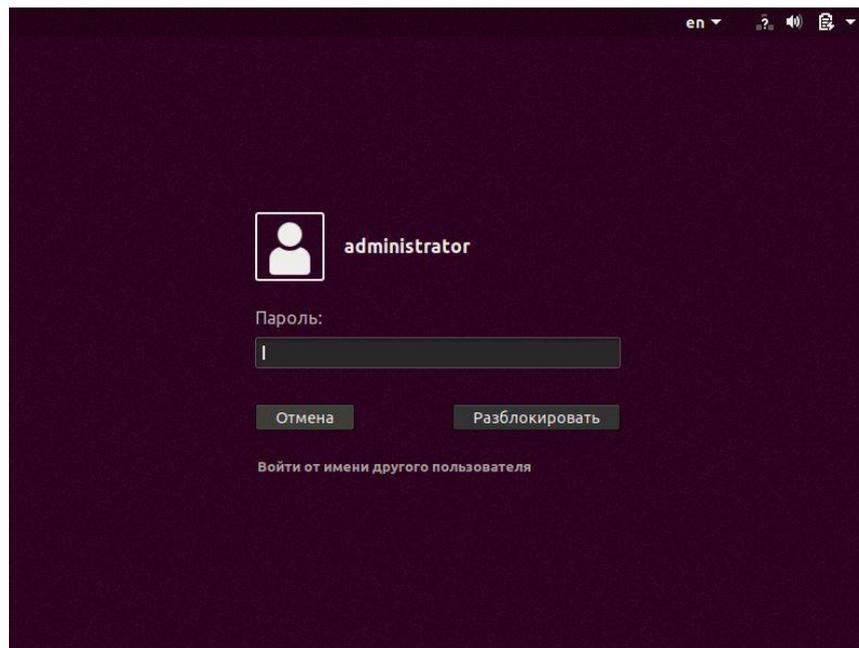


Рисунок 28. Вход в Linux Ubuntu 18.04.5 LTS

Система установлена и готова к использованию (рисунок 29):

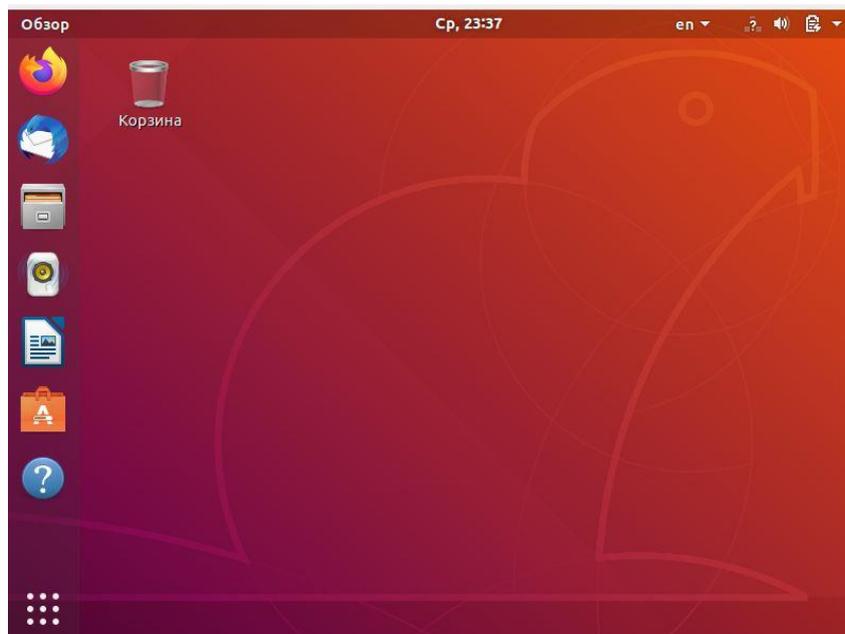


Рисунок 29. Рабочий стол Linux Ubuntu 18.04.5 LTS

Контрольные вопросы

1. Перечислить основные функции операционных систем.
2. Что такое ресурс процесса?
3. Что в себя включает дистрибутив операционной системы Linux?
4. По какой лицензии распространяется операционная система Linux?
5. Что такое PID в Linux?
6. Какой формат файла пакета обеспечивает совместимость между различными дистрибутивами ОС Linux?
7. Перечислить основные этапы установки ОС Linux.
8. Перечислить рекомендуемые разделы диска в Linux.

Практическая работа №2. Файловая система.

Краткая теория

Файловая система (ФС) - предоставляет пользователям (и процессам) ресурсы долговременной памяти компьютера. Операционная система Windows может быть установлена на файловую систему NTFS, поэтому обычно у пользователей не возникает вопросов какую ФС лучше использовать. ОС Linux имеет значительные отличия. Так, в ядро системы встроены и могут использоваться несколько файловых систем, каждая из которых оптимизирована для решения определенных задач.

Чтобы на каждом разделе можно было работать с файлами и каталогами, необходима файловая система. Кроме записи содержимого файлов на диск нужно еще хранить данные о папках, имена файлов, их размер, адрес на жестком диске, атрибуты доступа. Всем этим занимается файловая система. От файловой системы зависит очень многое: скорость работы с файлами, скорость записи и даже размер файлов. Также от стабильности файловой системы будет зависеть сохранность файлов.

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы. Далее будут рассмотрены типы файловых систем Linux, включая специальные файловые системы [2].

Основные файловые системы

Каждый дистрибутив Linux позволяет использовать одну из следующих файловых систем, имеющих свои преимущества и недостатки:

Ext2;

Ext3;

Ext4;

JFS;

ReiserFS;

XFS;

Btrfs;

ZFS.

Все они включены в ядро и могут использоваться в качестве корневой файловой системы. Рассмотрим каждую из них более подробно.

Ext2, Ext3, Ext4 (Extended Filesystem) - это стандартная файловая система для ОС Linux. Была разработана еще для Minix. Она самая стабильная из всех существующих ФС, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux.

В 2001 году вышла файловая система ext3, которая стала стабильнее в работе благодаря использованию журналирования. В 2006 была выпущена версия ext4, использующаяся сейчас во всех дистрибутивах Linux. В ней было внесено много изменений, в том числе увеличен максимальный размер раздела до одного экзбайта.

JFS (Journaled File System) была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

ReiserFS была разработана в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она разработана под

руководством Ганса Райзера и поддерживает только Linux. Особенностью ФС является динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, предотвращая фрагментацию и улучшая работу с небольшими файлами. Еще одно преимущество ФС - возможность изменять размеры разделов во время работы. Тем не менее, имеется недостаток в некоторой нестабильности работы и риске потери данных при отключении питания.

XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы в 2001 году. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов во время работы и незначительный размер служебной информации. XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе RedHat. Из недостатков ФС следует отметить отсутствие возможности уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания.

Btrfs или *B-Tree File System* - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановлении данных. Файловая система объединяет в себе очень много новых возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера во время работы, создание мгновенных снимков, а также высокая производительность. Однако многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

Другие файловые системы, такие как NTFS, FAT, HFS могут использоваться в ОС Linux, но корневая файловая система на них не устанавливается, поскольку они для этого не предназначены.

Специальные файловые системы

Ядро Linux использует специальные файловые системы, чтобы предоставить доступ пользователю и программам к своим настройкам и информации. Наиболее часто используются следующие специальные ФС:

tmpfs;

procfs;

`sysfs`;

Файловая система `tmpfs` позволяет размещать любые пользовательские файлы в оперативной памяти компьютера. Достаточно создать блочное устройство нужного размера, затем подключить его к каталогу, и пользователь может записывать файлы в оперативную память.

`procfs` - по умолчанию смонтирована в каталог `proc` и содержит всю информацию о запущенных в системе процессах, а также о самом ядре.

`sysfs` - с помощью этой файловой системы пользователь может задавать различные настройки ядра во время выполнения.

Основные каталоги (директории)

В этом разделе будут рассмотрены основные каталоги (директории), которые имеются у большинства дистрибутивов ОС Linux.

/ - корень

Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Здесь нет дисков, как в Windows. Вместо этого, адреса всех файлов начинаются с корня, а дополнительные разделы, usb-накопители или оптические диски подключаются в папки корневого каталога. Только пользователь "root" имеет право читать и изменять файлы в этом каталоге. Обратите внимание, что у пользователя "root" домашний каталог `/root`, но не `/`.

/bin - (binaries) бинарные файлы пользователя

Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. То есть утилиты, которые могут использоваться, пока не подключен каталог `/usr/`. Это такие общие команды, как `cat`, `ls`, `tail`, `ps` и т. д.

/sbin - (system binaries) системные исполняемые файлы

Так же как и `/bin` содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не подключен (не подмонтирован) каталог `/usr`. Здесь находятся программы, которые можно выполнять только с правами суперпользователя ("root"). Это разные утилиты для обслуживания системы, например, `iptables`, `reboot`, `fdisk`, `ifconfig`, `swapon` и т. д.

/etc - (etcetera) конфигурационные файлы

В этой директории содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, при использовании системы инициализации Init Scripts здесь находятся скрипты запуска и завершения системных сервисов, монтирования файловых систем и автозагрузки программ.

/dev - (devices) файлы устройств

В ОС Linux внешние устройства являются файлами. Таким образом, все подключенные usb-накопители, клавиатуры, микрофоны, камеры - файлы в каталоге /dev/. Этот каталог содержит не совсем обычную файловую систему. Структура файловой системы Linux и содержащиеся в каталоге /dev файлы инициализируются при загрузке системы сервисом udev. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. Это такие устройства, как: /dev/sda, /dev/sr0, /dev/tty1, /dev/usbmon0 и т. д.

/proc - (process) информация о процессах

Это тоже необычная файловая система, а подсистема, динамически создаваемая ядром. Здесь содержится вся информация о запущенных процессах в реальном времени. По сути это псевдофайловая система, содержащая подробную информацию о каждом процессе, его PID, имени исполняемого файла, параметрах запуска, доступе к оперативной памяти и так далее. Также в этом каталоге можно найти информацию об использовании системных ресурсов, например, /proc/cpuinfo, /proc/meminfo или /proc/uptime.

/var (variable) - переменные файлы

Название каталога /var указывает на то, что он содержит файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные csh-файлы, базы данных и так далее. Далее рассмотрим назначение каталогов Linux в директории /var/.

/var/log - файлы логов (файлы журналов)

В этой директории содержится большинство файлов логов всех программ, установленных в операционной системе. У многих программ есть свои подкаталоги в этой директории, например, /var/log/apache - логи веб-сервера, /var/log/squid -

файлы журналов сервера squid. Если в системе что-либо перестало работать, скорее всего, информацию об этом располагается в этой директории.

/var/lib - базы данных

В этой директории находятся файлы баз данных, пакеты, сохраненные пакетным менеджером и т.д.

/var/mail - почта

В эту директорию почтовый сервер складывает все полученные или отправленные электронные письма, здесь же могут находиться его логи и файлы конфигурации.

/var/spool - принтер

Каталог отвечает за очереди печати на принтере и работу набора программ cups.

/var/lock - файлы блокировок

В этой директории находятся файлы блокировок. Эти файлы означают, что определенный ресурс или файл занят и не может быть использован другим процессом. apt-get, например, блокирует свою базу данных, чтобы другие программы не могли ее использовать, пока программа с ней работает.

/var/run - PID процессов

Каталог содержит файлы с PID процессов, которые могут быть использованы, для взаимодействия между программами. В отличие от каталога /run данные сохраняются после перезагрузки.

/tmp (temp) - временные файлы

В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. Файлы удаляются при каждой перезагрузке.

/usr - (user applications) программы пользователя

Каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений и документация.

/usr/bin/ - исполняемые файлы

Каталог содержит исполняемые файлы различных программ, которые не нужны на первых этапах загрузки системы, например, музыкальные плееры, графические редакторы, браузеры и так далее.

/usr/sbin/

Каталог содержит двоичные файлы программ для системного администрирования, которые нужно выполнять с правами суперпользователя. Например, такие как Gparted, sshd, useradd, userdel и т. д.

/usr/lib/ - библиотеки

Каталог содержит библиотеки для программ из каталогов /usr/bin или /usr/sbin.

/usr/local - файлы пользователя

Каталог содержит файлы программ, библиотек, и настроек созданные пользователем. Например, здесь могут храниться программы собранные и установленные из исходников и скрипты, написанные вручную.

/home - домашняя директория

В этой директории хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т. д. Например: /home/имя_учетной_записи и т. д.

/boot - файлы загрузчика

Каталог содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящие в каталоге /boot/grub.

lib (library) - системные библиотеки

Каталог содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin. Библиотеки имеют имена файлов с расширением *.so и начинаются с префикса lib*. Например, libncurses.so.5.7. Папка /lib64 в 64 битных системах содержит 64 битные версии библиотек из /lib.

/opt (optional applications) - дополнительные программы

В эту директорию устанавливаются проприетарные программы или драйверы. Это программы, созданные в виде отдельных исполняемых файлов самими производителями. Такие программы устанавливаются в подкаталоги /opt/, они

очень похожи на программы Windows, все исполняемые файлы, библиотеки и файлы конфигурации находятся в одной папке.

/mnt (mount) - монтирование

В этот каталог системные администраторы могут монтировать(подключать) внешние или дополнительные файловые системы.

/media - съемные носители

В этот каталог система монтирует все подключаемые внешние накопители - usb-накопители, оптические диски и другие носители информации.

/srv (server) - сервер

В этом каталоге содержатся файлы серверов и сервисов. Например, могут содержаться файлы веб-сервера apache.

/run - процессы

Еще один каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в файловом хранилище TMPFS, поэтому после перезагрузки все файлы удаляются.

/sys (system) - информация о системе

Назначение каталогов Linux из этой директории - получение информации о системе непосредственно от ядра. Это еще одна файловая система организуемая ядром и позволяющая просматривать и изменить многие параметры работы системы.

Файлы устройств и монтирование

Все устройства (HDD, DVD, USB и др.) представлены в виде файлов. Они хранятся в каталоге /dev. Все HDD-диски называются однотипно /dev/sdX. X – буква от «a» до «z» соответствует порядку подключения физического диска к контроллеру. Например, если диск подключен к контроллеру первым, то его имя /dev/sda. Если подключить usb –диск, то ему будет дано имя /dev/sdb. На каждом диске может быть несколько разделов. Им даются имена, начиная с «1»: /dev/sda1, /dev/sda2 и т.д.

Для получения доступа к файлам и каталогам, находящимся на другом разделе, этот раздел нужно подмонтировать. Сменные носители монтируются

автоматически (обычно к каталогу /media/<UUID-устройства>, UUID – уникальный идентификатор устройства).

Для монтирования используется команда (в Ubuntu):

sudo mount <имя_устройства> <точка_монтирования>

(например, **sudo mount /dev/sdb1 /mnt/files**).

Точка монтирования – каталог, через который будет осуществляться доступ к монтируемой файловой системе. На момент осуществления операции монтирования точка монтирования должна существовать.

Командный интерпретатор

Все команды исполняются командным интерпретатором. Командный интерпретатор - это просто программа, принимающая команды пользователя и исполняющая их. Синтаксис команд в различных интерпретаторах может отличаться. В системе Linux наиболее популярны два командных интерпретатора - bash и tcsh, которые немного отличаются друг от друга.

Пользователи и группы

Linux является многопользовательской системой. Несколько пользователей могут работать с ней одновременно. Один пользователь может быть зарегистрирован в системе локально, другой – по сети, например через SSH.

Среди учетных записей нужно выделить пользователя "root" (для использования этой учетной записи необходимо ввести команду sudo su в Ubuntu, а для установки программного обеспечения достаточно только применить права "root" командой sudo). Он обладает максимальными правами в системе, поэтому при работе следует соблюдать осторожность. При работе с использованием обычных пользовательских учетных записей (без прав "root") максимальный вред может быть нанесен только файлам самих пользователей. Системные файлы останутся неизменными, т.к. система не позволит с ними ничего сделать. Ввод команды от имени "root" может привести к нерабочему состоянию всей системы.

Для более эффективного управления пользователей объединяют в группы. По умолчанию один пользователь не имеет права доступа к домашнему каталогу другого пользователя, т.к. они находятся в разных группах. При объединении их в одну группу пользователи получают эти права.

Права доступа к файлам

Создатель файла должен иметь возможность управлять списком допустимых операций над файлом и списком пользователей, которым они разрешены. В ОС Linux существуют следующие типы доступа:

- Read (r) – чтение;
- Write (w) – запись, переименование, удаление;
- Execute (x) – выполнение (запуск).

Права доступа состоят из трех наборов:

- 1) для владельца (кто создал файл), первые три символа;
- 2) для группы владельцев, вторые три символа;
- 3) для прочих пользователей, последние три символа.

Права доступа к файлу имеют следующий вид:

-rwxrwxrwx (первый символ признак каталога «-»). Пример команды для просмотра прав доступа:

ls -l file.txt

Пусть у файла "file.txt" в результате выполнения команды были определены права доступа -rw-r--r-- (или 110 100 100). Для выяснения этих прав необходимо воспользоваться следующим соответствием символьного обозначения и прав доступа:

- 100 – только чтение;
- 110 – чтение и запись;
- 101 – чтение и выполнение;
- 111 – чтение, запись и выполнение.

Таким образом, для рассматриваемого файла создатель, владелец файла имеет права на чтение и запись, группа владельца файла и другие пользователи имеют права только на чтение.

Для изменения прав доступа используется команда `chmod`. Права доступа у пользователя "root" максимальные, он может управлять любым файлом всех пользователей.

Контрольные вопросы

1. Какие типы файловых систем вы знаете?
2. Перечислите основные каталоги системы Linux.
3. Какие типы доступа есть в ОС Linux?
4. Какие типы пользователей системы вы знаете? Какие права имеет пользователь с логином root?
5. Как найти подключенный съемный накопитель в Linux?
6. Где располагаются исполняемые файлы в Linux?
7. В какой каталог системные администраторы могут монтировать(подключать) внешние или дополнительные файловые системы?
8. Как называется домашняя директория в ОС Linux?
9. Какое символьное обозначение прав будет соответствовать разрешению на чтение и запись?
10. В какой директории содержится большинство файлов логов всех программ, установленных в операционной системе?

Практическая работа №3. Основные команды для администрирования ОС Linux

Краткая теория

Основные команды для администрирования ОС Linux

Для работы с ОС Linux используется командная строка терминала. Ниже представлены основные команды, использующиеся для администрирования [6]. *Linux является чувствительным к регистру. Команды `ls -a` и `ls -A`, а также файлы `text_file.txt` и `Text_file.txt` не обозначают один и тот же объект.*

Команды для управления файлами и каталогами

ls

Утилита для просмотра содержимого каталогов. По умолчанию показывает текущий каталог. Если в параметрах указать путь, то она перечислит содержимое конечного каталога. Полезные опции `-l` (List) и `-a` (All). Первая форматирует вывод в виде списка с более подробной информацией, а вторая включает показ скрытых файлов.

cat

Печатает содержимое файла, переданного в параметре, в стандартный вывод. Если передать несколько файлов, команда склеит их. Также можно перенаправить вывод в ещё один файл с помощью символа '>'. Если нужно вывести только определенное количество строк, используйте опцию -n (Number).

tac

В отличие от cat печатает файл в обратном виде.

cd

Позволяет перейти из текущего каталога в указанный пользователем. Если запустить без параметров - возвращает в домашний каталог. Вызов с двумя точками возвращает на уровень вверх относительно текущего каталога. Вызов с тире (cd -) возвращает к предыдущему каталогу.

touch

Команда изменяет время модификации файла, а если файл не существует, создает его.

pwd

Печатает на экран текущий каталог, в котором находится пользователь. Это может быть полезно, если командная строка Linux не выводит такую информацию. Эта команда будет востребована в Bash программировании, где для получения ссылки на каталог выполняется скрипт.

mkdir

Создание новых каталогов. Опция -p (Parents), позволяет создать всю структуру подкаталогов одной командой, даже если они ещё не существуют.

file

Показывает тип файла. В Linux файлы не всегда должны иметь расширения, чтобы с ними работать.

cp

Копирование файлов и каталогов.

mv

Перемещение или переименование файлов и каталогов. Важно отметить, что в Linux это одна и та же операция. Переименование - это перемещение файла в ту же директорию с другим именем.

chmod

Изменяет права доступа к файлу. Это чтение, запись и выполнение. Каждый пользователь может изменять права для своих файлов.

chown

Изменяет владельца файла. Только суперпользователь может изменять владельцев.

find

Поиск в файловой системе, файлах и папках.

locate

В отличие от find, команда locate ведёт поиск в базе данных updatedb для шаблонов имён файлов.

du

Показывает размер файла или каталога. Самые полезные опций: -h (Human), которая преобразует размеры файлов в легко читаемый формат, -s (Summarize), которая выводит минимум данных, и -d (Depth), устанавливающая глубину рекурсии по каталогам.

df

Анализатор дискового пространства. По умолчанию вывод достаточно подробный: перечислены все файловые системы, их размер, количество использованного и свободного пространства. Для удобства есть опция -h, делающая размеры легко читаемыми.

free

Вывод информация об использовании оперативной памяти. Для удобства есть опция -h, делающая размеры легко читаемыми.

mount / umount

Это команды консоли Linux для подключения и отключения файловых систем Linux. Только у суперпользователя есть права для этого.

Команды консоли для работы с текстом

more / less

Это две команды терминала для просмотра длинных текстов, которые не помещаются на одном экране. Если ваш терминал не поддерживает прокрутки, вы можете сделать это с помощью команды `less`, которая поддерживает больше опций.

head / tail

Команда `head` выводит несколько первых строк из файла, а `tail` выдает несколько последних строк. По умолчанию каждая утилита выводит десять строк, но это можно изменить с помощью опции `-n`. Ещё один полезный параметр `-f`, это сокращение от `follow` (следовать). Утилита постоянно выводит изменения в файле на экран. Например, если нужно следить за лог-файлом, вместо того, чтобы постоянно открывать и закрывать его, используйте команду `tail -nf`.

grep

Команда `grep` ищет текст по шаблону. По умолчанию она принимает стандартный ввод, но возможно искать в файлах. Шаблон может быть строкой или регулярным выражением. Утилита может вывести как совпадающие, так и не совпадающие строки и их контекст.

sort

Сортировка строк текста по различным критериям. Наиболее полезные опции: `-n` (Numeric), по числовому значению, и `-r` (Reverse), которая переворачивает вывод.

wc

Утилита командной строки Linux для подсчёта количества слов, строк, байт и символов.

diff

Показывает различия между двумя файлами в построчном сравнении. Причём выводятся только строки, в которых обнаружены отличия. Измененные строки отмечаются символом "с", удаленные - "d", а новые - "а".

Команды для управления процессами

ps / pgrep

Один из способов получить идентификатор процесса, это утилита `ps`, которая печатает информацию о запущенных процессах. По умолчанию вывод очень длинный, поэтому следует использовать опцию `-e`, чтобы увидеть информацию об определённом процессе. Выводится снимок состояния на момент вызова, и

информация не будет обновляться. Команда `ps` с ключом `aux` выводит полную информацию о процессах. `ps` работает следующим образом: задается имя процесса, а утилита показывает его идентификатор.

top / htop

Обе команды отображают процессы и могут быть использованы как консольные системные мониторы. Возможно не только просматривать, но и контролировать процессы через интерактивный интерфейс `htop`.

time

Время выполнения процесса. Это секундомер для выполнения программы. Не сообщает текущее время.

Команды окружения пользователя

su / sudo

`su` и `sudo` - это два способа выполнить одну и ту же задачу: запустить программу от имени "root". В зависимости от дистрибутива используется `su` или `sudo`. Разница в том, что `su` переключает на пользователя "root", а `sudo` только выполняет команду от его имени. Поэтому использование `sudo` будет наиболее безопасным вариантом работы.

date

Выводит дату и время в стандартный вывод. Его можно форматировать в зависимости от потребностей: вывести год, месяц, день, установить 12-ти или 24-ти часовой формат, получить наносекунды или номер недели. Например, `date +"%j %V"`, выведет день в году и номер недели в формате ISO.

alias

Команда создаёт синонимы для других команд Linux. Возможно делать новые команды или группы команд, а также переименовывать существующие.

uname

Выводит основную информацию о системе. Если задать параметр `-a` (All), то можно получить информацию о ядре, имени хоста и узнать архитектуру процессора.

uptime

Сообщает время работы системы.

sleep

Для выключения компьютера через определенный промежуток времени или использования в качестве импровизированной тревоги.

poweroff или **shutdown –h now**

Завершить работу системы прямо сейчас и выключить питание.

reboot **shutdown –r now**

Перезагрузить систему прямо сейчас.

users

Информация о том, кто вошел в систему.

w

Выводит информацию о том, откуда вошел пользователь (т.е. его IP-адрес), как именно, когда, а также информацию об использовании процессора.

who

Выводит только имя пользователя, название консоли, время и дату входа и IP-адрес.

Команды для управления пользователями

useradd / userdel / usermod

Эти команды консоли Linux позволяют добавлять, удалять и изменять учетные записи пользователей.

passwd

Эта команда позволяет изменить пароль учетной записи пользователя. Суперпользователь может сбросить пароли всех пользователей, даже несмотря на то, что не может их увидеть.

Команды для просмотра документации

man / whatis

Команда man открывает руководство по определённой команде. Для всех основных команд Linux есть man-страницы. whatis показывает, какие разделы руководств есть для данной команды.

whereis

Показывает полный путь к исполняемому файлу программы. Также может показать путь к исходникам, если они есть в системе.

Команды для управления сетью

ip

В пакете net-tools содержится множество утилит: ipconfig, netstat и прочие устаревшие, например, iproute2. Всё это заменяет одна утилита - ip.

ping

Быстро проверяет доступность узла, подключение к маршрутизатору или к Интернету и дает представление о качестве связи.

nethogs

Позволяет узнать, сколько трафика использует какая-либо программа в Linux или какая программа потребляет всю скорость Интернет соединения. Для того чтобы задать сетевой интерфейс используется опция -i.

traceroute

Это усовершенствованная версия ping. Позволяет увидеть не только полный маршрут сетевых пакетов, но и доступность узла, а также время доставки этих пакетов на каждый из узлов.

route

Вывод локальной таблицы маршрутизации (route -n).

ifconfig

Вывод информации о сетевом адаптере.

tracert

Аналог traceroute.

Установка, обновление и полное удаление утилит и пакетов

В ОС Windows есть понятие установщика программы, как правило, это файл с названием "setup.exe", который выполняет установку программы. В ОС Linux все программы распространяются в пакетах – в файлах с расширением ".rpm" (в дистрибутивах RedHat, Fedora, CentOS) или ".deb" (в дистрибутивах Debian, Ubuntu). Пакет содержит следующую информацию:

- программу – один или несколько исполнимых файлов;
- файлы, необходимые для работы программы (данные, конфигурации);
- документацию;
- инструкции, которые нужно выполнить при установке и удалении программы;
- информацию о зависимостях программы (некоторые программы требуют для своей установки наличия других программ);
- информацию о конфликтах программы (некоторые программы не могут работать одновременно с другими);
- информацию о разработчике программы.

Менеджер пакетов позволяет разрешать зависимости и конфликты при установке программ. Пакеты хранятся в репозиториях (электронное хранилище данных), у каждого дистрибутива есть собственный набор репозиторияев. Существует два основных менеджера пакетов: apt (в дистрибутивах Debian, Ubuntu) и dnf (в дистрибутивах Fedora, CentOS).

В Linux Ubuntu формат вызова менеджера пакетов имеет следующий вид:

sudo apt [опции] команды [пакет]

sudo apt-get [опции] команды [пакет]

Первая команда используется в более новых дистрибутивах, а вторая – как в новых, так и в старых. Для установки программы или утилиты необходимо ввести команду (для завершения команд нажмите клавишу **Tab**):

sudo apt-get install <название_программы>

Для обновления программы или утилиты необходимо ввести команду:

sudo apt-get upgrade <название_программы>

Для полного удаления программы или утилиты необходимо ввести команду (перед "auto-remove" два символа "-"):

sudo apt-get purge -- auto-remove <название_программы>

Важно отметить, что при обновлении системы и установке утилит требуется использовать права пользователя "root". Для этого необходимо перед командой установки или обновления ввести команду sudo. При этом будет запрошен пароль, который был введен при установке ОС Linux. Вводимые символы пароля не отображаются. Также стоит отметить, что во время установки или обновления

может потребоваться подтверждение действий, например, на вопрос «Хотите продолжить?» следует ввести ответ "y".

Контрольные вопросы

1. Какие вы знаете команды для управления каталогами? Перечислите.
2. Какие команды применяются для управления сетью?
3. Какой командой можно проверить подключение к интернету?
4. При помощи каких команд можно просмотреть руководство по определённой команде?
5. Как вывести дату и время в консоли Linux?
6. Как отсортировать строки по числовому значению?
7. Какая команда позволяет узнать имя текущего каталога?
8. Какие команды консоли Linux используются для подключения и отключения файловых систем Linux?
9. С помощью какой команды можно переименовать файл или каталог?
10. Как выглядит формат вызова менеджера пакетов в Linux Ubuntu?

Лабораторная работа №2. Изучение основ работы с каталогами

Цель работы

Ознакомиться с основными каталогами операционной системы Linux

Задание

1. Ознакомиться с основными каталогами операционной системы
2. Сделать скриншоты
3. Подготовить отчет

Краткая теория

После запуска терминала (сочетание клавиш "Ctrl+Alt+t") пользователь получает возможность вводить различные команды для работы с ОС Linux. Изначально пользователь находится в своей домашней директории "/home/имя_учетной_записи". Для того чтобы переместиться в каталог "/home" введите команду (рисунок 1):

```
administrator@administrator-VirtualBox: /
Файл Правка Вид Поиск Терминал Справка
administrator@administrator-VirtualBox:~$ cd /home
administrator@administrator-VirtualBox:/home$ cd administrator
administrator@administrator-VirtualBox:~$ pwd
/home/administrator
administrator@administrator-VirtualBox:~$ cd ..
administrator@administrator-VirtualBox:/home$ cd ..
administrator@administrator-VirtualBox:/home$ cd administrator
bash: cd: administrator: Нет такого файла или каталога
administrator@administrator-VirtualBox:/home$ cd /
administrator@administrator-VirtualBox:/home$ ls
bin      dev      initrd.img  lib64      mnt      root     snap     tmp     vmlinuz
boot    etc      initrd.img.old  lost+found  opt      run      srv      usr     vmlinuz.old
cdrom   home    lib         media      proc     sbin     sys      var
administrator@administrator-VirtualBox:/home$ ls /home/administrator
examples.desktop  Документы  Изображения  Общедоступные  Шаблоны
Видео             Загрузки  Музыка        'Рабочий стол'
administrator@administrator-VirtualBox:/home$ ls home
administrator  lost+found
administrator@administrator-VirtualBox:/home$
```

Рисунок 1. Изучение основ работы с каталогами

cd /home

Для возвращения обратно в домашний каталог `"/home/имя_учетной_записи"` используйте команду, позволяющую перейти в каталог, находящийся внутри данного каталога (внутри каталога `"/home"` имеется вложенный каталог `"имя_учетной_записи"`):

cd имя_учетной_записи

или команду, которая позволяет перейти в домашний каталог из любого другого каталога системы:

cd ~

Для определения каталога, в котором находится сейчас пользователь, введите команду

pwd

Для перехода из вложенного каталога на один уровень вверх, например, из вложенного каталога `"/home/имя_учетной_записи"` в каталог `"/home"`, введите команду:

cd ..

Перейдите в корневой каталог при помощи команды:

cd /

Просмотрите содержимое каталога (без вывода в терминал скрытых папок и информации о правах доступа) при помощи команды:

ls

Пользователь, находясь в одном каталоге, имеет возможность просматривать содержимое других каталогов. Так, находясь в корневом каталоге "/", пользователь может просмотреть содержимое каталога "/home/имя_учетной_записи". Введите команду:

ls home/имя_учетной_записи

Для просмотра содержимого только каталога "/home" следует ввести команду:

ls home

Получение информации о пользователях в системе

Во время начала работы с терминалом пользователю выводится приветствие операционной системы

```
имя_учетной_записи@имя_домена-VirtualBox:~$
```

В зависимости от того, в каком каталоге находится пользователь, оно может иметь разный вид:

```
имя_учетной_записи@имя_домена-VirtualBox:~$           или  
имя_учетной_записи@имя_домена-VirtualBox:/var$.
```

Из приветствия операционной системы можно получить следующую информацию:

имя_учетной_записи – имя текущего пользователя;

имя_домена-VirtualBox – имя хоста (например, сервера);

~ или /var – имя каталога, в котором сейчас находится пользователь ("~" - домашний каталог пользователя, "/var" – каталог переменных файлов);

\$ - символ, отделяющий приветствие от последующей команды.

Для определения количества пользователей, их имен, способа входа в операционную систему и времени пребывания в сеансе работы существуют специальные команды (рисунок 2). Для определения пользователя, который вошел в систему применяется команда:

users

```
administrator@administrator-VirtualBox:/$ users
administrator
administrator@administrator-VirtualBox:/$ who
administrator :0                2021-08-25 12:31 (:0)
administrator@administrator-VirtualBox:/$
```

Рисунок 2. Информация о пользователях в системе

Для определения информации о том, кто работает в системе: имя пользователей, название консоли, время и дату входа и IP-адрес (если пользователь подключился удаленно) применяется команда:

who

Для вывода информации о том, откуда вошел пользователь (его IP-адрес), как именно он вошел, когда, информацию об использовании процессора можно получить с помощью команды:

w

Контрольные вопросы

1. Какое сочетание клавиш можно использовать для вызова терминала?
2. Как создать папку или директорию?
3. Какая информация содержится при выводе команды **who**?
4. Для чего применяется команда **cd**?
5. Какую информацию можно получить из приветствия операционной системы?
6. Как вернуться обратно в домашний каталог?
7. Как узнать откуда вошел пользователь?
8. Как перебраться из каталога на 1 уровень выше?
9. Какая команда используется для определения, какой пользователь авторизован сейчас в системе?
10. Напишите имя хоста своей виртуальной машины.

Лабораторная работа №3. Операции с пользователями в системе

Цель работы

Научиться управлять пользователями в операционной системе Linux

Задание

1. Ознакомиться с теоретическим материалом;
2. Создать нового пользователя;
3. Создать и присвоить новому пользователю группу;
4. Проверить создание пользователя;
5. Создать пользователя в директории, отличной от дефолтной;
6. Удалить созданных пользователей;
7. Сделать скриншоты;
8. Ответить на вопросы;
9. Сформировать отчет.

Краткая теория

В ОС Linux существует три типа пользователей:

- **root** (от англ. root - корень) - суперпользователь, аккаунт в UNIX-подобных системах, владелец которого имеет право на выполнение всех операций без исключения. Присутствует в системе по умолчанию.

- **Системные пользователи** - системные процессы у которых есть учетные записи для управления привилегиями и правами доступа к файлам и каталогам. Создаются системой автоматически.

- **Обычные пользователи** - учетные записи пользователей, допущенных к управлению системой. Создаются системным администратором.

Каждый пользователь помимо имени имеет числовой идентификатор пользователя **UID** (User IDentificator). Пользователь root имеет идентификатор 0. Системные пользователи имеют идентификаторы от 1 до 100. Обычные пользователи имеют UID от 100.

Пользователи могут объединяться в группы. Каждый пользователь обязательно входит в ту или иную группу. Группы имеют числовой идентификатор группы **GID** (Group IDentificator).

Информация пользователей

В системе присутствует следующая информация о каждом пользователе:

- Имя пользователя (**user name**) - в рамках системы имя должно быть уникальным. В именах должны использоваться только английские буквы, числа и символы и . (точка).
- Идентификационный номер пользователя (**UID**) - является уникальным идентификатором пользователя в системе. Система отслеживает пользователей по UID, а не по именам.
- Идентификационный номер группы (**GID**) - обозначает группу, к которой относится пользователь. Каждый пользователь может принадлежать к одной или нескольким группам. Принадлежность пользователя к группе устанавливает системный администратор, чтобы иметь возможность ограничивать доступ пользователей к тем или иным ресурсам системы.
- Пароль (**password**) - пароль пользователя в зашифрованном виде.
- Полное имя (**full name**) - помимо системного имени может присутствовать полное имя пользователя, например фамилия и имя.
- Домашний каталог (**home directory**) - каталог, в который попадает пользователь после входа в систему. Подобный каталог имеется у каждого пользователя, все пользовательские каталоги хранятся в директории /home.
- Начальная оболочка (**login shell**) - командная оболочка, которая будет запускаться при входе в систему. Например, /bin/bash.

Вся информация о пользователях хранится в следующих файлах:

passwd (etc/passwd) - содержит информацию о пользователях, имеет следующий формат записи:

"user_name:password:UID:GID:full_name:home_directory:login_shell". Элементы записи должны разделяться символом - ":" (двоеточие) и записываются без пробелов. Если пароль хранится в зашифрованном виде в файле /etc/shadow, то вместо пароля указывается - "x".

group (etc/group) - информация о группах, формат - "group_name:password:GID:user1,user2,user3". Элементы записи должны разделяться символом - ":" (двоеточие) записываются без пробелов. Имена пользователей записываются через запятую.

У файлов /etc/passwd и /etc/group всегда определенные права доступа: чтение и запись для root, для остальных только чтение.

shadow (etc/shadow) - в этом файле хранятся так называемые "теневые пароли", информация о паролях пользователей в зашифрованном виде. Сделано это для безопасности, так как файл /etc/passwd может читаться кем угодно, а файл /etc/shadow может прочитать только root.

gshadow (etc/gshadow) - то же самое что и **shadow**, только для паролей групп.

Помимо основных, в системе присутствуют дополнительные файлы.

useradd (etc/default/useradd) – файл, задающий свойства "по умолчанию" для всех добавляемых пользователей. Можно просмотреть командой - useradd -D.

login.defs (/etc/login.defs) - содержит настройки для создания новых пользователей.

/etc/skel - каталог с дефолтными файлами, которые копируются в домашний каталог каждого пользователя при его создании.

Команды управления

Для управления пользователями используются следующие команды:

useradd или **adduser** - добавить нового пользователя.

passwd - задать пароль для пользователя.

usermod - изменить параметры учетной записи пользователя.

userdel или **deluser** - удалить учетную запись пользователя.

Для управления группами используются следующие команды:

groupadd - добавляет новую группу.

gpasswd - устанавливает пароль группы.

groupmod - изменение параметров группы.

groupdel - удаление группы.

Обычное добавление пользователя

При создании нового пользователя автоматически указываются несколько параметров. В файле /etc/passwd, заводится запись с указанием имени пользователя, домашнего каталога, UID, GID. В каталог помещаются файлы инициализации командной оболочки. Все можно указать вручную, при помощи дополнительных опций. Список опций можно просмотреть командой **adduser -help** или **adduser -h** (рисунок 1).

Для создания пользователя достаточно выполнить команду **adduser** и указать его имя.

Синтаксис команды: **adduser options username**

При использовании данной команды необходимо ответить на несколько вопросов, в процессе ее выполнения будут видны все шаги по созданию нового пользователя.

```
administrator@administrator-VirtualBox:~$ adduser -h
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] [--add_extra_groups]
[--encrypt-home] USER
  Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]
[--disabled-login] [--add_extra_groups] USER
  Add a system user

adduser --group [--gid ID] GROUP
addgroup [--gid ID] GROUP
  Add a user group

addgroup --system [--gid ID] GROUP
  Add a system group

adduser USER GROUP
  Add an existing user to an existing group

general options:
--quiet | -q          don't give process information to stdout
--force-badname      allow usernames which do not match the
                     NAME_REGEX[_SYSTEM] configuration variable
--extrausers         uses extra users as the database
--help | -h          usage message
--version | -v       version number and copyright
```

Рисунок 1 – Опции команды **adduser**

На рисунке 2 представлены шаги по созданию пользователя *iriska*.

```
root@administrator-VirtualBox:~# adduser iriska
Добавляется пользователь «iriska» ...
Добавляется новая группа «iriska» (1001) ...
Добавляется новый пользователь «iriska» (1001) в группу «iriska» ...
Создаётся домашний каталог «/home/iriska» ...
Копирование файлов из «/etc/skel» ...
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: пароль успешно обновлён
Изменение информации о пользователе iriska
Введите новое значение или нажмите ENTER для выбора значения по умолчанию
  Полное имя []: iriska
  Номер комнаты []: 1
  Рабочий телефон []: 111
  Домашний телефон []: 1234567
  Другое []:
Данная информация корректна? [Y/n] y
root@administrator-VirtualBox:~#
```

Рисунок 2 – добавление пользователя *iriska*

Далее при помощи команды `grep -E` проверим как создался пользователь (рис. 3)

```
root@administrator-VirtualBox:~# grep -E 'iriska' /etc/passwd
iriska:x:1001:1001:iriska,1,111,1234567:/home/iriska:/bin/bash
root@administrator-VirtualBox:~#
```

Рисунок 3 – проверка создания пользователя

Как видно из рисунка пользователь `iriska` создан с домашним каталогом `/home/iriska`.

Создание пользователя с определенным UID

Каждому пользователю при его создании присваивается идентификационный номер пользователя (UID), присвоение идет по порядку: 1000, 1001 и т.д. При использовании опции `-u` пользователю можно задать определенный UID. Для примера создадим пользователя `iriska` с UID равным 111.

Создание пользователя с домашней директорией, отличной от дефолтной директории

По умолчанию команда `useradd` создает домашний каталог пользователя в директории `/home`, имя домашнего каталога всегда совпадает с именем пользователя. Например, при создании пользователя `techlist`, создается домашняя директория `/home/techlist`.

Допустим, что возникла необходимость задать пользователю домашний каталог с размещением и именем отличающимися от стандартных настроек. При использовании опции `-d` можно задать свой каталог для пользователя.

Пример: `useradd -d /opt/users you_user_name`

Далее выполняем проверку при помощи команды:

```
grep -E 'you_user_name' /etc/passwd
```

Создание пользователя с определенными идентификаторами UID и GID

Каждому пользователю при его создании присваивается UID, присвоение идет по порядку: 1000, 1001 и т.д. При использовании опции `-u` пользователю можно задать определенный UID. Для примера создадим нового пользователя с UID равным 1111.

Для этого используем следующие параметры:

```
useradd -d 1111 you_user_name
```

Пользователь создан с UID равным 1111, но числовой идентификатор группы (GID) был задан дефолтный - 1001, для того чтобы это исправить можно поступить следующим образом. Сначала создадим группу `users` с GID равным 1111:

```
groupadd -g 1111 users
```

далее следует обратить внимание, что группа создана, но пользователей в ней нет. Теперь можно создать пользователя с UID=1111 и указать его принадлежность к группе users с помощью опции -g и указания GID 1111.

```
useradd -u 1111 -g 1111 you_user_name
```

Изменение пользователя

Иногда может потребоваться внести некоторые изменения в аккаунт пользователя. Сделать это можно при помощи команды **usermod**.

Синтаксис команды: **usermod options username**

Команда **usermod** имеет дополнительные опции, посмотреть список доступных опций можно командой **usermod -h**.

Удаление пользователей

```
userdel options username
```

Для того чтобы удалить пользователя, выполняем команду **userdel**. Команда **userdel** не удаляет пользователя если он работает в системе, сначала необходимо завершить все связанные с ним процессы. Для просмотра дополнительных опций команды **userdel**, используется ключ **-h**.

Контрольные вопросы

1. Какая информация о пользователе содержится в системе?
2. Как создать нового пользователя?
3. Что такое UID?
4. Как удалить пользователя?
5. Может ли пользователь принадлежать к нескольким группам?
6. Можно ли удалить активного пользователя командой **userdel**?
7. Как посмотреть дополнительные опции команды в Linux?
8. Каким образом присваивается UID?
9. Как проверить: создан ли пользователь?
10. Как выглядит синтаксис команды для добавления пользователя с параметрами?

Цель работы

Ознакомиться с основными утилитами, используемыми в ОС Linux

Задание

1. Ознакомиться с теорией;
2. Поочередно установить и вызвать каждую из предложенных утилит;
3. Сделать скриншоты;
4. Ответить на вопросы;
5. Сформировать отчет.

Краткая теория

Большинство команд в ОС Linux являются утилитами. Они хранятся в каталоге `"/bin"`. Если пользователь вводит в терминале команду для выполнения, то система начинает искать ее в этом каталоге, а также в других системных каталогах. Для выполнения работы необходимо перейти в каталог, где хранятся утилиты:

cd /bin

Просмотрите содержание каталога с помощью команды:

ls

Как видно из выведенной информации, в этом каталоге содержатся утилиты команд `"cp"`, `"grep"` и других утилит (рисунок 1).

```
administrator@administrator-VirtualBox:/bin$ ls
bash          fuser          nisdomainname stty
brltty        fusermount    ntfs-3g        su
bunzip2       getfacl        ntfs-3g.probe sync
busybox       grep           ntfscluster   systemctl
bzip2         gunzip         ntfscluster   systemd
bzcat         gzexe          ntfscluster   systemd-ask-password
bzcmp         gzip           ntfscluster   systemd-escape
bzdiff        hciconfig     ntfscluster   systemd-hwdb
bzegrep       hostname      ntfscluster   systemd-inhibit
bzexe         ip            ntfscluster   systemd-machine-id-setup
bzfgrep       journalctl    ntfscluster   systemd-notify
bzgrep        kbd_mode      ntfscluster   systemd-sysusers
bzip2         kill          ntfscluster   systemd-tmpfiles
bzless        kmod          ntfscluster   systemd-tty-ask-password-agent
bzmore        less          ntfscluster   tar
cat           lessecho      ntfscluster   tempfile
chacl         lessfile      ntfscluster   touch
chgrp         lesskey       ntfscluster   true
chmod         lesspipe      ntfscluster   udevadm
```

Рисунок 1. Содержание каталога с утилитами

Если команда была введена с ошибкой или утилита еще не была установлена, то система сообщит об этом (рисунок 1). Попробуйте ввести команду `lss` (рис. 2).

```
administrator@administrator-VirtualBox:~$ lss
Command 'lss' not found, did you mean:
  command 'lsd' from snap lsd (0.16.0)
  command 'lsw' from deb suckless-tools
  command 'lvs' from deb lvm2
  command 'lsm' from deb lsm
  command 'lrs' from deb lrslib
  command 'les' from deb atm-tools
  command 'ss' from deb iproute2
  command 'ls' from deb coreutils
  command 'less' from deb less
  command 'ass' from deb irpas
  command 'lssu' from deb nilfs-tools
  command 'gss' from deb libgss-dev
  command 'lsh' from deb lsh-client
  command 'lsc' from deb livescript
  command 'lsns' from deb util-linux

See 'snap info <snapname>' for additional versions.
administrator@administrator-VirtualBox:~$
```

Рисунок 2 - Ошибка при поиске команды "lss"

w - Утилита позволяет отобразить кто авторизован в системе, а также запущенные от имени этого пользователя процессы и оболочки (рисунок 3).

```
administrator@administrator-VirtualBox:/bin$ w
12:43:27 up 12 min, 1 user, load average: 0,04, 0,18, 0,25
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
administ :0          :0           12:31    ?xdm?  22.73s 0.02s /usr/lib/gdm3/
administrator@administrator-VirtualBox:/bin$
```

Рисунок 3 – отображение авторизованных пользователей

nmon - инструмент, показывающий информацию о производительности системы (рисунок 4).

Для установки данного инструмента наберите:

sudo apt-get install nmon

После завершения установки можно запускать команду:

nmon

nmon может отобразить информацию, связанную с сетью, центральным процессором, памятью или дисковым пространством.

```
nmon-16g [H for help] Hostname=administrator Refresh= 2secs 12:48.29
-----
          For help type H or ...
          nmon -? - hint
          nmon -h - full details
          To stop nmon type q to Quit

Use these keys to toggle statistics on/off:
c = CPU          l = CPU Long-term      - = Faster screen updates
C = " WideView  U = Utilisation    + = Slower screen updates
m = Memory       V = Virtual memory  j = File Systems
d = Disks        n = Network          . = only busy disks/procs
r = Resource     N = NFS                h = more options
k = Kernel       t = Top-processes      q = Quit
```

Рисунок 4 – запущенная утилита nmon

ncdu - это утилита, которая используется для анализа дискового пространства, занимаемого различными каталогами (рисунок 5)

Для установки в Ubuntu наберите:

sudo apt-get install ncdu

Затем для ее запуска:

ncdu

```
Scanning...
Total items: 61256 size: 403,2 MiB
Current item: /usr/src/linux-hwe-5.4-he...erators/use_after_iter.cocci
nning... Press q to abort
```

Рисунок 5 – анализ дискового пространства

```
ncdu 1.12 ~ Use the arrow keys to navigate, press ? for help
-----
 3,2 GiB [#####] /usr
. 1,6 GiB [#### ] /snap
. 1,4 GiB [#### ] /var
898,7 MiB [## ] /lib
. 116,8 MiB [ ] /boot
. 12,4 MiB [ ] /etc
. 12,2 MiB [ ] /bin
. 11,0 MiB [ ] /sbin
. 5,6 MiB [ ] /home
. 1,3 MiB [ ] /run
. 56,0 KiB [ ] /tmp
! 16,0 KiB [ ] /lost+found
e 4,0 KiB [ ] /lib64
! 4,0 KiB [ ] /srv
! 4,0 KiB [ ] /root
e 4,0 KiB [ ] /opt
e 4,0 KiB [ ] /mnt
e 4,0 KiB [ ] /media
e 4,0 KiB [ ] /cdrom
. 0,0 B [ ] /proc
. 0,0 B [ ] /sys
. 0,0 B [ ] /dev
@ 0,0 B [ ] initrd.img.old
@ 0,0 B [ ] initrd.img
@ 0,0 B [ ] vmlinuz.old
Total disk usage: 7,2 GiB Apparent size: 128,0 TiB Items: 408383
```

Рисунок 6 – результат анализа дискового пространства

sudo lshw -C network

Данная команда показывает подключенные сетевые устройства (рисунок 7)

```
administrator@administrator-VirtualBox:/$ sudo lshw -C network
*-network
описание: Ethernet interface
продукт: 82540EM Gigabit Ethernet Controller
производитель: Intel Corporation
физический ID: 3
сведения о шине: pci@0000:00:03.0
логическое имя: enp0s3
версия: 02
серийный №: 08:00:27:73:51:c3
размер: 1Gbit/s
capacity: 1Gbit/s
разрядность: 32 bits
частота: 66MHz
возможности: pm pci_x bus_master cap_list ethernet physical tp 10bt 10bt-
fd 100bt 100bt-fd 1000bt-fd autonegotiation
конфигурация: autonegotiation=on broadcast=yes driver=e1000 driverversio
n=7.3.21-k8-NAPI duplex=full ip=10.0.2.15 latency=64 link=yes mingnt=255 multic
ast=yes port=twisted pair speed=1Gbit/s
ресурсы: IRQ:19 память:f4200000-f421ffff ioport:d020(размер=8)
administrator@administrator-VirtualBox:/$
```

Рисунок 7 – данные о подключенных сетевых устройствах

findmnt – команда, которая используется для поиска подключенных файловых систем. Она используется для поиска подключенных устройств, а также может монтировать или размонтировать их при необходимости. Команда входит в набор полезных программы для linux - util-linux.

Для запуска findmnt наберите команду (рисунок 8):

findmnt

```
administrator@administrator-VirtualBox:/$ findmnt
TARGET                                SOURCE                                FSTYPE  OPTIONS
/                                       /dev/sda1                             ext4    rw,relatime,errors=rem
-/sys                                  sysfs                                   sysfs   rw,nosuid,nodev,noexec
-/sys/kernel/security                 securityfs                               securi  rw,nosuid,nodev,noexec
-/sys/fs/cgroup                       tmpfs                                   tmpfs   ro,nosuid,nodev,noexec
-/sys/fs/cgroup/unified                cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/systemd                cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/cpu,cpuacct            cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/freezer                cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/hugetlb                cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/rdma                   cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/blkio                  cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/net_cls,net_prio       cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/pids                   cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/devices                 cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/perf_event              cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/memory                  cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/cgroup/cpuset                  cgroup                                  cgroup  rw,nosuid,nodev,noexec
-/sys/fs/pstore                         pstore                                  pstore  rw,nosuid,nodev,noexec
-/sys/kernel/debug                     debugfs                                  debugfs  rw,relatime
-/sys/kernel/config                    configfs                                  config  rw,relatime
-/sys/fs/fuse/connections               fusectl                                  fusectl  rw,relatime
-/proc                                   proc                                     proc    rw,nosuid,nodev,noexec
-/proc/sys/fs/binfmt_misc               systemd-1                                autofs  rw,relatime,fd=34,pgrp
-/proc/sys/fs/binfmt_misc               binfmt_misc                               binfmt  rw,relatime
-/dev                                    udev                                       udev    rw,nosuid,relatime,siz
-/dev/pts                               devpts                                    devpts  rw,nosuid,noexec,relat
-/dev/shm                               tmpfs                                   tmpfs   rw,nosuid,nodev
```

Рисунок 8 – вывод подключенных файловых систем

Для от отображения в виде списка необходимо набрать следующую команду:

findmnt -l

Также можно выводить данные, фильтруя по типу файловой системы или по формату.

netstat – команда для получения сведений об активности и статистике сетевых соединений.

Пример использования команды **netstat -a** (вывод всех активных портов) показан на рисунке 9.

```
administrator@administrator-VirtualBox:/$ netstat -a
Активные соединения с интернетом (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp    0      0 localhost:domain 0.0.0.0:* LISTEN
tcp    0      0 localhost:ipp    0.0.0.0:* LISTEN
tcp6   0      0 ip6-localhost:ipp [::]:* LISTEN
udp    0      0 0.0.0.0:ipp     0.0.0.0:*
udp    0      0 0.0.0.0:mdns    0.0.0.0:*
udp    0      0 0.0.0.0:40955   0.0.0.0:*
udp    0      0 localhost:domain 0.0.0.0:*
udp    0      0 0.0.0.0:bootpc  0.0.0.0:*
udp6   0      0 [::]:39813      [::]:*
udp6   0      0 [::]:mdns       [::]:*
raw6   0      0 [::]:ipv6-icmp  [::]:*
Активные сокеты домена UNIX (servers and established)
Proto RefCnt Flags Type State I-Node Путь
unix  2 [ ACC ] STREAM LISTENING 20838 @/tmp/dbus-j04IjPod
unix  2 [ ACC ] STREAM LISTENING 20837 @/tmp/dbus-y610klsh
unix  2 [ ] DGRAM 20400 /run/user/1000/syste
md/notify
unix  2 [ ACC ] SEQPACKET LISTENING 14712 /run/udev/control
unix  2 [ ACC ] STREAM LISTENING 20403 /run/user/1000/syste
md/private
unix  2 [ ACC ] STREAM LISTENING 22896 @/tmp/dbus-f0Sxiiy9
unix  2 [ ACC ] STREAM LISTENING 20410 /run/user/1000/bus
```

Рисунок 9 - Список всех портов

Контрольные вопросы

1. Каким образом можно посмотреть подключенные сетевые устройства?
2. Как получить сведения об активности и статистике сетевых соединений?
3. Как посмотреть список всех портов?
4. Какая утилита используется для анализа дискового пространства?
5. Для чего используется утилита nmon?
6. Как посмотреть содержимое какого-либо каталога?
7. В каком каталоге хранятся утилиты?

Лабораторная работа №5. Изучение основ работы с файлами

Цель работы

Изучить основные команды для работы с файлами

Задание

1. Ознакомиться с теорией;
2. Выполнить предложенные команды в консоли;
3. Сделать скриншоты;
4. Ответить на вопросы;
5. Сформировать отчет.

Краткая теория

В данной работе мы рассмотрим основные команды для работы с каталогами в Linux, а именно:

- **ls** - список файлов в директории;
- **cd** - переход между директориями;
- **rm** - удалить файл;
- **mkdir** - создать папку;
- **rmdir** - удалить папку;
- **touch** - создать пустой файл;
- **mv** - переместить файл;
- **cp** - скопировать файл;

Рассмотрим подробно каждую команду.

ls

Команда ls позволяет вывести список файлов заданной папки, по умолчанию, будет выведен список файлов текущей папки (рисунок 1).

```
administrator@administrator-VirtualBox:~$ ls
examples.desktop  Документы  Изображения  Общедоступные  Шаблоны
Видео             Загрузки  Музыка        'Рабочий стол'
```

Рисунок 1 – список файлов в домашней директории пользователя

Также вы можете вывести список файлов из всех подкаталогов рекурсивно, для этого используйте опцию -R (рисунок 2)

```
administrator@administrator-VirtualBox:~$ ls -R
.:
examples.desktop  Документы  Изображения  Общедоступные  Шаблоны
Видео             Загрузки  Музыка        'Рабочий стол'

./Видео:
./Документы:
./Загрузки:
./Изображения:
'Снимок экрана от 2021-08-25 12-55-00.png'
./Музыка:
./Общедоступные:
'./Рабочий стол':
./Шаблоны:
```

Рисунок 2 – список файлов из всех подкаталогов

Чтобы вывести список файлов нужной папки, можно передать ее адрес утилите, например, /home (рисунок 3).

```
administrator@administrator-VirtualBox:~$ ls /home
administrator  iriska  lost+found
```

Рисунок 3 – список файлов в папке home

Чтобы получить больше информации и вывести все имена файлов в виде списка используйте опцию -l (рисунок 4).

```

administrator@administrator-VirtualBox:~$ ls -l /home
итого 24
drwxr-xr-x 14 administrator administrator 4096 авг 30 03:37 administrator
drwxr-xr-x  2          1001          1001 4096 авг 29 17:44 iriska
drwx-----  2 root          root          16384 авг  4 23:18 lost+found
administrator@administrator-VirtualBox:~$

```

Рисунок 4 – подробная информация о файлах

cd

Для навигации по файлам и каталогам Linux используйте команду `cd`. Она требует либо полный путь, либо имя каталога, в зависимости от текущего рабочего каталога, в котором вы находитесь. Допустим, вы находитесь в домашней директории и хотите перейти в Изображения (Рисунок 5). Для этого просто введите следующую команду:

```
cd ./Изображения
```

```

administrator@administrator-VirtualBox:~$ cd ./Изображения
administrator@administrator-VirtualBox:~/Изображения$

```

Рисунок 5 – переход в директорию изображения

Также стоит отметить, что оболочка Linux чувствительна к регистру. Важно точно вводить имена каталогов.

rm (данную команду выполнять не надо)

Команда **rm** используется для удаления файлов. Если вы хотите удалить каталог со всем его содержимым, в качестве альтернативы `rmdir` используйте `rm` с опцией `-r`. Будьте очень осторожны с этой командой и всегда проверяйте, в каком каталоге вы находитесь. Она удаляет всё и её невозможно отменить.

mkdir

`mkdir` используется для создания новых каталогов. Чтобы создать папку Music необходимо ввести команду

```
mkdir Music
```

команда создаст каталог с именем Music (рисунок 6).

```

administrator@administrator-VirtualBox:~$ mkdir Music
administrator@administrator-VirtualBox:~$ ls
examples.desktop  nano.save  Загрузки  Общедоступные
hello_world       Видео     Изображения  'Рабочий стол'
Music             Документы Музыка     Шаблоны
administrator@administrator-VirtualBox:~$

```

Рисунок 6 – создание каталога с именем Music

Дополнительные опции `mkdir`: Используйте опцию `p` (`parents`), чтобы создать каталог между двумя существующими каталогами. Например, `mkdir -p Music/2021/Newfile` создаст новый файл «2021».

rm -r

Если вам нужно удалить каталог, используйте команду `rm -r`. Однако `rm -r` позволяет удалять только пустые директории. Удалим созданный каталог с именем `Music` (рисунок 7)

```
administrator@administrator-VirtualBox:~$ rm -r Music
administrator@administrator-VirtualBox:~$ ls
examples.desktop  Видео      Изображения  'Рабочий стол'
hello_world       Документы  Музыка       Шаблоны
nano.save         Загрузки   Общедоступные
administrator@administrator-VirtualBox:~$ █
```

Рисунок 7 – удаление каталога с именем `Music`

touch

Команда `touch` позволяет создать новый пустой файл через командную строку Linux. В качестве примера введите (рисунок 8):

```
touch new_file.txt
```

```
administrator@administrator-VirtualBox:~$ touch new_file.txt
administrator@administrator-VirtualBox:~$ ls
examples.desktop  new_file.txt  Загрузки      Общедоступные
hello_world       Видео        Изображения   'Рабочий стол'
nano.save         Документы    Музыка        Шаблоны
administrator@administrator-VirtualBox:~$ █
```

Рисунок 8 – создание пустого файла

mv

Для перемещения файла вам нужно ввести `mv`, имя файла и каталог назначения (рисунок 9). Например:

```
mv new_file.txt ./Документы
```

```
administrator@administrator-VirtualBox:~$ mv new_file.txt ./Документы
administrator@administrator-VirtualBox:~$ ls
examples.desktop  Видео      Изображения  'Рабочий стол'
hello_world       Документы  Музыка       Шаблоны
nano.save         Загрузки   Общедоступные
administrator@administrator-VirtualBox:~$ █
```

Рисунок 9 – перемещение файла из корневого каталога в каталог `Документы`

cp

`cp` используется для копирования файлов из текущего каталога в другой каталог. Например (Рисунок 10):

```
cp new_file.txt ./Изображения
```

```
administrator@administrator-VirtualBox:~/Документы$ cp new_file.txt ./Изображения
administrator@administrator-VirtualBox:~/Документы$ █
```

Рисунок 10 – Копирование файла в другой каталог

Для проверки можно перейти в каталог «Изображения» и выполнить команду ls

Контрольные вопросы

1. Что такое директория?
2. Для чего применяется опция -I?
3. Какая команда копирует файлы?
4. Чем отличаются команды rm -и rmdir?
5. Что делает команда mv?
6. Как создать пустой файл?

Лабораторная работа №6. Изучение флагов в командах

Цель

Ознакомиться с флагами файлов

Задание

1. Ознакомиться с теорией;
2. Разобраться что такое флаги и как их просматривать;
3. Создать файл и изменить для него ограничения;
4. Ответить на вопросы;
5. Сформировать отчет.

Краткая теория

Флаг – это атрибут, позволяющий ограничить возможные действия с файлом. В Unix, права доступа и участники, сгруппированы три по три, read/write/execution для user/group/other. Кроме вышеупомянутых прав доступа, операционные системы Unix, поддерживает "файловые флаги", для обеспечения дополнительного уровня защиты, контроля и управления объектами системы.

Например, с помощью флагов, можно запретить удаление файла, даже пользователю root. Вообще, правильней говорить не о правах пользователей, на тот или иной файл/директорию, а о правах процессов, так как все манипуляции с файлами (создание, чтение, изменение, удаление), производятся с помощью какой-то программы, например, текстового редактора, который в запущенном виде, собственно и является процессом.

Каждому пользователю системы, присваиваются идентификаторы UserID (идентификатор пользователя) и GroupID (идентификатор группы), именно по ним операционная система определяет владельца и группу для того или иного объекта системы. Символические имена назначаются исключительно для удобства чтения пользователями. Каждый процесс в операционной системе Unix, имеет не два, а четыре идентификатора, это, реальный UserID (UID) и эффективный UserID (UID) а также реальный и эффективный идентификатор группы GroupID (GID).

Реальный UserID (Реальный GroupID) - реальные номера применяются для учета использования системных ресурсов. Эффективный UserID (Эффективный GroupID) - новый идентификатор, полученный процессом уже во время выполнения, применяются для определения прав доступа. Процессы запущенные автоматически (например, при старте системы), также получают UserID, какой именно, определяется запускающей программой (cron, inetd и т.д.).

В обычных случаях, UserID наследуется от родительского процесса. Некоторые процессы - "родители", могут присваивать, дочернему процессу - "потомку" UserID, не совпадающий со своим. Иногда, запущенный процесс, может во время выполнения, менять свои идентификаторы. Это происходит, когда в правах доступа на файл, установлены дополнительные биты - SUID (Set user ID - бит смены идентификатора пользователя) и SGID (Set group ID - бит смены идентификатора группы).

Когда пользователь или процесс запускает файл, у которого установлен один из этих битов, процессу, временно присваиваются права владельца данного файла (а не того, кто его запускает). Таким образом, обычный пользователь, может выполнять команды от имени пользователя root. Как было упомянуто выше, права доступа к файлам и директориям, определяются по Реальному UserID и Реальному GroupID. Если идентификатор не меняется в процессе выполнения программы, Реальный UserID (реальный GroupID) и Эффективный UserID (эффективный GroupID) совпадают. Система обеспечения прав доступа в операционных системах Unix, имеет числовое и символическое выражения. Числовое выражение выглядит следующим образом (табл. 1).

Таблица 1 – Числовые значение системы прав доступа

Значение	Права доступа	В списке файлов каталога
0 (000)	Ничего не разрешено	---
1 (001)	запрещено читать и записывать, разрешено исполнять	--x
2 (010)	Запрещено читать и исполнять, разрешено записывать	-w-
3 (011)	Запрещено читать, разрешено записывать и исполнять	-wx
4 (100)	Разрешено читать, запрещено записывать и исполнять	r--
5 (101)	Разрешено читать и исполнять, запрещено записывать	r-x
6 (110)	Разрешено читать и записывать, запрещено исполнять	rw-
7 (111)	Разрешено все	rwx

Плюс к этому, биты смены идентификаторов SUID и SGID, восьмеричные значения которых 4000 и 2000. Они обеспечивают программам доступ к файлам и процессам, к которым в обычных условиях, пользователь доступа не имеет.

В случае установки бита SGID для директории, созданные в ней файлы, при запуске, будут принимать идентификатор группы каталога, а не группы в которую входит владелец файла. Существует еще один бит с восьмеричным значением 1000, так называемый Sticky-bit. Если данный бит установлен на директорию, содержащиеся в ней файлы, разрешено удалять и переименовывать, только в случае, если пользователь является, владельцем каталога, владельцем файла или это пользователь root.

Чтобы посмотреть права доступа и владельцев на файлы и директории в режиме листинга, можно использовать команду ls с флагом -l (рисунок 1)

```

administrator@administrator-VirtualBox:~$ ls -l
итого 44
-rw-r--r-- 1 administrator administrator 8980 авг  4 23:23 examples.desktop
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Видео
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Документы
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Загрузки
drwxr-xr-x 2 administrator administrator 4096 авг 25 12:55 Изображения
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Музыка
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Общедоступные
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 'Рабочий стол'
drwxr-xr-x 2 administrator administrator 4096 авг  4 23:36 Шаблоны
administrator@administrator-VirtualBox:~$ █

```

Рисунок 1 – Команда ls -l

Узнать все возможные для использования флаги можно, если ввести саму команду и флаг --help. Введите команду для просмотра всех возможных флагов команды "ls":

ls --help

```
administrator@administrator-VirtualBox:~$ ls --help
Использование: ls [КЛЮЧ]... [ФАЙЛ]...
Выдаёт информацию о ФАЙЛАХ (по умолчанию о текущем каталоге).
Сортирует в алфавитном порядке, если не задан ни --sort, ни один из
ключей -cftuvSUX.

Аргументы, обязательные для длинных ключей, обязательны и для коротких.
-a, --all                не скрывать файлы начинающиеся с .
-A, --almost-all       не выдавать подразумеваемые . и ..
--author                вместе с -l, печатать автора каждого файла
-b, --escape            печатать экранирующие последовательности
                        в стиле C для не графических символов
--block-size=РАЗМЕР    использовать блоки указанного РАЗМЕРА; например,
                        «--block-size=M» выводит размеры в единицах,
                        равных 1048576 байтам; см. формат РАЗМЕРА далее.
-B, --ignore-backups    не выдавать файлы, оканчивающиеся на ~
-c                      с -lt: сортировать и показывать по stime (времени
                        последнего изменения файла);
                        с -l: показывать stime и сортировать по имени;
                        иначе: сортировать по stime, сначала самые новые
-C                      выдавать список в несколько колонок
--color[=КОГДА]        расцвечивать вывод;
                        КОГДА может быть «always» (по умолчанию, если не
                        указано), «auto» или «never». Подробнее см. ниже
-d, --directory        выдавать имена каталогов, а не их содержимое
-D, --dired             генерировать вывод для режима Emacs dired
-f                      не сортировать, включает -aU,
```

Рисунок 2 – Команда ls –help

Для изменения прав используется команда `chmod`.

К примеру, вы являетесь владельцем файла с именем `new_file` и хотите установить его разрешения таким образом, чтобы: пользователь мог читать, писать и выполнять его; члены группы могут прочесть и выполнить его; а также другие могут только читать его. Эта команда будет выглядеть вот так:

```
chmod u=rwx,g=rx,o=r new_file
```

В качестве альтернативы можно использовать запись в восьмеричном виде для обозначения прав

```
chmod 754 new_file
```

Контрольные вопросы

1. Что такое флаг?
2. Для чего нужны флаги?
3. Что означает запись `rw-?`?
4. Как выдать новые права для пользователя?
5. Что такое Sticky-bit?
6. Как будет выглядеть разрежение на чтение и запуск в числовом виде?
7. Какая команда позволяет изменить права на файл?
8. Какие есть варианты для выполнения команды `chmod`?

Практическая работа №4. Сетевая модель

Краткая теория

Эталонная сетевая модель OSI

OSI (Open System Interconnection) - Сетевая модель взаимодействия открытых систем. Именно этой модели придерживаются производители сетевых устройств, когда разрабатывают новые продукты.

Сетевая модель OSI состоит из 7 уровней, причем принято начинать отсчёт с нижнего уровня (рисунок 1).

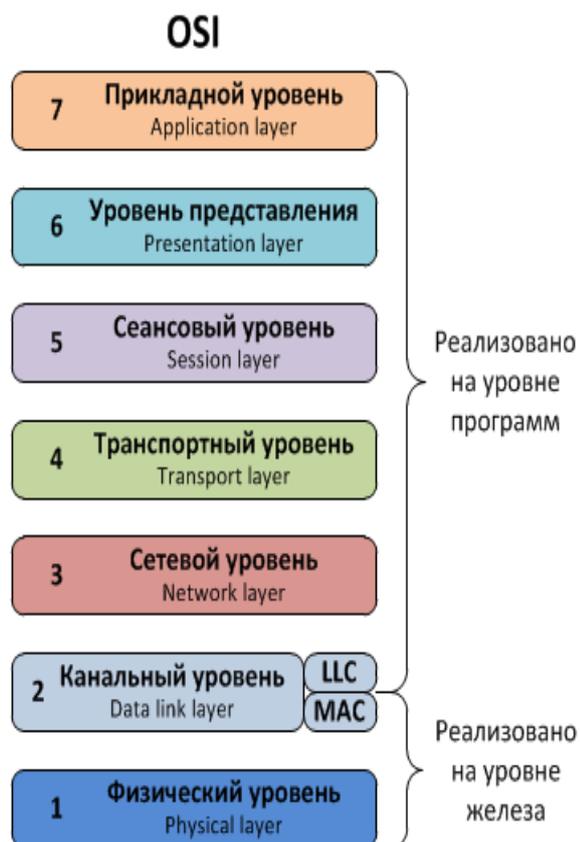


Рисунок 1 – сетевая модель OSI

Перечислим эти уровни:

7. Прикладной уровень (application layer);
6. Уровень представления (presentation layer) ;
5. Сеансовый уровень (session layer);
4. Транспортный уровень (transport layer);
3. Сетевой уровень (network layer);
2. Канальный уровень (data link layer);
1. Физический уровень (physical layer).

Прикладной уровень

Прикладной уровень или уровень приложений (application layer) – это самый верхний уровень модели. Он осуществляет связь пользовательских приложений с сетью.

Уровень представлений

Уровень представления (presentation layer) – преобразует данные в соответствующий формат.

Сеансовый уровень

Сеансовый уровень или уровень сессий (session layer) – организует сеанс связи между компьютерами.

Транспортный уровень

Транспортный уровень (transport layer) – этот уровень обеспечивает надёжность передачи данных от отправителя к получателю.

Сетевой уровень

Сетевой уровень (network layer) – этот уровень определяет путь, по которому данные будут переданы.

Канальный уровень

Канальный уровень (data link layer) – он нам нужен для взаимодействия сетей на физическом уровне.

Физический уровень

Физический уровень (physical layer) – самый нижний уровень, непосредственно осуществляющий передачу потока данных.

IP-адреса и маска подсети

Маска подсети помогает маршрутизатору понять, как и куда передавать пакет. Подсетью может являться любая сеть со своими протоколами. Маршрутизатор передает пакет напрямую, если получатель находится в той же подсети, что и отправитель. Если же подсети получателя и отправителя различаются, пакет передается на второй маршрутизатор, со второго на третий и далее по цепочке, пока не достигнет получателя.

Протокол интернета — IP (Internet Protocol) используется маршрутизатором, чтобы определить, к какой подсети принадлежит получатель. Свой уникальный IP-адрес есть у каждого сетевого устройства, при этом в глобальной сети не может существовать два устройства с одинаковым IP. Он имеет два подвида, первым был принят IPv4 (IP version 4, версии 4) в 1983 году. IPv4 предусматривает назначение каждому устройству 32-битного IP-адреса, что

ограничивало максимально возможное число уникальных адресов 4 миллиардами (232). В более привычном для человека десятичном виде IPv4 выглядит как четыре блока (октета) чисел от 0 до 255, разделенных тремя точками. Первый октет IP-адреса означает его класс, классов всего 4: А, В, С, D. Рассмотрим, например, IPv4 адрес класса С 223.135.100.7. Первые два октета 223.135 определяют класс, третий 100 — это номер подсети, а последний означает номер сетевого оборудования. Например, если необходимо отправить информацию с компьютера номер 7 с IPv4 адресом 223.135.100.7 на компьютер номер 10 в той же подсети, то адрес компьютера получателя будет следующий: 223.135.100.10.

В связи с быстрым ростом сети Интернет остро вставала необходимость увеличения числа возможных IP-адресов. В 1998 впервые был описан IPv6 (IP версии 6), который использует 128-битные адреса, и позволяет назначить уникальные адреса для $3,4 \times 10^{38}$ устройств.

IPv6 имеет вид восьми блоков по четыре шестнадцатеричных значения, а каждый блок разделяется двоеточием. IPv6 выглядит следующим образом: 2DAB:FFFF:0000:0000:01AA:00FF:DD72:2C4A.

Так как IPv6 адреса длинные, их разрешается сокращать по следующим правилам: ведущие нули допускается опускать, например в адресе выше :00FF: допускается записывать как :FF:, группы нулей, идущие подряд тоже допустимо сокращать и заменять на двойное двоеточие, например, 2DAB:FFFF::01AA:00FF:DD72:2C4A. Допускается делать не больше одного подобного сокращения в адресе IPv6. IP предназначен для определения адресата и доставки ему информации, он предоставляет услугу для вышестоящих уровней, но не гарантирует целостность доставляемой информации.

Контрольные вопросы

1. Что собой представляет модель OSI?
2. Сколько уровней в модели OSI?
3. На каком уровне модели OSI находится “Уровень представлений”?
4. Что находится на физическом уровне модели OSI?
5. Что такое маска подсети? Зачем она нужна?
6. На каком уровне модели OSI используется протокол IP?
7. Какова длина IP-адреса используется в протоколе IPv6?
8. Что собой представляет MAC-адрес?
9. Как принято записывать адреса IPv6?

10. Как можно сокращать адреса IPv6?

Практическая работа №5. Изучение команд для настройки сети

Краткая теория

Утилиты для работы с сетью:

- `curl` & `wget` - загрузка ресурсов из сети Интернет;
- `ping` - проверка доступности хоста;
- `mtr` - комбинация `traceroute` и `ping`;
- `host` - выполнение DNS запросов;
- `whois` - поиск данных в базе данных WHOIS;
- `ifplugstatus` - определение подключен ли сетевой кабель;
- `ifconfig` - настройка сетевого интерфейса;
- `ifdown` & `ifup` - активация/деактивация сетевого интерфейса;
- `netstat` - отображение сетевой информации.

Программы **curl** и **wget** используются для загрузки файлов из интернет, прямо в терминале. Если вы используете `curl`, для загрузки файла наберите выполняйте команду `curl -O url_загружаемого_файла`. Если вы используете `wget`, в этом случае не нужны никакие дополнительные параметры командной строки. После загрузки, файл будет находиться в домашней директории.

ping - это диагностическая утилита, которая отправляет сетевые ECHO_REQUEST пакеты по указанному вами адресу, получает от него ответ и измеряет задержку между моментом отправки пакета и его получением. Если наблюдаются проблемы со связью - программа для всех пакетов, на которые не пришел ответ, вернет сообщение "request timeout". Утилита `ping` - это первое средство, которое применяется для проверки подключен ли компьютер к сети, как своего, так и удаленного, например, сервер, на котором размещен ваш интернет сайт, или почтовый сервер. Команда `ping` есть во всех операционных системах, работающих с сетью. В отличие от Windows, где команда "`ping адрес_хоста`" отправит только четыре пакета, в Linux такая команда будет непрерывно посылать пакеты, пока вы не прервете ее работу.

Для того, чтоб ограничить количество пакетов, следует указать параметр командной строки "-с количество".

Команда **mtr** объединяет в себе функции двух, ранее рассмотренных команд - ping и tracerpath. mtr непрерывно отсылает ECHO_REQUEST пакеты каждому узлу из маршрута, позволяя в реальном времени видеть задержки, прохождения пакетов по каждому узлу. Эта утилита облегчает задачу диагностики - легко обнаружить узел, из-за которого возникают проблемы и пропадают пакеты.

Команда **host** выполняет DNS запросы для прямого и обратного разрешения доменных имен. Запустите команду с IP адресом в качестве параметра, и она вернет доменное имя, ассоциированное с ним. Также можно по доменному имени определить IP-адрес. Для этого в качестве параметра укажите доменное имя.

Команда **whois** отображает whois записи, ассоциированные с доменным именем. Эта информация может пригодиться, если вам нужно узнать кто, где и когда зарегистрировал доменное имя и до какой даты действительна регистрация.

ifup активирует сетевой интерфейс, делая его доступным для передачи и приема данных. Команда **ifdown** отключает сетевой интерфейс, переводя его в состояние, при котором он не может передавать или получать данные. Команда **ifquery** отображает информацию о конфигурации сетевого интерфейса.

Для проверки подключения сетевого кабеля к компьютеру используется команда **ifplugstatus**. Эта программа не установлена по умолчанию - для использования, ее нужно установить:

sudo apt-get install ifplugd

Запустите команду без параметров, для того, чтоб узнать информацию по всем интерфейсам, или укажите в качестве параметра командной строки интерфейс, статус которого вас интересует.

“Link beat detected” обозначает, что кабель подключен. Если не подключен - программа скажет “unplugged”.

Команда **ifconfig** позволяет настроить, оптимизировать, а в случае необходимости - отладить работу сетевого интерфейса. Также эта программа позволяет быстро определить IP-адрес компьютера и другую информацию о

сетевых интерфейсах, включая их имена, скорость и режим подключения и так далее.

Контрольные вопросы

1. Что делает команда `mtr`?
2. Что означает "Link beat detected"?
3. Для чего используется опция "-c" в утилите `ping`?
4. Что такое хост?
5. Для чего нужен сервер DNS?
6. Опишите функции протокола DHCP?
7. Какие программы используются для загрузки файлов из интернета?
8. Как выключить сетевой интерфейс?
9. Что выполняет команда `host`?

Лабораторная работа №7. Изучение команд для настройки сети

Цель работы

Изучить основные команды для настройки сети.

Задание

1. Выполнить команды в консоли;
2. Сделать скриншоты;
3. Описать использованные команды;
4. Сформировать отчет.

Порядок выполнения лабораторной работы

Для проверки сетевой связности устройств используется команда "`ping`". Эта команда позволяет посылать эхо-запросы к узлу сети и получать информацию о его доступности и времени прохождения пакетов между ПК и узлом. Команда имеет следующий синтаксис: `ping <символьный или IP-адрес>`. Для выяснения маршрута следования пакетов используются команды "`tracert`" (в старых дистрибутивах) и "`tracert`" (в новых дистрибутивах) с синтаксисом аналогичным команде "`ping`".

Проверьте доступность узлов в сети Интернет (с IP-адресом 8.8.8.8 и с сайтом "yandex.ru") с помощью команды:

```
ping yandex.ru  
ping 8.8.8.8
```

Для завершения посылки эхо-запросов нажмите сочетание клавиш "Ctrl+c" (Рисунок 1).

```
root@administrator-VirtualBox:/home/administrator# ping 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=22.2 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=20.8 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=20.2 ms  
^C  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2014ms  
rtt min/avg/max/mdev = 20.206/21.102/22.291/0.876 ms  
root@administrator-VirtualBox:/home/administrator#
```

Рисунок 1 – команда ping

Для установки отсутствующих утилит необходимо использовать команду **apt install имя_утилиты**

На рисунке 2 показан пример использования команды **traceroute**

```
administrator@administrator-VirtualBox:~$ sudo traceroute mtuci.ru  
[sudo] пароль для administrator:  
traceroute to mtuci.ru (185.193.139.130), 30 hops max, 60 byte packets  
 1  _gateway (192.168.1.1)  3.521 ms  3.486 ms  4.721 ms  
 2  msk-b22-b36.ti.ru (212.1.254.187)  4.937 ms  4.922 ms  4.908 ms  
 3  orl-b1-te-1-0-0-4000.ti.ru (212.1.251.82)  8.634 ms  5.550 ms  8.606 ms  
 4  212.1.251.0 (212.1.251.0)  4.849 ms  4.829 ms  4.816 ms  
 5  * * *  
 6  * * *  
 7  * * *  
 8  92.53.94.87 (92.53.94.87)  3.792 ms  3.779 ms  3.744 ms  
 9  92.53.94.45 (92.53.94.45)  4.499 ms  4.482 ms  4.832 ms  
10  92.53.92.8 (92.53.92.8)  4.565 ms  4.495 ms  4.466 ms  
11  185.193.139.130 (185.193.139.130)  4.915 ms !X  4.900 ms !X  5.002 ms !X  
administrator@administrator-VirtualBox:~$
```

Рисунок 2 – Пример использования команды traceroute

Команда traceroute отправляет пакет с временем жизни равном 1 и смотрит адрес ответившего узла, дальше время жизни = 2 и так пока не достигнет цели. Каждый раз отправляется по три пакета и для каждого из них измеряется время прохождения. Пакет отправляется на случайный порт, который, скорее всего, не занят. Когда утилита traceroute получает сообщение от целевого узла о том, что порт недоступен трассировка считается завершенной. На данном рисунке можно увидеть, что отправленный пакет прошел 10 узлов прежде чем достичь конечной цели. На каждый узел отправлялось по три пакета и для каждого из них было

засечено время прохождения. И если на одном из узлов возникнет проблема, теперь вы будете знать на каком.

Далее рассмотрим команду **host**. Команда `host` в системе Linux используется для операций поиска в DNS (системе доменных имен). Проще говоря, эта команда используется для поиска IP-адреса определенного доменного имени или, если вы хотите узнать доменное имя определенного IP-адреса (рисунок 3).

```
root@administrator-VirtualBox:/home/administrator# host 8.8.8.8
8.8.8.8.in-addr.arpa domain name pointer dns.google.
root@administrator-VirtualBox:/home/administrator#
```

Рисунок 3 - Команда `host`

На рисунке 4 представлен пример выполнения команды **whois**. Команда **whois** отображает `whois`-записи, ассоциированные с доменным именем. Данная команда показывает кто, где и когда зарегистрировал доменное имя и до какой даты действительна регистрация

```
root@administrator-VirtualBox:/home/administrator# whois ya.ru
% By submitting a query to RIPN's Whois Service
% you agree to abide by the following terms of use:
% http://www.ripn.net/about/servpol.html#3.2 (in Russian)
% http://www.ripn.net/about/en/servpol.html#3.2 (in English).

domain:            YA.RU
nserver:          ns1.yandex.ru.
nserver:          ns2.yandex.ru.
state:            REGISTERED, DELEGATED, VERIFIED
org:              YANDEX, LLC.
registrar:        RU-CENTER-RU
admin-contact:    https://www.nic.ru/whois
created:          1999-07-12T14:40:22Z
paid-till:        2022-07-31T21:00:00Z
free-date:        2022-09-01
source:           TCI

Last updated on 2021-10-31T17:11:31Z
```

Рисунок 4 – Команда `whois`

Использование утилиты **mtr** (рисунок 5). С помощью данной утилиты, как и с помощью **traceroute** можно обнаружить проблемы на сети, если они есть.

```

My traceroute [v0.92]
administrator-VirtualBox (10.0.2.15) 2021-05-31T20:15:07+0300
Keys: Help Display mode Restart statistics Order of fields quit
          Packets
Host      Loss%  Snt   Last   Avg   Best  Wrst  StDev
1.  _gateway  0.0%   4    0.4   0.3   0.3   0.4   0.1
2.  192.168.1.1 0.0%   4    1.7   1.7   1.5   1.9   0.2
3.  msk-b22-b36.ti.ru 0.0%   4    2.7   8.8   2.5  26.5  11.8
4.  orl-b1-te-1-0-0-4000.ti.ru 0.0%   4    4.1  18.7   2.9  36.8  17.7
5.  212.1.251.64 0.0%   4    8.8   4.7   3.3   8.8   2.7
6.  209.85.149.28 0.0%   4    4.0   4.9   4.0   5.9   0.9
7.  209.85.253.53 0.0%   4    3.9   4.5   3.7   7.0   1.6
8.  108.170.250.83 25.0%   4    3.6   3.8   3.6   4.0   0.2
9.  ???
10. 74.125.253.109 0.0%   4   25.3  23.3  18.9  29.7   5.1
11. 172.253.51.187 0.0%   4   18.2  18.8  17.9  21.1   1.5
12. ???
13. ???
14. ???
15. ???
16. ???

```

Рисунок 5 – Утилита mtr

Контрольные вопросы

1. Как узнать доменное имя хоста и дату его регистрации?
2. Какие команды помогут определить доступность узла?
3. Как работает команда traceroute?
4. Для чего нужна команда host?
5. Опишите назначение утилиты mtr?

Лабораторная работа №8 Изучение работы пользователей в системе Linux с различными правами доступа

Цель работы

Изучить работу пользователей в системе Linux с различными правами доступа

Задание

1. Создать и удалить файл под обычным пользователем;
2. Попытаться удалить файл, принадлежащий суперпользователю с помощью учетной записи обычного пользователя;
3. Удалить файл под учетной записью root;
4. Сделать скриншоты;
5. Ответить на вопросы;
6. Сформировать отчет.

Порядок выполнения лабораторной работы

В операционной системе Linux возможно создавать, редактировать и удалять пользователей. Изначально в системе присутствует пользователь "root" с

максимальными правами. Он может осуществлять любые операции над всеми файлами и каталогами (директориями) в системе. При установке системы был создан пользователь "administrator". Его домашний каталог "/home/ administrator ". У пользователя " administrator " нет прав "root". Попробуйте удалить какой-нибудь файл, который принадлежит пользователю "root". Просмотрите содержание каталога "/bin" (рисунок 1):

ls /bin

Попробуйте удалить утилиту "nano" (сейчас зарегистрирован пользователь " administrator "):

rm /bin/nano

Подтвердите удаление, введите символ "y".

```
administrator@administrator-VirtualBox:~$ rm /bin/nano
rm: удалить защищенный от записи обычный файл '/bin/nano'? y
rm: невозможно удалить '/bin/nano': Отказано в доступе
administrator@administrator-VirtualBox:~$
```

Рисунок 1 - Содержание директории "/bin", сообщение об отказе в удалении утилиты "nano" обычным пользователем

Команда "rm" указывает на то, что у пользователя "administrator" нет прав на удаление утилиты "nano". Посмотрите подробную информацию об утилите "nano", введите команду (рисунок 2):

ls -la /bin

```
administrator@administrator-VirtualBox:~$ ls -la /bin
итого 12644
drwxr-xr-x  2 root root    4096 окт 24 14:05 .
drwxr-xr-x 24 root root    4096 окт 24 14:01 ..
-rwxr-xr-x  1 root root 1113504 июн  7  2019 bash
-rwxr-xr-x  1 root root  748968 авг 29  2018 brltty
-rwxr-xr-x  1 root root   34888 июл  4  2019 bunzip2
-rwxr-xr-x  1 root root 2062296 сен 18  2020 busybox
-rwxr-xr-x  1 root root   34888 июл  4  2019 bzip2
lrwxrwxrwx  1 root root      6 авг  4 23:18 bzip2 -> bzip2
-rwxr-xr-x  1 root root   2140 июл  4  2019 bzip2
lrwxrwxrwx  1 root root      6 авг  4 23:18 bzip2 -> bzip2
-rwxr-xr-x  1 root root   4877 июл  4  2019 bzip2
lrwxrwxrwx  1 root root      6 авг  4 23:18 bzip2 -> bzip2
-rwxr-xr-x  1 root root   3642 июл  4  2019 bzip2
-rwxr-xr-x  1 root root   34888 июл  4  2019 bzip2
```

Рисунок 2 - Права доступа к утилите "nano"

Первое обозначение "root" указывает на то, что утилита принадлежит пользователю "root", а второе – на то, что она принадлежит группе пользователей "root". Первые три символа прав доступа к файлу "rwx" – права владельца утилиты (r - чтение, w - запись (удаление, переименование и др.) и x – запуск), вторые три символа "r-x" – права группы "root" (только r – чтение и x – запуск) и последние три символа "r-x" – права других пользователей, включая "administrator" (только r – чтение и x – запуск). Пользователь "administrator" не может переименовать или удалить утилиту "nano".

Зарегистрируйтесь под пользователем "root". Для этого введите команду:

sudo su

Введите пароль, который был задан при установке системы. Вводимые символы пароля не отображаются. Теперь в приветствии отображается пользователь "root" (рисунок 3):

```
administrator@administrator-VirtualBox:~$ sudo su
[sudo] пароль для administrator:
root@administrator-VirtualBox:/home/administrator#
```

Рисунок 3 - Переключение на пользователя "root"

Создайте в корневой директории два файла (они будут созданы пользователем "root"):

cd /

touch text_file_1.txt

touch text_file_2.txt

Посмотрите на права, присвоенные этим файлам (рисунок 4):

ls -la

```
dr-xr-xr-x 13 root root 0 окт 31 16:42 sys
-rw-r--r-- 1 root root 0 окт 31 21:03 text_file_1.txt
-rw-r--r-- 1 root root 0 окт 31 21:03 text_file_2.txt
drwxrwxrwt 17 root root 4096 окт 31 21:02 tmp
```

Рисунок 4 - Обзор прав доступа к файлам, созданным пользователем "root"

Видно, что оба файла принадлежат пользователю "root" и у него есть права rw-, т.е. на чтение и запись (удаление, переименование). Попробуйте удалить файл "root_file_1.txt":

rm text_file_1.txt

Проверьте, удалился ли он (рисунок 5):

ls -la

```
drwxr-xr-x 14 root root 4096 окт 24 13:53 snap
drwxr-xr-x  2 root root 4096 авг  7  2020 srv
dr-xr-xr-x 13 root root    0 окт 31 16:42 sys
-rw-r--r--  1 root root    0 окт 31 21:03 text_file_2.txt
```

Рисунок 5 - Удаление файлов, созданных пользователем "root"

Выйдете из учетной записи пользователя "root":

exit

Теперь активен пользователь "administrator". Перейдите в корневой каталог и попытайтесь удалить второй файл " root_file_2.txt " (рисунок 6):

cd /

rm root_file_2.txt

```
administrator@administrator-VirtualBox:~$ cd /
administrator@administrator-VirtualBox:/$ rm text_file_2.txt
rm: удалить защищенный от записи пустой обычный файл 'text_file_2.txt'? y
rm: невозможно удалить 'text_file_2.txt': Отказано в доступе
administrator@administrator-VirtualBox:/$
```

Рисунок 6 - Отказ в удалении файла, созданного пользователем "root", обычным пользователем

Файл не может быть удален пользователем "administrator", т.к. он не является его владельцем. При создании файла его владельцем становится тот пользователь, который его создал. Теперь создайте в домашней директории пользователя "administrator" ("/home/administrator") файл "adm_file.txt" и просмотрите права доступа к этому файлу (рисунок 7):

cd ~

touch adm_file.txt

ls -la

```
administrator@administrator-VirtualBox:~$ touch adm_file.txt
administrator@administrator-VirtualBox:~$ ls -la
итого 88
drwxr-xr-x 14 administrator administrator 4096 окт 31 21:10 .
drwxr-xr-x  5 root                root    4096 авг 29 17:44 ..
-rw-rw-r--  1 administrator administrator    0 окт 31 21:10 adm_file.txt
-rw-----  1 administrator administrator  485 окт 31 20:50 .bash_history
```

Рисунок 7 - Просмотр прав доступа к файлу, созданному пользователем "administrator"

Видно, что владельцем файла является "administrator". Попробуйте удалить этот файл, зарегистрировавшись под пользователем "root" (рисунок 8):

sudo su

```
rm adm_file.txt
```

```
ls -la
```

```
administrator@administrator-VirtualBox:~$ sudo su
root@administrator-VirtualBox:/home/administrator# rm adm_file.txt
root@administrator-VirtualBox:/home/administrator# ls -la
итого 88
drwxr-xr-x 14 administrator administrator 4096 окт 31 21:12 .
drwxr-xr-x  5 root root 4096 авг 29 17:44 ..
-rw-----  1 administrator administrator  485 окт 31 20:50 .bash_history
-rw-r--r--  1 administrator administrator  220 авг  4 23:23 .bash_logout
-rw-r--r--  1 administrator administrator 3771 авг  4 23:23 .bashrc
drwx----- 13 administrator administrator 4096 авг 25 18:52 .cache
drwx----- 12 administrator administrator 4096 авг  5 00:10 .config
-rw-r--r--  1 administrator administrator 8980 авг  4 23:23 examples.desktop
drwx-----  3 administrator administrator 4096 авг  4 23:36 .gnupg
-rw-----  1 administrator administrator 1942 окт 31 16:43 .ICEauthority
drwx-----  3 administrator administrator 4096 авг  4 23:36 .local
-rw-r--r--  1 administrator administrator  807 авг  4 23:23 .profile
-rw-r--r--  1 administrator administrator   0 авг 25 12:48 .sudo_as_admin_successful
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Видео
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Документы
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Загрузки
drwxr-xr-x  2 administrator administrator 4096 авг 25 12:55 Изображения
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Музыка
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Общедоступные
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 'Рабочий стол'
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Шаблоны
```

Рисунок 8 - Пользователь "root" удалил файл, созданный пользователем "administrator"

Как видно, файл был удален пользователем "root", т.к. он имеет максимальные права в системе. Выйдете из учетной записи пользователя "root" с помощью команды:

exit.

Контрольные вопросы

1. Что делает команда exit?
2. Может ли пользователь user удалить файлы пользователя admin?
3. Что такое права пользователя?
4. Что такое права группы?
5. Что показывает команда rm?
6. Можно ли удалить файлы пользователя root из под учетной записи admin?
7. Какие файлы может удалять пользователь root?
8. Как создать новый файл в директории?

Лабораторная работа №9 Изучение инструмента для перенаправления результатов работы команды

Цель работы

Освоить перенаправление результатов работы одной команды на другую

Задание

1. Применить автоматизацию для команды (каждый выбирает свою);
2. Сделать скрипты;
3. Ответить на вопросы;
4. Сформировать отчет.

Порядок выполнения лабораторной работы

При работе с Linux часто возникает необходимость найти в системных файлах (файлах логирования), собирающих информацию о всех событиях произошедших в системе, информацию, относящуюся к работе конкретной программы. Если открыть такой файл при помощи утилиты "cat" или редактора "nano", то может понадобится значительное время при ручном поиске. Откройте при помощи утилиты "cat" и редактора "nano" файл логирования "syslog" и попробуйте найти в нем сведения о работе программы "cron":

```
cat /var/log/syslog
```

```
nano /var/log/syslog
```

Как видно из вывода содержимого файла "syslog" в терминал (рисунок 1) и в редактор "nano" (рисунок 2), имеется большое число записей, в которых сложно найти нужную информацию.

```

root@administrator-VirtualBox:/home/administrator# cat /var/log/syslog
Oct 31 16:47:59 administrator-VirtualBox rsyslogd: [origin software="rsyslogd"
swVersion="8.32.0" x-pid="833" x-info="http://www.rsyslog.com"] rsyslogd was H
UPed
Oct 31 16:48:00 administrator-VirtualBox anacron[787]: Job `cron.daily' termina
ted
Oct 31 16:48:35 administrator-VirtualBox NetworkManager[832]: <info> [16356881
15.3045] connectivity: (enp0s3) timed out
Oct 31 16:48:35 administrator-VirtualBox NetworkManager[832]: <info> [16356881
15.3047] manager: NetworkManager state is now CONNECTED_SITE
Oct 31 16:48:35 administrator-VirtualBox whoopsie[982]: [16:48:35] offline
Oct 31 16:48:35 administrator-VirtualBox dbus-daemon[797]: [system] Activating
via systemd: service name='org.freedesktop.nm_dispatcher' unit='dbus-org.freede
sktop.nm-dispatcher.service' requested by ':1.14' (uid=0 pid=832 comm="/usr/sbi
n/NetworkManager --no-daemon " label="unconfined")
Oct 31 16:48:35 administrator-VirtualBox systemd[1]: Starting Network Manager S
cript Dispatcher Service...
Oct 31 16:48:35 administrator-VirtualBox dbus-daemon[797]: [system] Successfull
y activated service 'org.freedesktop.nm_dispatcher'
Oct 31 16:48:35 administrator-VirtualBox systemd[1]: Started Network Manager Sc
ript Dispatcher Service.
Oct 31 16:48:35 administrator-VirtualBox nm-dispatcher: req:1 'connectivity-cha
nge': new request (1 scripts)
Oct 31 16:48:35 administrator-VirtualBox nm-dispatcher: req:1 'connectivity-cha
nge': start running ordered scripts...
Oct 31 16:52:56 administrator-VirtualBox anacron[787]: Job `cron.weekly' starte
d

```

Рисунок 1 - Вывод командой "cat" информации из файла "syslog" в терминал

```

GNU nano 2.9.3 /var/log/syslog
Oct 31 16:47:59 administrator-VirtualBox rsyslogd: [origin software="rsyslogd$
Oct 31 16:48:00 administrator-VirtualBox anacron[787]: Job `cron.daily' termin$
Oct 31 16:48:35 administrator-VirtualBox NetworkManager[832]: <info> [1635688$
Oct 31 16:48:35 administrator-VirtualBox NetworkManager[832]: <info> [1635688$
Oct 31 16:48:35 administrator-VirtualBox whoopsie[982]: [16:48:35] offline
Oct 31 16:48:35 administrator-VirtualBox dbus-daemon[797]: [system] Activating$
Oct 31 16:48:35 administrator-VirtualBox systemd[1]: Starting Network Manager $
Oct 31 16:48:35 administrator-VirtualBox dbus-daemon[797]: [system] Successfull$
Oct 31 16:48:35 administrator-VirtualBox systemd[1]: Started Network Manager S$
Oct 31 16:48:35 administrator-VirtualBox nm-dispatcher: req:1 'connectivity-ch$
Oct 31 16:48:35 administrator-VirtualBox nm-dispatcher: req:1 'connectivity-ch$
Oct 31 16:52:56 administrator-VirtualBox anacron[787]: Job `cron.weekly' start$
Oct 31 16:52:58 administrator-VirtualBox anacron[2892]: Updated timestamp for $
Oct 31 16:52:59 administrator-VirtualBox anacron[787]: Job `cron.weekly' termi$
Oct 31 16:52:59 administrator-VirtualBox anacron[787]: Normal exit (2 jobs run)
Oct 31 16:53:06 administrator-VirtualBox NetworkManager[832]: <info> [1635688$
Oct 31 16:53:06 administrator-VirtualBox dbus-daemon[797]: [system] Activating$
Oct 31 16:53:06 administrator-VirtualBox systemd[1]: Starting Network Manager $
Oct 31 16:53:06 administrator-VirtualBox whoopsie[982]: [16:53:06] The default$
Oct 31 16:53:06 administrator-VirtualBox whoopsie[982]: [16:53:06] Not a paid $

```

Рисунок 2 - Открытие файла "syslog" в редакторе "nano"

Для автоматизации можно использовать инструмент, перенаправляющий результаты работы одной команды на вход другой. Так, добавление символа "|" после команды "cat /var/log/syslog" позволит передать ее выполнение другой команде, например, "grep", для автоматизации поиска строк, содержащих информацию о работе программы "cron". Введите команду (рисунок 3):

```
cat /var/log/syslog | grep cron
```

```

Oct 31 16:52:59 administrator-VirtualBox anacron[787]: Normal exit (2 jobs run)
Oct 31 17:17:03 administrator-VirtualBox CRON[3008]: (root) CMD ( cd / && run
-parts --report /etc/cron.hourly)
Oct 31 18:17:03 administrator-VirtualBox CRON[3103]: (root) CMD ( cd / && run
-parts --report /etc/cron.hourly)
Oct 31 19:01:56 administrator-VirtualBox systemd[1]: Started Run anacron jobs.
Oct 31 19:01:56 administrator-VirtualBox anacron[3158]: Anacron 2.3 started on
2021-10-31
Oct 31 19:01:56 administrator-VirtualBox anacron[3158]: Normal exit (0 jobs run
)
Oct 31 19:17:02 administrator-VirtualBox CRON[3186]: (root) CMD ( cd / && run
-parts --report /etc/cron.hourly)
Oct 31 20:01:56 administrator-VirtualBox systemd[1]: Started Run anacron jobs.
Oct 31 20:01:56 administrator-VirtualBox anacron[4002]: Anacron 2.3 started on
2021-10-31
Oct 31 20:01:56 administrator-VirtualBox anacron[4002]: Normal exit (0 jobs run
)
Oct 31 20:17:01 administrator-VirtualBox CRON[4626]: (root) CMD ( cd / && run
-parts --report /etc/cron.hourly)

```

Рисунок 3 - Перенаправление вывода данных командой "cat" команде "grep"

Как видно из рисунка 3, выведены данные, содержащие информацию только о работе программы "cron". Если дополнительно необходимо отфильтровать информацию еще по какому-нибудь критерию, то можно добавить символ "|" и после него указать дополнительный атрибут поиска. Например, если нужно дополнительно отфильтровать информацию по дате, тогда введите следующую команду:

cat /var/log/syslog | grep cron | grep "Feb 9" (между Feb и 9 два пробела, критерий поиска выделен красным цветом).

При необходимости сохранить вывод в редакторе "nano" введите следующую команду:

cat /var/log/syslog | grep cron | grep "Feb 9" | nano

Информация будет сохранена автоматически в домашней директории пользователя.

```

Буфер записан в nano.save
root@administrator-VirtualBox: /home/administrator#

```

Рисунок 4 - Сохранения отфильтрованной информации в домашней директории пользователя

Откройте сохраненный файл:

nano nano.save

Из рисунка 5 видно, что все необходимые данные сохранены.

```
GNU nano 2.9.3 nano.save
Oct 31 16:48:00 administrator-VirtualBox anacron[787]: Job `cron.daily'
terminated Oct 31 16:52:56 administrator-VirtualBox anacron[787]: Job
`cron.weekly' started Oct 31 16:52:58 administrator-VirtualBox
anacron[2892]: Updated timestamp for job `cron.weekly' to 2021-10-31
Oct 31 16:52:59 administrator-VirtualBox anacron[787]: Job
`cron.weekly' terminated Oct 31 16:52:59 administrator-VirtualBox
anacron[787]: Normal exit (2 jobs run) Oct 31 17:17:03
administrator-VirtualBox CRON[3008]: (root) CMD ( cd / && run-parts
--report /etc/cron.hourly) Oct 31 18:17:03 administrator-VirtualBox
CRON[3103]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Oct 31 19:01:56 administrator-VirtualBox systemd[1]: Started Run
anacron jobs. Oct 31 19:01:56 administrator-VirtualBox anacron[3158]:
Anacron 2.3 started on 2021-10-31 Oct 31 19:01:56
administrator-VirtualBox anacron[3158]: Normal exit (0 jobs run) Oct 31
19:17:02 administrator-VirtualBox CRON[3186]: (root) CMD ( cd / &&
run-parts --report /etc/cron.hourly) Oct 31 20:01:56
```

Рисунок 5. Содержание файла с информацией

Контрольные вопросы

1. Что такое логи?
2. С помощью каких команд осуществляется просмотр логов?
3. Как посмотреть версию системы?
4. Как осуществляется открытие файлов логов?
5. Где хранятся логи?
6. Как можно отфильтровать логи?
7. Что выполняет команда `cat`?
8. Что такое `nano`?

Практическая работа №6 Изучение файла настроек Shell

Краткая теория

Shell или командная оболочка - это программа, которая организывает среду для выполнения других программ и команд. Командная оболочка имеет свои встроенные команды, арифметические операторы и другие синтаксические выражения, но основная её задача упрощать запуск других программ. Именно командная оболочка занимается поиском программ в текущем каталоге и в путях, указанных в переменной среды `PATH`, управляет сменой текущего каталога и переменными окружения. Таким образом, основная задача оболочки - интерпретировать команды пользователя и выполнять их независимо от того, внешние ли это программы или внутренние команды. Чтобы понять с чем вы имеете дело, командой оболочки или внешней программой можно воспользоваться командой `whereis`. Например команда `cd`, это не программа, а встроенная команда оболочки, как и `bg` и `fg`:

whereis cd

А вот команда `ls` - уже внешняя программа, исполняемый файл которой расположен в файловой системе. Командная оболочка - это традиционный способ ввода данных в Unix подобных операционных системах. Командные оболочки появились почти сразу же после появления ОС Unix и привычных нам интерфейсов взаимодействия с компьютером. Первой командной оболочкой была Thompson Shell, разработанная Кеном Томсоном в Bell Labs в 1971 году. Уже тогда поддерживалось перенаправление ввода-вывода команд с помощью туннелей, а также поддерживались простые условные операторы. Циклы `while` в командных оболочках появились чуть позже, в оболочке PWB Shell. Но современный вид командная оболочка приобрела только с выходом оболочки Борна (Bourne Shell) в 1979 году. В ней было уже автодополнение имен файлов и команд, стандартные переменные окружения и привычные управляющие структуры. Исполняемый файл этой оболочки имел такой путь в системе - `/bin/sh`. В современных системах это обычно ссылка на используемую оболочку. Именно вдохновляясь идеями этой оболочки потом была разработана оболочка Bash (Bourne Again Shell), используемая сейчас по умолчанию в большинстве дистрибутивов Linux. Она уже была разработана Браеном Фоксом в рамках проекта GNU в 1989 году. С тех пор прошло много времени, и эта оболочка сильно устарела. Поэтому были разработаны и другие более современные командные оболочки, такие как ZSH и Fish.

Если вы не меняли командную оболочку, то очень высока вероятность, что у вас используется Bash. Самый простой способ узнать какая оболочка у вас используется на данный момент, посмотреть содержимое переменной окружения SHELL:

echo \$SHELL

Ещё один способ узнать текущую оболочку - воспользоваться командой `ps`. Если ей передать опцию `-p` и идентификатор процесса, то она покажет информацию о нём. Вывести идентификатор процесса текущей оболочки можно с помощью символов `$$`. Поэтому выполните следующую команду:

```
ps -p $$
```

Контрольные вопросы

1. Что такое shell?
2. Кто является разработчиком PWB shell?
3. Что делает команда echo?
4. Чем отличается команда echo от ps?
5. В каком году была разработана оболочка shell?
6. Где лежит исполняемый файл оболочки shell?
7. Какая оболочка в Linux используется по умолчанию?
8. Что в себе содержит командная оболочка?

Лабораторная работа №10 Изучение файла настроек Shell (.bashrc) и команды alias

Цель работы

Ознакомиться с возможностями Shell (.bashrc) и команды alias

Задание

1. Найти файл .bashrc
2. Создать свой вариант сокращения команды
3. Сделать скриншоты
4. Сформировать отчет

Порядок выполнения лабораторной работы

Посмотрите содержание домашней директории с помощью команды (рисунок 1):

ls -la

```
administrator@administrator-VirtualBox:~$ ls -la
.                .config          .profile         Музыка
..              examples.desktop .sudo_as_admin_successful  Общедоступные
.bash_history   .gnupg           Видео            'Рабочий стол'
.bash_logout    .ICEauthority   Документы       Шаблоны
.bashrc         .local           Загрузки
.cache          nano.save        Изображения
administrator@administrator-VirtualBox:~$
```

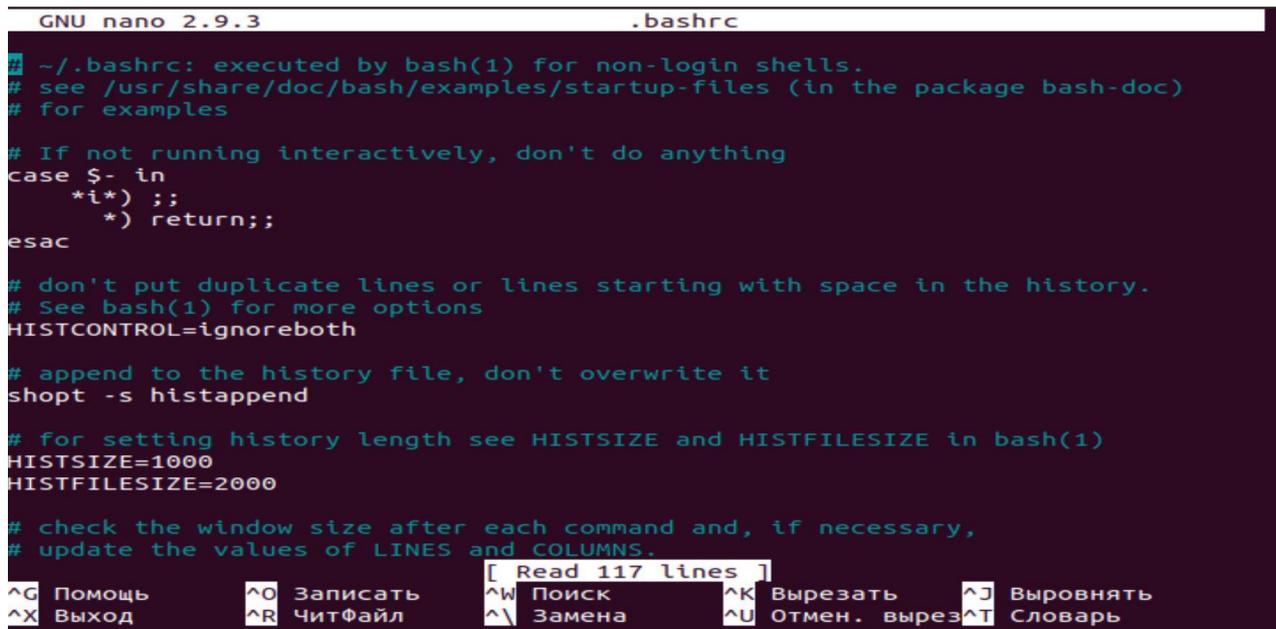
Рисунок 1 - Содержание домашней директории

В директории имеется специальный файл ".bashrc". Это файл с настройками shell – программы, позволяющей вводить все команды. Сейчас установлена и

используется версия, написанная на языке bash. Откройте этот файл с помощью редактора "nano":

nano .bashrc

Найдите строчки с тестом alias (рисунок 2):



```
GNU nano 2.9.3 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.

[ Read 117 lines ]
^G Помощь      ^O Записать
^X Выход       ^R ЧитФайл
^W Поиск      ^K Вырезать
^_ Замена    ^U Отмен. вырез
^J Выворнять  ^T Словарь
```

Рисунок 2 - Содержание файла ".bashrc"

alias – команда, позволяющая задать псевдонимы для других команд. Так, псевдоним "ll" является короткой версией команды "ls -aLF". Закройте редактор "nano", нажав сочетание клавиш "Ctrl+x", и введите сначала короткую, а затем полную версию команды, сравните выведенную информацию (рисунок 3 и рисунок 4):

ll

```

administrator@administrator-VirtualBox:~$ ll
итого 92
drwxr-xr-x 14 administrator administrator 4096 окт 31 21:36 ./
drwxr-xr-x 5 root root 4096 авг 29 17:44 ../
-rw----- 1 administrator administrator 485 окт 31 20:50 .bash_history
-rw-r--r-- 1 administrator administrator 220 авг 4 23:23 .bash_logout
-rw-r--r-- 1 administrator administrator 3771 авг 4 23:23 .bashrc
drwx----- 13 administrator administrator 4096 авг 25 18:52 .cache/
drwx----- 12 administrator administrator 4096 авг 5 00:10 .config/
-rw-r--r-- 1 administrator administrator 8980 авг 4 23:23 examples.desktop
drwx----- 3 administrator administrator 4096 авг 4 23:36 .gnupg/
-rw----- 1 administrator administrator 1942 окт 31 16:43 .ICEauthority
drwx----- 3 administrator administrator 4096 авг 4 23:36 .local/
-rw----- 1 root root 1769 окт 31 21:26 nano.save
-rw-r--r-- 1 administrator administrator 807 авг 4 23:23 .profile
-rw-r--r-- 1 administrator administrator 0 авг 25 12:48 .sudo_as_admin_successful
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Видео/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Документы/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Загрузки/
drwxr-xr-x 2 administrator administrator 4096 авг 25 12:55 Изображения/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Музыка/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Общедоступные/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 'Рабочий стол'/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Шаблоны/
administrator@administrator-VirtualBox:~$

```

Рисунок 3 - Результат работы команды "ll"

Далее введите команду (рисунок 4):

ls -aLF

```

administrator@administrator-VirtualBox:~$ ls -aLF
итого 92
drwxr-xr-x 14 administrator administrator 4096 окт 31 21:36 ./
drwxr-xr-x 5 root root 4096 авг 29 17:44 ../
-rw----- 1 administrator administrator 485 окт 31 20:50 .bash_history
-rw-r--r-- 1 administrator administrator 220 авг 4 23:23 .bash_logout
-rw-r--r-- 1 administrator administrator 3771 авг 4 23:23 .bashrc
drwx----- 13 administrator administrator 4096 авг 25 18:52 .cache/
drwx----- 12 administrator administrator 4096 авг 5 00:10 .config/
-rw-r--r-- 1 administrator administrator 8980 авг 4 23:23 examples.desktop
drwx----- 3 administrator administrator 4096 авг 4 23:36 .gnupg/
-rw----- 1 administrator administrator 1942 окт 31 16:43 .ICEauthority
drwx----- 3 administrator administrator 4096 авг 4 23:36 .local/
-rw----- 1 root root 1769 окт 31 21:26 nano.save
-rw-r--r-- 1 administrator administrator 807 авг 4 23:23 .profile
-rw-r--r-- 1 administrator administrator 0 авг 25 12:48 .sudo_as_admin_successful
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Видео/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Документы/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Загрузки/
drwxr-xr-x 2 administrator administrator 4096 авг 25 12:55 Изображения/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Музыка/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Общедоступные/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 'Рабочий стол'/
drwxr-xr-x 2 administrator administrator 4096 авг 4 23:36 Шаблоны/
administrator@administrator-VirtualBox:~$

```

Рисунок 4 - Результат работы команды "ls -aLF"

Выведенная информация полностью совпадает. Создайте свой собственный alias. Откройте файл с помощью редактора "nano":

nano .bashrc

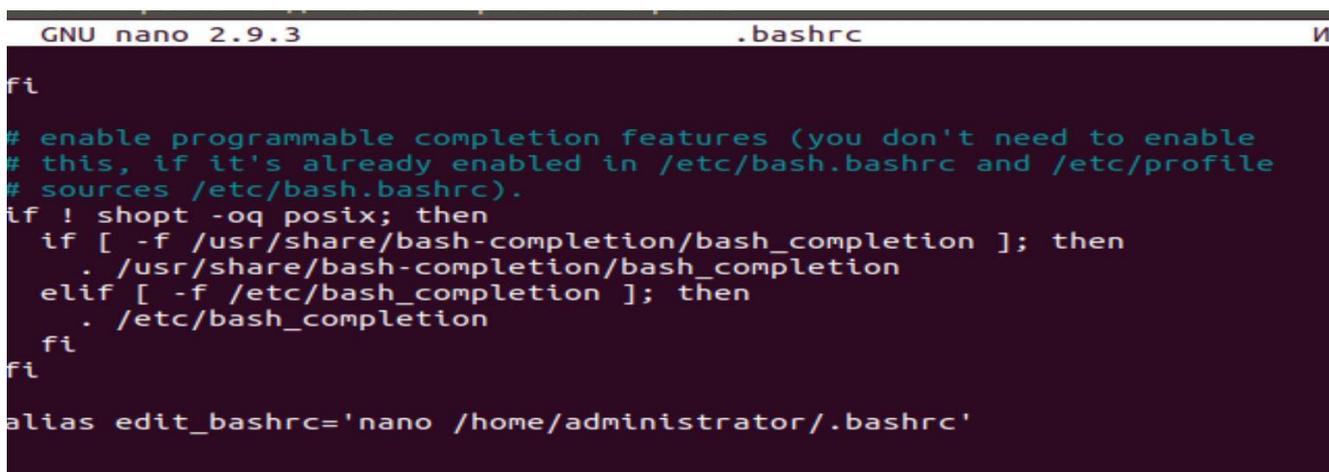
Синтаксис команды для создания псевдонима:

alias <псевдоним='команда_которая_должна_быть_выполнена'>

Создайте псевдоним для команды, позволяющей редактировать файл ".bashrc". В самом конце файла введите команду (рисунок 5):

```
alias edit_bashrc='nano /home/administrator/.bashrc'
```

Для сохранения изменений нажмите сочетание клавиш "Ctrl+o", затем подтвердите имя файла, нажав клавишу "Enter", закройте редактор, нажав сочетание клавиш "Ctrl+x".



```
GNU nano 2.9.3 .bashrc И:
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
alias edit_bashrc='nano /home/administrator/.bashrc'
```

Рисунок 5 - Добавление псевдонима команды в файл ".bashrc"

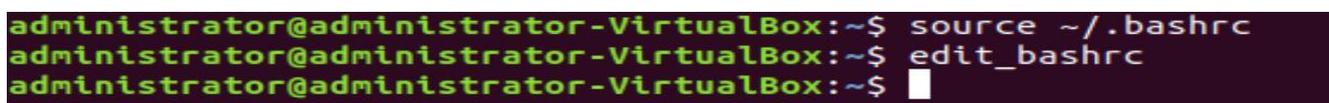
Теперь необходимо произвести перезагрузку настроек командой

"source ~/.bashrc", иначе команда "edit_bashrc" не будет найдена.

Попробуйте ввести команду "edit_bashrc", перезагрузите настройки и повторите (рисунок 6): Кавычки, наверное, лучше убрать.

```
source ~/.bashrc
```

```
edit_bashrc
```



```
administrator@administrator-VirtualBox:~$ source ~/.bashrc
administrator@administrator-VirtualBox:~$ edit_bashrc
administrator@administrator-VirtualBox:~$ █
```

Рисунок 6 - Перезагрузка настроек

Псевдоним полностью выполняет команду для открытия в редакторе файла ".bashrc" (рисунок 7).

```
GNU nano 2.9.3 .bashrc И:
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
alias edit_bashrc='nano /home/administrator/.bashrc'
```

Рисунок 7. Открытие файла ".bashrc" с помощью созданной команды-псевдонима

Контрольные вопросы

1. Что такое alias?
2. Для чего нужен файл bashrc?
3. Зачем нужно проводить перезагрузку bashrc?
4. Для чего используются сокращенные команды?
5. Как сохранить файл в редакторе nano?
6. Можно ли спрятать под псевдонимом несколько команд?
7. Можно ли посмотреть какие псевдонимы уже есть в системе? Если да, то как?

Лабораторная работа №11 Изучение переменных окружения

Цель работы

Приобрести навыки использования переменных окружения.

Задание

1. Узнать, что такое переменные окружения в Linux;
2. Создать переменную окружения;
3. Сделать скриншоты;
4. Сформировать отчет.

Порядок выполнения лабораторной работы

Переменные окружения в Linux - это специальные переменные, определенные оболочкой и используемые программами во время выполнения. Они могут определяться системой и пользователем. Системные переменные окружения Linux определяются системой и используются программами системного уровня. Пользовательские переменные окружения устанавливаются пользователем, для текущей оболочки, временно или постоянно. Вся концепция добавления и удаления переменных оболочки крутится вокруг нескольких файлов, команд и различных оболочек.

Переменная окружения может быть трех типов:

1. Локальные переменные окружения

Эти переменные определены только для текущей сессии. Они будут безвозвратно стерты после завершения сессии, будь то удаленный доступ или эмулятор терминала. Они не хранятся ни в каких файлах, а создаются и удаляются с помощью специальных команд.

2. Пользовательские переменные оболочки

Эти переменные оболочки в Linux определяются для конкретного пользователя и загружаются каждый раз, когда он входит в систему при помощи локального терминала, или же подключается удаленно. Такие переменные, как правило, хранятся в файлах конфигурации: `.bashrc`, `.bash_profile`, `.bash_login`, `.profile` или в других файлах, размещенных в директории пользователя.

3. Системные переменные окружения

Эти переменные доступны во всей системе, для всех пользователей. Они загружаются при старте системы из системных файлов конфигурации: `/etc/environment`, `/etc/profile`, `/etc/profile.d/` `/etc/bash.bashrc`.

Переменные окружения в Linux похожи на глобальные переменные в языках программирования. Во-первых, они записываются заглавными буквами, например, "HOME". Во-вторых, переменные окружения используют специальные значения, которые применяются в программах Linux. Так, чтобы переменной окружения "HOME", в которой содержится текущее значение домашней директории (каталога), определить новое местоположение, необходимо ввести команду "export", затем название переменной окружения ("HOME"), символ присваивания "=" и указать директорию, которая будет считаться домашней. Вывод значения

переменной окружения осуществляется с помощью команды "echo", символа "\$" и названия самой переменной окружения.

Выведите значения переменной окружения "HOME":

echo \$HOME

В результате выполнения команд будет получен вывод
/home/administrator – путь к домашней директории пользователя (рис. 1).

Теперь назначьте в качестве домашней директории пользователя каталог "root":

export HOME=/root

Изменилось отображение приветствия операционной системы (рисунок 1):

administrator@administrator-VirtualBox:~\$

administrator@administrator-VirtualBox:/home/administrator\$

Это произошло, т.к. теперь домашним каталогом пользователя считается "/root", а пользователь находится в каталоге "/home/administrator". Попробуйте перейти в домашнюю директорию пользователя:

cd~

При попытке перехода появляется сообщение о том, что пользователю отказано в доступе, т.к. сейчас работает пользователь с учетной записью "administrator". Верните исходное местоположение домашнего каталога пользователя ("/home/administrator"):

export HOME=/home/administrator

Изменилось отображение приветствия операционной системы:

administrator@administrator-VirtualBox:~\$

Выведите значение переменной окружения "EDITOR", которая содержит текстовый редактор по умолчанию для работы с файлами (рисунок 1):

echo \$EDITOR

```

administrator@administrator-VirtualBox:~$ echo $HOME
/home/administrator
administrator@administrator-VirtualBox:~$ export HOME=/root
administrator@administrator-VirtualBox:/home/administrator$ cd ~
bash: cd: /root: Отказано в доступе
administrator@administrator-VirtualBox:/home/administrator$ export HOME=/home/a
dministrator
administrator@administrator-VirtualBox:~$ echo $EDITOR
administrator@administrator-VirtualBox:~$ █

```

Рисунок 1 - Вывод и изменение значений переменной окружения "HOME"

Сейчас переменная не определена. Установку переменной окружения можно выполнить для одной команды и глобально. Выполните установку переменной окружения "EDITOR" и отредактируйте файл с настройками для утилиты "cron", который имеет название "crontab". Команда "**crontab -e**" позволяет открыть файл редактором по умолчанию. Чтобы установить переменную окружения для одной команды и открыть файл "crontab" редактором по умолчанию, введите команды, показанные на рисунке 2:

EDITOR=nano crontab -e

```

GNU nano 2.9.3 /tmp/crontab.ommEDi/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command

```

Рисунок 2 - Открытие файла "crontab" при определении переменной окружения "EDITOR" для одной команды

Файл "crontab" открылся в тестовом редакторе "nano". Закройте файл нажатием сочетания клавиш "Ctrl+x". Выведите значения переменной окружения "EDITOR" (рисунок 3):

echo \$EDITOR

Сейчас переменная не определена, т.к. она не была установлена глобально. Установите переменную окружения "EDITOR" глобально, чтобы она использовалась всегда по умолчанию (рисунок 3):

```
export EDITOR=nano
```

Выведите значение переменной окружения "EDITOR" (рисунок 3):

```
echo $EDITOR
```

Теперь переменная установлена и выводится название текстового редактора "nano".

```
administrator@administrator-VirtualBox:~$ echo $EDITOR
administrator@administrator-VirtualBox:~$ export EDITOR=nano
administrator@administrator-VirtualBox:~$ echo $EDITOR
nano
administrator@administrator-VirtualBox:~$
```

Рисунок 3 - Глобальное определение значения переменной окружения "EDITOR" и вывод ее значения

Попробуйте открыть файл "crontab" редактором по умолчанию, введите команду (рисунок 4):

```
crontab -e
```

```
GNU nano 2.9.3 /tmp/crontab.Rap6An/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

[Read 22 lines]

^G Помощь	^O Записать	^W Поиск	^K Вырезать	^J Выровнять
^X Выход	^R ЧитФайл	^_ Замена	^U Отмен. вырез	^I Словарь

Рисунок 4 - Открытие файла "crontab" при глобальном определении переменной окружения "EDITOR"

Файл "crontab" открылся в тестовом редакторе "nano". Закройте файл нажатием сочетания клавиш "Ctrl+x".

Контрольные вопросы

1. Что такое переменная окружения?
2. С помощью, какой комбинации клавиш, закрывается файл?
3. Сколько типов переменного окружения вы знаете?
4. Что происходит при вводе команды `echo $EDITOR`?
5. Как можно определить переменную окружения?
6. Как записываются переменные окружения?
7. Как устанавливаются переменные окружения?

Лабораторная работа №12 Установка и удаление утилит

Цель работы

Изучить возможности для установки и удаления утилит

Задание

1. Разобраться с последовательностью действий при установке и удалении утилит
2. Установить утилиту `bleachbit`
3. Обновить и запустить `bleachbit`
4. Удалить утилиту

Порядок выполнения лабораторной работы

Рассмотрим последовательность действий при работе с утилитами на примере "`bleachbit`". При работе с файлами, установке и удалении программ в операционной системе Linux накапливается большое количество служебной информации, которая занимает место на жестком диске. Удалить ненужную информацию может утилита "`bleachbit`". Для установки утилиты введите команду (для завершения команд нажмите клавишу **Tab**):

```
sudo apt-get install bleachbit
```

Важно отметить, что при установке, обновлении и удалении утилит требуется использовать права пользователя "`root`". Для этого необходимо перед командой установки, обновления и удаления ввести команду `sudo`. При этом будет запрошен пароль, который был введен при установке ОС Linux. Вводимые символы пароля

не отображаются (рисунок 1). Также стоит отметить, что может потребоваться подтверждение действий, например, на вопрос «Хотите продолжить?» следует ввести ответ "y".

```
administrator@administrator-VirtualBox:~$ sudo apt-get install bleachbit
[sudo] пароль для administrator:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
 linux-hwe-5.4-headers-5.4.0-42 linux-hwe-5.4-headers-5.4.0-80
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
 libpython-stdlib python python-cairo python-gobject-2 python-gtk2
 python-minimal python-notify python2.7 python2.7-minimal
Предлагаемые пакеты:
```

Рисунок 1 - Установка утилиты "bleachbit"

Для обновления утилиты необходимо ввести команду (рисунок 2):

sudo apt-get upgrade bleachbit

```
administrator@administrator-VirtualBox:~$ sudo apt-get upgrade bleachbit
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет bleachbit самой новой версии (2.0-2).
Расчёт обновлений... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
 linux-hwe-5.4-headers-5.4.0-42 linux-hwe-5.4-headers-5.4.0-80
Для их удаления используйте «sudo apt autoremove».
Следующие пакеты будут обновлены:
 apport apport-gtk bind9-host binutils binutils-common
 binutils-x86-64-linux-gnu dnsutils libbind9-160 libbinutils
 libdns-export1100 libdns1100 libirs160 libisc-export169 libisc169
 libisccc160 libiscfg160 liblwres160 python3-apport python3-problem-report
 tzdata
Обновлено 20 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 паке
тов, и 0 пакетов не обновлено.
20 standard security updates
Необходимо скачать 0 B/5 401 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 16,4 k
B.
Хотите продолжить? [Д/н] █
```

Рисунок 2 - Обновление утилиты "bleachbit"

Запустить утилиту можно через терминал. Причем либо с правами пользователя "study", либо с правами пользователя "root" (рисунок 3).

```
study@study-VirtualBox:~$ bleachbit
study@study-VirtualBox:~$ sudo bleachbit

(bleachbit:7114): IBUS-WARNING **: The owner of /home/study/.config/ibus/bus is
not root!
study@study-VirtualBox:~$
```

Рисунок 3 - Запуск утилиты "bleachbit" в терминале

В операционной системе Linux Ubuntu 18.05.4 LTS Desktop имеется графический интерфейс пользователя. Это дает возможность запускать программу при помощи кнопки "Поиск на компьютере" (рисунок 4):

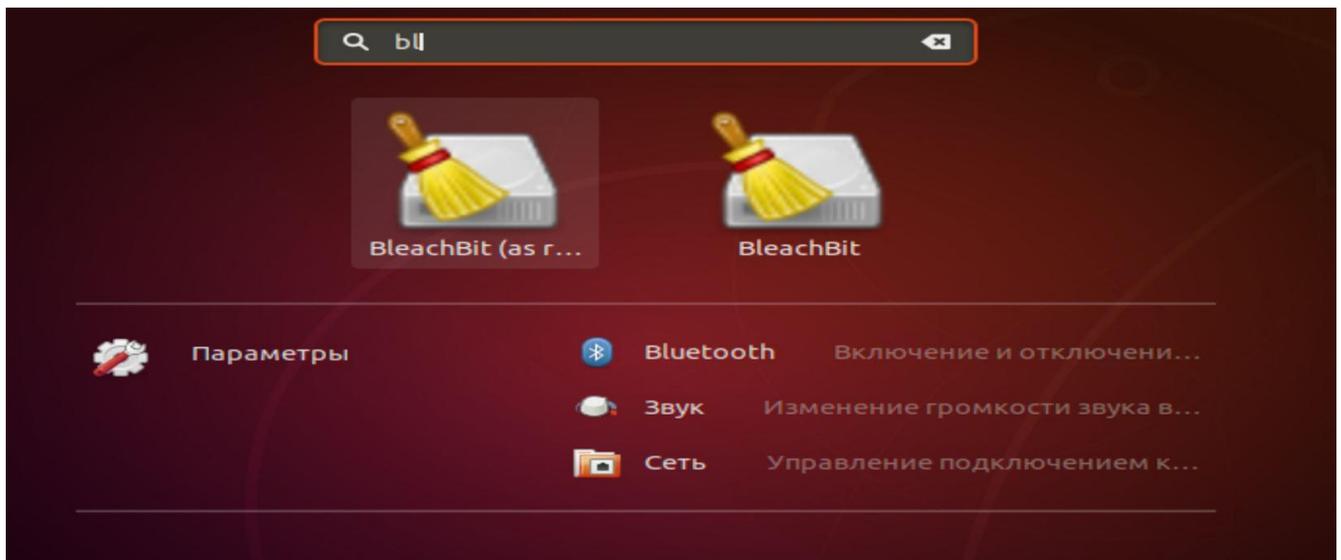


Рисунок 4 - Запуск утилиты "bleachbit" при помощи графического интерфейса

Для полного удаления утилиты необходимо ввести команду (перед "auto-remove" два символа "-"):

sudo apt-get purge --auto-remove bleachbit

Для подтверждения удаления на вопрос «Хотите продолжить?» следует ввести ответ "y" (рисунок 5).

```

administrator@administrator-VirtualBox:~$ sudo apt-get purge --auto-remove bleachbit
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты будут УДАЛЕНЫ:
 bleachbit* libpython-stdlib* linux-hwe-5.4-headers-5.4.0-42*
 linux-hwe-5.4-headers-5.4.0-80* python* python-cairo* python-gobject-2*
 python-gtk2* python-minimal* python-notify* python2.7* python2.7-minimal*
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 12 пакетов,
и 0 пакетов не обновлено.
После данной операции объем занятого дискового пространства уменьшится на 154 МВ.
Хотите продолжить? [Д/н] y
(Чтение базы данных ... на данный момент установлено 202044 файла и каталога.)
Удаляется bleachbit (2.0-2) ...
Удаляется python-notify (0.1.1-4) ...
Удаляется python-gtk2 (2.24.0-5.1ubuntu2) ...
Удаляется python-gobject-2 (2.28.6-12ubuntu3) ...
Удаляется python-cairo:amd64 (1.16.2-1) ...
Удаляется python (2.7.15-rc1-1) ...
Удаляется libpython-stdlib:amd64 (2.7.15~rc1-1) ...
Удаляется linux-hwe-5.4-headers-5.4.0-42 (5.4.0-42.46~18.04.1) ...
Удаляется linux-hwe-5.4-headers-5.4.0-80 (5.4.0-80.90~18.04.1) ...

```

Рисунок 5 - Удаление утилиты "bleachbit"

Контрольные вопросы

1. Как производится удаление утилит?
2. Как произвести полное удаление утилиты?
3. С помощью какой команды можно обновить утилиты?

4. Можно ли установить утилиту не используя права пользователя root?
5. Можно ли запускать утилиты не только через терминал?

Лабораторная работа №13 Запуск собственных скриптов

Цель работы

Создать и запустить собственный скрипт.

Задание

1. Создать скрипт;
2. Запустить;
3. Сделать скриншоты;
4. Сформировать отчет.

Порядок выполнения лабораторной работы

Создайте скрипт, который будет выводить фразу "hello world". Для этого создайте новый файл с названием "hello_world":

nano hello_world

В команде указано, что файл создается с помощью текстового редактора "nano". В первой строке файла необходимо указать интерпретатор, который будет читать код создаваемого скрипта, а затем выполнять его. Будем использовать интерпретатор bash, который является оболочкой в системе Linux. Первая строка должна состоять из символов "#", "!" и пути к интерпретатору "/bin/bash". Далее идет тело программы. Для вывода в терминал строки "hello world" с помощью написанного скрипта необходимо использовать команду "echo" и передать ей строку "hello world" (рисунок 1):

```
#!/bin/bash
```

```
echo "hello world"
```

Сохраните изменения, нажав сочетание клавиш "Ctrl+o", затем нажмите клавишу "Enter" для подтверждения названия файла. Закройте редактор, нажав сочетание клавиш "Ctrl+x". Файл будет сохранен в домашней директории пользователя.

```
GNU nano 2.9.3          hello_world
#!/bin/bash
echo "HELLO WORLD!!!!!!!"

```

[Wrote 2 lines]

^G Помощь	^O Записать	^W Поиск	^K Вырезать	^J Вы
^X Выход	^R ЧитФайл	^\ Замена	^U Отмен. вырез	^T Сл

Рисунок 1 - Содержание созданного файла с именем "hello_world"

Убедитесь в наличии созданного файла "hello_world" в домашней директории:

ls

Теперь необходимо запустить созданный скрипт. Выведите в терминал список файлов, содержащихся в домашней директории, с указанием прав доступа к файлам (рисунок 2):

ls -al

По выведенным данным видно, что файл "hello_world" не является исполняемым, т.к. у его прав доступа ("-rw-rw-r--") отсутствует символ "x".

```
administrator@administrator-VirtualBox:~$ ls -al
итого 100
drwxr-xr-x 15 administrator administrator 4096 ноя  1 00:29 .
drwxr-xr-x  5 root root 4096 авг 29 17:44 ..
-rw-r----- 1 administrator administrator 485 окт 31 20:50 .bash_history
-rw-r----- 1 administrator administrator 220 авг  4 23:23 .bash_logout
-rw-r----- 1 administrator administrator 3826 окт 31 21:42 .bashrc
drwx----- 13 administrator administrator 4096 авг 25 18:52 .cache
drwx----- 12 administrator administrator 4096 авг  5 00:10 .config
-rw-r----- 1 administrator administrator 8980 авг  4 23:23 examples.desktop
drwx----- 3 administrator administrator 4096 ноя  1 00:18 .gnupg
-rw-rw-r--  1 administrator administrator  39 ноя  1 00:28 hello_world
-rw-r----- 1 administrator administrator 1942 окт 31 16:43 .ICEauthority
drwx----- 3 administrator administrator 4096 авг  4 23:36 .local
-rw-r----- 1 root root 1769 окт 31 21:26 nano.save
-rw-r----- 1 administrator administrator 807 авг  4 23:23 .profile
drwx----- 2 administrator administrator 4096 ноя  1 00:18 .ssh
-rw-r----- 1 administrator administrator  0 авг 25 12:48 .sudo_as_admin_successful
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Видео
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Документы
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Загрузки
drwxr-xr-x  2 administrator administrator 4096 авг 25 12:55 Изображения
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Музыка
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Общедоступные
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 'Рабочий стол'
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Шаблоны
administrator@administrator-VirtualBox:~$
```

Рисунок 2 - Содержание домашней директории пользователя с выводом информации о правах доступа к файлам

Попробуйте запустить скрипт. Запуск программы или скрипта в текущей директории осуществляется введением команды `./<название_программы>`:

`./hello_world`

Необходимо сделать файл исполняемым, иначе при попытке его запуска будет выведено сообщение, что пользователю отказано в доступе (рисунок 3). Чтобы изменить права доступа и сделать файл исполняемым, нужно ввести команду

`chmod +x <название_файла>`:

`chmod +x hello_world`

```
administrator@administrator-VirtualBox:~$ ./hello_world
bash: ./hello_world: Отказано в доступе
administrator@administrator-VirtualBox:~$ chmod +x hello_world
administrator@administrator-VirtualBox:~$
```

Рисунок 3 - Изменение прав доступа к файлу "hello_world"

Выведите в терминал список файлов, содержащихся в домашней директории, с указанием прав доступа к файлам (рисунок 4):

ls -al

```
administrator@administrator-VirtualBox:~$ ls -al
итого 100
drwxr-xr-x 15 administrator administrator 4096 ноя  1 00:29 .
drwxr-xr-x  5 root root 4096 авг 29 17:44 ..
-rw-----  1 administrator administrator 485 окт 31 20:50 .bash_history
-rw-r--r--  1 administrator administrator 220 авг  4 23:23 .bash_logout
-rw-r--r--  1 administrator administrator 3826 окт 31 21:42 .bashrc
drwx----- 13 administrator administrator 4096 авг 25 18:52 .cache
drwx----- 12 administrator administrator 4096 авг  5 00:10 .config
-rw-r--r--  1 administrator administrator 8980 авг  4 23:23 examples.desktop
drwx-----  3 administrator administrator 4096 ноя  1 00:18 .gnupg
-rwxrwxr-x  1 administrator administrator  39 ноя  1 00:28 hello_world
-rw-----  1 administrator administrator 1942 окт 31 16:43 .ICEauthority
drwx-----  3 administrator administrator 4096 авг  4 23:36 .local
-rw-----  1 root root 1769 окт 31 21:26 nano.save
-rw-r--r--  1 administrator administrator  807 авг  4 23:23 .profile
drwx-----  2 administrator administrator 4096 ноя  1 00:18 .ssh
-rw-r--r--  1 administrator administrator  0 авг 25 12:48 .sudo_as_admin_successful
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Видео
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Документы
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Загрузки
drwxr-xr-x  2 administrator administrator 4096 авг 25 12:55 Изображения
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Музыка
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Общедоступные
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 'Рабочий стол'
drwxr-xr-x  2 administrator administrator 4096 авг  4 23:36 Шаблоны
administrator@administrator-VirtualBox:~$
```

Рисунок 4 - Вывод содержимого домашней директории пользователя после изменения прав доступа к файлу "hello_world"

По выведенным данным видно, что файл "hello_world" стал исполняемым ("-rwxrwxr-x"). Попробуйте запустить скрипт.

./hello_world

Скрипт запускается успешно и в терминал выводится сообщение "hello world" (рисунок 5).

```
administrator@administrator-VirtualBox:~$ ./hello_world
HELLO WORLD!!!!!!!
administrator@administrator-VirtualBox:~$ █
```

Рисунок 5 - Успешный вывод информации в результате работы скрипта

Контрольные вопросы

1. Что такое скрипт?
2. В каком файле создаются скрипты?
3. Как сделать файл исполняемым?
4. Как написать свой скрипт?
5. Какие права доступа должны быть у исполняемого файла?
6. Что будет если при запуске скрипта у него не будет прав на исполнение?

7. Какой командой можно запустить скрипт?

Практическая работа №7 Удаленное управление ОС

Цель работы:

Подключиться к рабочему столу Ubuntu с помощью протокола удаленного рабочего стола.

Задание

1. Выяснить IP-адрес машины с Linux;
2. Установить xrdp на Ubuntu;
3. Запустить приложение Remote Desktop в Windows;
4. Подключиться к машине с Ubuntu;
5. Сделать скриншоты;
6. Сформировать отчет.

Выполнение

Прежде чем установить соединение с устройством Ubuntu, необходимо знать IP-адрес. Для этого необходимо войти в терминал Ubuntu и набрать команду (рисунок 1):

ifconfig

```
administrator@administrator-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.45 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::be82:3fdd:9e3a:72dd prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:73:51:c3 txqueuelen 1000 (Ethernet)
    RX packets 39 bytes 4450 (4.4 KB)
    RX errors 0 dropped 1 overruns 0 frame 0
    TX packets 89 bytes 10085 (10.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Локальная петля (Loopback))
    RX packets 128 bytes 10049 (10.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 128 bytes 10049 (10.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

administrator@administrator-VirtualBox:~$
```

Рисунок 1 - IP-адрес виртуальной машины

В данном примере IP-адрес виртуальной машины 192.168.1.45.

Другой способ найти IP-адрес - найти значок подключения на панели, щелкнуть правой кнопкой мыши и выбрать Сведения о подключении. Здесь вы также найдете IP-адрес.

Самым простым вариантом является использование протокола Remote Desktop Protocol или RDP. Встроенный в Windows, этот инструмент можно использовать для создания подключения к удаленному рабочему столу в вашей домашней сети.

Для успешного подключения к удаленному рабочему столу, необходимо установить инструмент xrdp на Ubuntu. Для этого откройте окно терминала (**Ctrl + Alt + T**) и введите (рисунок 2):

sudo apt install xrdp

```
administrator@administrator-VirtualBox:~$ sudo apt install xrdp
[sudo] пароль для administrator:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Предлагаемые пакеты:
  guacamole xrdp-pulseaudio-installer
Рекомендуемые пакеты:
  xorgxrdp
Следующие НОВЫЕ пакеты будут установлены:
  xrdp
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакет
ов, и 0 пакетов не обновлено.
Необходимо скачать 419 кВ архивов.
После данной операции объем занятого дискового пространства возрастёт на 2 851
кВ.
Пол:1 http://ru.archive.ubuntu.com/ubuntu bionic/universe amd64 xrdp amd64 0.9.
5-2 [419 кВ]
Получено 419 кВ за 0с (1 876 кВ/с)
```

Рисунок 2 –Установка xrdp

Затем (Рисунок 3):

sudo systemctl enable xrdp

```
administrator@administrator-VirtualBox:~$ sudo systemctl enable xrdp
Synchronizing state of xrdp.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable xrdp
administrator@administrator-VirtualBox:~$ █
```

Рисунок 3 –Активация xrdp

Подождите, пока он установится, затем запустите приложение Remote Desktop в Windows с помощью меню "Пуск" или "Поиск". Введите rdp и нажмите на Remote Desktop Connection (Подключение к удаленному рабочему столу). Когда приложение будет открыто, введите IP-адрес в поле Компьютер (рис. 5).

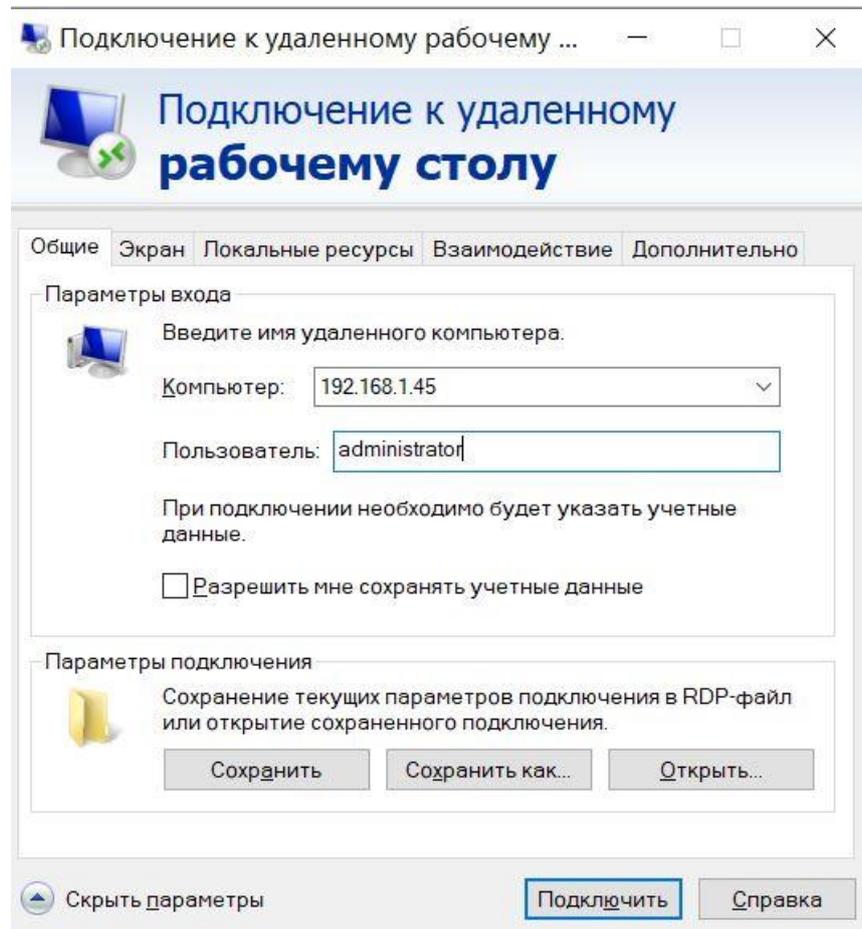


Рисунок 5 – Подключение к удаленному рабочему столу

Затем нажмите кнопку «Показать опции» и добавьте имя пользователя для Ubuntu PC. Вы можете нажать «Сохранить», чтобы сохранить эти настройки для повторного использования в следующий раз. Нажмите кнопку «Подключить», чтобы начать подключение и введите пароль учетной записи Ubuntu, когда это будет предложено. После этого соединение будет установлено, что даст вам полный доступ к удаленному компьютеру Ubuntu с помощью мыши и клавиатуры. Если вы планируете часто использовать это соединение, вы можете создать для него конфигурационный файл, чтобы сэкономить время.

Хотя RDP является хорошим вариантом для удаленного подключения к компьютеру Ubuntu, с приходом Ubuntu 18.04 LTS, такой способ менее надежен. Похоже, что существует проблема с установлением удаленного соединения, когда в учетной записи уже выполнен вход на машине Ubuntu. Таким образом, простой способ обойти это - просто выйти из пользователя. Если это невозможно, попробуйте переключить RDP-соединение с использования сервера Xorg на использование X11rdr, подождите, пока это не сработает, а затем снова попробуйте Xorg. Вы также можете попытаться установить соединение после перезагрузки машины с Linux.

Контрольные вопросы

1. Как узнать IP-адрес виртуальной машины?
2. Что такое удаленное подключение?
3. Что такое RDP?
4. Как осуществляется подключение к другому компьютеру?
5. Для чего применяется команда **ifconfig**?
6. Как активировать xrdp?

Лабораторная работа №14 Подключение по SSH

Цель работы:

Подключиться к виртуальной машине по SSH

Задание

1. Узнать IP-адрес виртуальной машины;
2. Скачать Putty;
3. Установить SSH-соединение;
4. Сделать скриншоты;
5. Сформировать отчет.

Выполнение

SSH - это основной протокол для удаленного управления серверами на базе операционной системы Linux. Все действия при подключении к SSH выполняются в командной строке, но при достаточном уровне знаний и привилегий в системе там можно сделать практически все что угодно. Протокол SSH позволяет вам выполнять команды в удаленной системе так, как будто вы это делаете в своей системе. Вам доступен буфер обмена, вы вводите команды и можете использовать их вывод. Недоступны разве что файлы из вашей файловой системы. Например, когда вы подключитесь к серверу по SSH из Ubuntu, то все будет выглядеть так, как будто вы открыли терминал в своей системе.

Для подключения по SSH нам необходимо знать такие данные:

- IP- адрес сервера, к которому мы собираемся подключиться;
- порт, на котором ожидает подключения SSH сервер, по умолчанию используется 22, но в целях безопасности порт подключения ssh часто изменяют;
- имя и пароль пользователя на удаленном сервере.

В Linux подключение по SSH выполняется с помощью утилиты ssh. Для подключения к удаленному компьютеру ее синтаксис будет выглядеть следующим образом:

```
$ ssh имя_пользователя@айпи_адрес
```

Это самый простой вариант, если вам также нужно задать порт, используйте опцию -p:

```
$ ssh имя_пользователя@айпи_адрес -p порт
```

Для подключения по SSH с Windows необходимо установить программу putty. Это позволит вам установить SSH-соединение, которое дает удаленный

доступ к командной строке Ubuntu. Однако SSH часто отключен по умолчанию, так что если он не установлен, вам придется это исправить.

После установки через терминал (**sudo apt install openssh-server**) вы сможете установить удаленное соединение (просто используя IP-адрес, имя пользователя и пароль Ubuntu) и использовать терминал для установки необходимых инструментов.

Контрольные вопросы

1. Что такое SSH?
2. Что надо знать, чтоб подключиться по SSH?
3. Какие действия позволяет совершить подключение по SSH?
4. Какой порт по умолчанию используется для подключения по SSH?
5. Какие протоколы для подключения вы знаете?
6. Как осуществить подключение с помощью Putty? Какие у нее аналоги?
7. Как установить защищённое подключение?

Лабораторная работа №15 Изменение пароля пользователя

Цель работы:

Научиться изменять пароль пользователя в ОС Linux.

Задание

1. Поменять пароль через терминал;
2. Сделать скриншоты;
3. Сформировать отчет.

Выполнение

В Linux есть несколько утилит с помощью которых может быть выполнена смена пароля Linux. Здесь мы рассмотрим только способы сделать это с помощью терминала.

Список пользователей в Linux хранится в файле `/etc/passwd`, вы можете без труда открыть его и посмотреть, пароли же выделены в отдельный файл -

/etc/shadow. Этот файл можно открыть только с правами суперпользователя, и, более того, пароли здесь хранятся в зашифрованном виде, поэтому узнать пароль Linux не получится, а поменять вручную будет сложно. В большинстве случаев смена пароля выполняется с помощью утилиты `passwd`. Это очень мощная утилита, она позволяет не только менять пароль, но и управлять сроком его жизни. У неё такой синтаксис:

\$ passwd опции пользователь

Рассмотрим опции, чтобы лучше ориентироваться в использовании утилиты:

- d - удалить пароль пользователя, после этого он не сможет войти;
- e - сделать пароль устаревшим;
- i - через сколько дней после того, как пароль устарел, отключить аккаунт, если пользователь не сменил пароль;
- l - запретить пользователю входить в систему;
- n - минимальное количество дней между сменами пароля;
- S - отобразить информацию об аккаунте;
- u - отменяет действие параметра `-l`;
- x - максимальное количество дней, пока пароль можно использовать;
- w - количество дней, после которых нужно предупреждать пользователя о том, что надо сменить пароль.

Смена пароля с помощью passwd

Откройте терминал. Для этого можно нажать сочетание клавиш `Ctrl+Alt+T`. Введите команду (вместо `administrator` укажите свое имя пользователя):

sudo passwd administrator

Появится запрос на ввод текущего пароля пользователя, введите его и нажмите `Enter`. Обратите внимание на то, что при вводе пароля на экран не выводится никаких символов. Затем введите новый пароль и нажмите `Enter`. Затем еще раз введите новый пароль и нажмите `Enter`. Если появилось сообщение

passwd: password updated successfully

это означает, что пароль изменен.

Общие требования для пароля такие:

должен содержать от 6 до 8 символов, причём один или несколько из них должны относиться как минимум к двум из таких множеств:

- Буквы нижнего регистра
- Буквы верхнего регистра
- Цифры от нуля до девяти
- Знаки препинания и знак _

Теперь рассмотрим, как изменить пароль Linux для другого пользователя.

Пароль для другого пользователя можно изменить только из учетной записи суперпользователя, в остальном порядок действий аналогичен.

Контрольные вопросы

1. Назовите основную команду для просмотра пароля?
2. Что делает опция `-e`?
3. Какие требования предъявляются к паролям?
4. Какие действия необходимо осуществить для изменения пароля?
5. Какие опции у команды `passwd`?
6. Где хранится список пользователей в Linux?
7. Где хранятся пароли пользователей в Linux?
8. В каком виде хранятся пароли в Linux?

Лабораторная работа №16 Планирование выполнения команд

Цель работы

Запланировать однократное выполнение команды в заданное время

Задание

1. Установить и включить утилиту `at`;

2. Запланировать выполнение какой-либо команды на определенное время;
3. Сделать скриншоты;
4. Сформировать отчет.

Выполнение

Команда **at** предпочтительнее для ситуаций, когда ваша задача «одноразовая». Ниже приведена таблица с некоторыми основными командами, связанными с **at**.

Таблица 1 – Основные параметры команды at

Команда	Функция
at	позволяет пользователю планировать задачу
atq	выводит список заданий в очередь для вошедшего в систему пользователя или всех пользователей, если они запускаются как sudo
atrm	удаляет задания по указанному номеру задания
batch	указывает системе запускать задание только в указанное время, если загрузка системы находится на определенном уровне (средняя загрузка <1,5)

Использование команды **at** имеет свой уникальный формат. Когда вы хотите запланировать работу, вы наберете в свой терминал:

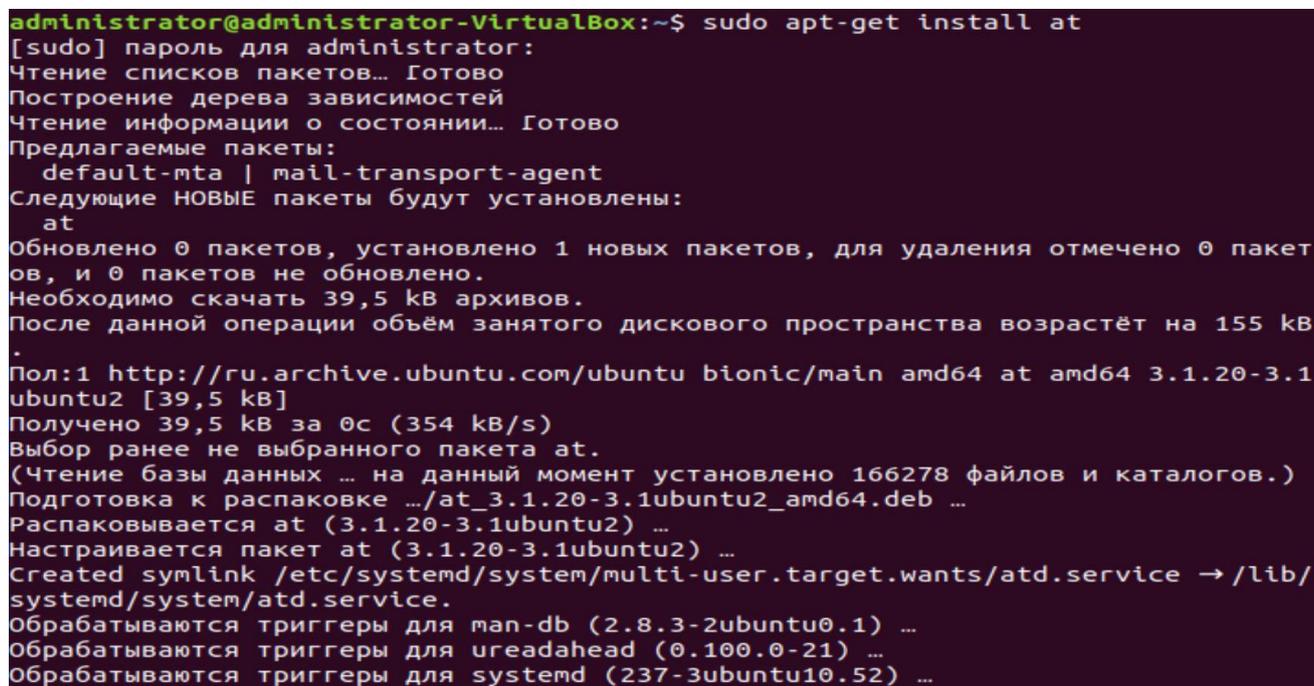
at [time] [date/day]

Время является обязательным, но дата является необязательной. Если ничего не введено, будет сделано предположение на основе текущей даты и системного времени. Когда вы нажмете Enter, вам будет предложено ввести команды. Вы

можете ввести столько, сколько хотите. Введите Ctrl + D, чтобы сохранить и выйти.

Для установки утилиты at необходимо выполнить:

```
$ sudo apt-get install at
```



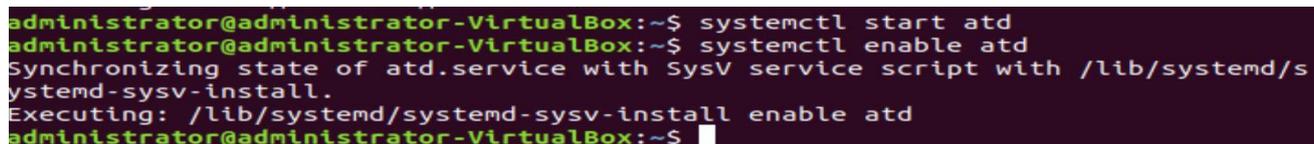
```
administrator@administrator-VirtualBox:~$ sudo apt-get install at
[sudo] пароль для administrator:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Предлагаемые пакеты:
  default-mta | mail-transport-agent
Следующие НОВЫЕ пакеты будут установлены:
  at
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 39,5 кВ архивов.
После данной операции объем занятого дискового пространства возрастёт на 155 кВ.
.
Пол:1 http://ru.archive.ubuntu.com/ubuntu bionic/main amd64 at amd64 3.1.20-3.1ubuntu2 [39,5 кВ]
Получено 39,5 кВ за 0с (354 кВ/с)
Выбор ранее не выбранного пакета at.
(Чтение базы данных ... на данный момент установлено 166278 файлов и каталогов.)
Подготовка к распаковке .../at_3.1.20-3.1ubuntu2_amd64.deb ...
Распаковывается at (3.1.20-3.1ubuntu2) ...
Настраивается пакет at (3.1.20-3.1ubuntu2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/atd.service → /lib/systemd/system/atd.service.
Обрабатываются триггеры для man-db (2.8.3-2ubuntu0.1) ...
Обрабатываются триггеры для ureadahead (0.100.0-21) ...
Обрабатываются триггеры для systemd (237-3ubuntu10.52) ...
```

Рисунок 1 – Установка утилиты at

Далее необходимо запустить и добавить службу в автозагрузку (рисунок 2) с помощью команд

```
systemctl start atd
```

```
systemctl enable atd
```



```
administrator@administrator-VirtualBox:~$ systemctl start atd
administrator@administrator-VirtualBox:~$ systemctl enable atd
Synchronizing state of atd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable atd
administrator@administrator-VirtualBox:~$
```

Рисунок 2 – Добавление в автозагрузку

Проверяем статус (рисунок 3):

```
systemctl status atd
```

```
administrator@administrator-VirtualBox:~$ systemctl status atd
● atd.service - Deferred execution scheduler
   Loaded: loaded (/lib/systemd/system/atd.service; enabled; vendor preset: ena
   Active: active (running) since Mon 2021-11-01 02:19:59 MSK; 8min ago
     Docs: man:atd(8)
    Main PID: 1792 (atd)
      Tasks: 1 (limit: 4663)
   CGroup: /system.slice/atd.service
           └─1792 /usr/sbin/atd -f

ноя 01 02:19:59 administrator-VirtualBox systemd[1]: Started Deferred execution
lines 1-10/10 (END)
```

Рисунок 3 – Проверка состояния atd

Когда atd запущен, вы можете запланировать любую команду или задачу следующим образом. Мы хотим отправить 4 ping запроса на www.google.com, через 5 минут, используем следующую команду:

```
echo "ping -c 4 google.com" | at -m now +10 minute
```

Если вы решите не использовать параметр *-m*, команда будет выполнена, но в выводе ничего не будет напечатано. Однако вы можете использовать перенаправление вывода в файл. Команда at позволяет также указывать точное время для выполнения команды (например: отключить систему в 22:15 сегодня: `echo "shutdown -h now" | at -m 22:15`).

Контрольные вопросы

1. Какие основные команды at вы знаете?
2. Что они делают?
3. Что такое at?
4. Как посмотреть статус at?
5. Что выполняет команда `echo "shutdown -h now" | at -m 22:15`?
6. Как запускать службы в ОС?
7. Какой формат имеет at?
8. В чем отличие at от atd?

Практическая работа №8 Использование CRON

Краткая теория

Cron - это сервис, как и большинство других сервисов Linux, он запускается при старте системы и работает в фоновом режиме. Его основная задача выполнять нужные процессы в нужное время. Существует несколько конфигурационных файлов, из которых он берет информацию о том, что и когда нужно выполнять. Сервис открывает файл `/etc/crontab`, в котором указаны все нужные данные. Часто, в современных дистрибутивах там прописан запуск утилиты `run-parts`, которая запускает нужные скрипты из следующих папок:

- `/etc/cron.minutely` - каждую минуту;
- `/etc/cron.hourly` - каждый час;
- `/etc/cron.daily` - каждый день;
- `/etc/cron.weekly` - каждую неделю;
- `/etc/cron.monthly` - каждый месяц.

В этих папках должны находиться скрипты, которые нужно выполнять с указанным интервалом. Скрипты должны иметь права на выполнение и их имя не должно содержать точки. Это очень сильно облегчает работу с планировщиком для новых пользователей. Также в файле `crontab` прописан запуск команды `anacron`, которая работает, так же как и `cron`, только предназначена для задач, которые нужно выполнять один раз в длительный период, например, раз в день, неделю, месяц, год. Она позволяет выполнять их даже если компьютер работает не всегда и время от времени выключается. Дата выполнения задания последний раз записывается в файл `/var/spool/anacron`, а затем, при следующем запуске `anacron` проверяет был ли запущен нужный процесс в нужное время, и если нет, то запускает его. Сам же сервис `cron` больше рассчитан на выполнение задач в течение дня или с точно расписанным временем и датой.

Для настройки времени, даты и интервала выполнения задания используется специальный синтаксис файла `crontab` и специальная команда. Конечно, вы всегда можете отредактировать файл `/etc/crontab`, но этого делать не рекомендуется.

Вместо этого, есть команда `crontab: crontab -e`. Ее всегда желательно выполнять с опцией `-e`, тогда для редактирования правил будет использован ваш текстовый редактор по умолчанию. Команда открывает вам временный файл, в котором уже представлены все текущие правила `crontab` и вы можете добавить новые. После завершения работы команды `crontab` файл будет обработан и все правила будут добавлены в

`/var/spool/cron/crontabs/имя_пользователя`

При этом добавленные процессы будут запускаться именно от того пользователя, от которого вы их добавляли. Поэтому тут нужно быть аккуратным, и если вам нужно выполнять скрипты от `root`, то и `crontab` нужно выполнить от `root`, а не от пользователя. Это часто становится причиной ошибок.

Давайте рассмотрим синтаксис настройки одной задачи `crontab`: минута час день месяц день_недели /путь/к/исполняемому/файлу. При этом обязательно нужно писать полный путь к команде, потому что для команд, запускаемых от имени `crontab`, переменная среды `PATH` будет отличаться, и сервис просто не сможет найти вашу команду. Это вторая самая распространенная причина проблем с `Cron`. Дата и время указываются с помощью цифр или символа `*`. Этот символ означает, что нужно выполнять каждый раз, если в первом поле - то каждую минуту и так далее.

Контрольные вопросы

1. Что такое `Cron`?
2. Для чего используется программа `crontab`?
3. В какую директорию сохраняются файлы?
4. В чем отличие `Cron` от `at`?
5. Какие проблемы самые распространенные при использовании `crontab`?
6. Можно ли редактировать файл `/etc/crontab`?

Лабораторная работа №17 Анализ сетевого трафика

Цель работы:

Ознакомиться с основными возможностями программы Wireshark.

Задание

1. Установить программу Wireshark на виртуальную машину;
2. Совершить захват трафика при выходе в Интернет;
3. Сделать скриншоты;
4. Описать основные протоколы;
5. Сформировать отчет.

Выполнение

Для выполнения данной работы необходимо скачать и установить программу Wireshark (рисунок 1). С помощью данной программы необходимо будет захватить и проанализировать захваченный трафик.

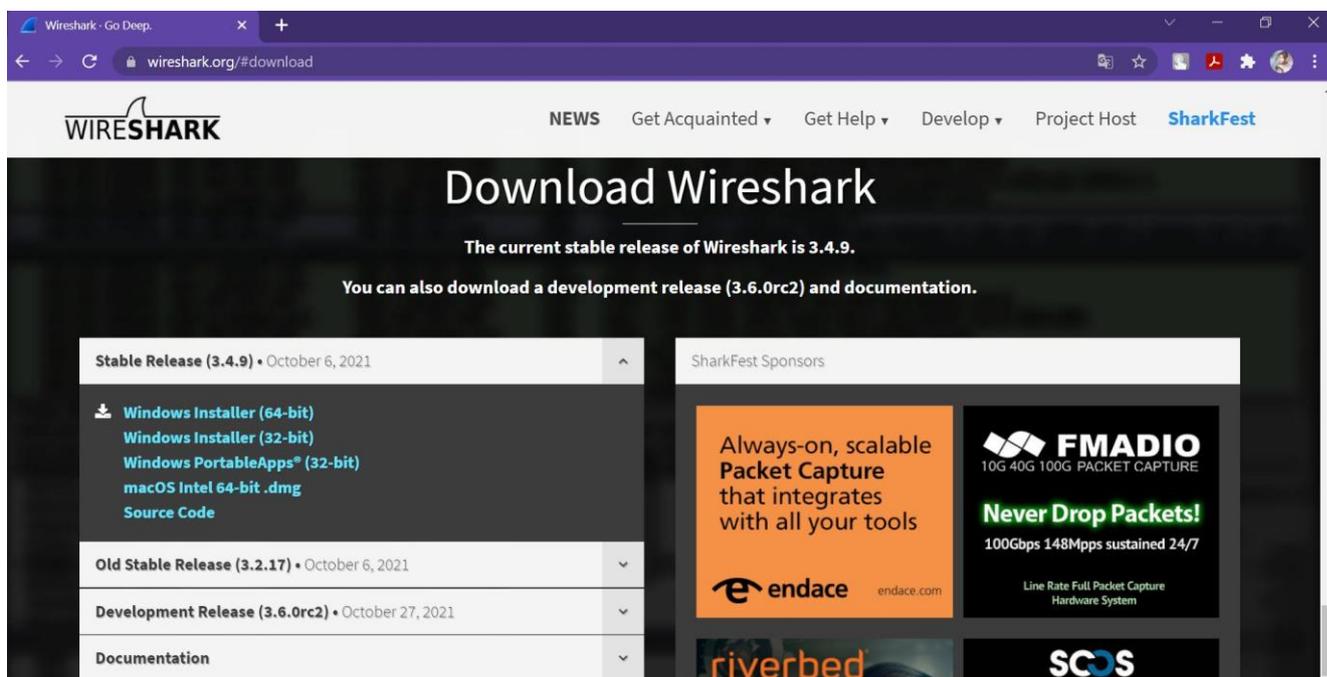


Рисунок 1 –Страница для скачивания Wireshark

Все параметры оставлять по умолчанию

После установки программы Wireshark необходимо ее запустить (рисунок 2):

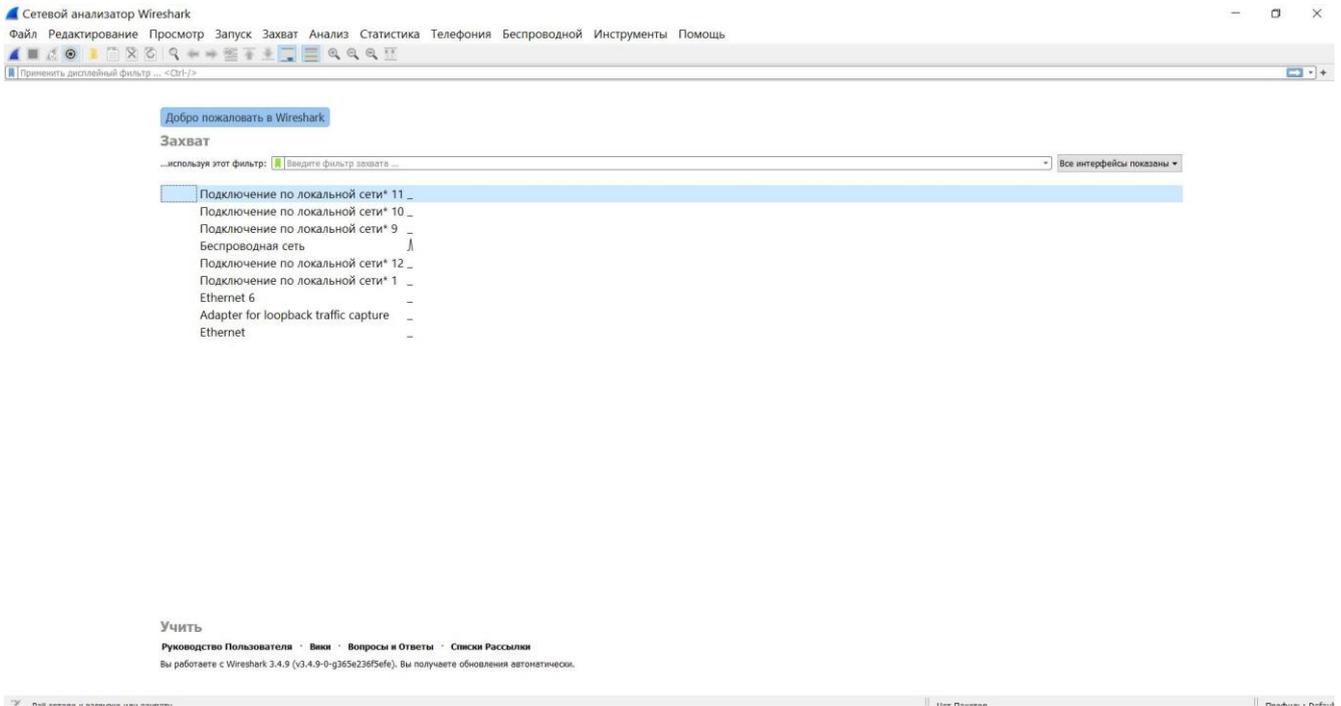


Рисунок 2 – Запущенная программа

После запуска программы необходимо выбрать сетевой интерфейс в соответствии с подключением (беспроводная или локальная сеть) и запустить захват трафика. Далее нужно открыть любую страницу в браузере, затем вернуться в Wireshark (рисунок 3).

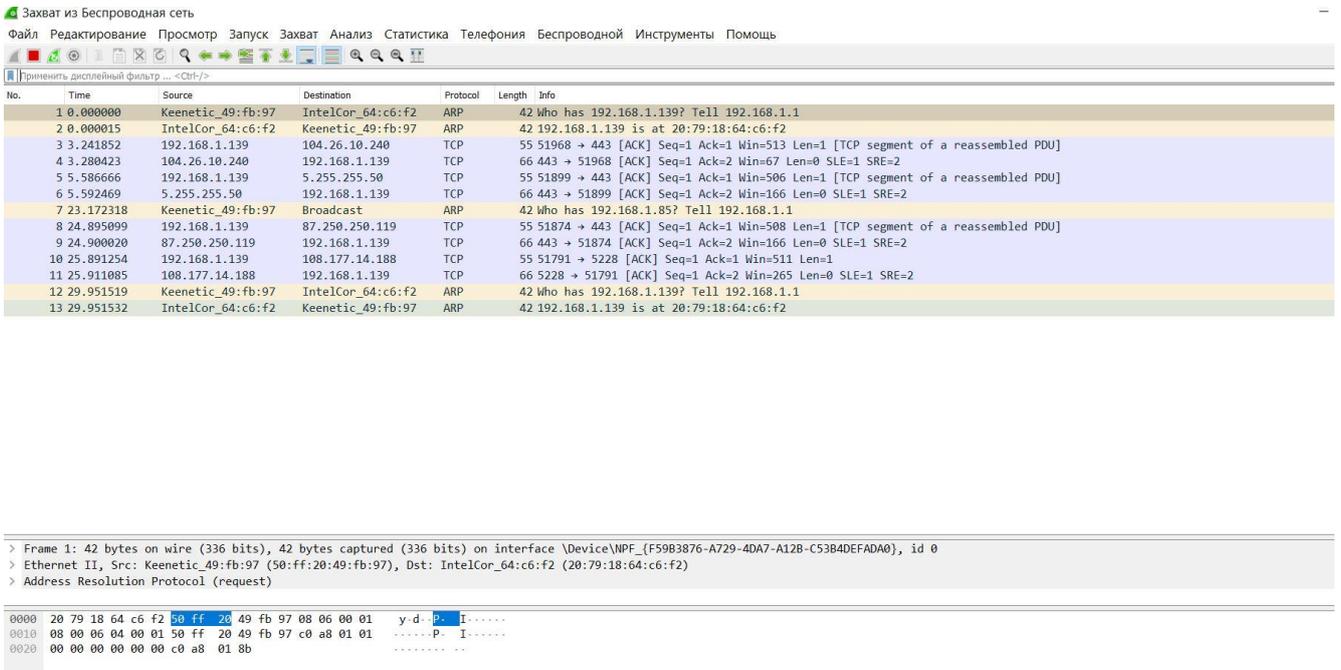


Рисунок 3 – Захват трафика

После возвращения остановить захват трафика (нажать красный квадратик), затем посмотреть какие протоколы были захвачены и описать их.

Контрольные вопросы

1. Что такое трафик?
2. С помощью каких программ осуществляется захват трафика?
3. На каком уровне модели OSI применяется протокол TCP?
4. Какие действия необходимы для захвата трафика?
5. Что означает поле length в программе Wireshark?
6. Можно ли осуществлять захват трафика в беспроводной сети?
7. Какие протоколы вы знаете?

Список литературы

1. Ахматов А. А. «Начальный курс пользователя операционной системы Linux». - М.: Альянс-пресс, 2003. - 448 с.
2. Баррет Д. Д. «Linux : основные команды: карманный справочник». - М.: Кудиц - Образ, 2005. - 288 с.
3. Колисниченко Д.Н. LINUX-сервер своими руками. — СПб: Наука и Техника, 2002. — 576 с: ил.
4. Колисниченко Д.Н., Аллен Питер В. LINUX: полное руководство. — СПб: Наука и Техника, 2006. — 784 с: ил.
5. Кофлер М. Linux. Установка, настройка, администрирование. — СПб.: Питер, 2014. — 768 с.: ил.
6. Скловская С.Л. Команды LINUX. Справочник, 3-е изд., перераб. И доп. / СПб.: ООО «ДиаСофтЮП», 2004. – 848 с.
7. Маликова Е.Е., Панов А.Е. Фатхулин Т.Д., Учебное пособие "Основы работы с операционной системой Linux" по направлению подготовки магистров 11.04.02 - Инфокоммуникационные технологии и системы связи / МТУСИ. – М., 2019. – 92 с.