

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики
Кафедра вычислительной техники и электроники (ВТиЭ)

Отчёт по производственной эксплуатационной практике

Выполнил студент 595 гр.:

_____ Осипов А.В.

Проверил: проф. д.т.н. каф. ВТиЭ

_____ Белозерских В.В.

Оценка _____

«_____» _____ 2022г.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	3
1.1. Общие цели производственной эксплуатационной практики.....	3
1.2. Постановка задачи производственной эксплуатационной практики.....	3
2. ОБЩИЕ СВЕДЕНИЯ.....	4
2.1. Постановка задачи.....	4
2.2. Описание выполненных работ.....	4
3. ЗАКЛЮЧЕНИЕ	7

1. ВВЕДЕНИЕ

1.1. Общие цели производственной эксплуатационной практики

Целью проведения практики является приобретение студентами универсальных и профессиональных навыков, практического опыта, закрепление, систематизация и расширение теоретических знаний по использованию, администрированию, настройке и наладке программно-аппаратными комплексов. Производственная эксплуатационная практика ориентирована на формирование у бакалавров универсальных и профессиональных компетенций.

1.2. Постановка задачи производственной эксплуатационной практики

В ходе выполнения практики обучающийся приобретает навыки научно-исследовательской деятельности и решает одну или несколько профессиональных задач:

- изучение научно-технической информации, отечественного и зарубежного опыта по тематике исследования;
- математическое моделирование процессов и объектов на базе стандартных пакетов автоматизированного проектирования и исследований;
- проведение экспериментов по заданной методике и анализ результатов;
- проведение измерений и наблюдений, составление описания проводимых исследований, подготовка данных для составления обзоров, отчетов и научных публикаций;
- составление отчета по выполненному заданию, участие во внедрении результатов исследований и разработок.

Время прохождения практики: 16.05.–11.06.2022.

2. ОБЩИЕ СВЕДЕНИЯ

Производственная эксплуатационная практика проходила в АО «БСКБ «Восток». Руководителям практики являлись Белозерских Василий Вениаминович и Кошманов Дмитрий Сергеевич.

2.1. Постановка задачи

Создать простую тестовую базу данных на PostgreSQL из двух связанных таблиц. На C++ реализовать сервер и клиент (для клиента использовать Qt). Клиент должен получать данные из БД от сервера по стандарту REST API. Для реализации API найти библиотеки с открытым исходным кодом и активной поддержкой со стороны сообщества разработчиков. Результаты работы продемонстрировать в виде рабочей информационной системы и отчёта руководителю практики от предприятия.

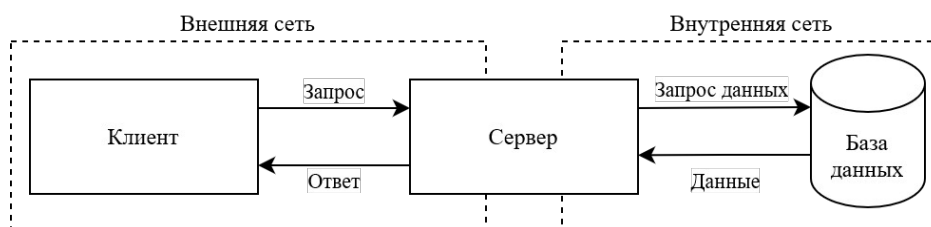


Рис. 1. Схема обмена данными в информационной системе

В рамках решения данной задачи мне было отведена её часть по созданию сервера и настройке базы данных. Реализацией клиента занимался другой студент 595 группы Лаптев А.В.

2.2. Описание выполненных работ

В качестве операционной системы, на которой работает сервер, была установлена отечественная операционная система AstraLinux «Орёл», один из дистрибутивов GNU/Linux. Для компиляции серверного приложения, написанного на C++, используется стандартное для этой платформы средство – компилятор G++ из набора GNU Compiler Collection. Поскольку на сервере была установлена версия AstraLinux без графической оболочки, в качестве редактора кода пришлось использовать консольную версию программы Emacs, которая имеет встроенную поддержку синтаксиса C++.

В качестве системы управления базой данных (СУБД) в задаче была обозначена PostgreSQL, которая входит в комплект программ «СУБД» в установщике AstraLinux. В ней была создана база данных из 3 таблиц (две основных и одна вспомогательная), хранящие в себе список студентов, список групп и записи о посещаемости студентами

занятий. Таблица студентов и таблица посещаемости связаны общим числовым ключом, идентифицирующим студента в обеих таблицах. При этом идентификатор студента в таблице посещаемости является частью её составного ключа вместе с датой и временем занятия, к которому относится отметка посещаемости.

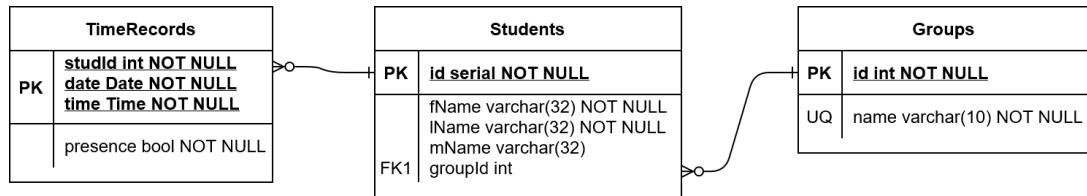


Рис 2. Модель данных, хранимых в базе данных информационной системы

Суть работы сервера заключается в получении запросов от клиента и преобразование их в запросы для базы данных, а также кэширование часто запрашиваемых данных. В ответ сервер присылает клиенту запрашиваемые данные или отчёт о статусе запросов на изменение данных в БД. API сервера, через который происходит обмен запросами и данными с клиентом, должен быть спроектирован согласно принципам REST: клиент-серверная архитектура, отсутствие хранения состояния клиента на сервере, кэширование данных, единообразие доступа к ресурсам сервера и слоистая архитектура. В качестве протокола обмена данными между клиентом и сервером используется HTTP. Передаваемая по этому протоколу информация хранится в текстовом формате JSON, а также в URI-параметрах запроса (для запросов на получение данных).

Для реализации всего необходимо для работы сервера функционала (обработка HTTP-запросов, чтение/запись JSON, работа с БД) была использована многоцелевая библиотека Росо для C++. Данная библиотека состоит из множества модулей, каждый из которых решает одну из обозначенных ранее задач. Данная библиотека отсутствовала в официальном репозитории AstraLinux, поэтому она была собрана из открытого исходного кода в виде динамически подключаемых библиотек для Linux. На функциях Росо завязана значительная часть работы сервера: от асинхронной обработки поступающих HTTP-запросов до системы динамической типизации, использующей при работе с данными из JSON.

Созданный сервер обладает следующими возможностями для работы с БД:

- получение информации о группах студентов в БД и списка студентов в каждой из групп или во всех группах;
- получение записей о посещаемости согласно заданным параметрам отбора (фильтрам) по группе, дате и времени занятия и текущей отметки посещаемости;

- возможность добавлять, изменять и удалять записи о посещаемости и данных студентов в базе данных через POST-запросы;
- кэширование недавно запрашиваемых данных с целью ускорения доступа к ним и защиты от возможных кратковременных сбоев связи с БД;
- возможность одновременного внесения изменения сразу в несколько записей БД за один запрос клиента (параметры запроса подаются в виде массива).

В итоге за 3 недели разработки серверного приложения (от алгоритма его работы до написания и компиляции кода) и 1 неделю его тестирования и документирования была создана часть программного комплекса, которая вместе с клиентом и СУБД формирует информационную систему для контроля посещаемости студентов учебного заведения. Задействованные в разработке архитектурные и технические решения обеспечивают эффективную работу данной системы и позволяют использовать разные виды клиентов, не ограничиваясь тем клиентом, который был разработан в рамках совместного с Лаптевым А.В. решения поставленной задачи.

3. ЗАКЛЮЧЕНИЕ

В ходе производственной эксплуатационной практики была разработана серверная программа для работы с базой данных на базе PostgreSQL и передачи полученной из неё данных клиенту в рамках работы сетевой информационной системы.

Для достижения поставленной цели надо было решить задачи:

1. Определение целевого назначения создаваемой информационной системы
2. Выбор среды разработки и запуска серверной программы
3. Выбор сетевого протокола и библиотек для реализации API сервера
4. Разработка алгоритма работы сервера
5. Настройка базы данных и написание программы
6. Соединение сервера с клиентом и проверка их взаимодействия

Созданная серверная программа обеспечивает быстрый и при этом безопасный доступ к функциям базы данных, предоставляя понятный для использования в клиентских приложениях API. Это позволяет в дальнейшем легко создать на базе любой платформы клиент ИС, адаптированный под нужды пользователя. Использование широко распространённого протокола HTTP и текстового формата обмена данными JSON позволяют делать запросы даже из веб-браузера и общих утилит для проверки веб-ресурсов, по типу curl.

Таким образом, поставленные передо мной задачи были выполнены в срок и в полном объеме. Все необходимые компетенции в области разработки ПО для сетевого оборудования были освоены. Итоговая система, включая реализованный Лаптевым А.В. клиент, может быть использована с некоторыми изменениями для простого контроля посещаемости в учебных заведениях.