ПРАКТИКА №1. НАСТРОЙКА И ИСПОЛЬЗОВАНИЕ ОС LINUX В КАЧЕСТВЕ ФАЙЛОВОЙ СИСТЕМЫ

ОГЛАВЛЕНИЕ

1. Операционная система Linux

Для чего нужна операционная система Linux

Операционная система Linux (ОС Linux) – одно из самых популярных серверных решений и ОС для широкого использования в корпоративном частном секторе, основанное на использовании открытого исходного кода. Данная операционная система ввиду своей политики распространения ядра находит множество проявлений в виде настольных операционных систем (самые популярные из них: Linux Mint, Ubuntu Linux, Feedora Workstation), так и в виде портативных или серверных решений.

Необходимость существования Linux на рынке и в сфере ИТ можно объяснить просто – во многих частных решениях на основе собственных аппаратных ресурсов необходима система управления данными ресурсами. Проще всего написать систему управления вычислительными ресурсами на основе открытого, отлаженного, заранее написанного ядра, которое поддерживается сотнями тысяч разработчиков по всему миру.

Ввиду своей широкой применимости данную ОС используют:

1. в работе домашних и рабочих компьютеров;
2. в серверах, распределенных кластерных системах и суперкомпьютерах;

3. в мобильных устройствах (Android) – смартфоны на базе Android являются представителями устройств на семействе ОС Linux;
4. во встраиваемых системах – многие гаджеты умного дома, вроде умных холодильников, стиральных машин и тому подобных;
5. в веб-серверах – использовать Linux на серверах выгодно, потому что эта операционная система бесплатна, вам не надо платить за лицензию, и вы можете сразу же развернуть нужный образ Linux на сервере;
6. в сетевом оборудовании – для обеспечения работы роутеров тоже нужна операционная система и как правило, все эти прошивки и системы работают на ядре Linux;
7. а также в транспорте, авиации, игровых консолях и многих других устройствах узкого назначения.

В Linux-системах пользователи работают через интерфейс командной строки (CLI), графический интерфейс пользователя (GUI), или, в случае встраиваемых систем, через элементы управления соответствующих аппаратных средств. Настольные системы, как правило, имеют графический пользовательский интерфейс, в котором командная строка доступна через окно эмулятора терминала или в отдельной виртуальной консоли. Большинство низкоуровневых компонентов Линукс, включая пользовательские компоненты GNU, использует исключительно командную строку. Командная строка особенно хорошо подходит для автоматизации повторяющихся или отложенных задач, а также предоставляет очень простой механизм межпроцессного взаимодействия. Программа графического эмулятора терминала часто используется для доступа к командной строке с рабочего стола Linux.

Виртуализация операционной системы

В рамках настоящей практической работы студенту предлагается на собственных рабочих местах развернуть систему виртуализации, на основе

которой будет установлена виртуализируемая серверная операционная система Linux под управлением с помощью только лишь командной строки. Данное решение экономит ресурсы рабочего места и ресурсы на использование ОС.

Операционная система хоста – экземпляр операционной системы, работающей на устройстве и в данный момент управляющий аппаратной частью компьютера.

Система виртуализации на уровне операционной системы – позволяет запускать программное обеспечение в изолированных на уровне операционной системы пространствах.

Используемая в рамках данной работы система виртуализации: **Oracle VirtualBox**, которую можно получить по ссылке: https://www.virtualbox.org/. Данная система доступна для установки на всех пользовательских и некоторых специальных операционных системах, среди которых: Windows, Linux, macOS, Solaris, а также Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris, OpenSolaris, OS/2, OpenBSD.

Система виртуализации Oracle VirtualBox является свободно распространяемой в рамках личного использования и позволяет на базе вашей собственной операционной системы, без риска для её удаления или изменения, разворачивать одну или несколько других операционных систем.

Причины использования VirtualBox:

1. простота установки на любую операционную систему (из перечисленных выше);
2. наличие обширного сообщества пользователей, что подразумевает возможность поиска ответа на, практически, любой вопрос;
3. поддержка создания множества виртуальных машин под разными ОС (Windows, FreeBSD, Ubuntu, CentOS, RedHat и др.);
4. создание отдельных виртуальных сетей между виртуальными машинами;

5. создание и восстановление снимков (резервных копий) системы и обширные возможности взаимодействия с операционной системой, на базе которой используется VirtualBox.

Принцип использования VirtualBox. Разворачиваете любой дистрибутив операционной системы. Настраиваете количество ресурсов вашего компьютера (оперативная память, процессор, диски), которые и будет использовать ваша виртуализируемая система. Используете настроенную систему в отдельном открытом окне или подключаетесь к ней через *терминал вашей операционной системе по протоколу ssh*.

Существует также возможность настройки общей папки между системой хоста и виртуализируемой ОС для обмена данными между устройствами.

Установка Oracle VirtualBox

Используемая в рамках данной работы система виртуализации: **Oracle VirtualBox**, которую можно получить по ссылке: https://www.virtualbox.org/.

Загрузим VirtualBox с сайта перейдя во вкладку **downloads** (Рисунок 1).

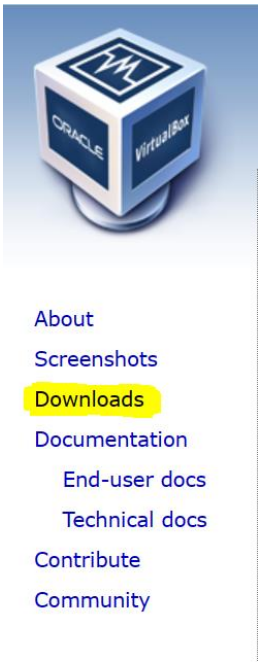

Рисунок 1. Вкладка загрузки

Затем выберите операционную систему, установленную на вышем устройстве для установки дистрибутива (Рисунок 2).

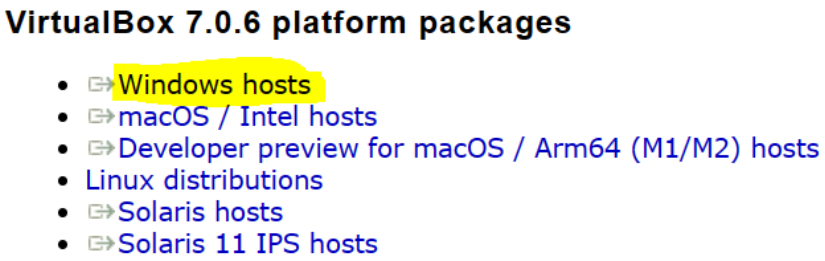

Рисунок 2. Выбор операционной системы, на которой будет устанавливаться VirtualBox

Откройте файл на своем компьютере и начните установку. На вкладке «custom setup» (Рисунок 3) вы можете выбрать свой путь для установки, нажав на **browse**.

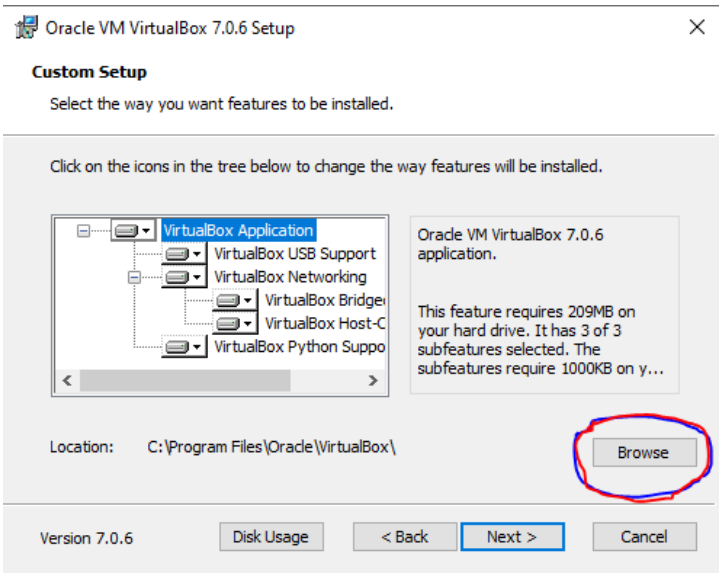

Рисунок 3. Путь, в который будет установлена программа. Browse для изменения пути

Нажимая везде «Next >», а потом открыв приложение мы увидим окно приложения Oracle VM VirtualBox (Рисунок 4). В нем вы можете управлять виртуальными машинами, создавать, экспортировать, настраивать и запускать тестовые операционные системы. Данная программа просто полезна для использования сторонних ОС, а также для тестирования поведения разрабатываемых программ на других платформах.

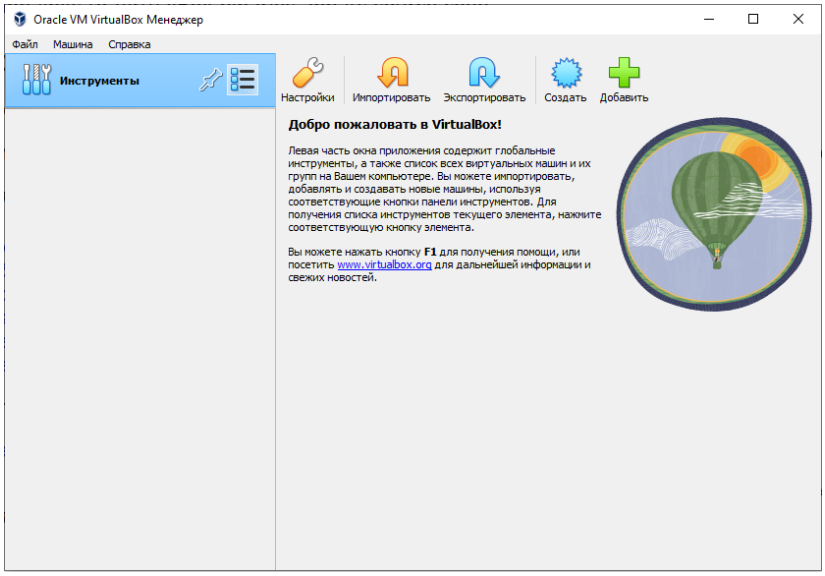

Рисунок 4. Окно приложения VirtualBox

Далее перейдем к настройке заранее заготовленной операционной системы в системе виртуализации.

Настройка конфигурации ОС в Oracle VirtualBox

Для продолжения практики необходимо скачать файл с конфигурацией серверной операционной системы на свой компьютер с диска: https://disk.yandex.ru/d/oe6INm5yyRn1Xw. Это настроенный образ системы, который вы сможете импортировать в VirtualBox (Рисунок 5).

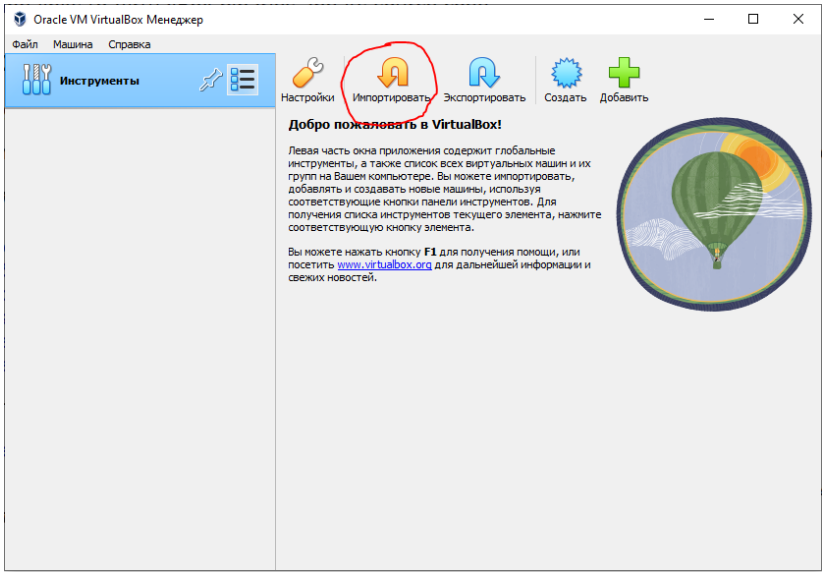

Рисунок 5. Импорт готовой конфигурации ОС Linux

Далее выберите файл, который вы только что скачали через кнопку, выделенную ниже на картинке (Рисунок 6).

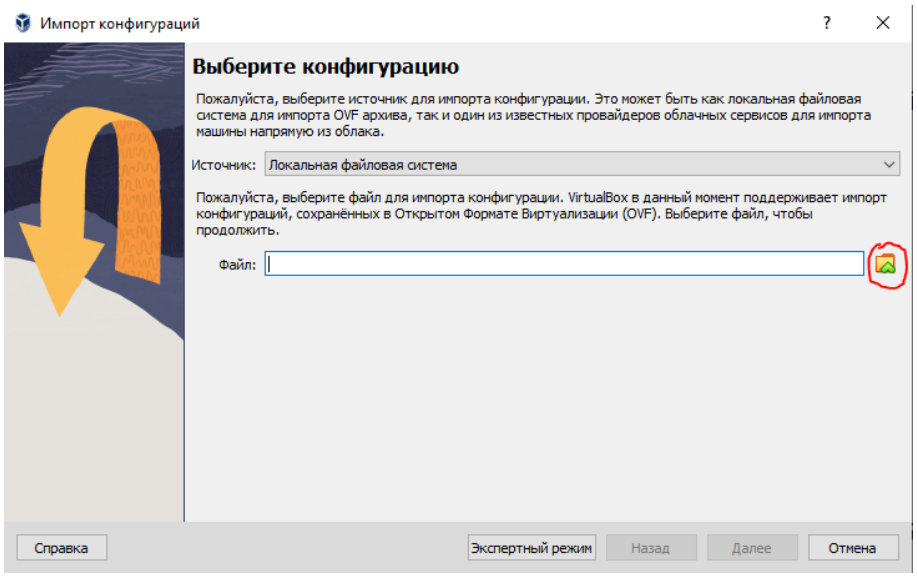

Рисунок 6. Выбор готовой конфигурации операционной системы Linux

Выберите загруженный файл по его полному пути (Рисунок 7).

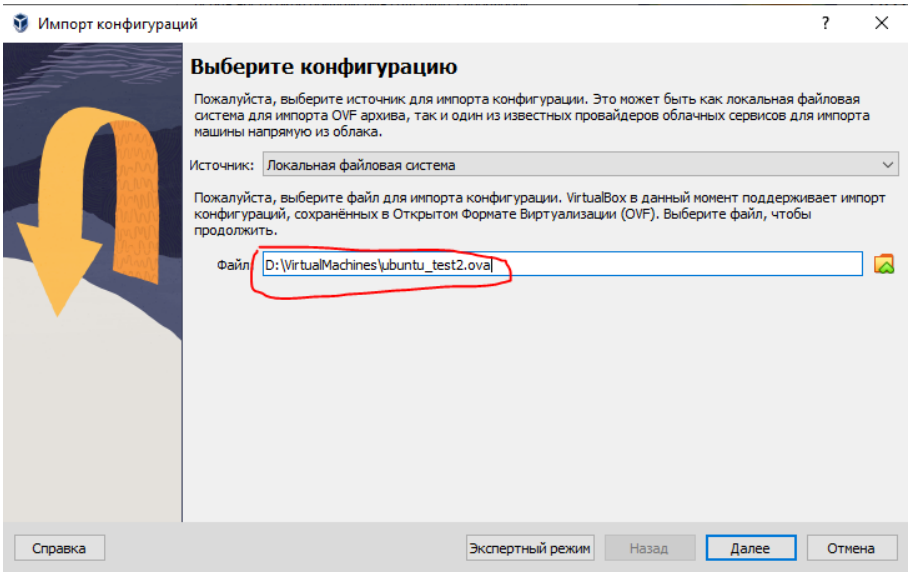

Рисунок 7. Выбор файла

На экране параметров импорта вы можете просмотреть параметров операционной системы, импортируемой в систему виртуализации. На данной вкладке (Рисунок 8) можно настроить папку, в которой будут находится файлы новой системы и её логи.

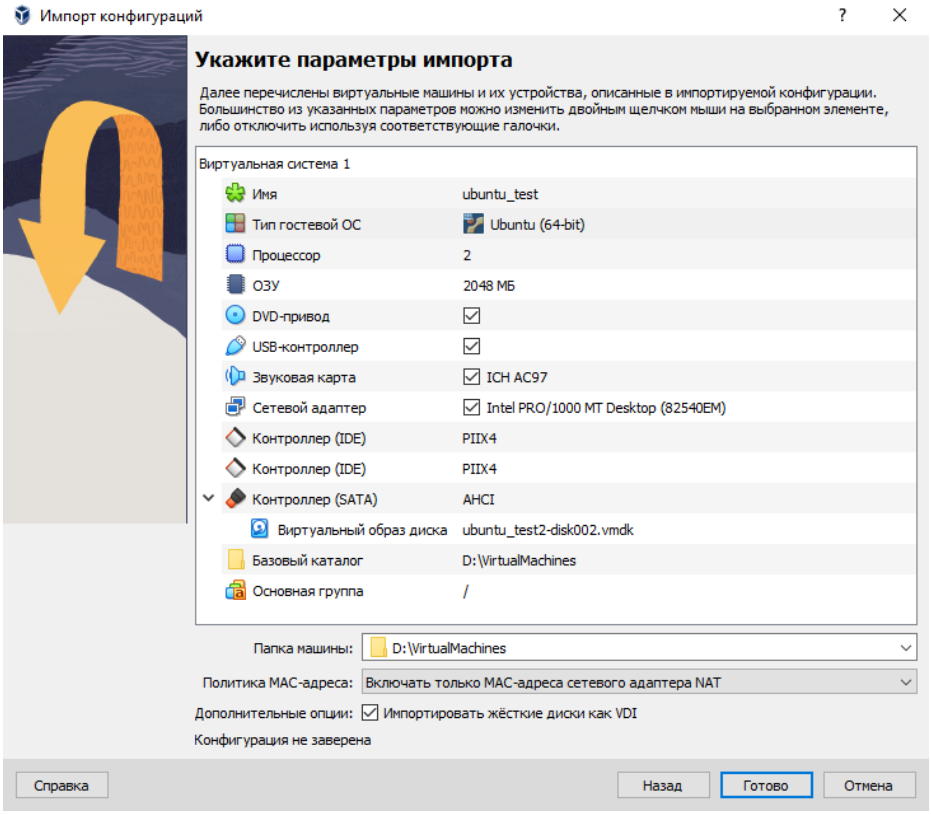

Рисунок 8. Стандартная конфигурация операционной системы

Нажимаем «Готово» и дожидаемся процесса завершения импорта конфигурации. Нажав на кнопку «Настроить» можно перенастроить параметры операционной системы.

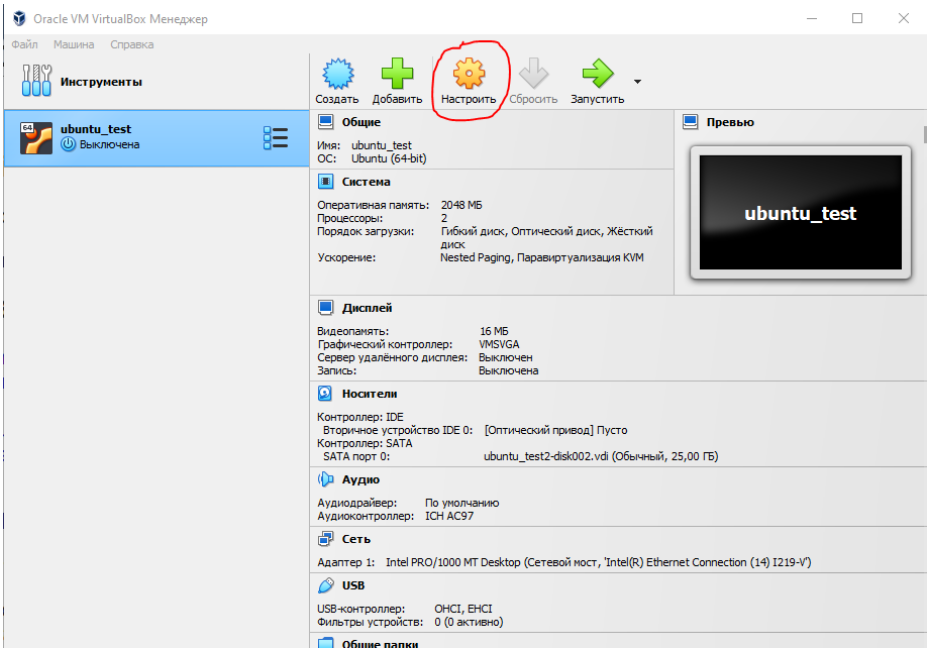

Рисунок 9. Настройки операционной системы

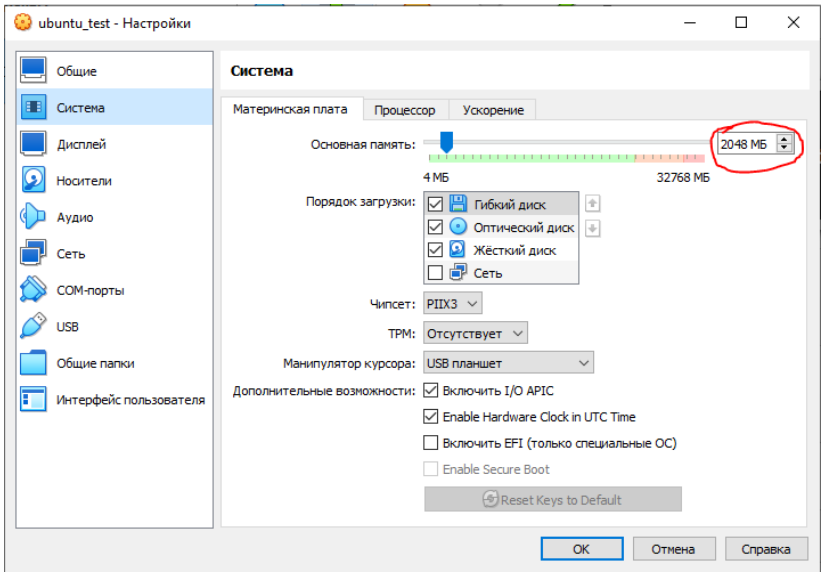

Рисунок 10. Количество ОЗУ, которое вы хотите выделить. Минимум 1024 МБ

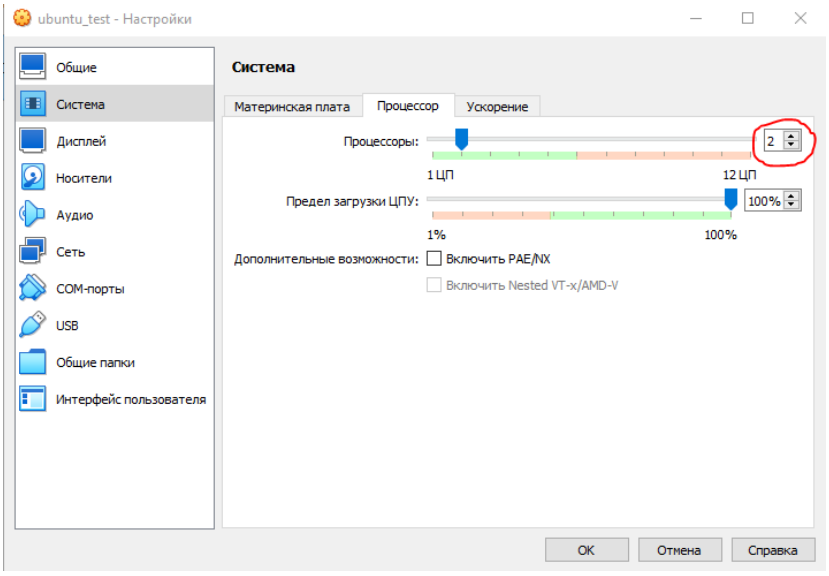

Рисунок 11. Количество ядер процессора для обслуживания ОС

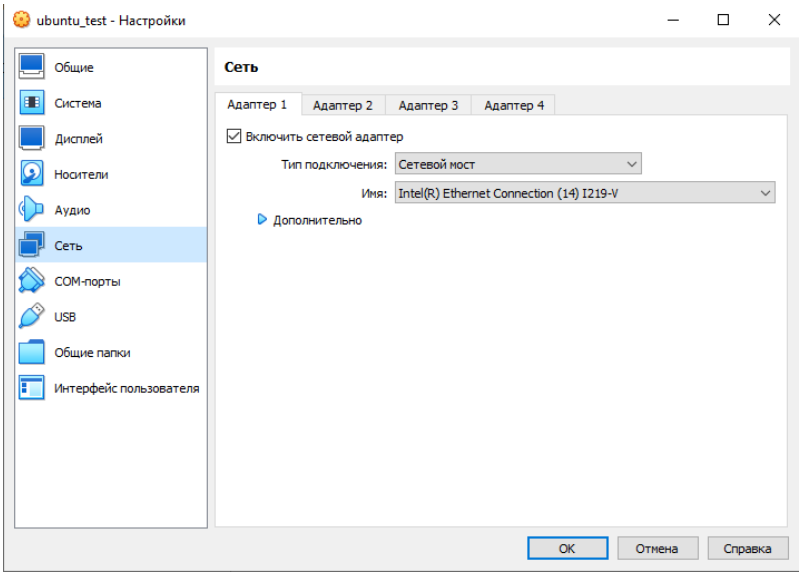

Рисунок 12. Настройка сетевого интерфейса системы

Общая папка – способ обмена файлами и данными между ОС хоста и виртуализируемой операционной системой, который эмулирует сетевую папку. В данной папке со стороны вашей системы «Путь к папке» и со стороны гостевой ОС «Точка подключения» хранятся общие файлы.

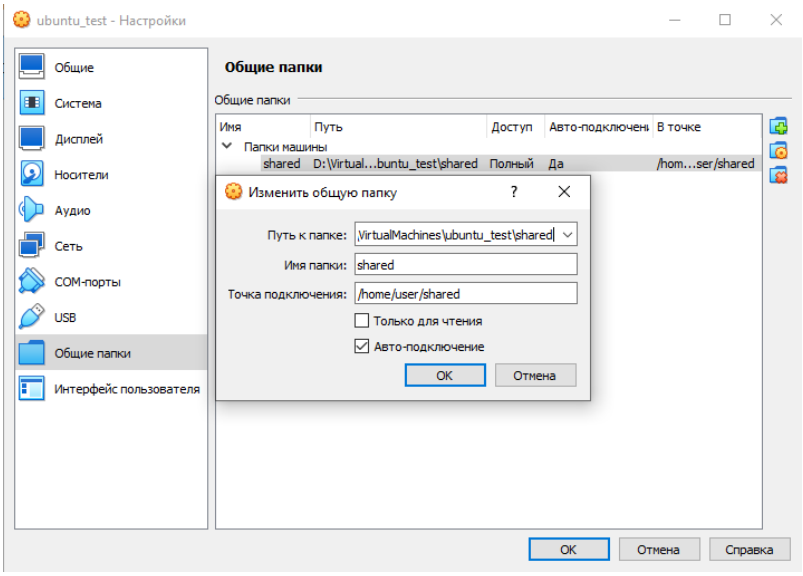

Рисунок 13. Настройка общей папки между вашей ОС и гостевой ОС

Затем нажимаем кнопку «запустить».

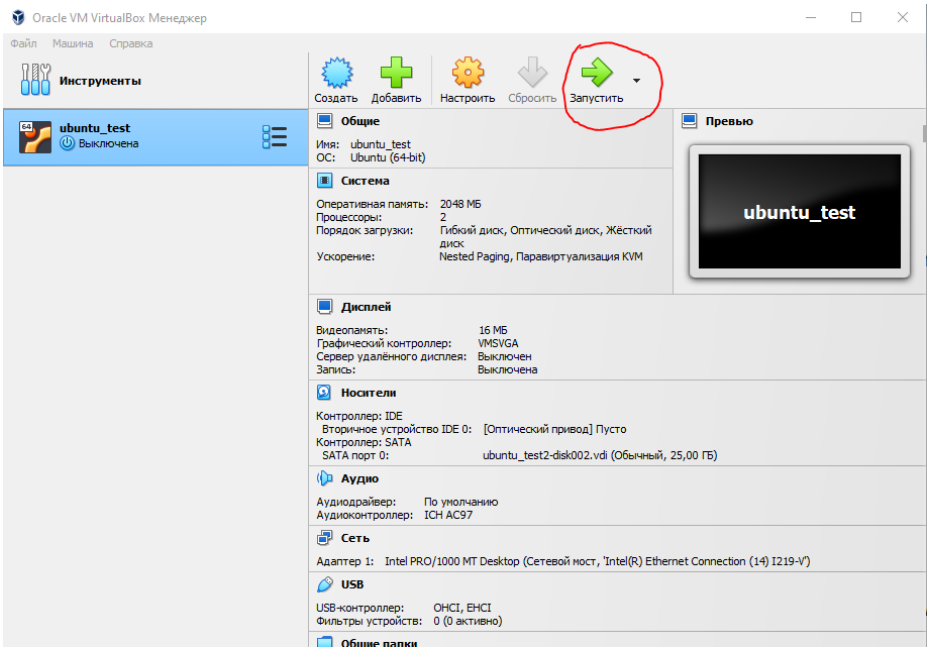

Рисунок 14. Запуск ОС

Возможные проблемы при виртуализации операционной системы в VirtualBox

Если вдруг в ходе вашей установки и запуска операционной системы на базе среды виртуализации у вас возникла проблема, то данный раздел призван решить большую часть данных проблем.

Технологии виртуализации в BIOS. Включите компьютер, а затем сразу же нажмите клавишу F2, F10 или DEL (в зависимости от производителя компьютера), чтобы открыть утилиту настройки BIOS. Нажимая клавиши со стрелками, выберите вкладку. Configuration (Конфигурация), затем выберите. Virtualization Technology (Технология виртуализации). Выберите Enable (Включить). Нажмите клавишу F10 для сохранения настроек и выхода из настройки BIOS.

Платформа виртуализации Windows. В поиске Windows найдите раздел «Включение или отключение компонентов Windows». Далее нажмите галочку рядом с «Платформа виртуальной машины» и «Платформа низкоуровневой оболочки Windows». Нажмите «ОК».

Вход в операционную систему

При входе в операционную систему, пользователю будет предложено ввести логин и пароль для заданной учетной записи пользователя:

Login: <набор_лог.имени> <Enter>

Password: <набор_пароля> <Enter>

[представление системы]

Далее пользователь, успешно вошедший в систему, может взаимодействовать с системой посредством ввода команд в терминал построчно:

$ <ввод_команды> <Enter>

[сеанс работы с системой]

Командой **exit** необходимо завершать сеанс работы с системой, так как только при этом завершаются все процессы, обслуживавшие данный терминал пользователя.

При работе с образом, предложенным в курсе используемое имя пользователя: **user**, и пароль также: **user**.

Команды терминала

Управление серверной редакцией операционной системы осуществляется через терминал командной строки, что в свою очередь порождает необходимость использования встроенных команд для взаимодействия с компьютером.

Синтаксис команд интерпретатора можно представить в следующем обобщенном виде:

```
$ имя_команды [-ключи] [аргумент [аргументы]] <Enter>
```

Приглашение **$** и управляющая клавиша **<Enter>** необходимы для синхронизации работы операционной системы и пользователя. Квадратные скобки (**[ ]**) в записи команды указывают на необязательные параметры, угловые скобки (< >) – на обязательные параметры.

Скобки используются только при описании синтаксиса команд и не вводятся при их выполнении. Символ | означает несколько возможных вариантов, а многоточие (**...**) - то, что параметр может повторяться.

Горячие клавиши

При выполнении практической работы вы можете пользоваться следующими клавишами для быстрого выполнения команд:

- Клавиша «Вверх» — просмотр предыдущей выполненной команды.
- Shift + PageUp — прокрутить экран вверх.
- Shift + PageDown — прокрутить экран вниз.
- Tab — дополнить название команды или файла, начинающееся с введенных букв.
- Двойной Tab — вывести доступные названия команд и/или файлов, начинающиеся с введенных букв.
- Ctrl + Insert — копировать выделенный текст.
- Shift + Insert — вставить выделенный текст.
- Ctrl + R — поиск по истории выполненных команд.

Горячие клавиши предназначены для удобного использования операционной системы посредством командной строки.

Подключение к серверу через терминал командной строки

Работа с серверной операционной системой может осуществляться двумя способами: через окно виртуальной машины VirtualBox напрямую или через командную строку операционной системы хоста. Первый вариант ничем не отличается по функционалу от второго. Однако, реальные задачи работы с сервером Linux подразумевают использование внешнего удаленного сервера по сети через протокол **ssh** с использованием командной строки и сетевых папок для обмена файлами.

Опишем здесь принцип подключения к включенной серверной операционной системе через терминал командной строки операционной системы хоста.

1. Узнаем ip-адрес нашей виртуальной операционной системы. Для Ubuntu Server 18.04 данную информацию можно получить при написании в командную строку команды **ifconfig** или **ip a**.

```
user@ubuntu1804-test:~$ ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UP group default qlen 1000
    link/ether 08:00:27:f1:2b:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 86274sec preferred_lft 86274sec
    inet6 fe80::a00:27ff:fef1:2baf/64 scope link
       valid_lft forever preferred_lft forever
```

Нас интересует подсвеченный желтым текст в ячейке BROADCAST.

2. Откроем на нашей операционной системе хоста терминал командной строки. В windows терминал может быть открыт посредством комбинации клавиш **«Win+R»** и затем написанием слова **«cmd»**. Также в поиске или в меню «пуск» можно найти приложение «Командная строка» или «Windows PowerShell».
3. В командной строке необходимо ввести следующую команду:

ssh <имя пользователя linux>@xxx.xxx.xxx.xxx

password: <пароль>

где <имя пользователя linux> и <пароль> — совпадают с теми, что используются при входе в систему,

xxx.xxx.xxx.xxx — ip-адрес виртуальной операционной системы.

В результате проделанных шагов мы можете использовать все возможности операционной системы из выбранной вами программы работы с терминалом командной строки.

Пользователи и суперпользователь

По умолчанию первый пользователь Linux обладает правами суперпользователя, так как он относится к группе **root**. Такая возможность позволяет ему вносить изменения в корневые директории системы. Это позволяет устанавливать программы на операционную систему и перемещаться между каталогами корневой системы.

Так как вы используете образ операционной системы внутри системы виртуализации VirtualBox, то имеете возможность пользоваться правами суперпользователя. Для этого достаточно перейти в режим суперпользователя:

```
$ sudo su
```

Для выхода из режима суперпользователя используем команду **exit.**

```
$ exit
```

Справка по командам в командной строке

Для получения информации о порядке использования команд, изучаемых в ходе выполнения лабораторного практикума, используйте команду **man**:

```
$ man команда
```

Если команда встроена в оболочку (вместо руководства по команде открывается справочная информация по bash), информацию о ней можно получить, используя команду **help**:

```
$ help команда
```

Для анализа проделанной лабораторной работы в целом и подготовки её к защите можно использовать команду **history** — вывод на экран выполненных команд.

2. Управление файлами и каталогами

Используемые команды:

- **>** — создание нового файла или перенаправление потока вывода в файл
- **cat** — вывод содержимого текстового файла
- **cd** — переход в другой каталог
- **cp** — копирование файлов и каталогов
- **du** — вывод информации о месте, занимаемом на диске файлом или каталогом
- **find** — поиск файлов в каталоге
- **head** — вывод первых строк файла
- **less** — вывод содержимого файла на экран с возможностью прокрутки
- **ln** — создание ссылки на файл или каталог
- **ls** — вывод списка файлов и подкаталогов в текущем каталоге
- **mkdir** — создание нового каталога
- **more** — постраничный вывод содержимого файла
- **mv** — перемещение файла или каталог
- **pwd** — вывод пути до текущего каталога
- **rmdir** — удаление каталога
- **rm** — удаление файла
- **tail** — вывод последних строк файла
- **tree** — вывод на экран иерархии каталогов
- **sort** — сортировка строк в файлах и выводе команд
- **touch** — создание нового файла
- **wc** — подсчет количества строк, слов и байт в файле

Иерархическая файловая система

Все данные, хранимые на диске, представлены в виде файлов. Файлы имеют имена, позволяющие пользователям обращаться к данным файла.

Каталог — это совокупность файлов. Каталоги организованы в древовидную структуру.

Текущий рабочий каталог — это каталог, в котором находится пользователь, вызывая ту или иную команду. Узнать текущий рабочий каталог можно, используя команду **pwd**.

Домашний каталог — каталог, в котором хранятся личные файлы пользователя. Каждый пользователь в системе имеет свой личный каталог. В системе ваш домашний каталог обозначается спецсимволом «~», который вы можете использовать для перехода в домашний каталог (**cd** ~), копирования файлов в домашний каталог и в других случаях.

Корневой каталог – первый каталог в древовидной структуре системы, для которого все остальные каталоги являются вложенными. Обратиться к корневому каталогу можно, используя спецсимвол «/». Например, для перехода в корневой каталог необходимо использовать команду **cd /**. Основные каталоги:

/bin — каталог содержит исполняемые файлы, например, ls, vi, cd, cp.

/boot — каталог содержит файлы, необходимые для загрузки системы, например ядро linux и файлы загрузчика (lilo, grub или другого).

/dev — каталог содержит файлы устройств, присоединенных к системе, или файлы виртуальных устройств, созданных ядром.

/etc — каталог содержит файлы конфигурации большей части программ и приложений.

/home — каталог содержит домашние каталоги пользователей.

/lib — каталог содержит библиотеки, необходимые для исполнения приложений из каталогов /bin и /sbin.

/root — домашний каталог пользователя root.

/sbin — каталог содержит исполняемые файлы, используемые при загрузке системы или для ее администрирования суперпользователем root.

/tmp — общий каталог, используемый для хранения временных файлов.

/usr — каталог содержит пользовательские приложения и библиотеки. Большая часть программ и библиотек, не требующихся для загрузки или восстановления системы, хранятся в этом каталоге.

/var — каталог содержит часто изменяемые файлы: системные журналы (/var/log), конфигурационные файлы, сообщения электронной почты, веб-сайты, файловые архивы ftp и другие данные.

Подробное описание структуры каталогов системы можно получить, используя следующую команду:

```
$ man 7 hier
```

Путь к файлу, находящемуся в корневом каталоге, записывается так: **/file**. Путь к файлам в других каталогах, также начинается от «/» и содержит список всех каталогов на пути от корневого каталога до файла. Например, /home/user/file.

Другой вариант записи пути на файл — начинать запись не от корня каталогов, а от текущего рабочего каталога. При этом текущий каталог обозначается символом «**.**», а родительский каталог - «**..**». Например, если вы находитесь в каталоге **/home/user/dir1/**, а хотите посмотреть файл **/home/user/dir2/file**, то *относительный путь* до него будет **../dir2/file**.

Еще один вариант записи пути - путь относительно домашнего каталога. Путь до вашего домашнего каталога можно заменить на символ «~». Например, путь к файлу **/home/user/dir2/file** можно записать так: **~/dir2/file**.

Таким образом, к файлу всегда можно обратиться тремя способами. Путь к файлу относительно корневого каталога всегда начинается с «/» и

называется **абсолютным**. Путь к файлу относительно текущего каталога записывается без «/» и называется **относительным**.

Запись пути относительно домашнего каталога обычно используется при обращении к файлам, находящимся в домашнем каталоге и его подкаталогах.

Команда перемещения по директориям

Перемещаться по каталогам можно с помощью команды **cd**, о которой рассказывалось ранее. В конкретный момент времени один пользователь терминала может находиться в одном каталоге или директории. Данное состояние указывается в приветственной строке перед вводом команды.

Для изменения положения пользователя относительно каталога операционной системы можно использовать команду **cd < путь>**. Примеры использования команды:

```
$ cd .
```

Выполнение команды выше не переводит пользователя ни в одну из директорий (в итоге он окажется в /home/user/ или ~).

```
$ cd ..
```

Выполнение данной команды перемещает пользователя на один каталог вверх по иерархической файловой системе (в итоге он окажется в /home).

```
$ cd ../..
```

Выполнение данной команды перемещает пользователя на два каталога вверх по иерархической файловой системе (в итоге он окажется в /).

```
$ cd <путь к каталогу>
```

Выполнение данной команды перемещает пользователя в выбранный каталог по пути.

Команда просмотра содержимого каталогов

Для просмотра содержимого каталогов используется команда **ls** (ls — сокращенная форма глагола «list»). Команда **ls** без аргументов отображает содержимое текущего рабочего каталога. В качестве аргумента можно указать ссылку на каталог, содержимое которого хочется просмотреть. Наиболее часто используемые ключи команды **ls**:

- **-l** — выводит «длинный список» каталогов и файлов, указывая для каждого элемента его тип (каталог или файл), права доступа, владельца, размер и другие данные.
- **-a** — выводит полный список каталогов и файлов, включая скрытые файлы (их названия начинаются с символа «.»)
- **-R** — используется для рекурсивного вывода содержимого каталога. При этом выводится не только содержимое каталога, указанного в качестве аргумента команды, но и содержимое всех подкаталогов.

Пример использования команды ls:

```
$ ls
dir1 file.txt
$ ls -la
drwx------ 31 user user 4096 Jun 19 00:32 .
drwxr-xr-x 6 root root 4096 Jul 12 23:19 ..
drwxr-xr-x 3 user user 4096 Aug 1 04:25 dir1
-rwxr-xr-x 1 user user 2252 Jul 30 20:07 file.txt
```

В результате вывода команды **ls -l** показывается «длинный список». В него входит:

- тип файла (d — каталог, - — простой файл, l — символьная ссылка, c — символьное устройство, b — блочное устройство, s — сокет, p — канал);
- права доступа к файлу;
- количество ссылок на файл;
- имя владельца;
- имя группы владельца;
- размер файла (в байтах);
- временной штамп;
- имя файла.

Создание каталогов

Для создания каталогов используется команда **mkdir** (**mkdir** — сокращенно от «make directory»). В качестве аргумента команде передается имя каталога, который требуется создать.

Для создания каталога с подкаталогами необходимо использовать ключ -p, так как иначе будет выведена ошибка, и команда не отработает. Пример использования команды **mkdir**:

```
$ mkdir -p dir1/dir2
```

В примере выше в текущей директории пользователя будет создана папка **dir1**, внутри которой будет находиться папка **dir2.**

Просмотр файлов

Для просмотра файлов существует несколько команд, простейшей из которых является **cat**. Если передать ей список файлов в качестве аргумента, то их содержимое будет «склеено» и выведено на экран. Если указать только один файл, то выведется содержимое этого файла.

Пример использования команды cat:

```
$ cat /etc/hosts
```

В результате выполнения данной команды в командную строку будет выведено содержание файла, который содержит список доступных сетевых хостов, определенных в системе в данный момент.

Перенаправление потока вывода в файл

При выполнении команд ls и cat результат их работы отображается на экране. В Linux большинство команд, которые выводят текст на экран, используют понятие стандартный поток вывода. По умолчанию он связан с терминалом. Оболочка bash командной строки позволяет перенаправлять стандартный поток вывода в другие места. Например, в файл. Для этого используется символ «>». Пример использования:

```
$ ls /tmp > file.txt
```

Если файл file.txt существует, то его содержимое будет перезаписано выводом команды ls. Если этого файла не существует, то он будет создан.

Для добавления данных в файл без удаления уже записанной в него ранее информации используются символы «>>». При этом новые данные будут добавлены в конец файла.

При написании Bash-скриптов часто используется перенаправление потока стандартного вывода или потока ошибок, возникших в ходе выполнения скрипта, в специальное устройство /dev/null (пустое устройство). Запись в него происходит успешно независимо от объема переданной информации.

Чтение из /dev/null эквивалентно считыванию конца файла (EOF). Например, для перенаправления потока стандартного вывода при выводе на экран содержимого файла file1.txt в /dev/null используется следующая команда:

```
$ cat file1.txt > /dev/null
```

При выполнении этой команды содержимое файла не будет выведено на экран.

Создание файлов

Пустые файлы можно создать несколькими способами. Например, команда **touch** используется для обновления данных о временах модификации и последнего доступа к файлу. При отсутствии файла, к которому обращается команда, будет создан пустой файл. Пример создания файла с применением команды **touch**:

```
$ touch file.txt
```

Аналогично перенаправлению потока вывода в файл можно создавать новые пустые файлы, перенаправляя в файл отсутствие данных. Пример создания пустого файла:

```
$ file2.txt
```

Таким образом можно создавать любые пустые файлы с любым заранее заданным расширением, например файлы баз данных «**example.db**» или пустые файлы изображений «**image.jpg**».

Копирование файлов и директорий Linux

С помощью команды cp (copy) можно создавать копии файлов и каталогов.

Примеры использования команды cp:

- **$ cp source target** — файл с именем source копируется в файл с именем target
- **$ cp source dir/** — файл с именем source копируется в каталог dir с тем же именем

- **$ cp -r source target** — каталог с именем source копируется в каталог с именем target

Аргумент команды **-r** позволяет копировать каталог вместе с его содержимым, что невозможно без его использования.

Перемещение и переименование файлов и директорий Linux

С помощью команды **mv** (move) файлы можно перемещать из одного каталога в другой или менять имя файла. Примеры использования команды **mv**:

- **$ mv source target** — файл source переименовывается в файл target
- **$ mv source dir/** — файл source перемещается в каталог dir/

Команда mv интересна тем, что принцип ее работы тесно связан с файловой системой: Linux воспринимает имя файла как нечто внешнее по отношению к его содержимому. Несмотря на то, что название команды происходит от слова «перемещение», она редко занимается перемещением данных. Вместо этого файловая система просто изменяет имя.

1. Если имя файла изменяется с /dir/file на /dir/newfile, то это называется **переименованием**.
2. Если имя файла изменяется с /dir/file на /newdir/file, то это называется **перемещением**.
3. Если имя файла изменяется с /dir/file на /newdir/newfile, то это — **перемещение** с **переименованием**.

Но, по сути, в Linux все это является одним и тем же: изменением полного имени файла.

Удаление файлов и директорий Linux

С помощью команды **rm** (remove) файлы можно удалять. Для удаления каталога используется команда **rmdir**. Для ее использования необходимо предварительно удалить все файлы и подкаталоги, так как rmdir удаляет только пустые каталоги.

Для рекурсивного удаления каталога с файлами и подкаталогами можно использовать команду **rm** с ключом **-r**. Пример использования команды rm:

- **$ rm file** — удаление файла с именем **file.**
- **$ rm -r dir1** — удаление каталога **dir1** и всех вложенных файлов.

Изменение файлов

Файлы в командной строке можно изменять с помощью стандартного встроенного редактора **nano**. Его открытие возможно посредством использования схожей по названию команды в командной строке. Пример использования команды **nano**:

- **$ nano** — открытие текстового редактора **nano** в командной строке для набора нового файла с нуля, при этом имя файла нужно будет указать в дальнейшем**.**
- **$ nano file.txt** — открытие файла с именем file.txt в текстовом редакторе или его создание с пустым содержанием, если такого файла нет в текущей директории.

Текстовый редактор **nano** командной строки поддерживает написание текстовой информации и её изменение. Для получения дополнительного функционала необходимо использовать следующие комбинации клавиш:

- «Ctrl + G» — справка по получению помощи;
- «Ctrl + X» — выход из редактора;
- «Ctrl + O» — сохранить изменения;
- «Ctrl + R» — читать файл с выбранным именем;
- «Ctrl + W» — поиск позиции файла или регулярного выражения Perl;
- «Ctrl + \» — поиск и замена символа, слова или регулярного выражения;
- «Ctrl + K» — вырезать выделенный текст;
- «Ctrl + U» — вставка текста;

- «Ctrl + C» — отменить действие или посмотреть текущую позицию символа в тексте;
- «Ctrl + Shift + -» — перейти к позиции в тексте на пересечении строки и столбца в тексте;
- «Alt + U» — отменить последнее действие с текстом;
- «Alt + E» — вернуть последнее действие после отмены
- «Alt + 6» — скопировать выделенный текст;

Более подробно о команде **nano** и её использовании можно прочитать из справки **man,** документации или тут

Поиск файлов и содержимого в них

Команда **find** используется для поиска в файловой системе файлов, которые соответствуют особым критериям. Почти все параметры файла могут быть указаны, например, его имя, размер, время последнего изменения, даже число ссылок. Единственное ограничение — **find** не позволяет искать файлы по их содержимому. Синтаксис использования команды **find**:

```
$ find [каталог поиска] [критерии поиска] [действие]
```

Для поиска файла в корне каталогов по имени и вывода списка файлов без выполнения каких-либо действий может использоваться следующая команда:

```
$ find / -name “*.conf”
```

Для более близкого знакомства с командой **find** воспользуйтесь справкой, вызвав ее командой **man find** на своей рабочей станции. Команда **grep** традиционно используется для выборки из всех данных только тех, что нас интересуют. Пример использования команды **grep**:

```
$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Первый аргумент команды — текст для поиска, остальные аргументы — файлы для поиска. Если команда вызывается только с одним аргументом, то в качестве источника данных она использует стандартный ввод.

Подстановка имен файлов (Pathname Expansion)

Для упрощения работы в оболочке bash можно использовать подстановки, обращаясь к файлам и каталогам.

«~» (Tilde Expansion) обозначает домашний каталог пользователя. Например, команда **cd ~** заменяет команду **cd /home/user** и перемещает вас в домашний каталог. Для перехода в домашний каталог другого пользователя укажите его имя после символа «~», например «~root». Пример:

```
[user@station /]$ cd ~
[user@station ~]$ cd ~user2
[user@station /home/user2]$
```

Символ «*» заменяет любое число произвольных символов в имени файла или каталога. Пример:

```
[user@station ~]$ ls
dir1 f3 file file1 file2 q1
[user@station ~]$ ls file*
file file1 file2
[user@station ~]$ ls *1
dir1 file1 q1
```

Символ «?» аналогичен по своим свойствам символу «*», но заменяет один произвольный символ. Пример:

```
[user@station ~]$ ls
dir1 f3 file file1 file2 q1
[user@station ~]$ ls file?
file1 file2
[user@station ~]$ ls ?1
```

```
q1
```

Использование квадратных скобок ([]) позволяет выбирать произвольные символы для подстановки. Пример:

```
[user@station ~]$ ls
dir1 f3 file file1 file2 q1
[user@station ~]$ ls [f,q]*1
file1 q1
```

Каналы

Поток вывода (**stdout**) из одного процесса может быть связан с потоком ввода (**stdin**) другого процесса. Это носит название «канал». Для создания канала между двумя командами в bash используется символ «|». Пример использования каналов:

Например, пользователю нужно найти самые большие файлы в каталоге /etc. Сначала он составляет команду **find**, которая найдет все файлы, размер которых больше 100Кб.

```
$ find /etc -size +100k
```

Заметив, что команда выводит список файлов без сортировки, он решает отсортировать их в алфавитном порядке.

Можно перенаправить вывод команды **find** в файл, а затем отсортировать список, используя команду **sort**. Вместо этого он решает создать канал, направив результат поиска на вход команды **sort**.

```
$ find /etc -size +100k | sort
```

При этом файлы будут отсортированы в алфавитном порядке.

3. Доступ к файлам, скачивание файлов в сети, кодировка и архивация

Изменение прав доступа к файлам

Механизм разграничения прав доступа позволяет ограничивать доступ пользователей к объектам системы – каталогам и файлам.

Права доступа подразделяются на три типа: чтение (read), запись (write) и выполнение (execute). Эти типы прав доступа могут быть предоставлены трем классам пользователей:

- пользователю-владельцу файла (u),
- выбранной группе пользователей (g) и всем остальным
- пользователям (o).

Владелец файла обычно является членом группы, пользователям которой выдаются права доступа. Разрешение на чтение позволяет пользователю читать содержимое файлов, а в случае каталогов - просматривать перечень имен файлов в каталоге (используя, например, ls).

Разрешение на запись позволяет пользователю писать в файл и изменять его. Для каталогов это дает право создавать новые файлы и подкаталоги, или удалять файлы в этом каталоге.

Разрешение на выполнение позволяет пользователю выполнять файлы (как бинарные программы, так и командные файлы). Разрешение на выполнение применительно к каталогам означает возможность переходить к каталогу, используя, например, команду cd.

Права доступа к обычным файлам и каталогам:

	Чтение (r)	**Запись (w)**	**Выполнение (x)**
Обычный файл	Просмотреть содержимое файла	Изменить файл	Использовать файл как команду
Каталог	Вывести список	Добавить или	Работа с

	содержащихся в каталоге подкаталогов и файлов	удалить файлы	известным файлом внутри каталога

Для просмотра прав доступа к файлу или каталогу можно воспользоваться командой **ls -l**.

Пользователь, создающий объект, становится его владельцем, а основная группа, в которую входит пользователь, становится группой, имеющий доступ с указанными правами к объекту. Пользователь, создавший объект, не может изменить владельца, но может изменить группу и права доступа всех категорий пользователей к объекту. Изменить владельца объекта может только суперпользователь root.

Для изменения владельца файла или каталога используется команда chown. Синтаксис команды:

```
$ chown [-ключи] новый пользователь: [новая группа] имя файла
```

Для рекурсивного изменения владельца каталога и вложенных файлов используется ключ **-R**. Если группу-владельца изменять не требуется, то эта часть команды опускается.

Для изменения группы-владельца файла или каталога воспользуйтесь командой **chgrp**. Синтаксис команды:

```
$ chgrp [-ключи] новая_группа имя_файла
```

Для изменения владельца каталога и всех вложенных файлов используется ключ **-R**.

Существует два формата записи прав доступа: символический, который мы видим, работая с командой **ls -l** (rwxrwxrwx или r-x------) и восьмеричный (777 или 002). Как работать с форматами записи прав доступа?

Файл имеет три разных типа разрешений к доступу чтение (r), запись (w) и выполнение (x) для трех классов пользователей: пользователь (u), группа (g) и остальные (o). При восьмеричной записи каждый тип пользователя с правами доступа получает разряд: место "сотен" - для пользователя (u), место "десяток" для группы (g), а место "единиц" - для остальных (o).

Каждый тип доступа получает цифровое обозначение: чтение (r) получает 4 (100), запись (w) получает 2 (010), а выполнение (x) получает 1 (001). Символы в восьмеричной записи — это совокупность прав доступа для данного класса пользователя с правами доступа.

Используя команду chmod можно изменять права доступа к файлу. Синтаксис команды:

```
$ chmod [-ключи] права_доступа имя_файла_или_каталога
```

Для рекурсивного изменения прав доступа используется ключ **-R**, изменяющий права для каталога и всех файлов внутри него.

Примеры использования команды chmod:

- Исходный файл:

```
-rw-rw-r-- 1 user user 42 Jan 16 08:09 file.txt
```

- Лишить группу права записи:

```
$ chmod g-w file.txt
-rw-r—r-- 1 user user 42 Jan 16 08:09 file.txt
```

- Наделить пользователя-владельца и группу правом исполнения:

```
$ chmod ug+x file.txt
-rwxrwxr-- 1 user user 42 Jan 16 08:09 file.txt
```

- Наделить всех других пользователей правом записи:

```
$ chmod o+w file.txt
```

```
-rw-rw-rw- 1 user user 42 Jan 16 08:09 file.txt
```

- Лишить группу и всех других пользователей всех прав:

```
$ chmod go-rwx file.txt
-rw------- 1 user user 42 Jan 16 08:09 file.txt
```

- Лишить всех права записи:

```
$ chmod a-w file.txt
-r--r--r-- 1 user user 42 Jan 16 08:09 file.txt
```

- Лишить пользователя-владельца и всех других пользователей права чтения:

```
$ chmod uo-r file.txt
--w-rw---- 1 user user 42 Jan 16 08:09 file.txt
```

- Наделить группу и других пользователей правами чтения и исполнения, но не записи

```
$ chmod go=rx file.txt
-rw-r-xr-x 1 user user 42 Jan 16 08:09 file.txt
```

- Наделить пользователя-владельца правами исполнения и лишить группу права записи:

```
$ chmod 744 file.txt
-rwxr--r-- 1 user user 42 Jan 16 08:09 file.txt
```

- Наделить всех пользователей всеми правами:

```
$ chmod 777 file.txt
-rwxrwxrwx 1 user user 42 Jan 16 08:09 file.txt
```

- Лишить других пользователей права чтения:

```
$ chmod 660 file.txt
-rw-rw-r-- 1 user user 42 Jan 16 08:09 file.txt
$ chmod 744 file.txt
-rw-rw-r-- 1 user user 42 Jan 16 08:09 file.txt
```

Таким образом мы можем настраивать доступ к файлам и каталогам внутри нашей системы.

Скачивание файлов из сети посредством wget

Для получения файлов и данных из внешних источников, к которым есть путь по сети, можно воспользоваться командой **wget**. Данная команда позволяет скачивать файлы на устройство не только из сети интернет, но и из совместной локальной сети. Для этого к данным файлам пользователь системы должен иметь доступ.

В качестве примера попробуем получить доступ к файлу на сайте нашего университета, а именно к гербу РТУ МИРЭА в цвете. Данный файл находится по следующей ссылке на сайте ВУЗа: https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar.

Пример использования команды **wget**:

- Загрузка одного файла

```
$ wget https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Загрузка файла и сохранение его с новым именем

```
$ wget -O mirea_logo_rgb.rar
https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Ограничение скорости загрузки файлов

```
$ wget –limit-rate=250K
https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Сохранить файл в папке

По умолчанию wget сохраняет файл в текущую папку, но это поведение очень легко изменить с помощью опции -P:

```
$ wget -P path_to_forder
https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Завершение прерванной загрузки

```
$ wget –c https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Фоновая загрузка файла

```
$ wget –b https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

- Загрузка нескольких файлов

```
$ wget -I url_list.txt
```

- Увеличение общего числа попыток загрузки файла

```
$ wget –-tries=100
https://www.mirea.ru/upload/medialibrary/1ed/TSvetnoy.rar
```

Архивация файлов в Linux

В Linux присутствует несколько утилит, предназначенных для работы с пользовательскими форматами архивов. Необходимость в них возникает в случае работы с файлами и файловыми системами, поскольку все каталоги по сети интернет объективно удобнее и быстрее передавать в формате сжатых каталогов (архивов).

Архив, с точки зрения пользователя, представляет собой директорию с файлами, преобразованную в один целостный файл и подверженный алгоритму сжатия данных для уменьшения его совокупного размера в байтах.

Существует несколько популярных пользовательских форматов архивов, с которыми пользователь мог часто сталкиваться в процессе работы за ПК, среди них: .rar, .zip, .7z, а также для пользователей macOS и Linux распространенным является формат архива .tar.gz.

Утилита tar для работы с архивами. Самой популярной для Linux утилитой для архивации есть **tar**. Она используется почти везде, для архивации исходников, упаковки пакетов. Для сжатия используются и другие утилиты, в зависимости от алгоритма сжатия, например, **zip**, **bz**, **xz**, **lzma** и т д. Сначала выполняется архивация, затем сжатие, отдельными программами. Автоматический запуск некоторых утилит сжатия для только что созданного архива поддерживается в **tar** и других подобных программах с помощью специальных опций.

Рассмотрим утилиту **tar** для работы с архивами:

```
$ tar [опции] <файл_для_записи> </папка_с_файлами_для_архива>
```

А теперь разберем основные опции:

- A — добавить файл к архиву
- c — создать архив в linux
- d — сравнить файлы архива и распакованные файлы в файловой системе
- j — сжать архив с помощью Bzip
- z — сжать архив с помощью Gzip
- r — добавить файлы в конец архива
- t — показать содержимое архива
- u — обновить архив относительно файловой системы
- x — извлечь файлы из архива
- v — показать подробную информацию о процессе работы
- f — файл для записи архива
- -C — распаковать в указанную папку

- --strip-components — отбросить n вложенных папок

Чтобы создать архив используйте такую команду:

```
$ tar -cvf archive.tar.gz /path/to/files
```

А чтобы распаковать архив tar linux:

```
$ tar -xvf archive.tar.gz
```

Чтобы заархивировать и сжать папку с помощью gzip воспользуемся командой:

```
$ tar -zcvf home.tar.gz path_to_folder
```

Чтобы добавить файл в архив используйте:

```
$ tar -rvf archive.tar file.txt
```

Для извлечения одного файла синтаксис тот же:

```
$ tar -xvf archive.tar file.txt
```

Утилита zip для работы с архивами. Кроссплатформенная утилита для создания сжатых архивов формата zip. Совместимая с Windows реализациями этого алгоритма. Zip архивы очень часто используются для обмена файлами в интернете. С помощью этой утилиты можно сжимать как файлы, так и сжать папку linux.

Синтаксис утилиты:

```
$ zip [опции] <файл_для_записи>
$ unzip [опции] <архив>
```

Команда **zip** работает для архивации файлов и каталогов, **unzip** наоборот для разархивирования.

Опции утилиты:

- -d — удалить файл из архива

- -r — рекурсивно обходить каталоги
- -0 — только архивировать, без сжатия
- -9 — наилучший степень сжатия
- -F — исправить zip файл
- -e — шифровать файлы

Чтобы создать Zip архив в Linux используйте:

```
$ zip -r /path/to/files/*
```

А для распаковки:

```
$ unzip archive.zip
```

Данная утилита работает намного проще чем **tar** ввиду того, что поддерживает мало опций и достаточно легковесна в плане разбиения обязанностей на две команды.

Утилита rar для работы с архивами. По умолчанию не в каждом дистрибутиве Linux присутствует утилита **rar**. В образе операционной системы, которая поставляется с данным практическим заданием, она имеется.

Распаковка .rar linux выполняется с помощью утилиты **unrar**. А упаковка архива командой **rar**. Обычно она не поставляется в системе по умолчанию, но вы можете очень просто установить ее из официальных репозиториев. Для установки утилиты в Ubuntu выполните такую команду:

```
$ sudo apt install unrar
```

Синтаксис команды rar:

```
$ unrar <команда> [опции] <архив|файлы> [путь_для_распаковки]
```

Команды unrar:

- e — распаковать архив;
- l — вывести список файлов внутри архива;

- p — вывести распакованный файл в стандартный вывод;
- t — проверить архив;
- v — вывести подробную информацию про архив;
- x — извлечь файлы, сохраняя полный путь внутри архива.

Основные опции unrar:

- -ad — добавить имя архива к пути распаковки;
- -ai — игнорировать атрибуты файлов;
- -inul — не выводить сообщения об ошибках;
- -p — указать пароль архива;
- -sl — распаковывать только файлы, меньше указанного размера;
- -u — обновить уже распакованные файлы;
- -y — отвечать y на все вопросы;
- -x — не распаковывать указанные файлы;
- -ts — сохранять временную метку исходных файлов.

Примеры:

Чтобы извлечь файлы из архива в текущую папку достаточно передать утилите имя архива и команду e:

```
$ unrar e file.rar
```

Вы можете вывести содержимое архива, ничего не распаковывая с помощью команды l:

```
$ unrar l file.rar
```

Чтобы сохранять полный путь, который прописан в архиве нужно использовать команду x:

```
$ unrar x file.rar
```

Вы можете указать папку, в которую стоит распаковывать файлы:

```
$ unrar e file.rar path_to_folder
```

Кодировка файлов

Говоря простыми словами, кодировка символов – это способ информирования компьютера о том, как интерпретировать исходные нули и единицы в реальные символы, где символ представлен набором чисел. Когда мы печатаем текст в файле, слова и предложения, которые мы формируем, готовятся из разных символов, а символы упорядочиваются в кодировку.

Распознавание кодировки файла. Стандартным способом узнать кодировку файла в Linux с помощью командной строки можно применив утилиту **file**. Применив к файлу “my_file_to_decode” следующую команду:

```
$ file -i my_file_to_decode
```

мы можем узнать его кодировку с целью дальнейшей работы с ним.

Изменение кодировки файла. В любом случае, нам необходимо уметь преобразовывать кодировки файлов из одной в другую с целью подготовки данных для прочтения другими программами. Для смены кодировки файла необходимо воспользоваться стандартной утилитой **iconv**. Рассмотрим принцип работы утилиты.

Для того чтобы узнать полный список поддерживаемых кодировок введите команду:

```
$ iconv -l
```

Можно получить подтверждение наличия именно нужной кодировки с помощью использования последовательности команд и использования **grep**:

```
$ iconv -l | grep encoding
```

Например, чтобы получить список всех кодировок, поддерживающих Windows, можно ввести команду, соответствующую желаемой подстановке символов:

```
$ iconv -l | grep WINDOWS*
```

Чтобы перевести файл из выбранной кодировки в другую, поддерживаемую **iconv**, воспользуемся командой:

```
$ iconv [опции] -f из-кодировки -t в-кодировку файл(ы) ввода -o файлы вывода
```

Пример для файла «my_file.txt» в кодировке CP866:

```
$ iconv -f CP866 -t UTF-8//TRANSLIT my_file.txt -o new_file.txt
```

Добавление постфикса //TRANSLIT позволяет переводить неопознанные символы в транслитерацию. Новый файл «new_file.txt» находится уже в кодировке utf-8.

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

1. Задание на выполнение

1. Развернуть операционную систему Ubuntu Linux Server 18.04 с помощью средства для виртуализации ОС на персональном компьютере. Разрешено как использовать готовый образ ОС по ссылке, так и вручную настроить серверную операционную систему с нуля. Войти в ОС Linux под созданной учетной записью пользователя (в случае с готовым образом системы пользователь/логин: **user**, пароль: **user**). (2 балла)
2. Получить справку по команде **top**. С помощью команды **top** просмотреть занимаемое операционной системой место в оперативной памяти. Выйдите из выполнения команды **top** в терминале. (2 балла)
3. Просмотреть путь к текущей директории. Создать в пользовательской директории папку для самостоятельной работы **цифры_шифра/my_test_folder**. Внутри нее создать директории в связи со следующей структурой:

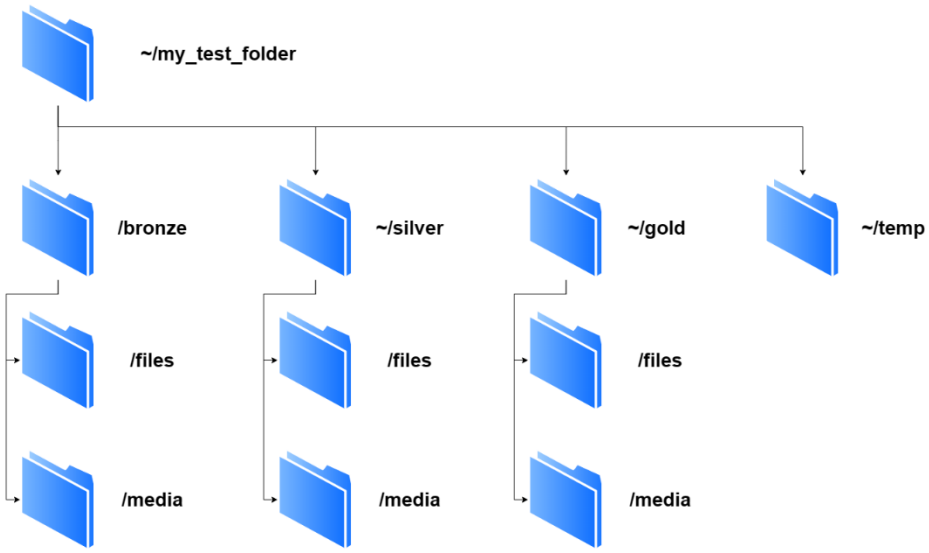

Рисунок 15. Структура директорий для самостоятельной работы

Вывести на экран всю созданную древовидную структуру в виде списка папок и подпапок в терминал командной строки.

Перевести вывод созданной структуры в файл **~/my_test_folder/temp/dirs.txt**. (2 балла)

4. Перейти в папку /my_test_folder/temp. Создайте в папке пустой файл test.txt. Ввести в файл информацию «Hello, its my first file in Linux!». Вывести сообщение из файла в консоль. Просмотрите размер созданного файла. (2 балла)
5. С помощью команды **wget** загрузить в папку **temp** файл архива https://github.com/qwerty29544/BigDataEssentials/raw/main/Practice1_LinuxCommands/data.tar.gz с данными для выполнения дальнейших пунктов практической работы. Разархивировать файлы в папку **temp**, не создавая новых каталогов. Просмотрите список файлов и их уровней доступа. (2 балла)
6. Перенесите все файлы изображений (расширения .eps, .png, .jpg) в папку **/bronze/media**. Выведите список файлов в данной директории продемонстрируйте количество занимаемого места на диске данными файлами, а также список прав на доступ к файлу. Переименуйте изображение герба РТУ МИРЭА в MIREA_gerb_rgb.* (2 балла)
7. Выведите в терминал первые строки из файлов изображений. Покажите какой формат изображений выводит информацию в структурированном виде. Чем визуально различаются префиксы во всех форматах файлов изображений. (2 балла)
8. Перенести файлы форматов .txt, .TXT, .csv, .db в папку /bronze/files. Для файла **TNVED1.TXT** выведите первые его строки. Проверьте кодировку файла **TNVED1.TXT** с помощью команды **file**. Создайте новый файл с именем **tnved1_utf.txt** переведя кодировку файла **TNVED1.TXT** из кодировки CP866 в кодировку UTF-8. Выведите ещё раз содержимое файла в терминал. Проделайте то же самое с

файлами TNVED2 и TNVED3. Перенесите файлы в читаемой новой кодировке в папку /silver/files. (4 балла)

9. В полученных файлах **tnved*_utf.txt** с помощью редактора **nano** заменить прямые разделители «|» на двойные «||» во всем тексте. Выведите содержимое на экран терминала. Переместите данные файлы далее в каталог **/gold/files**. Вывести итоговый результат работы в виде дерева каталога /**my_test_folder** (2 балла)

2. Вопросы для проверки знаний

1. Как получить справку по команде ssh?
2. Как посмотреть текущую рабочую директорию?
3. Как сменить рабочую директорию?
4. Как перейти в родительскую/корневую директорию?
5. Как вернуться в домашнюю директорию?
6. Как вывести список файлов в директории?
7. Как посмотреть время последнего изменения/доступа к файлу /tmp/test.txt?
8. Как создать новую директорию test?
9. Как создать пустой файл?
10. Как узнать тип файла?
11. Как посмотреть размер файла?
12. Как переименовать файл?
13. Как удалить файл/директорию?
14. Как как узнать размер директории?
15. Как посчитать количество строк в текстовом файле?
16. Как вывести на экран отсортированные строки текстового файла?
17. Как удалить дубли строк из файла?
18. Как дописать содержимое одного текстового файла в конец второго?
19. Как вывести на экран первые 30 строк файла?

20. Как вывести на экран последние 30 строк файла?
21. Как разбить текстовый файл на несколько по 100 строк в каждом?
22. Как посмотреть содержимое текстового файла?
23. Как загрузить файлы или несколько файлов из сети интернет?
24. Как можно произвести архивацию файлов директории в единый архив?
25. Как можно произвести разархивацию файлов из архива?