

***Методические указания по дисциплине
«Администрирование информационных систем»
Лабораторная работа № 1***

Система виртуализации. Система управления контентом.

Задание выполняется группой 4 человек.

Задание:

1. Установить систему виртуализации Virtualbox.
2. Установить ОС Linux Debian.
3. Осуществить базовую настройку сетевых интерфейсов виртуальной машины.
4. Установить и настроить пакеты для виртуальной машины.
5. Развернуть одностраничный сайт (можно использовать CMS WordPress).

Следующие данные должны быть отражены: концептуальное описание технического объекта или информационной системы. Здесь также должны быть представлены следующие положения ТЗ: цели создания системы и основные функциональные возможности системы; информация об участниках рабочей группы (роль в группе, контакты и т. д.).

ВАЖНО! Если есть желание, то вместо создания сайта проекта на *CMS WordPress*, можно создать собственный сайт, используя *html*, *css*, *js* и разместить его в рабочей папке веб-сервера. В таком случае выполнять п.6 методических рекомендаций не нужно.

Разработанный сайт должен соответствовать следующим требованиям:

- Структура *html*-документа должна соответствовать стандарту *html5*.
- Не должно быть *inline*-стилей *CSS*. Все *CSS*-стили должны быть оформлены в отдельных **.css* файлах и помещены в отдельную папку веб-сайта.
- Не должно быть *inline*-скриптов *JavaScript*. *JavaScript* код должен быть оформлен в отдельных **.js* файлах и помещен в отдельную папку веб-сайта.

Рекомендуемый ход выполнения работы:

1. Теоретическая часть:

1.1 Составить краткий конспект на тему: понятие CMS, основные функции CMS, предоставляемые возможности CMS, CMS и CMF, основные базовые понятия и термины, использующиеся в CMS для работы и функционирования. Привести ссылки на источники информации.

1.2 Обзор и анализ методов решения: коробочные коммерческие CMS, open-source CMS, студийные CMS. Выбрать наиболее популярные 3-5 CMS, оформить таблицу, где будут указаны достоинства и недостатки, тип лицензии, номер и дата выхода (релиза) текущей версии, поддерживаемая база данных и операционная система, веб сервер, языки программирования, использованные для разработки. Привести ссылки на источники информации.

1.3 Провести анализ (в табличной форме) наиболее популярных web-серверов (не менее четырех) по следующим критериям: поддерживаемая операционная система, тип лицензии, поддерживаемые языки программирования, производительность, основные особенности конфигурирования.

2. Практическая часть:

2.1 Настройка виртуальной машины и операционной системы.

2.2 Описание основных разделов сайта.

2.2 Установка CMS WordPress, основные этапы работы: установка и настройка программной оболочки для работы с локальным сервером, скачивание CMS, установка CMS на сервер: конфигурация сайта, конфигурация базы данных.

2.3 Конфигурация CMS: настройка оформления сайта (выбрать визуальную тему/шаблон, добавить необходимые плагины), настройка структуры сайта (согласно расположению виджетов), создание пользователей и обеспечение их доступа (посетители, редакторы, администраторы), к информации, добавление содержимого на сайт (в соответствии с заданием).

Результатом выполнения задания являются:

- Отчет, содержащий следующую информацию:

1) Пошаговое описание практической части.

2) Отчет по лабораторной работе, содержащий ход выполнения работы с описанием и скриншотами выполнения, результаты выполнения лабораторной работы.

Для успешной защиты лабораторной работы студенты должны предоставить проект и отчет к нему.

Требования к оформлению отчета:

Способ выполнения текста должен быть единым для всей работы. **Шрифт** – **Times New Roman**, кегль 14, **межстрочный интервал** – 1,5, **размеры полей**: левое – 30 мм; правое – 10 мм, верхнее – 20 мм; нижнее – 20 мм. Сокращения слов в тексте допускаются только общепринятые.

Абзацный отступ (1,25) должен быть одинаковым во всей работе. **Нумерация страниц** основного текста должна быть сквозной. Номер страницы на титульном листе не указывается, задание на производственную практику является второй страницей. Сам номер располагается внизу по центру страницы или справа.

Методические рекомендации к выполнению лабораторной работы

1. Необходимое программное обеспечение

1) Система виртуализации Virtualbox:

<https://download.virtualbox.org/virtualbox/6.1.36/VirtualBox-6.1.36-152435-Win.exe>

2) Выбираем установочный пакет в соответствии с ОС рабочей машины и устанавливаем (во время установки сетевые интерфейсы будут перезапущены!).

3) ISO-образ ОС Linux Debian для установки на виртуальную машину (ВМ):

<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/>

В нижней части страницы будет ссылка на скачивание актуальной версии iso-образа в формате debian-xx.x.x- amd64-netinst.iso

4) Клиент SSH - Putty (для Windows):

<https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>

5) Клиент для передачи файлов по протоколу SSH - WinSCP (для Windows):
<https://winscp.net/download/WinSCP-5.21.1-Setup.exe>

2. Создание новой ВМ.

Запускаем Virtualbox.

Создаем новую гостевую виртуальную машину (далее просто ВМ):

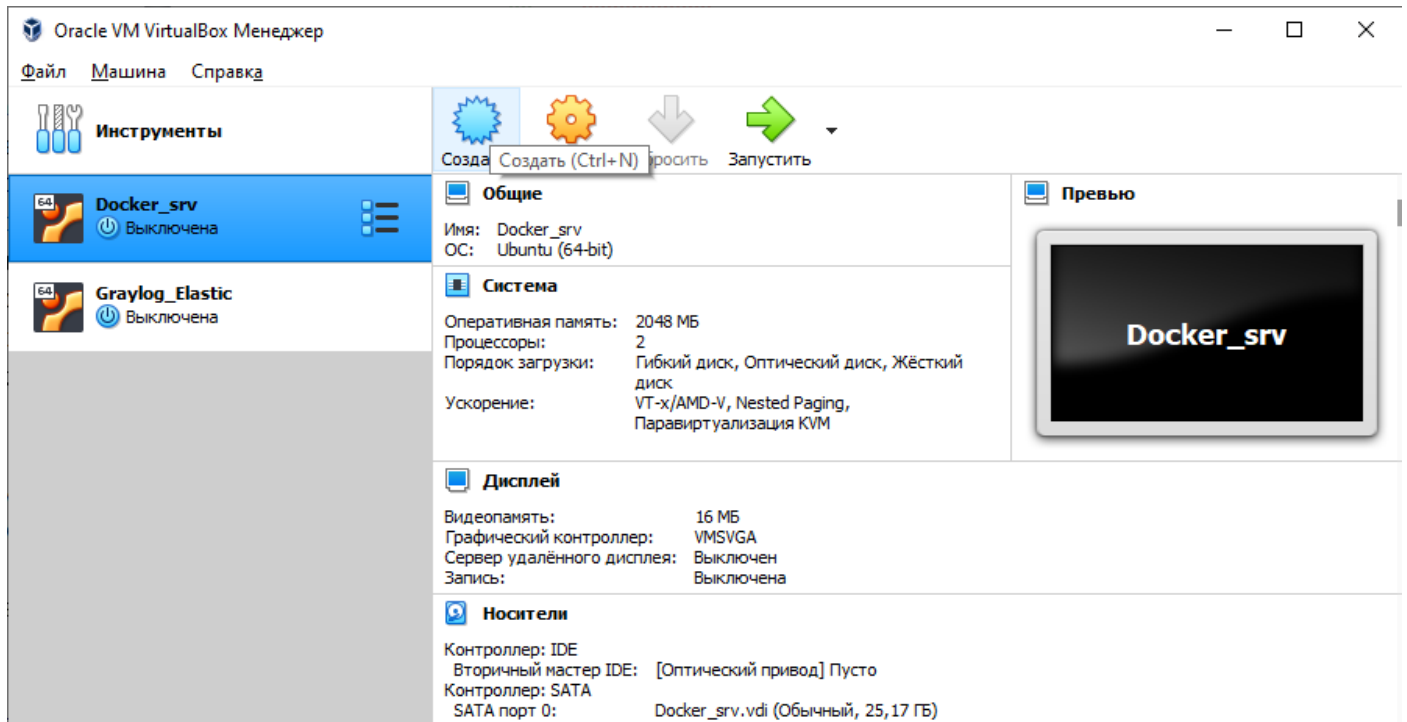


Рис. 2.1 – Главное окно Virtualbox

Выбираем тип ОС и задаем имя для новой ВМ (рис. 2.2):

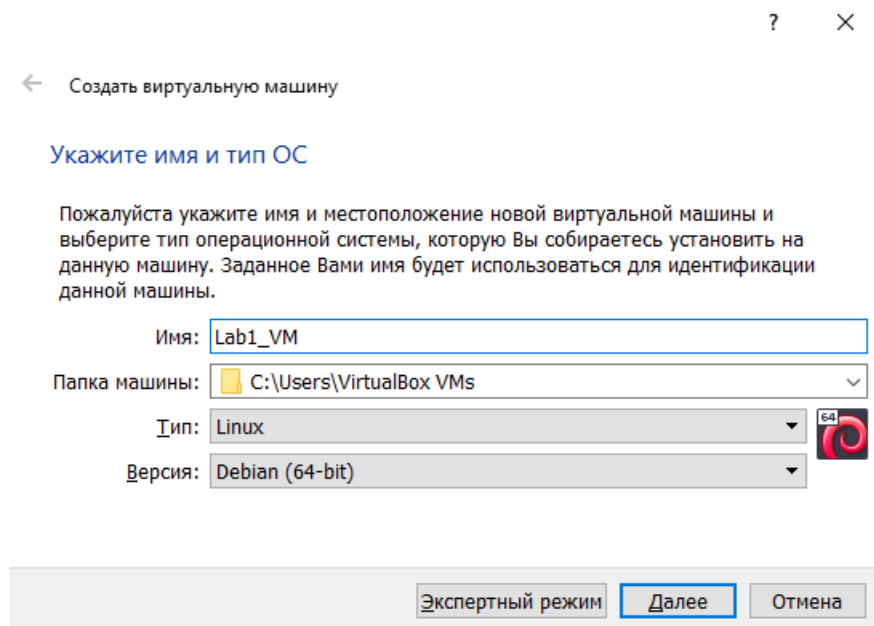


Рис. 2.2 – Параметры создания ВМ

Рекомендованный объем памяти: 1024Мб.

Создать новый виртуальный жёсткий диск (VDI, динамический, объемом 10Гб).

Запускаем созданную VM с помощью панели управления главного окна:

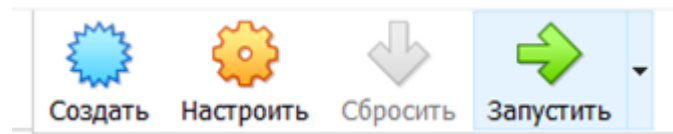


Рис. 2.3 – Запуск VM

Выбираем ISO-образ (*debian-11.4.0-amd64-netinst.iso*, полученный ранее в п. 1) в качестве загрузочного диска при старте:

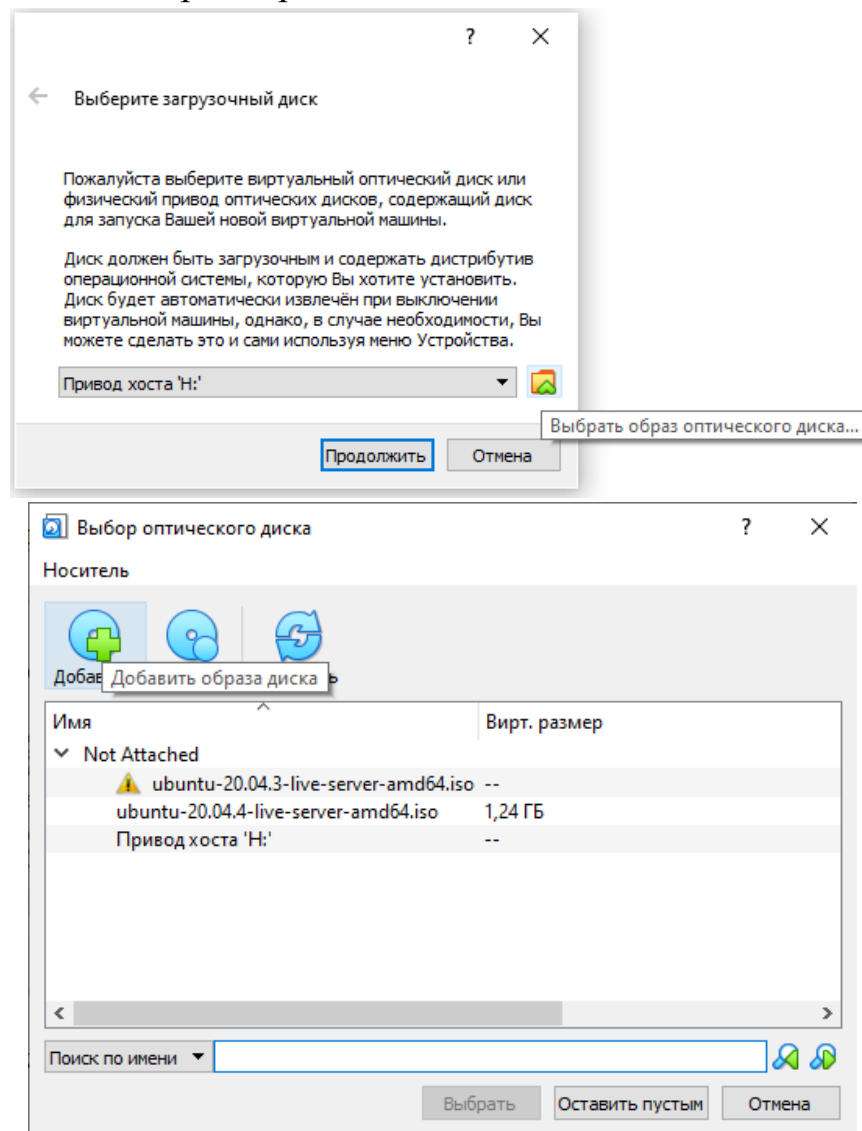


Рис. 2.4 – Выбор загрузочного образа

3. Установка ОС

После загрузки образа в появившемся установочном меню выбираем **Install** или **Graphical install**.

Параметры установки:

Имя компьютера: **aislab<N>** , где N – любое двузначное число.

Имя домена (локальное доменное имя системы): пропускаем или вводим, например: **aislab<N>.usatu.local**

Пароль суперпользователя (**root**): **Qwerty@**

Имя пользователя/учетной записи: **ais**

Пароль: *придумать свой (не более 8 символов)*

Разметка диска: автоматически (все файлы в одном разделе)

В результате разметки будут созданы 3 раздела: основной раздел ~90% диска, раздел подкачки ~10% диска и раздел для загрузчика ОС.

Записать изменения на диск: **Да**

...Установка базовой системы...

Сканирование других дисков: **Нет**

Менеджер пакетов: РФ (*deb.debian.org*)

Информация о прокси: **Нет**

Конфигурация дополнительных пакетов (рис. 3.1):

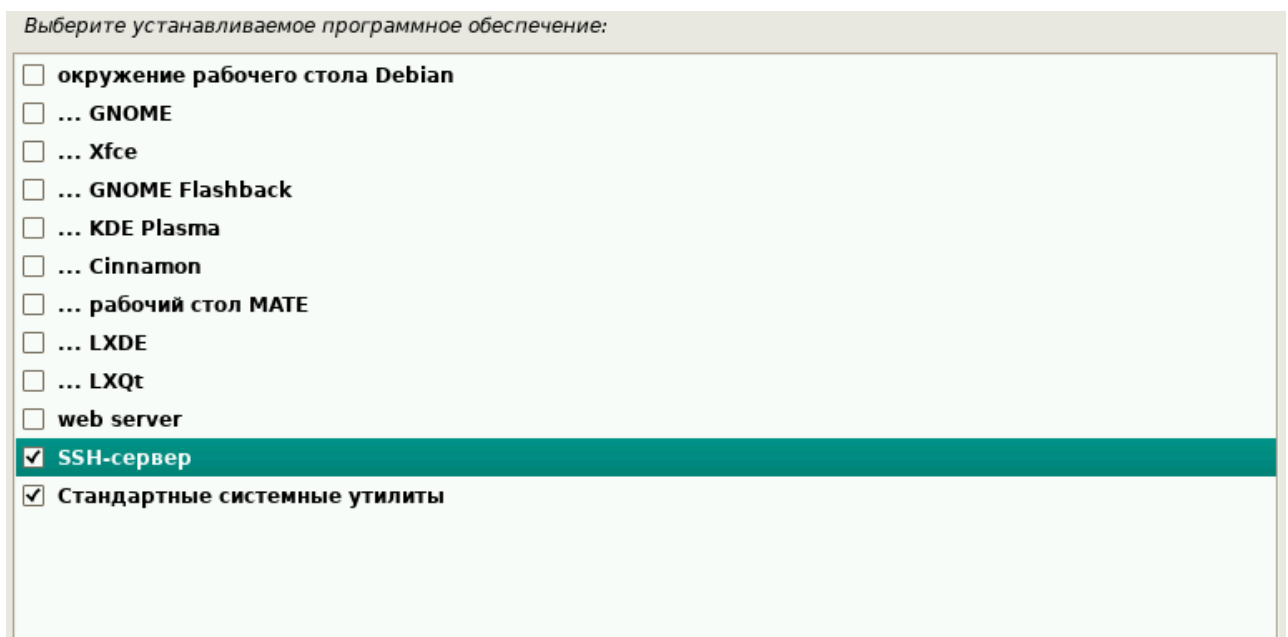


Рис. 3.1 – Выбор дополнительных программных пакетов

Установить загрузчик GRUB в главную загрузочную запись? **Да**

Устройство для установки системного загрузчика: **/dev/sda**

Машина будет перезагружена...

4. Базовая настройка сетевых интерфейсов VM

После базовой установки сетевой интерфейс VM будет по умолчанию работать в режиме NAT (преобразование адреса), когда Virtualbox перехватывает сетевой трафик

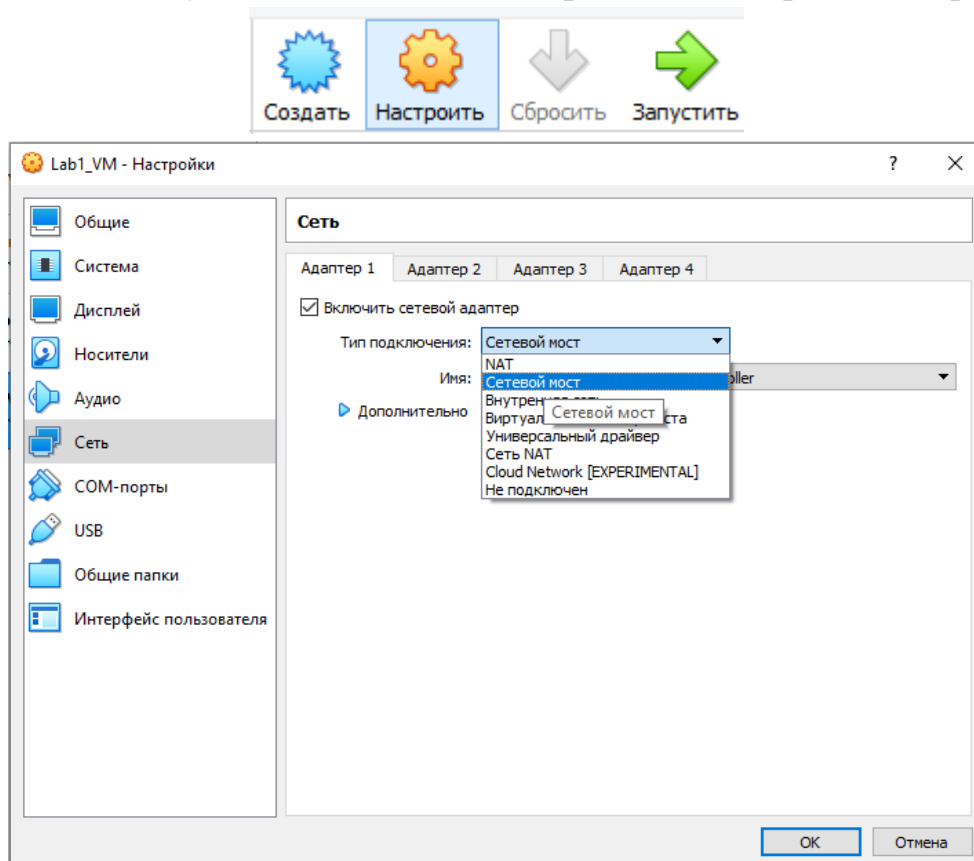
на физическом интерфейсе и для необходимых пакетов преобразует внутренний IP-адрес ВМ (обратный адрес «получателя») на внешний (доступный).

Необходимо настроить сетевые интерфейсы ВМ таким образом, чтобы предоставить доступ к сетевым ресурсам виртуальной системы как для компьютера, на котором запущена сама ВМ, так и для компьютеров, находящихся в той же локальной сети.

Для этого останавливаем ВМ, заходим в настройки, выбираем «Сеть» и во вкладке «Адаптер 1» меняем режим на «Сетевой мост», далее включаем «Адаптер 2» и устанавливаем режим «Виртуальный адаптер хоста» (рис. 4.1).

«Адаптер 1» в режиме «Сетевой мост» создаст дополнительный виртуальный интерфейс, использующий физическое подключение сетевой карты ПК. Таким образом, ВМ будет подключена к той же локальной сети (через кабель или точку доступа) и будут автоматически получены сетевые настройки из того же диапазона IP-адресов (при наличии активного DHCP-сервера).

«Адаптер 2» в режиме «Виртуальный адаптер хоста» создаст между ВМ и ОС физической машины внутреннюю виртуальную сеть (по умолчанию из пула адресов 192.168.56.0/24), что позволит осуществлять локальное сетевое взаимодействие между физической машиной и запущенной на ней ВМ в процессе тестирования и разработки.



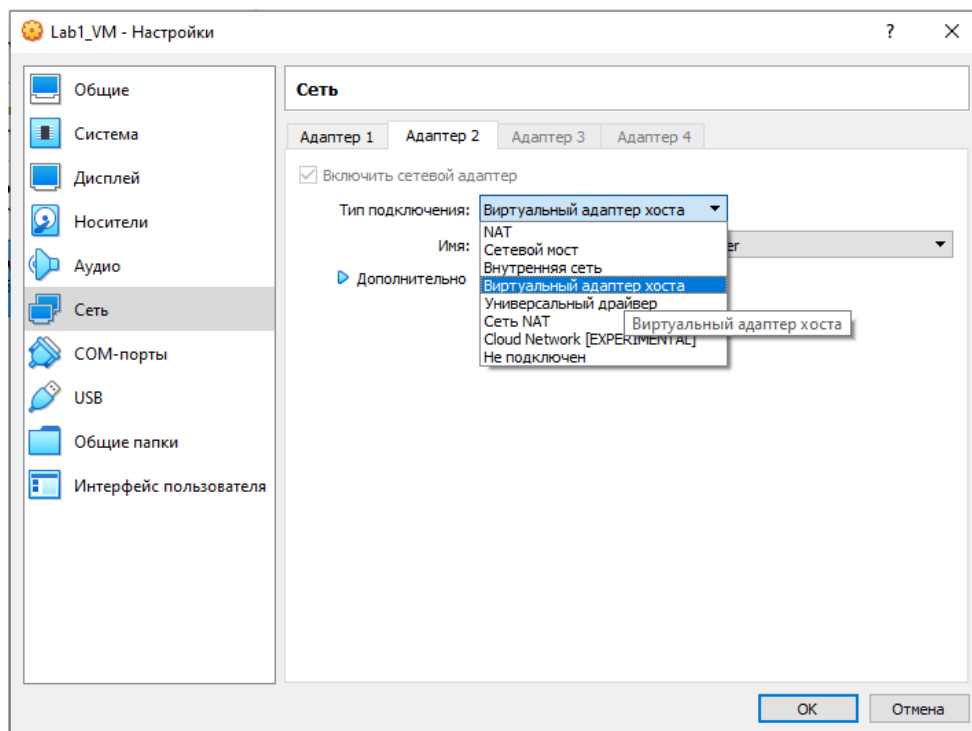


Рис. 4.1 – Настройка сетевых интерфейсов VM

4.1 Проверка корректности настройки сети. Для проверки корректности настройки сети включаем VM и заходим в систему, используя login: **root** и пароль суперпользователя, установленный в п.3.

Для получения информации о сетевых настройках в ОС Linux используется утилита **ip**, набираем консольную команду для получения информации обо всех сетевых интерфейсах:

ip a

Вывод команды будет примерно следующего содержания:

```
root@aislalab22:~# 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 pfifo_fast state UP group default qlen 1000
   link/ether 08:00:27:08:7f:6c brd ff:ff:ff:ff:ff:ff
   inet 192.168.4.101/24 brd 192.168.4.255 scope global dynamic enp0s3
       valid_lft 5682sec preferred_lft 5682sec
   inet6 fe80::a00:27ff:fe08:7f6c/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 08:00:27:19:6e:75 brd ff:ff:ff:ff:ff:ff
```

Строка:

`inet 192.168.4.101/24 brd 192.168.4.255 scope global dynamic enp0s3`

Т.е. основной сетевой интерфейс VM с именем **enp0s3** автоматически получил IP-адрес **192.168.4.101** (ip-адрес из той же сети, к которой подключен ПК).

enp0s8: ... state DOWN ... – означает, что Адаптер 2 обнаружен, но отключен.

Включаем второй интерфейс командой:


```
ip link set dev enp0s8 up
```

При повторном вводе команды `ip a`, состояние интерфейса **enp0s8** должно отобразиться как **state UP**.

4.2 Настройка статичного адреса на виртуальном сетевом интерфейсе.

Переходим в папку **/etc/network** с помощью команды:

```
cd /etc/network
```

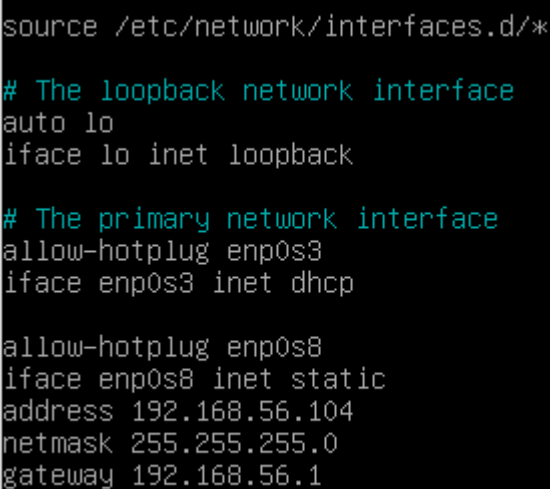
Редактируем файл **interfaces**, чтобы задать статичный адрес для интерфейса **enp0s8**. Используем утилиту **nano** в качестве консольного редактора,

```
nano interfaces
```

Добавляем следующие строки:

```
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.56.104
    netmask 255.255.255.0
    gateway 192.168.56.1
```

Базовые команды для работы с ОС Linux через терминал можно посмотреть здесь: <https://pingvinus.ru/note/10-commands-for-beginners>



```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.56.104
netmask 255.255.255.0
gateway 192.168.56.1
```

Рис. 4.2 – Файл конфигурации сетевых интерфейсов ВМ

Значение `address` можно установить любое в рамках подсети 192.168.56.x

Сохраняем (Ctrl+X) и перезапускаем ВМ командой **reboot**

Аналогичным образом (если необходимо), можно настроить статичный IP-адрес для Адаптера 1 (интерфейс **enp0s3**), но для сохранения доступа к интернет-ресурсам

также необходимо установить адрес DNS-сервера по умолчанию. Это можно сделать, отредактировав файл **/etc/resolv.conf** (например, публичный DNS-сервер Yandex: **nameserver 77.88.8.1**)

Если повторно вывести информацию о сетевых интерфейсах VM, то в выводе можно увидеть:

```
link/ether 08:00:27:75:90:d3 brd ff:ff:ff:ff:ff:ff
inet 192.168.56.104/24 brd 192.168.56.255 scope global dynamic enp0s8
    valid_lft 372sec preferred_lft 372sec
inet6 fe80::a00:27ff:fe75:90d3/64 scope link
    valid_lft forever preferred_lft forever
root@aislab1:~# _
```

Т.е. виртуальный сетевой интерфейс VM с именем **enp0s8** получил IP-адрес **192.168.56.104**

4.3 Удаленное подключение к VM. Т.к. при установке ОС был также предустановлен SSH-сервер (рис. 3.1), то можно подключиться к системе по IP-адресу, используя утилиту Putty через защищенный протокол SSH (рис. 4.3, 4.4):

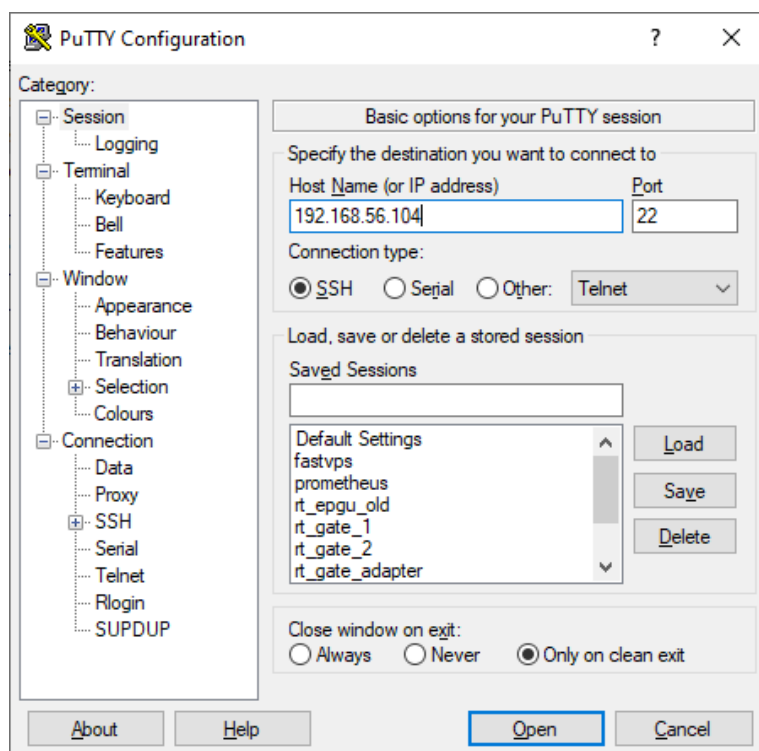


Рис. 4.3 – Настройка сессии подключения по SSH

Перед подключением убедитесь, что клиент использует кодировку UTF-8:

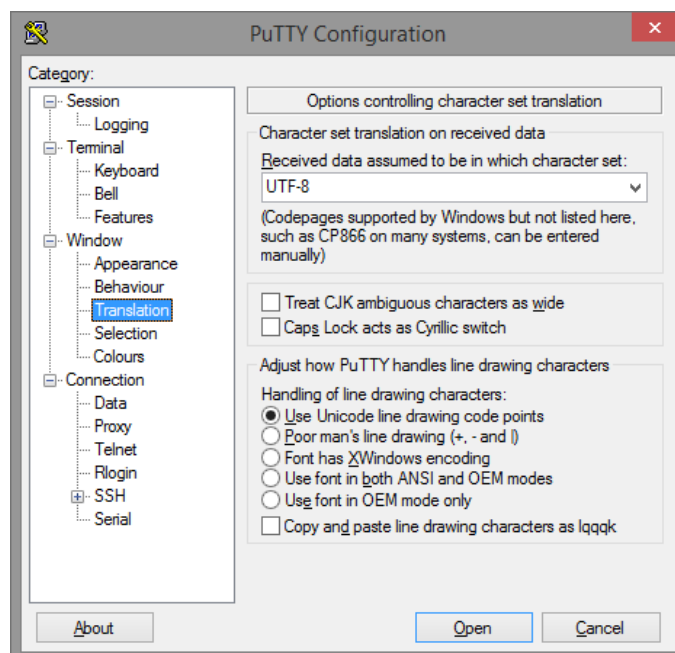


Рис. 4.4 – Настройка параметров кодирования для сессии SSH

Логинимся под обычным (ранее созданным в п. 3) пользователем **ais** и переходим в режим администратора командой: **su** и вводим пароль суперпользователя, установленный в п.3.

4.4 Настройка статической сетевой маршрутизации. Т.к. в системе настроено 2 сетевых интерфейса, необходимо проверить как распределяется (маршрутизируется) сетевой трафик между данными интерфейсами. Данная настройка ОС хранит в таблице маршрутизации, чтобы вывести активные правила маршрутизации вводим команду:

ip route

Примерный вывод команды:

```
default via 192.168.56.1 dev enp0s8 onlink
192.168.4.0/24 dev enp0s3 proto kernel scope link src 192.168.4.101
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.104
```

Вторая строка вывода интерпретируется следующим образом: для взаимодействия с адресами подсети 192.168.4.0/24 система использует интерфейс enp0s3 с адресом 192.168.4.101. Аналогично интерпретируется третья строка, но для подсети 192.168.56.0/24 и интерфейса enp0s8.

Ключевое правило маршрутизации здесь определяется в первой строке «default via ...», что означает, что по умолчанию взаимодействие с любыми неизвестными IP-адресами (т.е. не входящие в сети 192.168.4.0/24 и 192.168.56.0/24) система будет осуществлять через интерфейс enp0s8. Данный интерфейс (Адаптер 2) отвечает за подсеть 192.168.56.0/24, а эта подсеть, как мы знаем, является виртуальной локальной сетью между ПК и ВМ и, соответственно, не имеет доступа к интернет-ресурсам. В этом можно убедиться, запустив утилиту ping на любой публичный адрес, например:

ping yandex.ru

Выполнение команды «зависнет» или будет выводиться сообщение «No route to Host». Для отмены выполнения нажмите Ctrl+C.

Таким образом, маршрут по умолчанию необходимо переключить на интерфейс enp0s3, который через сетевую карту ПК подключен к внешней сети. Для выполнения этой настройки сначала удаляем текущий маршрут по умолчанию:

```
ip route del default
```

И устанавливаем новый маршрут через enp0s3:

```
ip route add default via 192.168.4.1 dev enp0s3
```

где адрес 192.168.4.1 – это IP-адрес вашего интернет шлюза по умолчанию (например, вашего Wi-Fi роутера). Посмотреть текущий адрес вашего шлюза можно в информации о подключении на вашем ПК, например, в Win10: «Параметры сети и интернет» → Просмотр свойств оборудования и подключения:

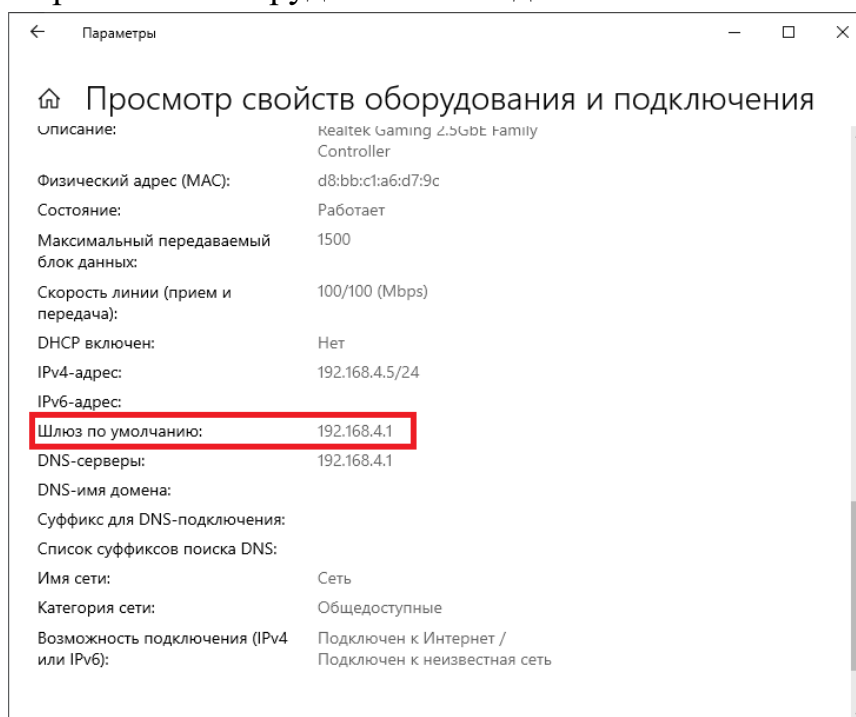


Рис. 4.5 – Вывод информации о сетевом подключении в Win10

После установки нового маршрута повторно проверьте доступность интернет-ресурсов утилитой ping.

ВАЖНО! Настройки статической маршрутизации, установленные с помощью консольных команд, будут сброшены после перезагрузки ВМ. Для сохранения правил маршрутизации необходимо настроить автоматическое выполнение маршрутных команд после загрузки сетевых интерфейсов при запуске системы. Для этого добавляем инструкцию **post-up** в файле конфигурации сетевых интерфейсов (/etc/network/interfaces):

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp
    post-up ip route add default via 192.168.4.1 dev enp0s3

allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.56.104
    netmask 255.255.255.0
    gateway 192.168.56.1
    post-up ip route del default dev enp0s8
```

Рис. 4.6 – Настройка автоматического выполнения (post-загрузка) команд маршрутизации в файле конфигурации сетевых интерфейсов

5. Установка и настройка пакетов для ВМ

Установка пакетов из репозитория осуществляется с помощью **менеджера пакетов apt**. На начальном этапе достаточно двух команд:

apt install <список пакетов> – установка

apt purge <список пакетов> – удаление

В старых версиях ОС Linux типа Debian (выпущенных до 2014) для работы с пакетами используется команда `apt-get` вместо `apt`.

Для установки пакетов переходим в режим администратора командой: **su**

Перед началом установки, необходимо проверить доступность репозитория (выбранного во время установки ОС – *deb.debian.org*) для менеджера пакетов (**apt**), используем команду:

`apt update`

Если чтение репозитория прошло успешно, то ВМ подключена к интернету и можно обновить установленные пакеты:

`apt upgrade`

5.1 Файловый менеджер. Устанавливаем файловый менеджер (ФМ) **Midnight Commander (MC)** и пакет **sudo** для выполнения администраторских команд:

`apt install mc sudo`

Запускаем ФМ: **mc**

При запуске ФМ откроется в личной папке текущего пользователя (/home/ais). Для просмотра всех директорий в основном разделе, переходим на 2 уровня выше (рис. 5.1):

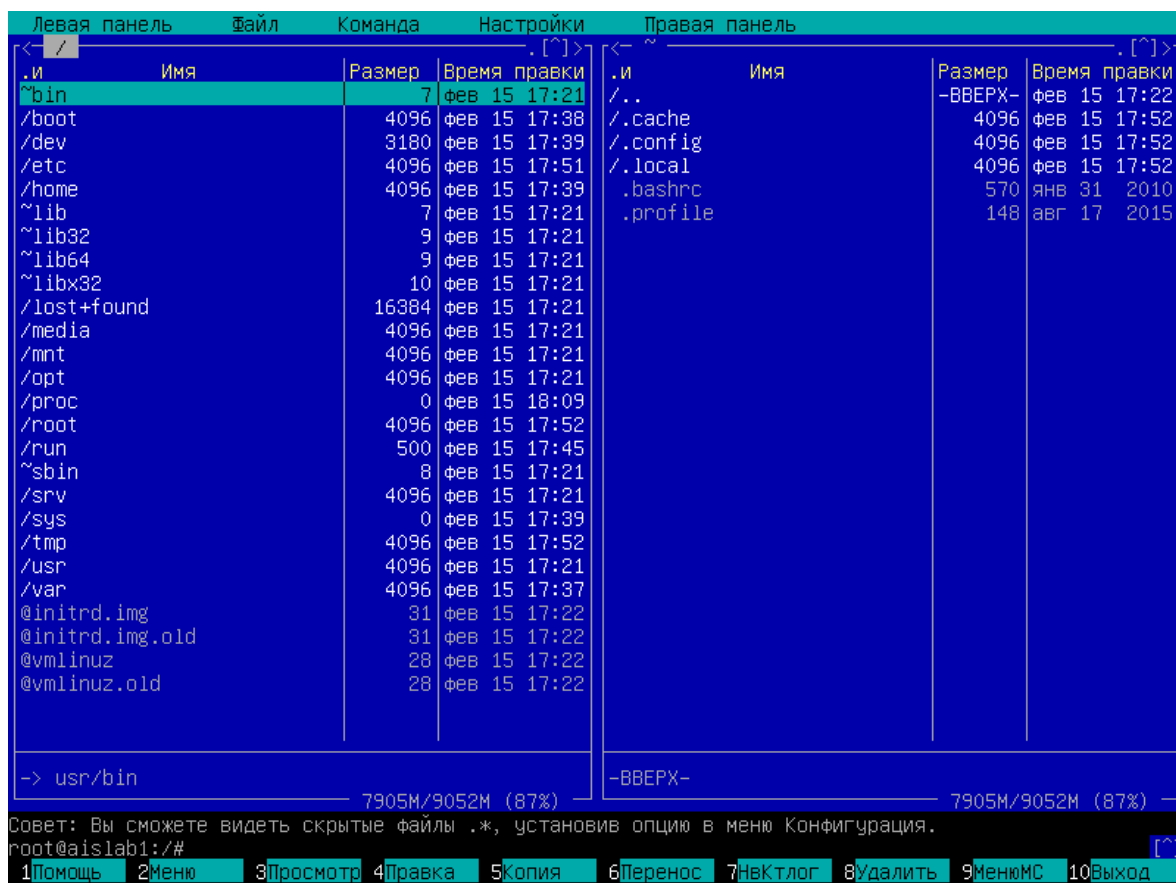


Рис. 5.1 – Список каталогов в основном разделе диска (левая панель MC)

Активация функций нижней панели инструментов MC осуществляется с помощью клавиш: F1 – помощь, F2 – меню, F5 – копировать... и т.д. Дополнительные комбинации: Shift+F6 – переименование файла, Shift+F5 – копия файла с возможностью изменить имя.

Назначение основных директорий:

/etc – содержит конфигурационные файлы для установленных пакетным менеджером приложений;

/home/<имя_пользователя> – личные папки обычных пользователей системы.

/root – домашняя директория встроенной учетной записи администратора системы.

/bin, /sbin – содержат исполняемые файлы (скрипты) установленных приложений;

/var – содержит переменные данные, такие как: системные журналы и журналы приложений (**/var/logs**), файлы баз данных (**/var/lib**), **контентные данные web-приложений** (**/var/www**).

/tmp – системная директория для хранения временных файлов, файлы, размещенные в этой папке будут удалены после перезапуска системы.

/opt – директория используется для установки сторонних системных приложений, которые не устанавливаются через менеджер пакетов apt (например, приложения собственной разработки).

Более подробную информация о структуре файловой системы Linux можно посмотреть здесь: <https://losst.ru/ctruktura-fajlovoj-sistemy-linux>

5.2 Выбор и установка веб-сервера. В общем случае под веб-сервером понимается специализированное программное обеспечение, предоставляющее доступ к контенту (html, видео, изображения, JavaScript и т.д.), по протоколу HTTP. Существует большое количество «встроенных» веб-серверов в рамках различных языков программирования и фреймворков, но большинство из данных решений не предназначены для работы в глобальной сети напрямую и используются только на этапе локального тестирования или работают совместно с «промышленными» веб-серверами.

Промышленный (production, жарг. «продуктивный») веб-сервер – специализированное решение для предоставления веб-контента, удовлетворяющее современным требованиям к безопасности, стабильности и производительности и предназначенное для работы с клиентскими запросами непосредственно в глобальной сети.

В данной лабораторной работе рассмотрим 3 основных промышленных веб-сервера, которые используются на более чем 80% серверов в глобальной сети: **Apache**, **Nginx**, **IIS**:

Таблица 1 – Ключевые особенности промышленных веб-серверов

Название	Особенности	Применение
Apache2 (или httpd в некоторых дистрибутивах Linux)	Является веб-сервером по умолчанию для ОС Linux. Имеет большое количество дополнительных модулей и простые механизмы для их динамического подключения. Прост в конфигурировании и настройке для шаблонных веб-приложений. Может работать под ОС Linux и Windows.	Используется для веб-приложений, использующих LAMP -стек, т.е. на основе Linux , Apache , MySQL , PHP (CRM-системы, конструкторы сайтов и т.п.) Используется как <u>проxy-сервер</u> , т.е. посредник между клиентом и целевым веб-приложением (например, на Java или Python).
Nginx	Имеет наилучшую производительность при работе со статическим	Используется как <u>проxy-сервер</u> , т.е. посредник между клиентом и целевым веб-приложением

	<p>контентом (html, видео, изображения, JavaScript и т.д.).</p> <p>Оптимизирован для потребления минимального количества ресурсов в условиях большого количества клиентских соединений.</p> <p>Может работать под ОС Linux и Windows.</p>	<p>(например, на java или python), при этом обычно проксируются только запросы к динамическому контенту приложения, а запросы к статическим файлам обрабатывает непосредственно Nginx.</p> <p>Используется как <u>балансировщик нагрузки</u> для целевых веб-приложений (т.е. перераспределяет клиентские запросы между несколькими приложениями).</p> <p>Используется как кэш-сервер для статического контента.</p>
<p>IIS (Microsoft Internet Information Services)</p>	<p>Предоставляет универсальную платформу для работы с различными веб-протоколами, в т.ч. почтовыми (POP3, SMTP), файловыми (FTP) и т.д.</p> <p>Тесная интеграция со средствами развёртывания и разработки от Microsoft (например, VisualStudio).</p> <p>Развитая система управления веб-приложениями (оснастка), интегрированная в ОС Windows Server.</p>	<p>Является основной платформой для приложений ASP.NET</p>

Т.к. в данной лабораторной работы требуется создание временного сайта проекта на CMS WordPress, которая использует LAMP-стек, устанавливаем веб-сервер Apache2:

```
apt install apache2
```

5.3 Настройка веб-сервера. Файлы конфигурации web-сервера находятся, соответственно, в **/etc/apache2**.

Описание основных конфигурационных файлов и папок:

apache2.conf – системные настройки веб-сервера.

ports.conf – настройки портов, на которых веб-сервер будет принимать соединения (по умолчанию: порт 80 для http запросов, порт 443 для защищённых https соединений).

conf-* – отвечают за конфигурацию базовых функций (безопасность, кодировка и т.п.).

mods-* – отвечают за конфигурацию дополнительных модулей.

sites-* – отвечают за конфигурацию сайтов (правила предоставления веб-контента).

Постфикс ***-available** означает, что данная папка содержит все доступные файлы конфигурации.

Папки с постфиксом ***-enabled** содержат ссылки, по которым можно определить, какая из доступных конфигураций активирована.

Для каждого web-сайта необходима настройка соответствующего конфигурационного файла в папке **sites-available**. По умолчанию в данной папке уже содержится файл с базовой настройкой **000-default.conf** с (примерно) следующим содержанием:

```
<VirtualHost *:80>
...
DocumentRoot /var/www/html
...
</VirtualHost>
```

Что означает, что Apache читает web-контент из корневой папки **/var/www/html** и контент доступен на всех интерфейсах (*) через порт 80.

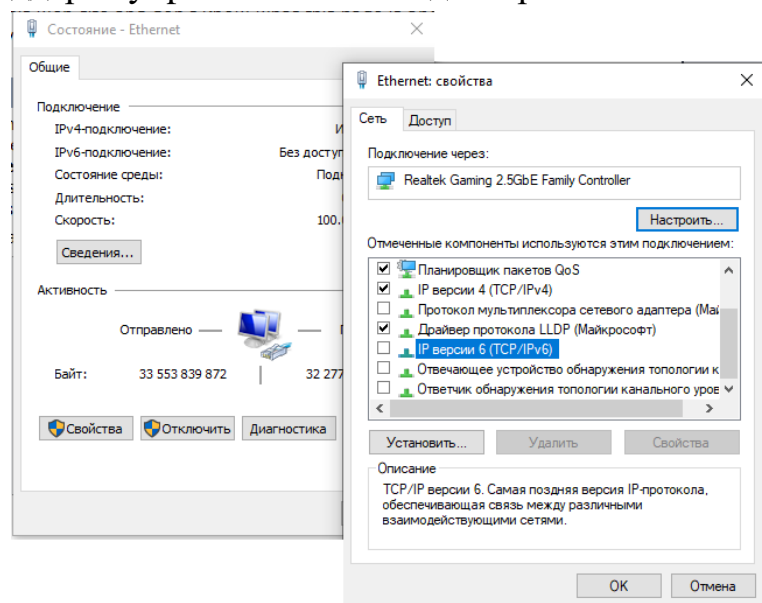
Проверяем, перейдя в браузере по адресу **http://192.168.56.104** (рис. 5.2):



Рис. 5.2 – Стартовая страница web-сервера Apache

ВАЖНО! Если при попытке перейти на стартовую страницу веб-сервера возникает таймаут соединения, то выполнить следующие шаги:

- 1) Отключить поддержку ipv6 на сетевом адаптере вашего ПК



- 2) В конфигурационном файле веб-сервера apache2 `/etc/apache2/ports.conf` изменить строку:

Listen 80

на

Listen 0.0.0.0:80

- 3) Перезагрузить ПК и виртуальную машину.

5.4 Настройка системы управления базой данных (СУБД). Устанавливаем СУБД **MariaDB**. MariaDB является ответвлением от MySQL и совместима (взаимозаменяема) с MySQL. В большинстве случаев рекомендуется устанавливать именно её как альтернативу MySQL ввиду упрощенного лицензирования и ряда дополнительных оптимизаций.

```
apt-get install mariadb-server
```

Выполним первоначальную настройку безопасности. Для этого существует команда `mysql_secure_installation`, после ввода которой запустится скрипт, в котором нужно будет выполнить следующий сценарий базовой настройки:

```
mysql_secure_installation
```

```
# Вводим пароль пароль пользователя root (администратор)
Enter current password for root (enter for none):
# Использовать локальные учетные записи системы для авторизации в БД ?
Switch to unix_socket authentication [Y/n] n
# Смена пароля суперпользователя root для MariaDB;
Change the root password? [Y/n] n
# Удаляем анонимного пользователя
Remove anonymous users? [Y/n] Y
# Запретить удаленное подключение к базе данных через root-аккаунт
Disallow root login remotely? [Y/n] Y
# Удаляем тестовую базу данных
Remove test database and access to it? [Y/n] Y
# Активируем новые правила доступа, перезагрузив таблицу привилегий:
Reload privilege tables now? [Y/n] Y
```

Подготовим пустую БД для развертывания CMS. Вводим команду **mariadb** и для доступа к консоли управления СУБД:

```
MariaDB [(none)]>
Создаем базу данных wpcmsdb:
MariaDB [(none)]> CREATE DATABASE wpcmsdb DEFAULT CHARACTER SET utf8;
Создадим пользователя cms для базы данных wpcmsdb.
MariaDB [(none)]> GRANT ALL ON wpcmsdb.* TO cms@localhost IDENTIFIED
BY 'ваш_пароль';
Обновляем права доступа:
MariaDB [(none)]> FLUSH PRIVILEGES;
Чтобы проверить результат, выводим список всех созданных БД:
MariaDB [(none)]> SHOW DATABASES;
Выводим список пользователей:
MariaDB [(none)]> SELECT User FROM mysql.user;
Завершаем работу с консолью управления БД:
MariaDB [(none)]> EXIT;
```

5.5 Настройка PHP. Установим необходимые пакеты для добавления функций обработки *.php скриптов на веб-сервере:

```
apt-get install php libapache2-mod-php php-mysql php-curl php-gd php-
mbstring php-xml php-xmlrpc php-soap php-intl php-zip
```

Далее необходимо настроить обработчик контента веб-сервера, чтобы в первую очередь читалась index.php страница (стартовая для CMS WordPress). Редактируем файл `/etc/apache2/mods-enabled/dir.conf` следующим образом:

```
<IfModule mod_dir.c>
```

```
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml
index.htm
</IfModule>
```

По умолчанию Apache в первую очередь обрабатывает index.html, и только четвертым будет обработка index.php. Для улучшения производительности лучше выставить приоритеты на обработку соответствующих индексных страниц для используемых типов веб-приложений. Для php-приложения перемещаем index.php на первое место соответственно.

Перезапустите службу web-сервера Apache:

```
systemctl restart apache2
```

Проверим статус web-сервера:

```
systemctl status apache2
```

ВАЖНО! **systemctl** – команда системного менеджера **systemd** для управления приложениями (сервисами), установленными в ОС Linux. Для управления или получения информации о приложении используйте аргументы **start** | **stop** | **restart** | **status** и имя сервиса. Например, для перезапуска mariadb:

```
systemctl restart mariadb
```

Проверим корректность установки php-модулей. Создаем тестовый скрипт php в корневой директории web-сервера:

```
nano /var/www/html/index.php
```

Содержание скрипта:


```
<?php phpinfo(); ?>
```

Т.к. служба веб-сервера Apache работает в системе от имени пользователя **www-data**, при создании новых контентных страниц в директории сервера вручную необходимо также предоставить права доступа для этого пользователя. Рекурсивно (параметр **-R**) меняем владельца на **www-data** для всех новых файлов в папке **/var/www/html**:

```
chown -R www-data /var/www/html
```

Открываем в браузере адрес ВМ. Должна появиться страница примерно следующего содержания (рис. 5.3):

PHP Version 7.4.30



System	Linux aislab22 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64
Build Date	Jul 7 2022 15:51:43
Server API	Apache 2.0 Handler

...

Рис. 5.2 – Содержание тестовой страницы index.php

ВАЖНО! На данном этапе сервер готов для развертывания PHP-приложений, НО данные настройки web-сервера являются недостаточными с точки зрения безопасности, т.о. сервер с подобными настройками можно использовать исключительно в безопасном сегменте сети за сетевым шлюзом!

6. Развертывание CMS WordPress. Переходим в каталог для временного хранения файлов:

```
cd /tmp
```

Скачиваем последнюю версию WordPress:

```
wget https://wordpress.org/latest.tar.gz
```

Распаковываем архив

```
tar -zxvf latest.tar.gz
```

Меняем название распакованной папки **wordpress** на более короткий вариант **wp**.

Копируем папку wp в корневой каталог web-сервера /var/www/html

```
cp -r /tmp/wp /var/www/html
```

Выставляем права для службы web-сервера:

```
chown -R www-data /var/www/html
```

Запускаем сценарий настройки CMS WordPress через браузер:

```
http://192.168.56.104/wp/wp-admin/install.php
```

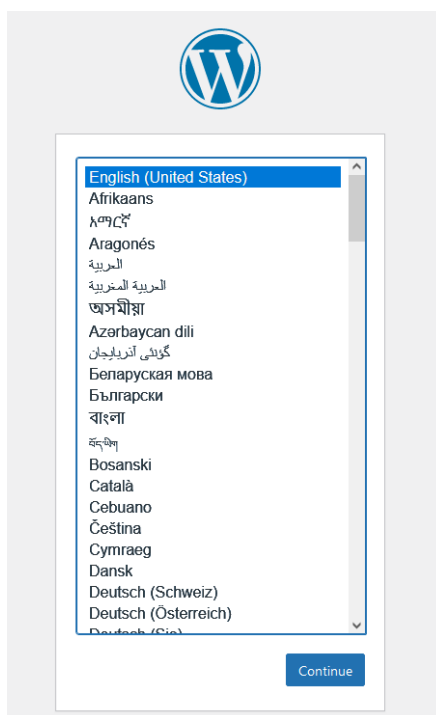


Рис. 5.3 – Стартовая страница сценария настройки CMS WordPress

В процессе выполнения сценария не забудьте использовать созданные ранее данные для подключения к БД (п.5.4)