

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
«Низкоуровневое программирование»
Вариант 2

Выполнил:
Студент группы Р33102
Гулямов Т.И.

Преподаватель:
Кореньков Ю.Д.

Санкт-Петербург
2023

Цель:

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.

Задачи:

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 выводящая его в человеко-понятном виде в стандартный вывод.

Описание работы:

Сборка и запуск:

```
git clone git@github.com:tplaymeow/itmo-low-level-programming-lab3.git
git submodule init
git submodule update
```

Для сборки и запуска необходимо использовать Cmake.

Серверная программа

```
cmake -B build -DCMAKE_BUILD_TYPE=Release
cmake --build build
build/server_app/server_app file.db 2002
```

Клиентская программа

```
cmake -B build -DCMAKE_BUILD_TYPE=Release
cmake --build build
build/client_app/client_app 127.0.0.1 2002
```

Пример работы:

Клиент:

```
create table ATable(a_text text, a_float float, b_bool bool);
create table BTable(b_text text, b_integer integer, b_float float);
insert into ATable("T1",1.0,false);
```

```

insert into ATable("T2",2.0,true);
insert into ATable("T3",3.0,false);
insert into BTable("T1",1,1.0);
insert into BTable("T2",2,2.0);
insert into BTable("T3",3,3.0);
select from ATable;
a_text      a_float      b_bool
T3           3.000000    false
T2           2.000000    true
T1           1.000000    false
select from BTable;
b_text      b_integer    b_float
T3           3           3.000000
T2           2           2.000000
T1           1           1.000000
select from ATable
join BTable on a_text == b_text
where a_text contains "1" or b_float > 2.0;
a_text      a_float      b_bool      b_text      b_integer
b_float
T3           3.000000    false      T3           3
3.000000
T1           1.000000    false      T1           1
1.000000

```

server:

Server started at port 2029

Cxema:

filter:

```

{
  "$schema":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter",
  "type": "object",
  "properties": {
    "comparison": {
      "type": "object",
      "properties": {
        "operator": {
          "type": "string"
        }
      }
    },
    "left": {
      "type": "object",
      "properties": {
        "column": {
          "type": "string"
        }
      }
    }
  }
}

```

```

        "literal": {
          "type": "object",
          "properties": {
            "type": {
              "type": "string"
            },
            "value": {
              "type": ["number", "string", "boolean"]
            }
          }
        },
      },
      "required": [],
    },
    "right": {
      "type": "object",
      "properties": {
        "column": {
          "type": "string"
        },
        "literal": {
          "type": "object",
          "properties": {
            "type": {
              "type": "string"
            },
            "value": {
              "type": ["number", "string", "boolean"]
            }
          }
        }
      },
      "required": []
    },
    "required": ["operator", "left", "right"]
  },
  "contains": {
    "type": "object",
    "properties": {
      "operator": {
        "type": "string"
      },
      "left": {
        "type": "object",
        "properties": {
          "column": {
            "type": "string"
          },
          "literal": {
            "type": "object",
            "properties": {
              "type": {
                "type": "string"
              },
              "value": {

```

```

        "type":["string"]
      }
    }
  },
  "required": [],
},
"right": {
  "type": "object",
  "properties": {
    "column": {
      "type": "string"
    },
    "literal": {
      "type": "object",
      "properties": {
        "type": {
          "type": "string"
        },
        "value": {
          "type":["string"]
        }
      }
    }
  }
},
"required": []
},
"required": ["operator", "left", "right"]
},
"logic": {
  "type": "object",
  "properties": {
    "operator": {
      "type": "string"
    },
    "left": {
      "$ref":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter"
    },
    "right": {
      "$ref":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter"
    }
  },
  "required": ["operator", "left", "right"]
}
},
"required": []
}

```

request:

```

{
  "$schema":

```

```
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/re
quest",
  "type": "object",
  "properties": {
    "drop": {
      "type": "object",
      "properties": {
        "table_name": {
          "type": "string"
        }
      },
    },
    "required": [
      "table_name"
    ]
  },
  "create": {
    "type": "object",
    "properties": {
      "table_name": {
        "type": "string"
      },
    },
    "columns": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "type": {
            "type": "string"
          }
        }
      },
    },
    "required": [
      "name",
      "type"
    ]
  },
  "required": [
    "table_name",
    "columns"
  ],
  "update": {
```

```

    "type": "object",
    "properties": {
      "table_name": {
        "type": "string"
      },
      "filter": {
        "$ref":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter"
      },
      "set": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "name": {
              "type": "string"
            },
            "literal": {
              "type": "object",
              "properties": {
                "type": {
                  "type": "string"
                },
                "value": {
                  "type": "number"
                }
              }
            },
            "required": [
              "type",
              "value"
            ]
          }
        },
        "required": [
          "name",
          "literal"
        ]
      }
    },
    "required": ["table_name"]
  },
  "delete": {
    "type": "object",
    "properties": {

```

```

        "table_name": {
            "type": "string"
        },
        "filter": {
            "$ref":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter"
        }
    },
    "required": ["table_name"]
},
"insert": {
    "type": "object",
    "properties": {
        "table_name": {
            "type": "string"
        },
        "values": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "type": {
                        "type": "string"
                    },
                    "value": {}
                }
            },
            "required": [
                "type",
                "value"
            ]
        }
    }
},
"required": ["table_name", "values"]
},
"select": {
    "type": "object",
    "properties": {
        "table_name": {
            "type": "string"
        },
        "filter": {
            "$ref":
"https://github.com/tplaymeow/itmo-low-level-programming-lab3/schemes/filter"

```



```

    },
    "join": {
      "type": "object",
      "properties": {
        "join_table": {
          "type": "string"
        },
        "table_column": {
          "type": "string"
        },
        "join_table_column": {
          "type": "string"
        }
      },
      "required": [
        "join_table",
        "table_column",
        "join_table_column"
      ]
    },
    "required": ["table_name"]
  },
  "required": []
}

```

Детали реализации:

Модели данных:

Для обмена информацией между клиентом и сервером используются общие модели данных описанные в [models.h](#) и [models.c](#).

Клиентское приложение:

- Парсит пользовательский ввод, преобразуя его в общие модели данных
- Производит преобразование этих моделей в JSON
- Отправляет на сервер

Серверное приложение:

- Декодирует и валидирует JSON, преобразуя его в общие модели данных
- Производит преобразование этих моделей в модели для работы с БД, разработанные в первой лабораторной работе
- Производит операции с БД
- Преобразуют ответ в общие модели
- Производит преобразование этих моделей в JSON
- Отправляет ответ на сервер

Преобразование в JSON:

Для работы с JSON была выбрана популярная библиотека cJSON, которая поддерживает преобразование в JSON и обратно. Работа с библиотекой описана в файлах [models_serialization.h](#) и [models_serialization.c](#).

Передача данных:

Для передачи данных используются системные API для TCP сокетов. Реализация находится в файлах [connection.h](#) и [connection.c](#).

Для каждого запроса:

- Открывается сокет
- Клиент отправляет хэдер с размером сообщения с запросом в бинарном формате
- Клиент отправляет текстовое сообщение с запросом
- Сервер обрабатывает запрос
- Сервер отправляет хэдер с размером сообщения с ответом в бинарном формате
- Сервер отправляет текстовое сообщение с ответом
- Сокет закрывается

Структура хедера

```
struct message_header {  
    int32_t content_length;  
};
```

Вывод:

Во время выполнения лабораторной работы, познакомился с работой с JSON на C, поработал с сокетами, значительно улучшил созданный в лабораторной работе 1 и лабораторной работе 2 функционал. Получил огромный опыт в интеграции отдельных систем и отладке.