МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 3 по курсу «Архитектура мобильных платформ»**

«Конфигурирование веб-сервера на одноплатном компьютере

Raspberry Pi 3»

Вариант №3

ВЫПОЛНИЛ студент группы 16-ИТ-3

Виноградова А.Д.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Ознакомиться с необходимым программным обеспечением для работы веб-сервера. Научиться конфигурировать веб-сервер на одноплатном компьютере Raspberry Pi 3.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. Назовите официальную операционную систему для Raspberry Pi 3.

С 2015 года дистрибутив Raspbian был официально предоставлен Raspberry Pi Foundation в качестве основной операционной системы для семейства одноплатных компьютеров Raspberry Pi.

1. Дайте определение понятию Веб-сервер.

Веб-сервер – это программа, которая принимает входящие HTTP-запросы, обрабатывает эти запросы, генерирует HTTP-ответ и отправляет его клиенту.

1. Перечислите и охарактеризуйте интерфейсы взаимодействия веб-сервера и веб-приложения.

Существует два главных типов интерфейс взаимодействия внешнего приложения и веб-сервера – *CGI* и *ISAPI.*

***CGI*** (Common Gateway Interface) – наиболее ранний способ взаимодействия веб-сервера и веб-приложения. Основная идея, которая лежит в основе CGI заключается в том, что при поступлении очередного HTTP-запроса, веб-сервер инициирует создание нового процесса и передает ему все необходимые данные HTTP-запроса. После того, как этот процесс отработает, он завершается, передав при этом результат обратно веб-серверу. Поскольку веб-сервер и приложение – это разные процессы с точки зрения операционной системы, то для обмена информации между ними используются средства межпроцессного взаимодействия (IPC) – зачастую это переменные окружения, именованные каналы и т.д.

***ISAPI*** (Internet Server API) – альтернативный способ взаимодействия веб-сервера и веб-приложения. В отличии от CGI, при взаимодействии в рамках интерфейса ISAPI, при поступлении очередного запроса, веб-сервер инициирует создание нового потока в рамках основного процесса, в котором работает веб-сервер. Поскольку с точки зрения операционной системы создание потока – это менее дорогостоящая операция, чем создание процесса, то такие приложения на практике оказываются более масштабируемыми. Кроме того, упрощается взаимодействие веб-сервера и веб-приложения, поскольку в этом случае используется единое адресное пространство в рамках операционной системы (поскольку весь код работает в одном и том же процессе). Однако, в случае серьезных неполадок в веб-приложении, которое взаимодействует с веб-сервером в рамках ISAPI, веб-сервер также потенциально подвергается риску быть завершенным. Поскольку веб-сервер и веб-приложение работают в одном и том же процессе, это действительно так. Поэтому разработчикам программного кода веб-сервера, поддерживающего ISAPI следует уделить этому вопросу особое внимание.

1. Перечислите наиболее популярные реализации веб-серверов и их особенности.

Наиболее популярный веб-сервер – это Apache, чуть менее распространены Nginx и IIS. Иногда с целью увеличения производительности устанавливается два веб-сервера: быстрый Nginx, который отдаёт пользователям статическое содержимое (физически существующие на сервере документы, не требующие обработки перед отправкой), а остальные запросы переадресовывает мощному Apache, который занимается генерацией динамических документов. Существуют и другие производительные связки (Nginx + FastCGI, например), о рациональности использования той или иной реализации принимают решение разработчики приложения и администраторы серверов.

1. Перечислите основные конфигурационные файлы Apache и доступные в них параметры.

Существуют два каталога для *conf*, *mods* и *site*. Это *available* и *enabled*. При включении модуля или хоста создается символическая ссылка из каталога *available* (доступно) в каталог *enable* (включено). Поэтому настройки лучше выполнять именно в каталогах *available*.

Главный файл конфигурации: *nano /eta/apache2/apache2.conf*

Директивы *Directory* отвечают за настройку прав доступа к той или иной директории в файловой системе.

В файле */etc/apache2/ports.conf* только одна директива, *Listen*, которая указывает программе на каком порту нужно работать.

В файле */etc/apache2/envvars* указаны переменные, которые можно использовать в других конфигурационных файлах.

1. Опишите механизм настройки виртуальных хостов Apache.

Настройки хостов Apache расположены в каталоге */etc/apache2/hosts-available/*. Для создания нового хоста достаточно создать файл с любым именем (лучше кончено с именем хоста) и заполнить его нужными данными. Обернуть все эти параметры нужно в директиву *VirtualHost*. Кроме рассмотренных параметров здесь будут использоваться такие:

* ServerName — основное имя домена
* ServerAlias — дополнительное имя, по которому будет доступен сайт
* ServerAdmin — электронная почта администратора
* DocumentRoot — папка с документами для этого домена

Например, *nano /etc/apache2/sites-available/test.site.conf*

Виртуальные хосты, как и модули нужно активировать. Для этого есть специальные утилиты. Чтобы активировать созданный виртуальный хост необходимо выполнить команду: *sudo a2ensite test.site*

Здесь *test.site* — имя файла виртуального хоста. Для отключения тоже есть команда: *sudo a2dissite test.site*

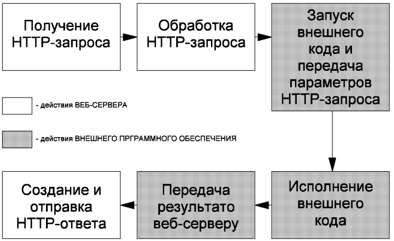
1. Перечислите основные особенности операционной системы Raspbian.

Детализированная настройка системы при первом запуске (тип клавиатуры, локализация, часовой пояс):

1. Обширный функционал с простотой в использовании;
2. Весит всего 440 МБ;
3. Специфицирован для обучения информатике.
4. Перечислите сопутствующие дополнительные задачи, решаемые веб-сервером.

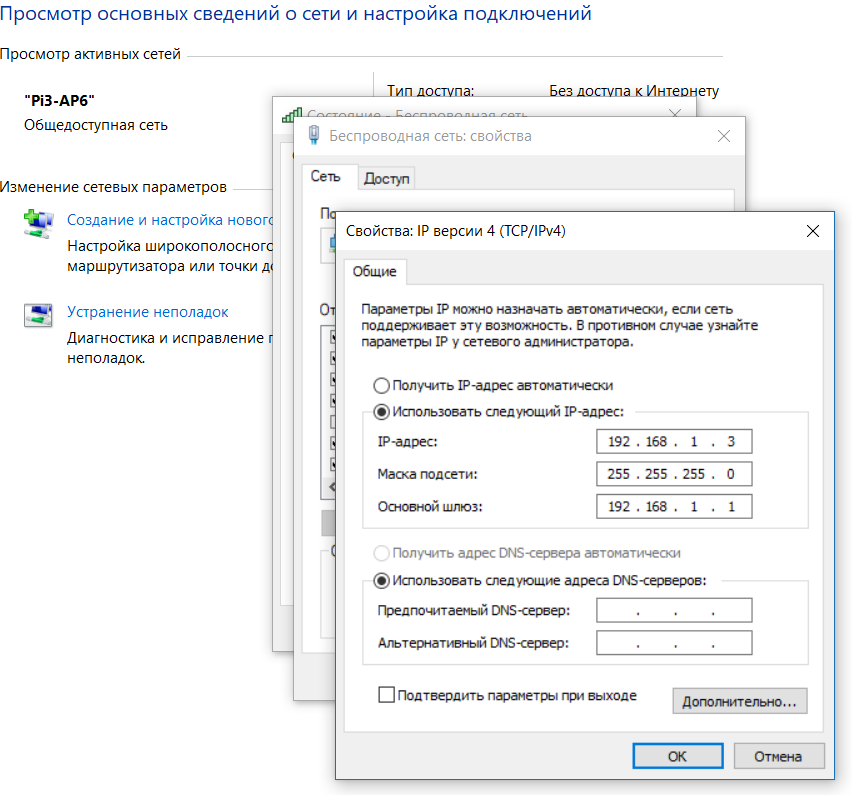
К этим задачам относится аутентфикация и авторизация пользователя, ведение серверного лога (для отладки работы веб-сервера), поддержка нескольких веб-сайтов на одном сервере (виртуальный хостинг), поддержка безопасных подключений по протоколу HTTPS и др. Эти функции в каждом конкретном случае зависят от реализации веб-сервера

1. Опишите механизм исполнения внешнего ПО веб-сервером.



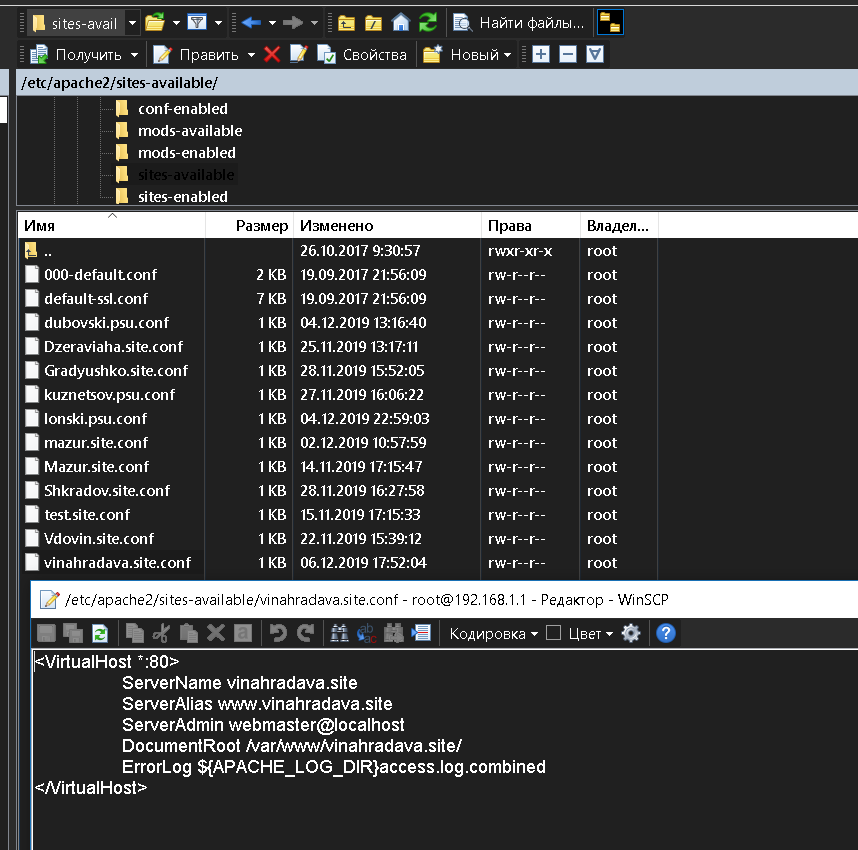
**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. Подключилась к Wi-Fi сети Raspberry Pi; по SFTP к Raspberry Pi по адресу 192.168.1.1 с помощью PuTTY и WinSCP, используя учётную запись пользователя root/root и настроила сеть:



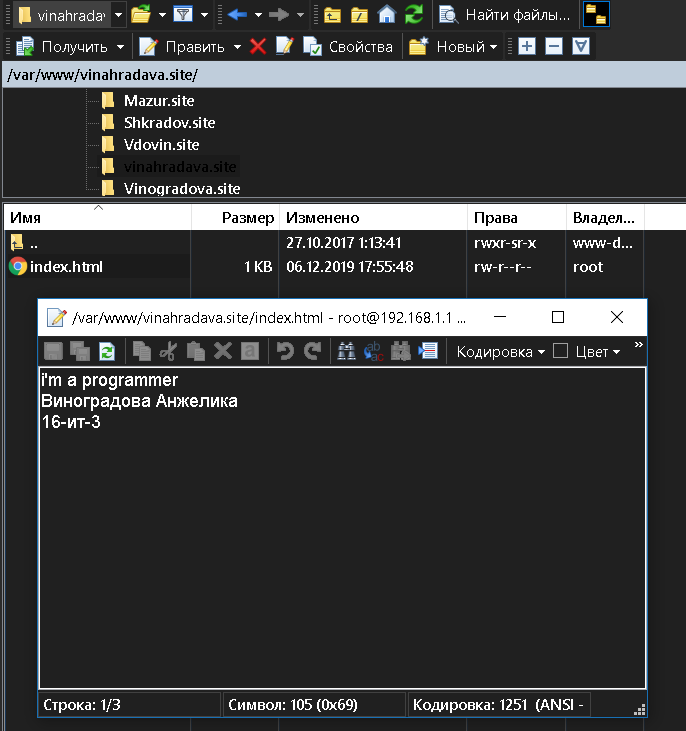
**Рисунок 1** – Настройка сети

1. Создала новый хост и заполнила данными, которые представлены на рисунке 2.



**Рисунок 2** – Заполнение файла хоста

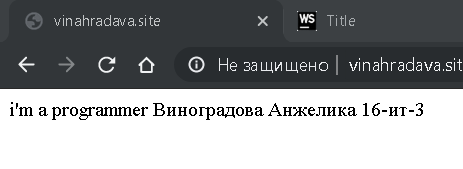
1. Активировала созданный виртуальный хост.
2. Создала index.html со следующим содержимым (рисунок 3).



**Рисунок 3** – Содержимое файла index.html

1. Перешла по адресу vinahradava.site.

**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:



**Рисунок 4** – Результат перехода по ссылке

**Листинг приложения**

1. #include <bcm2835.h> // подключение библиотеки bcm2835
2. #define PIN RPI\_V2\_GPIO\_P1\_12
3. void hightOff(){
4. bcm2835\_gpio\_write(PIN, LOW);
5. bcm2835\_delay(1000);
6. }
7. void hightOn(){
8. bcm2835\_gpio\_write(PIN, HIGH);
9. bcm2835\_delay(1000);
10. }
11. int main(int argc, char \*\*argv)
12. {
13. if (!bcm2835\_init()) // инициализация GPIO
14. return 1;
15. bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_OUTP);
16. for(int i=0;i<5;i++){
17. hightOn();
18. hightOff();
19. }
20. bcm2835\_close(); // завершение работы с GPIO
21. return 0;
22. }

**Вывод:** в результате выполнения лабораторной работы я ознакомилась с необходимым программным обеспечением для работы веб-сервера и научилась конфигурировать веб-сервер на одноплатном компьютере Raspberry Pi 3.