



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

## **ПРАКТИЧЕСКАЯ РАБОТА №3**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:**

**Студент группы ИКБО-16-19**

**Мурадов Н.Н.**

---

(подпись студента)

**Руководитель практической работы**

**преподаватель Волков М.Ю.**

---

(подпись руководителя)

Работа представлена

«\_\_\_» \_\_\_\_\_ 2021 г.

Допущен к работе

«\_\_\_» \_\_\_\_\_ 2021 г.

Москва 2021

## Содержание

Упражнение .....	3
Вывод: .....	8
Вопросы: .....	8
Список использованной литературы: .....	21

## Цель работы

Создать сложную серверную конфигурацию, состоящую из связки apache, nginx, php, mysql. Возможно использование связки apache, php как единый компонент. В данной конфигурации предполагается создание как минимум 3 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе №1. В этой конфигурации предполагается акселерированное проксирование без кэширования. Предполагается, что сервер nginx будет отображать статический контент, а apache динамический и в связке мы получим быстродействующую и эффективную систему. Для доступа к администрированию предусмотреть базовую авторизацию и аутентификацию с применением htaccess и хранением пользователей в БД.

## Упражнение

*Листинг 1. docker-compose.yml*

```
version: "3"

services:
  apache:
    build:
      context: httpd
    ports:
      - 4545:80
    volumes:
      - ./src:/var/www/html
```

- ./log:/var/log/apache2

depends\_on:

- bd

nginx:

image: nginx:1.13

ports:

- 4542:80

volumes:

- ./src:/var/www/html

- ./nginx/nginx.conf:/etc/nginx/nginx.conf

depends\_on:

- apache

bd:

image: mysql:5.7

#command: --default-authentication-plugin=mysql\_native\_password

volumes:

- ./bd/data:/var/lib/mysql

- ./bd:/docker-entrypoint-initdb.d

ports:

- 3306:3306

- 33060:33060

environment:

MYSQL\_ROOT\_PASSWORD: qwerty

MYSQL\_PASSWORD: WWW

## *Листинг 2. append.conf*

```
#IfModule mod_authn_dbd.c>

DBDriver mysql

DBDParams "host=bd,user=root,pass=qwerty,dbname=appDB"


<Directory "/var/www/html">
    AuthType Basic
    AuthName "Abrakadabra"
    AuthBasicProvider dbd


    AuthnCacheProvideFor dbd
    AuthnCacheContext auth
    Require valid-user
    AuthDBUserPWQuery "SELECT ENCRYPT(surname) FROM users WHERE name = %s"
</Directory>

#</IfModule>
```

## *Листинг 3. nginx.conf*

```
worker_processes 4;


events {
    worker_connections 1024;
}


http {
    include    mime.types;
    default_type application/octet-stream;
    sendfile   on;
    keepalive_timeout 65;


    server {
```

```

listen    80;

server_name localhost;

        root /var/www/html;


location ~
\.(jpg|jpeg|gif|png|ico|css|zip|tgz|gz|rar|bz2|doc|xls|exe|pdf|ppt|txt|tar|mid|midi|wav|bmp|rtf|j
s)$ {

        root /var/www/html;

}


location ~ /\. {

        deny all;

        access_log off;

        log_not_found off;

}


location ~ \.php$ {

        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        proxy_set_header X-Forwarded-Host $host;

        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_set_header Host $host;

        proxy_http_version 1.1;

        proxy_pass http://apache;

}


location @php-apache {

        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        proxy_set_header X-Forwarded-Host $host;

        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_set_header Host $host;

        proxy_http_version 1.1;

```

```

        proxy_pass http://apache;
    }

}
}

```

#### *Листинг 4. Dockerfile в httpd*

FROM php:7.2-apache

RUN a2enmod authn\_dbd && a2enmod authn\_socache && a2enmod socache\_shmcb && apt-get update && apt-get install -y apache2-utils && apt-get install -y libaprutil1-dbd-mysql && service apache2 restart && docker-php-ext-install pdo pdo\_mysql mysqli

COPY append.conf append.conf

RUN cat append.conf >> /etc/apache2/apache2.conf

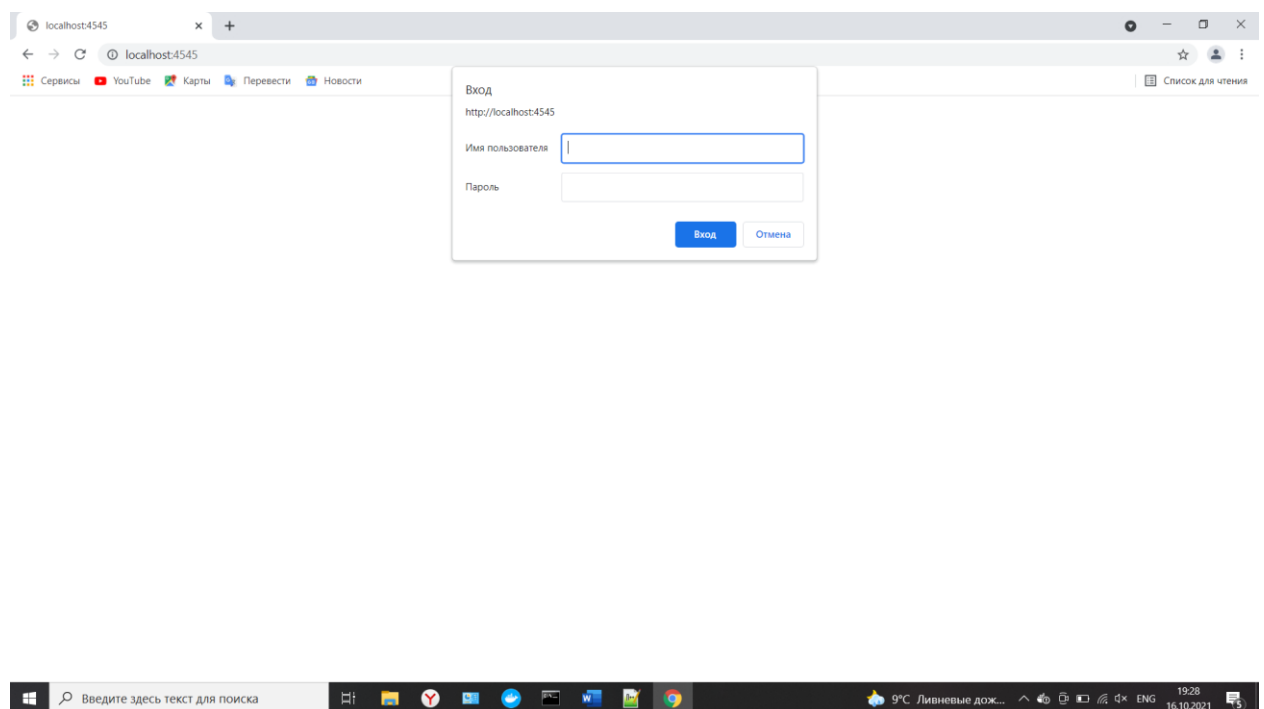


Рис.1 Окно аутентификации

**Таблица пользователей данного продукта**

Id	Name	Surname
1	Alex	Rover
2	Bob	Marley
3	Kate	Yandson
4	Lilo	Black

Рис.2 Содержимое страницы после аутентификации

## **Выводы**

В ходе выполнения работы была создана сложная серверная конфигурация, состоящая из связки apache+nginx+php+База данных. В качестве основы для сложной конфигурации была использована конфигурация из практического задания 1, а также добавлена базовая аутентификации.

## **Вопросы**

### **1. Что такое веб-сервер?**

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

### **2. Что такое сервер приложения и чем он отличается от веб-сервера?**

Основное различие между веб-сервером и сервером приложений заключается в том, что веб-сервер предназначен для обслуживания статических страниц, например HTML и CSS, тогда как сервер приложений отвечает за генерацию динамического содержимого путем выполнения кода на стороне сервера, например, JSP, сервлета или EJB.

### **3. Кратко опишите историю развития интернета в рамках развития веб-серверов.**

Первый сервер ARPANET был установлен 1 сентября 1969 года в Калифорнийском университете в Лос-Анджелесе. Компьютер Honeywell DP-516 имел 24 Кб оперативной памяти.

29 октября 1969 года в 21:00 между двумя первыми узлами сети ARPANET, находящимися на расстоянии в 640 км — в Калифорнийском университете Лос-Анджелеса (UCLA) и в Стэнфордском исследовательском институте



(SRI) — провели сеанс связи. Чарли Клайн (Charley Kline) пытался выполнить удалённое подключение к компьютеру в SRI. Успешную передачу каждого введённого символа его коллега Билл Дювалль (Bill Duvall) из SRI подтверждал по телефону.

В первый раз удалось отправить всего три символа «LOG», после чего сеть перестала функционировать. LOG должно было быть словом LOGON (команда входа в систему). В рабочее состояние систему вернули уже к 22:30 и следующая попытка оказалась успешной. Именно эту дату можно считать днём рождения Интернета. К 1971 году была разработана первая программа для отправки электронной почты по сети. Эта программа сразу стала очень популярна. В 1973 году к сети были подключены через трансатлантический телефонный кабель первые иностранные организации из Великобритании и Норвегии, сеть стала международной. В 1970-х годах сеть в основном использовалась для пересылки электронной почты, тогда же появились первые списки почтовой рассылки, новостные группы и доски объявлений. Однако в то время сеть ещё не могла легко взаимодействовать с другими сетями, построенными на других технических стандартах. К концу 1970-х годов начали бурно развиваться протоколы передачи данных, которые были стандартизированы в 1982—83 годах. Активную роль в разработке и стандартизации сетевых протоколов играл Джон Постел. 1 января 1983 года сеть ARPANET перешла с протокола NCP на TCP/IP, который успешно применяется до сих пор для объединения (или, как ещё говорят, «наслоения») сетей. Именно в 1983 году термин «Интернет» закрепился за сетью ARPANET. В 1984 году была разработана система доменных имён (англ. Domain Name System, DNS).

В 1984 году у сети ARPANET появился серьёзный соперник: Национальный научный фонд США (NSF) основал обширную межуниверситетскую сеть NSFNet (англ. National Science Foundation Network), которая была составлена из более мелких сетей (включая известные тогда сети Usenet и Bitnet) и имела

гораздо бо́льшую пропускную способность, чем ARPANET. К этой сети за год подключились около 10 тыс. компьютеров, звание «Интернет» начало плавно переходить к NSFNet. В 1988 году был разработан протокол Internet Relay Chat (IRC), благодаря чему в Интернете стало возможно общение в реальном времени (чат). В 1989 году в Европе, в стенах Европейского совета по ядерным исследованиям (фр. Conseil Européen pour la Recherche Nucléaire, CERN) родилась концепция Всемирной паутины. Её предложил знаменитый британский учёный Тим Бернерс-Ли, он же в течение двух лет разработал протокол HTTP, язык HTML и идентификаторы URI. В 1990 году сеть ARPANET прекратила своё существование, полностью проиграв конкуренцию NSFNet. В том же году было зафиксировано первое подключение к Интернету по телефонной линии (т. н. «дозвон» — англ. Dialup access). В 1991 году Всемирная паутина стала общедоступна в Интернете, а в 1993 году появился знаменитый веб-браузер NCSA Mosaic. Всемирная паутина набирала популярность. Можно считать что существует две ясно различимые эры в истории Web: [до браузера Mosaic] Марка Андриссена и после. Именно сочетание веб-протокола от Тима Бернерс-Ли, который обеспечивал коммуникацию, и браузера (Mosaic) от Марка Андриссена, который предоставил функционально совершенный пользовательский интерфейс, создало условия для наблюдаемого взрыва (интереса к Веб). За первые 24 месяца, истекшие после появления браузера Mosaic, Web прошел стадию от полной неизвестности (за пределами считанного числа людей внутри узкой группы ученых и специалистов лишь одного мало кому известного профиля деятельности) до полной и абсолютно везде в мире его распространенности. A Brief History of Cyberspace, Mark Pesce, ZDNet, 15 октября 1995 В 1995 году NSFNet вернулась к роли исследовательской сети, маршрутизацией всего трафика Интернета теперь занимались сетевые провайдеры, а не суперкомпьютеры Национального научного фонда.

В том же 1995 году Всемирная паутина стала основным поставщиком информации в Интернете, обогнав по трафику протокол пересылки файлов FTP. Был образован Консорциум всемирной паутины (W3C). Можно сказать, что Всемирная паутина преобразила Интернет и создала его современный облик. С 1996 года Всемирная паутина почти полностью подменяет собой понятие «Интернет». В 1990-е годы Интернет объединил в себе большинство существовавших тогда сетей (хотя некоторые, как Фидонет, остались обособленными). Объединение выглядело привлекательным благодаря отсутствию единого руководства, а также благодаря открытости технических стандартов Интернета, что делало сети независимыми от бизнеса и конкретных компаний. К 1997 году в Интернете насчитывалось уже около 10 млн компьютеров, было зарегистрировано более 1 млн доменных имён. Интернет стал очень популярным средством для обмена информацией. В настоящее время подключиться к Интернету можно через спутники связи, радио-каналы, кабельное телевидение, телефон, сотовую связь, специальные оптоволоконные линии или электропровода. Всемирная сеть стала неотъемлемой частью жизни в развитых и развивающихся странах.

В течение пяти лет Интернет достиг аудитории свыше 50 миллионов пользователей.

#### 4. Кратко опишите протокол HTTP.

HTTP — это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (web-browser).

#### 5. Опишите механизм взаимодействия HTTP-сервера, HTTP-клиента и пользователя.

Когда пользователь хочет перейти на страницу, браузер отправляет HTTP-запрос GET с указанием URL-адреса его HTML-страницы. Сервер извлекает запрошенный документ из своей файловой системы и возвращает HTTP-ответ, содержащий документ и код состояния HTTP Response status code 200 OK (успех). Сервер может вернуть другой код состояния, например, «404 Not Found», если файл отсутствует на сервере или «301 Moved Permanently», если файл существует, но был перемещён в другое место.

#### 6. Опишите цели и задачи веб-сервера.

Цель веб-сервера проста - обслуживать одновременно большое количество клиентов, максимально эффективно используя hardware. Главная задача веб-сервера принимать HTTP-запросы от пользователей, обрабатывать их, переводить в цифровой компьютерный код. Затем выдавать HTTP-ответы, преобразуя их из миллионов нулей и единиц в изображения, медиа-поток, буквы, HTML страницы. Любой веб-сервер, для удобства его использования пользователями, должен иметь удобный веб-браузер. Он передает веб-серверу запросы, преобразованные в URL-адреса интернет-ресурсов. Наряду со стандартными функциями, некоторые веб-серверы имеют дополнительные. Так, к примеру, соответствующее программное обеспечение может фиксировать число обращений пользователей к тому или иному ресурсу, записывать их в отдельный журнал. А еще они могут поддерживать HTTPS, что не маловажно для защищенного соединения между сайтами и пользователями. Зачастую веб-сервер устанавливается вместе с мейл-сервером. Это позволяет пользователям быстро переходить на страничку почты прямо с сайта, нажав всего лишь на одну гиперссылку.

#### 7. Опишите технологию SSI.

SSI – это простая и удобная технология организации динамических страничек. SSI экономит место на сервере, и одновременно делает администрирование сайта удобнее в десятки раз.

## 8. Что такое система управления контентом?

Система управления контентом (CMS) — это программное обеспечение, которое работает в вашем браузере. Она позволяет создавать, управлять и изменять веб-сайт и его содержимое, не имея никаких знаний в области программирования. Система управления контентом предоставляет вам графический интерфейс пользователя. В нём вы можете управлять всеми аспектами вашего сайта.

## 9. Верно ли, что сервер приложения умеет работать с протоколом HTTP?

Верно.

HTTP-клиентами чаще всего являются браузеры — Google Chrome, Mozilla Firefox, Safari, Opera, Yandex Browser и другие. А серверами являются веб-сервера. Вот эта приставка «веб-» и указывает на то, что это не просто какой-то сервер, а сервер, который умеет принимать запросы и отвечать на них по протоколу HTTP.

## 10. Что такое CGI?

CGI (от англ. Common Gateway Interface — «общий интерфейс шлюза») — стандарт интерфейса, используемого для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом

## 11. Как работает система с использованием интерфейс шлюза - CGI?

Сам интерфейс разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода-вывода. Такими возможностями обладают даже скрипты для встроенных командных интерпретаторов операционных систем, поэтому в простых случаях могут использоваться даже командные скрипты.

## 12. Назовите достоинства и недостатки CGI.

- CGI не налагает особых условий на платформу и web - сервер, поэтому работает на всех популярных платформах и web - серверах. Также технология не привязана к конкретному языку программирования и может быть использована на любом языке, работающем со стандартными потоками ввода/вывода.
- Производительность CGI - программ не высока. Основной причиной этого является то, что при очередном обращении к серверу для работы CGI – программы создается отдельный процесс, что требует большого количества системных ресурсов.
- Встроенных средств масштабируемости технология не предусматривает.
- CGI - программа представляет из себя готовый к исполнению файл, что препятствует легкому расширению системы.

### 13. Что такое FastCGI?

Интерфейс FastCGI — клиент-серверный протокол взаимодействия веб-сервера и приложения, дальнейшее развитие технологии CGI. По сравнению с CGI является более производительным и безопасным.

### 14. Назовите основные отличия CGI от FastCGI.

CGI-программа, запущенная в цикле. Если обычная CGI-программа заново запускается для каждого нового запроса, то в FastCGI-программе используется очередь запросов, которые обрабатываются последовательно.

### 15. Что такое менеджер процессов?

Поток `process_manager` отвечает за генерацию новых экземпляров процесса, за связь со средой и между экземплярами процесса, а также за выполнение экземпляров процесса.

### 16. Что такое PHP-FPM?

PHP-FPM — это альтернативная реализация PHP FastCGI с несколькими дополнительными возможностями, которые обычно используются для высоконагруженных сайтов.

#### 17. Что такое Spawn-fcgi?

spawn-fcgi — одна из составных частей проекта Lighttpd. Предназначен он для того, что бы запустить php, как FastCGI сервер, ну а с этим сервером может работать потом практически любой http сервер.

#### 18. Что такое Lighttpd?

lighttpd это веб-сервер с открытым исходным кодом, оптимизированный для критически важных сред, отвечает за предоставление доступа через HTTP или HTTPS протокол к статическому контенту.

#### 19. Что такое chroot окружение?

Chroot окружение — способ запуска программ, системный вызов и просто команда, позволяющая изменить корневой каталог в системе.

Chroot используется для создания jail и изоляции процессов, а также для сборки пакетов методом debootstrap. Командой chroot с указанием каталога запускается механизм, создающий систему директорий в этом каталоге идентичную той, что существует в корне. Команда выполняется только от имени суперпользователя.

#### 20. Опишите механизм взаимодействия сервисов с использованием FastCGI

FastCGI — открытый унифицированный стандарт, расширяющий интерфейс CGI и позволяющий создавать высокопроизводительные web-приложения без использования специфичных API web-сервера. Цель данной спецификации — с точки зрения FastCGI-приложения описать интерфейс взаимодействия между ним и web-сервером, также реализующим интерфейс FastCGI. Связь между web-сервером и FastCGI-процессом осуществляется

через один сокет, который процесс должен слушать на предмет входящих подключений от web-сервера.

После приема соединения от web-сервера FastCGI-процесс обменивается данными с использованием простого протокола, решающего две задачи: организация двунаправленного обмена в рамках одного соединения (для эмуляции STDIN, STDOUT, STDERR) и организация нескольких независимых FastCGI-сессий в рамках одного соединения.

21.Опишите процесс выбора встроенного или внешнего менеджера процессов.

Менеджер процессов тесно взаимодействует с микроядром, чтобы обеспечить услуги, составляющие сущность операционной системы. Менеджер процессов отвечает за создание новых процессов в системе и за управление основными ресурсами, связанными с процессом. Все эти услуги предоставляются посредством сообщений.

Встроенный менеджер процессов – удобная программа, с помощью которой можно определять запущенные на ПК приложения, процессы и службы, управлять вышеперечисленными компонентами (запускать, останавливать или завершать). С

помощью нее можно оценивать и влиять на быстродействие компьютера, а также выполнять другие задачи.

К сожалению, встроенный менеджер процессов для анализа и обработки запущенных процессов не всегда удобен и имеет очень ограниченные возможности, он не в полной мере отображает процессы, запущенные на компьютере, а средства анализа запущенных процессов вообще весьма ограничены. Из-за недостатков в применении менеджера процессов для анализа процессов, запущенных на компьютере, приходится прибегать к



внешним менеджерам процессов. Они более качественно отображают информацию и позволяют лучше контролировать процессы.

## 22. Что такое интерфейс шлюза?

Сетевой шлюз - аппаратный маршрутизатор или программное обеспечение для сопряжения компьютерных сетей, использующих разные протоколы (например, локальной и глобальной). Сетевой шлюз конвертирует протоколы одного типа физической среды в протоколы другой физической среды (сети).

## 23. Что такое SCGI?

SCGI (Simple Common Gateway Interface) - простой общий интерфейс шлюза - разработан как альтернатива CGI и во многом аналогичен FastCGI, но более прост в реализации.

## 24. Что такое PCGI?

PCGI (Perl Common Gateway Interface) — библиотека к языку программирования Perl для работы с интерфейсом CGI. Библиотека позволяет с высокой скоростью обрабатывать входящий поток данных. Основное достоинство заключается в том, что библиотека позволяет совершенно безопасно принимать сколь угодно крупные объёмы данных, при этом очень экономично потребляя оперативную память.

## 25. Что такое PSGI?

PSGI (Perl Web Server Gateway Interface) – это спецификация, предназначенная для отделения среды веб-сервера от кода веб-фреймворка. PSGI не является программным интерфейсом (API) для веб-приложений.

## 26. Что такое WSGI?

WSGI — стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером, например Apache. [WSGI предоставляет простой и универсальный интерфейс между большинством веб-серверов и веб-приложениями или фреймворками].

Это протокол, спецификация, которая была предложена в PEP 3333. Этот протокол предназначен для решения проблем совместимости многих веб-платформ и программного обеспечения веб-сервера. Благодаря WSGI вам больше не нужно выбирать конкретное программное обеспечение веб-сервера из-за используемой веб-платформы.

## 27. Опишите механизм взаимодействия серверов Apache и PHP

Когда компьютер обращается к web-серверу Apache, то он запускает интерпретатор PHP. Он выполняет скрипт записанный в файле index.php. То есть Apache это сервер, который взаимодействует с клиентом (принимает и отвечает на запросы клиента), а затем сервер выполняет запрос клиента, используя PHP

## 28. Опишите преимущества веб-сервера Apache.

Основные достоинства Apache - надежность, безопасность и гибкость настройки. Apache позволяет подключать различные модули, добавляющие в него новые возможности - например, можно подключить модуль, обеспечивающий поддержку PHP или любого другого Web-ориентированного языка программирования.

## 29. Опишите недостатки веб-сервера Apache.

Недостатки - отсутствие удобного графического интерфейса администратора. Настройка Apache осуществляется путем редактирования его конфигурационного файла. В Интернете можно найти простые конфигураторы Apache, но их возможностей явно не хватает для настройки всех функций Web-сервера.

## 30. Опишите архитектуру веб-сервера Apache.

Apache состоит из ядра и динамической модульной системы. Параметры системы изменяются с помощью конфигурационных файлов.

(В архитектуру Apache входит: простое ядро, платформо-зависимый уровень (APR), и модули.)

31.Опишите функции ядра веб-сервера Apache.

Ядро Apache включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей. Ядро (в отличие от модулей) полностью разрабатывается Apache Software Foundation, без участия сторонних программистов.

Теоретически ядро apache может функционировать в чистом виде, без использования модулей. Однако функциональность такого решения крайне ограничена.

Ядро Apache полностью написано на языке программирования C.

32.Опишите конфигурацию веб-сервера Apache.

Система конфигурации Apache основана на текстовых конфигурационных файлах. Имеет три условных уровня конфигурации:

- Конфигурация сервера
- Конфигурация виртуального хоста
- Конфигурация уровня каталога

33.Что такое URI, URL и чем они различаются.

URL — это URI, который, помимо идентификации ресурса, предоставляет ещё и информацию о местонахождении этого ресурса. А URN — это URI, который только идентифицирует ресурс в определённом пространстве имён (и, соответственно, в определённом контексте), но не указывает его местонахождение.

(URI: Это лишь обобщенное понятие (множество) идентификации ресурса, включающее в нашем случае как URL, так и URN, как по отдельности, так и

совместно. Т.е. мы можем считать, что:  $URI = URL$  или  $URI = URN$  или  $URI = URL + URN$ )

## Список использованной литературы:

1. Конспект лекций по дисциплине «Разработка сервисных частей интернет ресурсов», РТУ МИРЭА, лектор –преподаватель Куликов А.А.
2. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131719> (дата обращения: 29.09.2021). — Режим доступа: для авториз. пользователей.
3. Моуэт, Э. Использование Docker / Э. Моуэт ; научный редактор А. А. Маркелов ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-426-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/93576> (дата обращения: 16.09.2021). — Режим доступа: для авториз. пользователей.
4. Алибеков, Б. И. Лабораторный практикум по Web-программированию на PHP : учебное пособие / Б. И. Алибеков. — Махачкала : ДГУ, 2018. — 273 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/158357> (дата обращения: 16.09.2021). — Режим доступа: для авториз. пользователей.
5. Джош, Л. Современный PHP. Новые возможности и передовой опыт / Л. Джош ; перевод с английского Р. Н. Рагимов. — Москва : ДМК Пресс, 2016. — 304 с. — ISBN 978-5-97060-184-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/93269> (дата обращения: 16.09.2021). — Режим доступа: для авториз. пользователей.

