

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №3**  
по дисциплине «Низкоуровневое программирование»  
Вариант Protobuf

Выполнил:  
Студент группы Р33312  
Искандаров Ш.Х.

Преподаватель:  
Кореньков Юрий Дмитриевич

г. Санкт-Петербург  
2023

## Задание

На базе данного транспортного формата описать схему протокола обмена информацией и воспользоваться существующей библиотекой по выбору для реализации модуля, обеспечивающего его функционирование. Протокол должен включать представление информации о командах создания, выборки, модификации и удаления данных в соответствии с данной формой, и результатах их выполнения. Используя созданные в результате выполнения заданий модули, разработать в виде консольного приложения две программы: клиентскую и серверную части. Серверная часть – получающая по сети запросы и операции описанного формата и последовательно выполняющая их над файлом данных с помощью модуля из первого задания. Имя файла данных для работы получать с аргументами командной строки, создавать новый в случае его отсутствия. Клиентская часть – в цикле получающая на стандартный ввод текст команд, извлекающая из него информацию о запрашиваемой операции с помощью модуля из второго задания и пересылающая её на сервер с помощью модуля для обмена информацией, получающая ответ и выводорящая его в человеко-понятном виде в стандартный вывод.

Порядок выполнения:

1. Изучить выбранную библиотеку
  - a. Библиотека должна обеспечивать сериализацию и десериализацию с валидацией в соответствии со схемой
  - b. Предпочтителен выбор библиотек, поддерживающих кодогенерацию на основе схемы
  - c. Библиотека может поддерживать передачу данных посредством TCP соединения
    - i. Иначе, использовать сетевые сокеты посредством API ОС
  - d. Библиотека может обеспечивать диспетчеризацию удалённых вызовов
    - i. Иначе, реализовать диспетчеризацию вызовов на основе информации о виде команды
2. На основе существующей библиотеки реализовать модуль, обеспечивающий взаимодействие
  - a. Описать схему протокола в поддерживаемом библиотекой формате
    - i. Описание должно включать информацию о командах, их аргументах и результатах
    - ii. Схема может включать дополнительные сущности (например, для итератора)
  - b. Подключить библиотеку к проекту и сформировать публичный интерфейс модуля с использованием встроенных или сгенерированных структур данных используемой библиотеки

- i. Поддержать установление соединения, отправку команд и получение их результатов
    - ii. Поддержать приём входящих соединений, приём команд и отправку их результатов
  - c. Реализовать публичный интерфейс посредством библиотеки в соответствии с п1
- 3. Реализовать серверную часть в виде консольного приложения
  - a. В качестве аргументов командной строки приложение принимает:
    - i. Адрес локальной конечной точки для прослушивания входящих соединений
    - ii. Имя файла данных, который необходимо открыть, если он существует, иначе создать
  - b. Работает с файлом данных посредством модуля из задания 1
  - c. Принимает входящие соединения и взаимодействует с клиентами посредством модуля из п2
  - d. Поступающая информация о запрашиваемых операциях преобразуется из структур данных модуля взаимодействия к структурам данных модуля управления данными и наоборот
- 4. Реализовать клиентскую часть в виде консольного приложения
  - a. В качестве аргументов командной строки приложение принимает адрес конечной точки для подключения
  - b. Подключается к серверу и взаимодействует с ним посредством модуля из п2
  - c. Читает со стандартного ввода текст команд и анализирует их посредством модуля из задания 2
  - d. Преобразует результат разбора команды к структурам данных модуля из п2, передаёт их для обработки на сервер, возвращаемые результаты выводит в стандартный поток вывода
- 5. Результаты тестирования представить в виде отчёта, в который включить:
  - a. В части 3 привести пример сеанса работы разработанных программ
  - b. В части 4 описать решение, реализованное в соответствии с пп.2-4
  - c. В часть 5 включить составленную схему п.2а

## Решение

С использованием модуля из лабораторной работы 2 клиентское приложение анализирует пользовательский ввод. В этом процессе формируется абстрактное синтаксическое дерево (AST). Затем происходит преобразование AST в структуры данных, описанные в файлах proto. Это осуществляется путем обхода дерева в глубину, где составляется запрос, а затем используются сгенерированные методы protobuf-c для представления запроса в бинарном формате.

Необходимые структуры для взаимодействия между клиентом и сервером расположены в директории proto. Для обмена данными между клиентом и сервером используется модуль net. В этом модуле представлены команды для работы с сокетами: открытие, установление соединения, прием и отправка сообщений, закрытие сокета.

Протокол protobuf используется для передачи данных в бинарном формате. После приема сообщения сгенерированные библиотекой методы применяются для преобразования бинарных данных в C-структуры.

На сервере происходит дальнейшее преобразование полученных данных в элементы модуля базы данных, который был разработан в первой лабораторной работе.

Определяется тип операции и используется интерфейс для выполнения необходимой операции.

## Схема данных

### Request:

```
syntax = "proto2";

import "operation.proto";
import "filter.proto";
import "entity.proto";
import "node.proto";
import "relationship.proto";
import "property.proto";
import "typeelement.proto";
import "iterator.proto";

message RequestMessage {
    required OperationProto operation = 1;
    required TypeOfElementProto type = 2;
    optional FilterMessage filter = 3;

    optional EntityMessage entity = 4;
    optional NodeMessage node = 5;
    optional RelationshipMessage relationship = 6;
    optional PropertyMessage property = 7;

    repeated string relationships = 8;
    repeated string properties = 9;

    optional IteratorMessage iterator = 10;
}
```

Response:

```
syntax = "proto2";

import "iterator.proto";
import "noderesponce.proto";

message ResponceMessage {
    required int32 status = 1;
    required string message = 2;
    optional NodeResponse node = 3;
    optional IteratorMessage iterator = 4;
}
```

Пример работы:

Запуск сервера:

```
> ./llp 8080 db
Database created in file - db
Server started on port - 8080
```

Необходимо было добавить данные из датасета neo4j в свое приложение. При запуске клиента указывается файл, в котором находятся запросы для добавления данных.

[illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible]

```
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
```

```
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
```

```
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
```

```
Response - Status: 200, Message: Successfully executed operation for entity
Response - Status: 200, Message: Successfully executed operation for entity
```

```
Response - Status: 200 Message: Successfully executed operation for entity
```

## Демонстрация работы операции выборки

- Запрос на сервер

```
{ }
query SelectNode(i: $i) {
  selectNode(i: $i) {
    id(id: 0)
    type(type: tweet)
    relationships {
      reply_to
      mentioned
    }
    properties {
      text
      created_at
    }
  }
  filter: {
    created_at="2021-03-13T17:23:42Z"
  }
}
;
```

- Ответ с сервера

```
Response: {
  status: 201,
  message: Find element,
  node: {
    type: tweet,
    id: 0,
    name: 395
    relationships: [
    ],
    properties: [
      {
        value_type: 2,
        type: text,
        value: @asynchio @anshublog @neo4j @emileifrem Yup. I have my own list. Just curious what others think ...
      }
      ,
      {
        value_type: 2,
        type: created_at,
        value:
      }
    ]
  }
}
```



- Запрос в базу данных neo4j и ответ

```
MATCH (n:Tweet)-[:REPLY_TO]-(r)
WHERE n.created_at = datetime("2021-03-13T17:23:42Z")
RETURN ID(n), n.text, n.created_at, ID(r)
```

	ID(n)	n.text	n.created_at	ID(r)
1	395	"@asynchio @anshublog @neo4j @emileifrem Yup. I have my own list. Just curious what others think ..."	"2021-03-13T17:23:42Z"	397

## Итог

В процессе выполнения лабораторной работы был освоен протокол Protocol Buffers, а также изучена библиотека protobuf-c для взаимодействия с этим протоколом на языке программирования С. Были созданы клиентская и серверная модели, предназначенные для анализа запросов пользователя и их обработки в базе данных.