|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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»

**Студент группы** ИКБО-02-20 Бессмертных И.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Благирев.М.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

Оглавление

[1. Цель работы 4](#_Toc118091132)

[2. Ход работы 4](#_Toc118091133)

[ВЫВОДЫ 13](#_Toc118091134)

[3. Ответы на вопросы к практической работе 13](#_Toc118091135)

[3.1. Что такое веб-сервер? 14](#_Toc118091136)

[3.2. Что такое сервер приложения и чем он отличается от веб-сервера? 14](#_Toc118091137)

[3.3. Кратко опишите историю развития интернета в рамках развития веб-серверов. 14](#_Toc118091138)

[3.4. Кратко опишите протокол HTTP. 15](#_Toc118091139)

[3.5. Опишите механизм взаимодействия HTTP-сервера, HTTP-клиента и пользователя. 16](#_Toc118091140)

[3.6. Опишите цели и задачи веб-сервера. 16](#_Toc118091141)

[3.7. Опишите технологию SSI. 16](#_Toc118091142)

[3.8. Что такое система управления контентом? 17](#_Toc118091143)

[3.9. Верно ли, что сервер приложения умеет работать с протоколом HTTP? 17](#_Toc118091144)

[3.10. Что такое CGI? 17](#_Toc118091145)

[3.11. Как работает система с использованием интерфейс шлюза - CGI? 17](#_Toc118091146)

[3.12. Назовите достоинства и недостатки CGI. 18](#_Toc118091147)

[3.13. Что такое FastCGI? 18](#_Toc118091148)

[3.14. Назовите основные отличия CGI от FastCGI. 18](#_Toc118091149)

[3.15. Что такое менеджер процессов? 18](#_Toc118091150)

[3.16. Что такое PHP-FPM? 19](#_Toc118091151)

[3.17. Что такое Spawn-fcgi? 19](#_Toc118091152)

[3.18. Что такое Lighttpd? 19](#_Toc118091153)

[3.19. Что такое chroot окружение? 19](#_Toc118091154)

[3.20. Опишите механизм взаимодействия серверов с использованием FastCGI. 20](#_Toc118091155)

[3.21. Опишите процесс выбора встроенного или внешнего менеджера процессов. 20](#_Toc118091156)

[3.22. Что такое интерфейс шлюза? 20](#_Toc118091157)

[3.23. Что такое SCGI? 20](#_Toc118091158)

[3.24. Что такое PCGI 20](#_Toc118091159)

[3.25. Что такое PSGI? 21](#_Toc118091160)

[3.26. Что такое WSGI? 21](#_Toc118091161)

[3.27. Опишите механизм взаимодействия серверов Apache и PHP. 21](#_Toc118091162)

[3.28. Опишите преимущества веб-сервера Apache. 21](#_Toc118091163)

[3.29. Опишите недостатки веб-сервера Apache. 22](#_Toc118091164)

[3.30. Опишите архитектуру веб-сервера Apache. 22](#_Toc118091165)

[3.31. Опишите функции ядра веб-сервера Apache. 23](#_Toc118091166)

[3.32. Опишите конфигурацию веб-сервера Apache. 23](#_Toc118091167)

[3.33. Что такое URI, URL и чем они различаются. 24](#_Toc118091168)

[4. Ссылка на удаленный репозиторий проекта 24](#_Toc118091169)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 25](#_Toc118091170)

1. Цель работы

В задании предлагается создать сложную серверную конфигурацию, состоящую из связки apache+nginx+php+База данных. Возможно использование связки apache+php как единый компонент. В данной конфигурации предполагается создание как минимум 3 элемента (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 быстродейственную и эффективную систему.

1. Ход работы

Для облегчения работы с рекомендуемыми инструментами используются предоставленные скрипт инициализации БД для СУБД MYSQL, данные перемещены по адресу «etc/db/init.sql».

Создан образ apache с php

Создан файл конфигурации для nginx, располагающийся по адресу «etc/nginx/app.conf»

Для совместной работы контейнера сервера и контейнера базы данных был создан файл “docker-compose.yml”. Для контейнера базы данных используется готовый образ MariaDB. Для контейнера apache с php используется Dockerfile. Для контейнера nginx используется готовый образ.

Создана страница “etc/httpd/app/admin/index.php” с админ-панелью, для доступа к которой необходимо пройти авторизацию. Данные для входа: логин – ivan, пароль – ivan123321

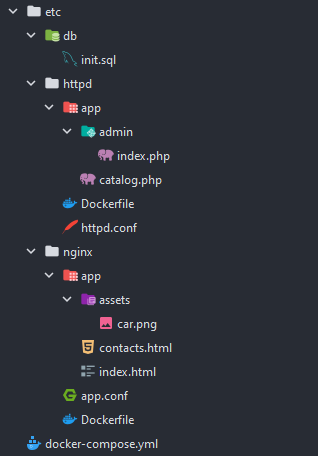


Рисунок 2.1 – Файлы проекта

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 2.2 – Файл docker-compose.yml

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.3 – Файл etc/nginx/app.conf

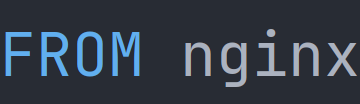


Рисунок 2.4 – Файл etc/nginx/Dockerfile

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 2.5 – Файл etc/httpd/httpd.conf

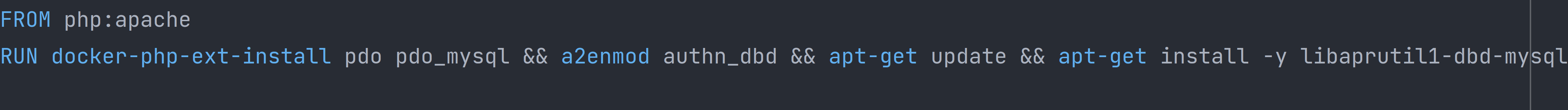


Рисунок 2.6 – Файл etc/httpd/Dockerfile

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 2.7 – etc/nginx/app/index.html – статическая страница 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.8 – etc/nginx/app/contacts.html – статическая страница 2

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.9 – etc/httpd/app/catalog.php – динамическая страница 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.10 - etc/httpd/app/admin/index.php – динамическая страница 2 и с админ-панелью

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.11 - etc/db/init.sql – дамп базы данных

Изображение выглядит как текст, автомобиль, внешний, оранжевый

Автоматически созданное описание

Рисунок 2.12 – результат работы статической страницы 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.13 – результат работы статической страницы 2

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.14 – результат работы динамической страницы 1

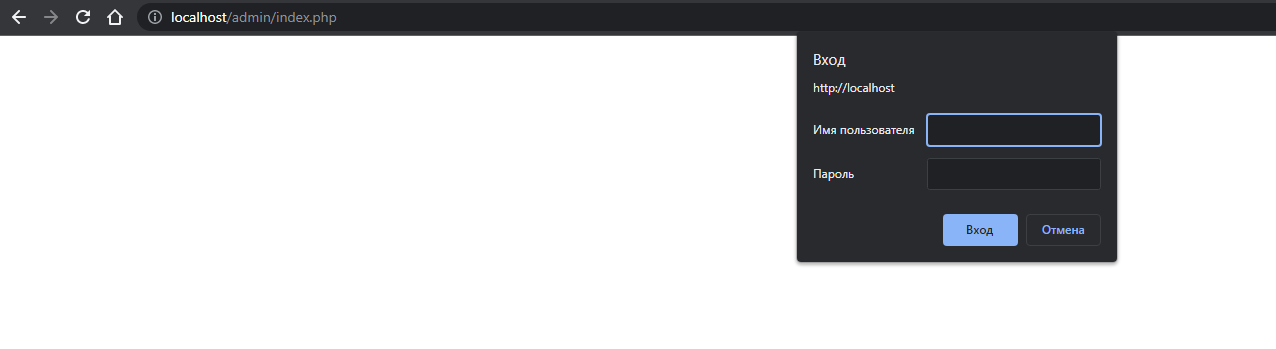


Рисунок 2.15 – требование авторизации на странице с админ-панелью

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.16 – результат работы динамической страницы 2 с админ-панелью

ВЫВОДЫ

Была создана конфигурация серверного программного обеспечения с сервисом для статических и динамических страниц. Это сделано с помощью nginx и apache, где первый обрабатывает входящие от пользователя запросы и является обратным прокси-сервером, который также обрабатывает статические страницы, а второй обрабатывает запросы nginx, когда запрашивается динамическая страница.

1. Ответы на вопросы к практической работе
   1. Что такое веб-сервер?

Веб-сервер - это компьютерное программное обеспечение и базовое оборудование, которое принимает запросы через HTTP (сетевой протокол, созданный для распространения веб-контента) или его безопасный вариант HTTPS.

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

Сервер приложений - это программа сервера, работающая в распределенной сети и обеспечивающая среду выполнения для прикладных программ. Сервер приложений представляет собой основной компонент среды выполнения во всех конфигурациях - на нем выполняются приложения. Основное отличие веб-сервера от [сервера приложений](https://ru.wikipedia.org/wiki/Сервер_приложений) заключается в том, что веб-сервер предназначен для обслуживания статических страниц, например [HTML](https://ru.wikipedia.org/wiki/HTML) и [CSS](https://ru.wikipedia.org/wiki/CSS), тогда как сервер приложений отвечает за генерацию динамического содержимого путём выполнения кода на стороне сервера

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

Интернет зародился в 60-х годах, в период Холодной войны. В это время США искали новые методы и способы безопасной передачи данных. В результате была разработана компьютерная сеть, получившая название ARPAnet – Advanced Research Projects Agency Work.

Шел 1989 год, 34 летний молодой и амбициозный математик из Англии — Тимоти Джон Бе́рнерс-Ли (далее — Бернерс-Ли Тим), работая в CERN (ЦЕРН — Европейская организация по ядерным исследованиям) над внутренней системой обмена документов Enquire, предложил своему начальству проект, известный как Всемирная паутина (англ. World Wide Web).

В основе проекта, который предлагал Джон, была идея публикации гипертекстовых документов, связанных между собой гиперссылками, что облегчило бы поиск и сбор информации. Реализация проекта требовала фундаментальной проработки всех аспектов проекта. Для реализации проекта были разработаны:

* идентификаторы URI (и, как частный случай, URL)
* протокол HTTP
* язык гипертекстовой разметки HTML

В рамках проекта Джон также написал первый в мире веб-сервер «httpd» и первый в мире гипертекстовый веб-браузер для компьютера NeXT, называвшийся «Nexus».

Спустя буквально 2 года, в 1993 году появился знаменитый веб-браузер с графическим интерфейсом NCSA Mosaic. Его основными разработками были Марк Андриссен (Marc Andreessen) и Эрик Дж. Бина (Eric J. Bina).

Mosaic имел интуитивно понятный интерфейс, стабильно работал, имел версию под Microsoft Windows, был бесплатен для загрузки, просто устанавливался, а главное был первым браузером, который отображал изображения в тексте, а не в отдельном окне.

Mosaic разделил историю становления www до своего появления и после. Так как позволил всем владельцам персональных компьютеров прикоснуться к интернету. Многие идеи воплощенные в NCSA Mosaic, получили развитие в браузерах следующего поколения: Netscape Navigator и Internet Explorer.

* 1. Кратко опишите протокол HTTP.

HTTP-протокол — это набор правил, по которым компьютеры обмениваются данными друг с другом. Его инициирует клиент (в данном случае — человек, заходящий в интернет с любого устройства), а обрабатывает сервер и возвращает обратно клиенту. Между ними могут находиться прокси-серверы, которые занимаются дополнительными задачами — шифрованием данных, перераспределением нагрузки или кэшированием.

Протокол HTTP реализован по клиент-серверной технологии и работает по принципу запрос-ответ без сохранения состояния. Целью запроса служит некий ресурс, который определяется единым идентификатором ресурса − URI (Uniform Resource Identifier), HTTP использует одну из разновидностей URI − URL (Uniform Resource Locator) − универсальный указатель ресурса, который помимо сведений о ресурсе определяет также его физическое местоположение. В процессе работы HTTP-сервера получает запрос клиента, обрабатывает его и либо выдает ему запрашиваемый ресурс, либо сообщает, что это сделать невозможно.

* 1. Опишите механизм взаимодействия HTTP-сервера, HTTP-клиента и пользователя.

HTTP-клиент, то есть браузер, инициирует HTTP-запрос и после этого ожидает ответ. Сервер обрабатывает запрос и отправляет ответ, после чего соединение прерывается. Получается, что клиент и сервер знают друг о друге только во время текущей сессии. Дальнейшие запросы выполняются уже при новом подключении, а клиент и сервер будут новыми друг для друга.

* 1. Опишите цели и задачи веб-сервера.

Цель веб-сервера проста - обслуживать одновременно большое количество клиентов. Главная цель веб сервера принимать HTTP-запросы от пользователей, обрабатывать их, переводить в цифровой компьютерный код. Затем выдавать HTTP-ответы, преобразуя их из миллионов нолей и единичек в изображения, медиа-потоки, буквы, HTML страницы.

* 1. Опишите технологию SSI.

Технология SSI (Server Side Includes) – это директивы, которые помещаются в HTML-страницы и оцениваются на сервере, когда обращаются к страницам. Она позволяет вам добавлять динамически сгенерированное содержание в существующую HTML-страницу без необходимости обработки всей страницы через CGI-программу или другую динамическую технологию.

* 1. Что такое система управления контентом?

Система управления контентом (CMS) - это компьютерное программное обеспечение, используемое для управления созданием и модификацией цифрового контента.

* 1. Верно ли, что сервер приложения умеет работать с протоколом HTTP?

Сервер приложений не умеет работать с протоколом HTTP и обрабатывать пользовательские запросы, так как это задача веб-сервера.

* 1. Что такое CGI?

CGI — стандарт интерфейса, используемого внешней программой для связи с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».

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.

* 1. Как работает система с использованием интерфейс шлюза - CGI?

Для передачи данных используются стандартные потоки ввода-вывода, от веб-сервера к СGI-приложению данные передаются через поток STDIN, принимаются обратно через поток STDOUT, а для передачи сообщений об ошибках используется поток STDERR.

Получив запрос от браузера пользователя, веб-сервер определяет, что запрошено динамическое содержимое и формирует специальный запрос, который через интерфейс CGI передается веб-приложению. При его получении приложение запускается и выполняет запрос, результатом которого служит HTML-код динамически сформированной страницы, который передается назад веб-серверу, после чего приложение завершает свою работу.

* 1. Назовите достоинства и недостатки CGI.
* CGI не налагает особых условий на платформу и веб-сервер, поэтому работает на всех популярных платформах и веб-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.
  1. 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. FastCGI снимает множество ограничений CGI-программ.

* 1. Назовите основные отличия CGI от FastCGI.

В CGI-режиме на каждый запрос создается отдельный процесс, "умирающий" после окончания обработки. В FastCGI процесс работает в качестве демона, то есть один и тот же процесс обрабатывает различные HTTP запросы один за другим. PHP из коробки умеет работать и в FastCGI режиме.

* 1. Что такое менеджер процессов?

Для управления FastCGI-процессами и распределением нагрузки служат менеджеры процессов, они могут быть как частью веб-сервера, так и отдельными приложениями. Популярные веб-сервера Apache и Lighttpd имеют встроенные менеджеры FastCGI-процессов, в то время как Nginx требует для своей работы c FastCGI внешний менеджер.

* 1. Что такое PHP-FPM?

PHP-FPM - это альтернативная реализация FastCGI режима в PHP с несколькими дополнительными возможностями, обычно используемыми для высоконагруженных сайтов.

* 1. Что такое Spawn-fcgi?

Spawn-fcgi является частью проекта, но в состав одноименного веб-сервера не входит, по умолчанию Lighttpd использует собственный, более простой, менеджер процессов. Разработчики рекомендуют использовать его в случаях, когда вам нужно управлять FastCGI-процессами, расположенными на другом хосте, либо требуются расширенные настройки безопасности.

* 1. Что такое Lighttpd?

Веб-сервер, библиотека, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам. Это свободное программное обеспечение, распространяемое по лицензии BSD. lighttpd работает в Linux и других Unix-подобных операционных системах, а также в Microsoft Windows.

* 1. Что такое chroot окружение?

Chroot-окружение – это системный вызов, который временно перемещает root каталог в новую папку. Как правило, root каталог находится в «/». Но при помощи chroot можно задать другой каталог, который будет служить как root-каталог в окружении chroot. Любые приложения, которые запускаются внутри изолированного окружения, в принципе не могут взаимодействовать с остальной операционной системой. Кроме того, не-рутовый пользователь (non-root), помещённый в chroot-окружение, не сможет перемещаться по иерархии каталогов

* 1. Опишите механизм взаимодействия серверов с использованием FastCGI.

FastCGI-процессы используют [Unix Domain Sockets](https://ru.wikipedia.org/wiki/Unix_domain_socket) или [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP) для связи с сервером. Это даёт следующее преимущество перед обычными CGI-программами: FastCGI-программы могут быть запущены не только на этом же сервере, но и где угодно в сети.

* 1. Опишите процесс выбора встроенного или внешнего менеджера процессов.

Выбирая между встроенным менеджером и внешним, нужно оценивать ситуацию и выбирать именно тот инструмент, который наиболее подходит вашим запросам. Например, создавая простой сервер для нескольких сайтов на типовых движках применение внешнего менеджера будет явно излишним. Этот подход хорош тем, что можно собирать то, что нужно для решения задачи прямо из конструктора

* 1. Что такое интерфейс шлюза?

Common Gateway Interface (Общий шлюзовой интерфейс) — этот сетевой стандарт позволяет Web-серверу запускать любую программу и передавать Web-браузеру данные в виде текстовой или двоичной (графической, звуковой) информации.

* 1. 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. Все, что применимо к FastGCI, справедливо и для SCGI.

* 1. Что такое PCGI

PCGI (Perl Common Gateway Interface) - библиотека Perl для работы с интерфейсом CGI, долгое время являлась основным вариантом работы с Perlприложениями через CGI, отличается хорошей производительностью при скромных потребностях в ресурсах и неплохой защиты от перегрузки.

* 1. Что такое PSGI?

PSGI (Perl Web Server Gateway Interface) − технология взаимодействия вебсервера и сервера приложений для Perl. Если PCGI представляет собой инструмент для работы с классическим CGI-интерфейсом, то PSGI более напоминает FastCGI. PSGI-сервер представляет среду для выполнения Perl-приложений которая постоянно запущена в качестве службы и может взаимодействовать с вебсервером через TCP/IP или UNIХ-сокеты и предоставляет Perl-приложениям те же преимущества, что и FastCGI.

* 1. Что такое WSGI?

WSGI (Web Server Gateway Interface) − предназначен для взаимодействия 87 веб-сервера с сервером приложений для программ, написанных на языке Python.

* 1. Опишите механизм взаимодействия серверов Apache и PHP.

Существует 3 способа вызова PHP из Apache:

1. как модуль
2. CGI
3. FastCGI
   1. Опишите преимущества веб-сервера Apache.

* Бесплатное программное обеспечение как для личного использования, так и для коммерческого;
* гибкость программы. Открытый исходный код позволяет подстроить программу под специфические нужды пользователя;
* поддержка многих языков программирования (PHP, Python, Ruby, Perl, ASP, Tcl), что дает доступ к различным серверным ПО;
* удобная и надежная конфигурация системы. Приложение легко настраивается через текстовые конфигурационные файлы и может быстро масштабироваться;
* Apache хорошо работает со многими системами управления контентом (CMS), такими как WordPress, Joomla, Drupal;
* поддерживает технологии FastCGI и CGI;
* может использовать СУБД для аутентификации пользователей;
* большая база инструкций. Так как продукт уже давно на рынке, разработчики и пользователи уже написали множество инструкций под разные ситуации.
  1. Опишите недостатки веб-сервера Apache.
* При высоком трафике снижается производительность. На высокопосещаемых сайтах может значительно снижаться скорость работы программы. Это связано с особенностью работы сервиса. Каждый пользовательский запрос обрабатывается отдельным потоком. Когда потоки заканчиваются, остальные операции ждут своей очереди. Если планируется большое количество посещений сайта, то лучше использовать Nginx;
* много параметров настройки. Если допустить ошибку при настройке конфигурации, это может привести к появлению уязвимостей во всей системе;
* риск заражения вирусом. Создавать модули могут не только добросовестные разработчики, но и злоумышленники. В модуль может быть внедрён вредоносный код. Устанавливать модули стоит только от проверенных разработчиков.
  1. Опишите архитектуру веб-сервера Apache.

Apache состоит из ядра и модулей. Ядро выполняет основные функции: обработка конфигурационных файлов, работа с протоколом HTTP, система загрузки модулей. Оно может работать самостоятельно без модулей, но тогда функционал программы будет крайне ограничен. Ядро создала команда компании Apache Software Foundation без вмешательства сторонних разработчиков.

* 1. Опишите функции ядра веб-сервера Apache.

Ядро HTTP сервера Apache разрабатывается фондом Apache Software Foundation, который поддерживает огромное количество разработчиков по всему миру. Его основными функциями являются:

1. Передача данных по HTTP.

2. Обработка файлов.

3. Загрузка и поддержка модулей.

Сервер может функционировать без дополнительных модулей, однако в этом случае, его возможности крайне ограничены.

* 1. Опишите конфигурацию веб-сервера Apache.

Система конфигурации web-сервера Apache состоит из 3-х уровней текстовых файлов.

httpd.conf — основной конфигурационный файл. В нём содержатся директивы, которые управляют работой веб-сервера. Местоположение файла зависит от операционной системы. Узнать, где он находится, можно с помощью команды:

.htaccess — дополнительный конфигурационный файл. Файл (или файлы) .htaccess задают правила работы веб-сервера в каталоге, в котором он размещен, и в его дочерних каталогах, то есть он не затрагивает глобальные изменения работы сервиса. Настройки .htaccess приоритетны перед настройками в httpd.conf. При каждом обращении к веб-серверу Apache просматривает все каталоги на наличие .htaccess.

Виртуальный хост. Он используется для запуска нескольких доменов на одном IP-адресе. Такие хосты нужны пользователям, которые запускают несколько сайтов на одном виртуальном сервере. На один сервер можно добавить неограниченное количество виртуальных хостов

* 1. Что такое URI, URL и чем они различаются.

URI – имя и адрес ресурса в сети, включает в себя URL и URN. URL – адрес ресурса в сети, определяет местонахождение и способ обращения к нему URN – имя ресурса в сети, определяет только название ресурса, но не говорит как к нему подключиться.

1. Ссылка на удаленный репозиторий проекта

Полный код проекта можно найти по ссылке:

https://github.com/Ivanbessmertnykh/Data-Servers

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Разработка серверных частей интернет-ресурсов [Электронный ресурс].

URL:<https://online-edu.mirea.ru/pluginfile.php?file=%2F980342%2Fmod_resource%2Fcontent%2F0%2Fpdf24_merged.pdf>

Дата обращения 15.10.2022

1. Руководство по PHP

URL: <https://www.php.net/manual/ru/index.php>

Дата обращения 15.10.2022

1. 3-я лекция по предмету

Дата проведения 08.10.2022