|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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: Настройка сервера по заданным**

**характеристикам, размещение проекта**

**Студент группы** ИКБО-30-20 Патина В.С.

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

(подпись студента)

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

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

(подпись руководителя)

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

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

Москва 2022

**СОДЕРЖАНИЕ**

[Постановка задачи 1](#_Toc113622084)

[Выполнение работы 1](#_Toc113622085)

[Ссылка на удаленный репозиторий 8](#_Toc113622086)

[Ответы на вопросы к практической: 9](#_Toc113622087)

[ВЫВОДЫ 15](#_Toc113622088)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 16](#_Toc113622089)

Постановка задачи

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 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе №1. В этой конфигурации предполагается акселерированное проксирование без кэширования.

Выполнение работы

Ссылка на удаленный репозиторий

<https://github.com/devilishgrapefruit/RSCIR/tree/main/prak2> (Рисунок 15).

Рисунок 15 - Репозиторий

Ответы на вопросы к практической:

1. **Что такое веб-сервер?**

Веб-сервер в процессе своей работы обрабатывает запросы браузера и выдает в ответ вебстраницы. Работа веб-сервера сводится к приему HTTP-запросы от веббраузеров, выступающих в качестве и выдачи им в ответ соответствующих ответов (как правило это HTTP-ответы, включая HTML-страницы) а также файлы изображений, медиа-потоки, файлы других типов и различные данными. Важно заметить, что, по сути, веб-сервер — это информационная система, так как его функционирование невозможно как без специального программного обеспечения, так и соответствующей аппаратной части, т.е. компьютера.

1. **Что такое сервер приложения и чем он отличается от веб-сервера?**
2. **Кратко опишите историю развития интернета в рамках развития веб-серверов.**
3. **Кратко опишите протокол HTTP.**

Для доступа к HTML документам посредством сети используется протокол HTTP (HyperText Transfer Protocol. Протокол HTTP реализован по клиент-серверной технологии и работает по принципу запрос-ответ без сохранения состояния.

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

В процессе работы HTTP-сервера получает запрос клиента, обрабатывает его и либо выдает ему запрашиваемый ресурс, либо сообщает, что это сделать невозможно.

Пользователь посредством HTTP-клиента (чаще всего это браузер) запрашивает у HTTP-сервера некий URL, сервер проверяет и отдает соответствующий этому URL-файл, обычно это HTML-страница. Запрашиваемый документ (в нашем примере это HTML-страница) может, в свою очередь, ссылаться на связанные ресурсы, например, изображения, файлы данных, документы. Для отображения таких связных ресурсов на странице, клиентом (браузером) последовательно запрашиваются у сервера все связные документы. Помимо изображений клиент может запрашивать таблицы стилей, скрипты, исполняемые на стороне клиента и т.д. После получения всех запрашиваемых ресурсов клиент (браузер) обработает их в соответствии с кодом HTML-документа и выдаст пользователю готовую страницу.

1. **Опишите цели и задачи веб-сервера.**

Основные цели и задачи веб-сервера сводятся к обработке HTTP-запросов клиента и возвращении пользователю результатов этой обработки. Заметим, что Веб-сервер не способен к самостоятельной генерации контента, он может обрабатывать только статическое содержимое.

1. **Опишите технологию SSI.**

SSI (от английского Server Side Includes – включение на стороне сервера) – это, как уже было сказано ранее, язык, разработанный для динамического создания и «сборки» веб-страниц на сервере из отдельных составных частей и выдачи клиенту полученного HTML-документа. Использование языка SSI дает возможность в код страницы инкапсулировать содержимое различных файлов. В этом случае, внесение дополнительных повторяющихся элементов, такие, например, как шапка (header), подвал (footer), меню и другие осуществляется в специальные обособленные файлы. Затем созданные таким образом файлы просто подключаются при окончательной сборке страницы.

1. **Что такое система управления контентом?**

Бурное развитие интернет-технологий и связанные с ним запросы потребителей требовали новых решений. Обработка запросов пользователей на стороне сервера стала возможно с появлением специального класса программ – СКРИПТОВ. Такие программы, как правило, создаются с использованием специальных скриптовых языков, первоначально в качестве такого языка использовался Perl, поскольку он создавался лингвистом Ларри Уолли и обладает богатыми возможностями для работы с текстом. Затем был разработан PHP, на который Perl, кстати, оказал существенное влияние. Дальнейшее увеличение объемов работ и требований к оперативности их выполнения привели к созданию особого класса программ, можно даже сказать – систем. Такие системы в настоящее время прочно заняли свое место в арсенале веб-мастера и называются CMS системы. CMS (от английского Content management system) − системы управления контентом, в виде полноценного веб-приложения. CMS предназначены для обеспечения динамической обработки запросов пользователя.

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

**HTTP?**

Сервер приложений не работает с протоколом HTTP и не обрабатывает пользовательские запросы. Все эти действия по-прежнему выполняет веб-сервер.

1. **Что такое CGI?**

CGI (от английского Common Gateway Interface) или интерфейс общего шлюза, представляет собой стандарт интерфейса, используемого внешней программой для связи с веб-сервером.

1. **Как работает система с использованием интерфейс шлюза - CGI?**
2. **Назовите достоинства и недостатки CGI.**

К достоинствам CGI можно отнести языковую и архитектурную независимость: CGI-приложение может быть написано на любом языке и одинаково хорошо работать с любым веб-сервером. Учитывая простоту и открытость стандарта, это привело к активному развитию веб-приложений.

Однако, кроме достоинств, CGI обладает и существенными недостатками. Основной из них − высокие накладные расходы на запуск и остановку процесса, что влечет за собой повышенные требования к аппаратным ресурсам и невысокую производительность. Использование стандартных потоков вводавывода ограничивает возможности масштабирования и обеспечения высокой доступности, так как требует, чтобы веб-сервер и сервер приложений находились в пределах одной операционной системы (ОС).

1. **Что такое FastCGI?**

FastCGI представляет собой клиент-серверный протокол для взаимодействия веб-сервера и сервера приложений, обеспечивающий высокую производительность и безопасность.

риложений, обеспечивающий высокую производительность и безопасность. FastCGI устраняет основную проблему CGI − повторный запуск процесса веб-приложения на каждый запрос, FastCGI процессы запущены постоянно, что позволяет существенно экономить время и ресурсы

1. **Назовите основные отличия CGI от FastCGI.**
2. **Что такое менеджер процессов?**
3. **Что такое PHP-FPM?**
4. **Что такое Spawn-fcgi?**
5. **Что такое Lighttpd?**
6. **Что такое chroot окружение?**
7. **Опишите механизм взаимодействия серверов с использованием FastCGI.**
8. **Опишите процесс выбора встроенного или внешнего менеджера процессов.**
9. **Что такое интерфейс шлюза?**
10. **Что такое SCGI?**
11. **Что такое PCGI**
12. **Что такое PSGI?**
13. **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.**
2. **Опишите преимущества веб-сервера Apache.**

К преимуществам HTTP сервера Apache, необходимо отнести высокий уровень надежности, гибкие настройки, свободный доступ к программе. В частности, к нему можно подключать большое количество внешних модулей, систем управления базами данных и т.п. Apache поддерживает интернетпротокол IPv6. Также к достоинствам Apache, необходимо отнести регулярно выпускаемые обновления и патчи (заплатки), которые позволяют быстро устранить различные проблемы с работой или безопасностью. Удобство и легкость настройки этого программного обеспечения, делают его одним из самых популярных вариантов как для начинающих, так и для опытных веб-мастеров.

1. **Опишите недостатки веб-сервера Apache.**

Все недостатки Apache, являются логическим продолжением его достоинств. Наиболее вескими недостатками являются: 1. Возможные проблемы с производительностью на высоконагруженных сайтах с большим трафиком, «тяжелым» контентом или приложениями, которые требуют высоких вычислительных мощностей. 2. Большое количество параметров настройки может привести к возникновению уязвимостей, из-за невнимательности при конфигурации. 3. Существует вероятность того, что в модули от независимых разработчиков будет внедрен вредоносный код.

1. **Опишите архитектуру веб-сервера Apache.**

В архитектуру Apache входит: простое ядро, платформо-зависимый уровень (APR), и модули. Любое приложение для Apache - даже простейшее, обслуживающее "дефолтную" страницу Apache "It worked" - использует несколько модулей. Пользователи Apache не нуждаются в знании этого, но для разработчика программ, понимание модулей и API модуля Apache является необходимым знанием при работе с Apache. Большинство, но не все модули, связаны с различными аспектами обработки НТТР запроса. Преимущество модульного подхода состоит в том, что он позволяет фокусировать модуль на специфическую задачу, игнорируя при этом другие аспекты НТТР, не касающиеся данной задачи.

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.**

Конфигурацию Apache, можно разделить на три основных уровня: 1. Конфигурация сервера. 2. Конфигурация виртуального хоста. 3. Конфигурация уровней каталога. Все настройки осуществляются при помощи специальных текстовых файлов, со своим собственным языком, который основан на блоках директив. С их помощью, могут быть изменены практически все параметры ядра. Большинство модулей, имеет свои собственные параметры. Также они достаточно часто используют для своей работы настроечные файлы операционных систем. Ряд настроек можно задавать с помощью параметров командной строки.

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

ВЫВОДЫ

В процессе выполнения практической работы мы создали проект с тремя сервисами: для кодирования и рисования svg-фигур, для сортировки и вывода массива, для вывода результата выполнения команд ls, whoami, id, ps. Для проверки работоспособности серверной конфигурации мы сгенерировали тестовые страницы с необходимыми данными.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Разработка серверных частей интернет-ресурсов [Электронный ресурс]: Учебное пособие / Волков М.Ю., Литвинов В.В., Лобанов А.А. и др. — М.: МИРЭА – Российский технологический университет, 2021.
2. VG элементы: rect, circle, ellipse, line, polyline, polygon, path URL: <http://shpargalkablog.ru/2015/11/svg.html> (Дата обращения 18.09.2022)
3. Create dynamic SVG with PHP URL: https://daext.com/blog/create-dynamic-svg-with-php/ (Дата обращения 18.09.2022)