

Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.01 Информатика и вычислительная техника

Лабораторная работа №3
по дисциплине «Низкоуровневое программирование»
Вариант №1 (XML)

Выполнил:
Студент группы Р33302
Иванов Н.Д.

Преподаватель:
Кореньков Юрий Дмитриевич

г. Санкт-Петербург
2023

Содержание

Цели.....	3
Задачи.....	4
Описание работы:.....	5
Выводы:.....	16

Цели

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 выводящая его в человеко-понятном виде в стандартный вывод.

Задачи

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
 - Иначе, использовать сетевые сокеты посредством API ОС
 - d. Библиотека может обеспечивать диспетчеризацию удалённых вызовов
 - 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
 - Описание должно включать информацию о командах, их аргументах и результатах
 - 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
 - Поддерживать установление соединения, отправку команд и получение их результатов
 - Поддерживать 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:
 - Адрес локальной конечной точки для прослушивания входящих соединений
 - 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:
 - d. В части 3 привести пример сеанса работы разработанных программ
 - e. В части 4 описать решение, реализованное в соответствии с пп.2-4
 - f. В часть 5 включить составленную схему п.2а

Описание работы:

Были созданы клиентский и серверный модули

Для парсинга была использована библиотека libxml2.

Основной цикл клиентского модуля

```
int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Использование: %s <server_ip> <server_port>\n", argv[0]);
        return 1;
    }

    const char *server_ip = argv[1];
    int server_port = atoi(argv[2]);

    int client_socket = create_client_socket(server_ip, server_port);

    char buffer[MAX_BUFFER_SIZE];

    int times = 5;
    while (times) {
        times--;
        printf("Enter a message to send to the server: ");
        input();
        char * xml = to_xml();

        strncpy(dest: buffer, src: xml, n: sizeof(buffer));
        freeAstTree();
        send_data(socket_fd: client_socket, data: buffer);
        receive_data(socket_fd: client_socket, buffer);
    }
    close_socket(socket_fd: client_socket);

    return 0;
}
```

Была написана небольшая обертка над сокетами со следующим интерфейсом:

```
14 → int create_client_socket(const char *server_ip, int server_port);
15 → void send_data(int socket_fd, const char *data);
16 → void receive_data(int socket_fd, char *buffer);
17 → void close_socket(int socket_fd);
18
19 #endif // SOCKET_WRAPPER_H
20
```

В ней 1 метод, который требует внимания:

```
106 → void receive_data(int socket_fd, char *buffer) {
107     memset(s: buffer, c: 0, n: MAX_BUFFER_SIZE);
108     recv(fd: socket_fd, buf: buffer, n: MAX_BUFFER_SIZE, flags: 0);
109     char *message = parseResponse(buffer);
110     while (trimmed_strcmp(message, "End") != 0) {
111         printf(format: "%s\n", message);
112         memset(s: buffer, c: 0, n: MAX_BUFFER_SIZE);
113         recv(fd: socket_fd, buf: buffer, n: MAX_BUFFER_SIZE, flags: 0);
114         message = parseResponse(buffer);
115     }
116 }
```

Клиент ожидает сообщения пока сервер не отправит сообщение с ключевым словом "End".

Это сделано потому что данные от сервера отсылаются пачками на случай если их будет много после какого-нибудь условного select-a.

Серверный модуль

Для сервера тоже была создана небольшая обертка над сокетами со следующим интерфейсом:

```
12
13 → int create_server_socket();
14 → void bind_socket(int socket_fd, int server_port);
15 → void listen_for_connections(int socket_fd);
16 → int accept_connection(int socket_fd);
17 → void send_data(int socket_fd, const char *data);
18 → void receive_data(int socket_fd, char *buffer);
19 → void close_socket(int socket_fd);
20
```

Основной цикл серверный:

```

29 int main(int argc, char *argv[]) {
30     if (argc != 3) {
31         fprintf(stderr, "Использование: %s <server_ip> <server_port>\n", argv[0]);
32         return 1;
33     }
34     const char *filePath = argv[1];
35     int server_port = atoi(argv[2]);
36     FILE *file = fopen(filePath, "rb+");
37
38
39     int server_socket = create_server_socket();
40     bind_socket(socket_fd: server_socket, server_port);
41     listen_for_connections(socket_fd: server_socket);
42
43     printf("Server waiting for connections...\n");
44
45     int client_socket = accept_connection(socket_fd: server_socket);
46     printf("Connection established with client.\n");
47
48     int times = 10;
49     while (times) {
50         times--;
51         char buffer[MAX_BUFFER_SIZE];
52         receive_data(socket_fd: client_socket, buffer);
53
54         printf("Received from client: %s\n", buffer);
55
56         char *xml = buffer;
57         from_xml(xml, file, client_socket);
58     }
59
60     close_socket(socket_fd: client_socket);
61     close_socket(socket_fd: server_socket);
62
63     return 0;
64 }

```

вся логика находится в метод `from_xml`, там просходит весь парсинг и создание структуры данных для взаимодействия с бд. Пример:

```

311 void *handleDelete(xmlNodePtr root, FILE *file, int client_socket) {
312     char *tableName;
313     for (xmlNodePtr node = root->children; node; node = node->next) {
314         if (xmlStrEqual(str1: node->name, str2: BAD_CAST "tableName")) {
315             tableName = (char *) xmlNodeGetContent(cur: node);
316         } else if (xmlStrEqual(str1: node->name, str2: BAD_CAST "filter")) {
317             PredMass *predMass = goDepth(node: node->children[0].next);
318             deleteRecordFromTable(file, tableName, predMass->predicate, predMass->predicateNumber);
319         }
320     }
321     send_data(socket_fd: client_socket, data: create_xml_document(message_content: "Your entity successfully deleted"));
322     send_data(socket_fd: client_socket, data: create_xml_document(message_content: "End"));
323 }
324

```

сначала мы пытаемся вытащить имя таблицы, затем пытаемся распарсить условия если они есть, создать по ним массив предикатов обходя рекурсивно дерево вглубь. И вызываем метод для удаления сущности из таблицы учитывая условия.

Ну и для примера предлагаю рассмотреть парсинг условий:

```
275 PredMass *mulCond(PredMass *predMass1, PredMass *predMass2) {
276     int newSize = predMass1->predicateNumber + predMass2->predicateNumber;
277
278     Predicate *mergedArray = (Predicate *) malloc(sizeof(Predicate) * newSize);
279     if (mergedArray == NULL) {
280         return NULL;
281     }
282
283     for (int i = 0; i < predMass1->predicateNumber; ++i) {
284         mergedArray[i] = predMass1->predicate[i];
285     }
286
287     for (int i = 0; i < predMass2->predicateNumber; ++i) {
288         mergedArray[predMass1->predicateNumber + i] = predMass2->predicate[i];
289     }
290     PredMass *predMass = malloc(sizeof(PredMass));
291     predMass->predicate = mergedArray;
292     predMass->predicateNumber = newSize;
293     return predMass;
294 }
295
296 PredMass *goDepth(xmlNodePtr node) {
297     if (xmlStrEqual(str1:node->name, str2:BAD_CAST "AND") || xmlStrEqual(str1:node->name, str2:BAD_CAST "OR")) {
298         return mulCond(predMass1:goDepth(node:traverseChildren(node, childNumber:0)), predMass2:goDepth(node:traverseChildren(node, childNumber:1)));
299     }
300     return parseCondition(condition:node);
301 }
```

У нас получается, что мы рекурсивно обходим наш xml документ, складывая условия в массив предикатов.

Схемы:

Была создана функция, которая принимает в себя строчное представление xml-файла и путь до файла со схемой, и возвращает то соответствует ли переданный xml схеме.

```
int validateXmlAgainstSchemaFile(const char *xmlString, const char *schemaFilePath) {
    LIBXML_TEST_VERSION

    xmlDocPtr doc = xmlReadMemory(xmlString, strlen(xmlString),
    "noname.xml", NULL, 0);
    if (doc == NULL) {
        fprintf(stderr, "Failed to parse the input XML.\n");
        return -1;
    }

    xmlSchemaParserCtxtPtr parserCtxt =
    xmlSchemaNewParserCtxt(schemaFilePath);
    if (parserCtxt == NULL) {
        fprintf(stderr, "Failed to parse the input XML schema.\n");
        xmlFreeDoc(doc);
        return -2;
    }

    xmlSchemaPtr schema = xmlSchemaParse(parserCtxt);
    if (schema == NULL) {
        fprintf(stderr, "Failed to parse the input XML schema.\n");
        xmlSchemaFreeParserCtxt(parserCtxt);
        xmlFreeDoc(doc);
        return -2;
    }
}
```



```

    xmlSchemaValidCtxtPtr validCtxt =
xmlSchemaNewValidCtxt(schema);
    if (validCtxt == NULL) {
        fprintf(stderr, "Failed to create a validation
context.\n");
        xmlSchemaFree(schema);
        xmlSchemaFreeParserCtxt(parserCtxt);
        xmlFreeDoc(doc);
        return -3;
    }
    int isValid = xmlSchemaValidateDoc(validCtxt, doc);

    xmlSchemaFreeValidCtxt(validCtxt);
    xmlSchemaFree(schema);
    xmlSchemaFreeParserCtxt(parserCtxt);
    xmlFreeDoc(doc);
    xmlCleanupParser();

    return isValid; // Возвращаем 0, если XML соответствует схеме,
и отрицательное число в противном случае
}

```

Простейшая схема для Response-a:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Root">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Message" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

Для response-a я решил использовать простую схему, потому что я по факту передаю лишь 1 сообщение строчное, которое уже заранее в удобочитаемом виде само по себе.

Схема для request-a:

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Root">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="requestType">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="SELECT_QUERY"/>
                            <xs:enumeration value="INSERT_QUERY"/>
                            <xs:enumeration value="DELETE_QUERY"/>
                            <xs:enumeration value="UPDATE_QUERY"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

```

        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="tableName" type="xs:string"/>
<xs:choice>
    <!-- If requestType is UPDATE -->
    <xs:sequence>
        <xs:element name="updateField">
            <xs:complexType>
                <xs:sequence>
                    <xs:element
name="updateFieldName" type="xs:string"/>
                    <xs:element
name="updateFieldValue" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="filter" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="filter"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element
name="leftOp">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="isColumnName" type="xs:boolean"/>
                                            <xs:element name="value" minOccurs="0">
                                                <xs:simpleType>
                                                    <xs:union
memberTypes="xs:string xs:int xs:double xs:boolean"/>
                                                </xs:simpleType>
                                            </xs:element>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

</xs:element>
<xs:element
name="operator">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="==" />
            <xs:enumeration value="<"/>
            <xs:enumeration value=">"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element
name="rightOp">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="isColumnName" type="xs:boolean"/>
            <xs:element name="value" minOccurs="0">
                <xs:simpleType>
                    <xs:union
memberTypes="xs:string xs:int xs:double xs:boolean"/>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

```

        </xs:sequence>

        <!-- If requestType is DELETE or SELECT -->
        <xs:sequence>
            <xs:element name="filter" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="filter"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element
name="leftOp">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="isColumnName" type="xs:boolean"/>
                                        <xs:element name="value" minOccurs="0">
                                            <xs:simpleType>
                                                <xs:union
memberTypes="xs:string xs:int xs:double xs:boolean"/>
                                            </xs:simpleType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        <xs:element
name="operator">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="="/>
                                    <xs:enumeration value="<"/>
                                    <xs:enumeration value=">"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```

</xs:restriction>

</xs:simpleType>
                                </xs:element>
                                <xs:element
name="rightOp">

<xs:complexType>

<xs:sequence>

<xs:element name="isColumnName" type="xs:boolean"/>

<xs:element name="value" minOccurs="0">

<xs:simpleType>

<xs:union
memberTypes="xs:string xs:int xs:double xs:boolean"/>

</xs:simpleType>

</xs:element>

</xs:sequence>

</xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
</xs:element>
<xs:element name="join" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="leftOperand">
        <xs:complexType>
          <xs:sequence>
            <xs:element
name="leftTable" type="xs:string"/>
            <xs:element
name="leftField" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        </xs:element>
        <xs:element
name="rightOperand">
            <xs:complexType>
                <xs:sequence>
                    <xs:element
name="rightTable" type="xs:string"/>
                    <xs:element
name="rightField" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="selectedVal"
minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="entity"
type="xs:string" minOccurs="0"/>
            <xs:element name="field"
type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
<!-- If requestType is INSERT -->
<xs:element name="insertValues" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="field"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="type"
type="Literal"/>
                        <xs:element
name="value" type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>

```

```
</xs:element>
<xs:simpleType name="Literal">
  <xs:restriction base="xs:string">
    <xs:enumeration value="STRING_LITERAL"/>
    <xs:enumeration value="INTEGER_LITERAL"/>
    <xs:enumeration value="DOUBLE_LITERAL"/>
    <xs:enumeration value="BOOL_LITERAL"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

Пример клиентской сессии:

```
Threads & Variables Console GDB Memory View
/home/iwaa0303/ClionProjects/llp_3/client/cmake-build-debug/client
Enter a message to send to the server: insert into Department values (3, "PIIKT", "Programming Engineeering");
Your insert was successfully completed

Enter a message to send to the server: from Department in Department select Department;
DepartmentId: 1; Name: Entropy; Description: Command develop Accounting and Invoicing modules;
DepartmentId: 2; Name: QuantumTeam; Description: Command develop loyalty module;
DepartmentId: 3; Name: BACKOFFICE; Description: Command develop backoffice module;
DepartmentId: 3; Name: "PIIKT"; Description: "Programming Engineeering";
Enter a message to send to the server: insert into Department values (5, "KT", "Computer Technologies");
Your insert was successfully completed

Enter a message to send to the server: from Department in Department select Department;
DepartmentId: 1; Name: Entropy; Description: Command develop Accounting and Invoicing modules;
DepartmentId: 2; Name: QuantumTeam; Description: Command develop loyalty module;
DepartmentId: 3; Name: BACKOFFICE; Description: Command develop backoffice module;
DepartmentId: 3; Name: "PIIKT"; Description: "Programming Engineeering";
DepartmentId: 5; Name: "KT"; Description: "Computer Technologies";
Enter a message to send to the server: from Employee in Employee select Employee;
Name: Nikita; Surname: Ivanov; Age: 20; DepartmentId: 1;
Name: Ivan; Surname: Bobrov; Age: 21; DepartmentId: 2;
Name: Boris; Surname: Kirillov; Age: 50; DepartmentId: 3;
Enter a message to send to the server: from Employee in Employee select Employee;
Name: Nikita; Surname: Ivanov; Age: 20; DepartmentId: 1;
Name: Ivan; Surname: Bobrov; Age: 21; DepartmentId: 2;
Name: Boris; Surname: Kirillov; Age: 50; DepartmentId: 3;
Enter a message to send to the server: from Department in Department where DepartmentId == 3 select Department;
DepartmentId: 3; Name: BACKOFFICE; Description: Command develop backoffice module;
DepartmentId: 3; Name: "PIIKT"; Description: "Programming Engineeering";
Enter a message to send to the server: from Department in Department where (DepartmentId == 3 && Name == "PIIKT") select Department;
DepartmentId: 3; Name: "PIIKT"; Description: "Programming Engineeering";
```

- 1) Добавили 1 сущность в таблицу Department
- 2) Посмотрели содержимое таблицы Department
- 3) Добавили еще сущность в таблицу Department
- 4) Посмотрели содержимое таблицы Department (убедились, что у нас все сохранилось)
- 5) Далее пару раз посмотрели содержимое таблицы Employee
- 6) Сделали выборку из таблицы Department с условиями (проверили работу conditions)

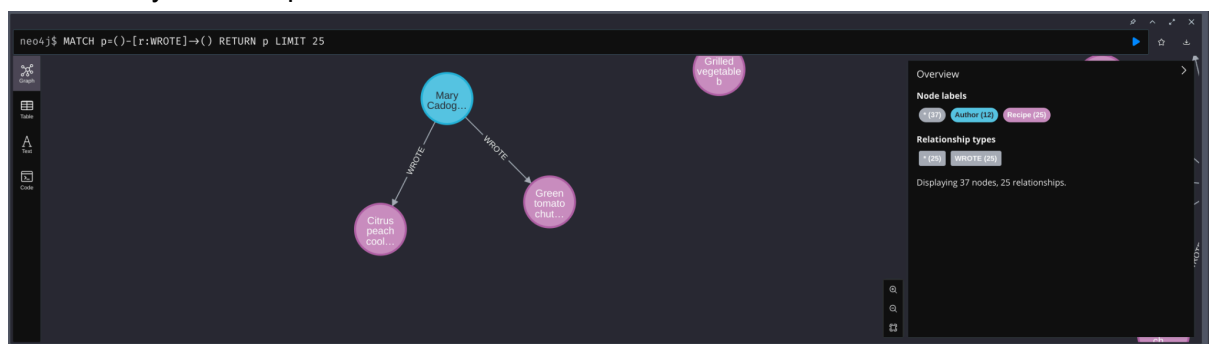
Дополнительное задание:

Задание:

На предметной области “Recipe” выбрать отношение между 2-мя нодами, загрузить датасет в свою базу данных и сделать запрос на выборку элементов с join-ом. Также можно использовать фильтр для выборки, чтобы в выводе данных было меньше.

Реализация:

Среди данных я решил выбрать отношение “WROTE”. Данное отношение отражает связь между рецептами и авторами этих рецептов. По своему типу оно один ко многим, одному автору может соответствовать множество рецептов, но одному рецепту соответствует 1 автор.



То как это выглядит в neo4j представлено на скриншоте выше.

Выгрузка данных:

Данные выгружались в формате json и парсились в обычный текстовый файл программой на python:

```
Author.py x
1 import json
2 import codecs
3
4 with codecs.open('Author.json', 'r', encoding='utf-8-sig') as json_file:
5     json_data = json.load(json_file)
6
7 with open('Author.txt', 'w', encoding='utf-8') as file:
8     file.write("identity|labels|name|elementId\n")
9
10    for item in json_data:
11
12        n = item.get('n', {})
13        properties = n.get('properties', {})
14        file.write(f"{n.get('identity', '')}|{'|'.join(n.get('labels', []))}|{properties.get('name', '')}|{n.get('elementId', '')}\n")
15
```

После обработки данные выглядели следующим образом:

```
11634|Tony Tobin|11634
11635|Jane Hornby|11635
11636|Sarah Cook|11636
11637|Good Food|11637
11638|Jennifer Joyce|11638
11639|Silvana Franco|11639
11640|Lucy Netherton|11640
11641|James Martin|11641
11642|Mary Cadogan|11642
11643|John Torode|11643
11644|Cassie Best|11644
11645|Barney Desmazery|11645
```

После этого была написана программа уже на языке C, которая создает необходимые структуры данных для выполнения insert-а в базу данных:

```
void insertAuthor(int32_t id, char *name, FILE *file) {
    FieldValue fieldValue1 = {.data: &id, .dataSize: sizeof(int32_t)};
    FieldValue fieldValue2 = {.data: name, .dataSize: sizeof(char) * strlen(s: name)};
    FieldValue fieldValues[2] = {[0]: fieldValue1, [1]: fieldValue2};
    EntityRecord entityRecord = {.fields: fieldValues, .linkNext: NULL};
    insertRecordIntoTable(file, &entityRecord, tableName: "Author");
}

void loadAuthorsData(FILE *file) {

    FILE *authorsFile = fopen(filename: AUTHORS_FILE_NAME, modes: "r");

    char line[MAX_LINE_LENGTH];
    int32_t authorId;
    int times = 303;
    while (times) {
        times--;
        fgets(s: line, n: sizeof(line), stream: authorsFile);
        char *token = strtok(s: line, delim: "|");

        if (token != NULL) {
            authorId = atoi(nptr: token);
            // здесь у нас есть id автора
            printf(format: "ID: %d, ", authorId);
        } else {
            break;
        }
        token = strtok(s: NULL, delim: "|");

        if (token != NULL) {
            while (isspace((unsigned char) token[strlen(token) - 1]))
                token[strlen(s: token) - 1] = '\0';

            insertAuthor(id: authorId, name: token, file);
        } else {
            break;
        }
    }
    fclose(stream: authorsFile);
}
```

Выполнение команды select загруженных данных:

```
Enter a message to send to the server: from Author in Author select Author;

Id: 11634; Name: Tony Tobin;

Id: 11635; Name: Jane Hornby;

Id: 11636; Name: Sarah Cook;

Id: 11637; Name: Good Food;

Id: 11638; Name: Jennifer Joyce;

Id: 11639; Name: Silvana Franco;

Id: 11640; Name: Lucy Netherton;

Id: 11641; Name: James Martin;

Id: 11642; Name: Mary Cadogan;

Id: 11643; Name: John Torode;

Id: 11644; Name: Cassie Best;

Id: 11645; Name: Barney Desmazery;

Id: 11646; Name: Tom Kerridge;

Id: 11647; Name: Sara Buenfeld;

Id: 11648; Name: Diana Henry;

Id: 11649; Name: Lulu Grimes;

Id: 11650; Name: Caroline Hire;

Id: 11651; Name: Orlando Murrin;

Id: 11652; Name: Emma Lewis;

Id: 11653; Name: Jack Monroe;

Id: 11657; Name: Carmelita Caruana;

Enter a message to send to the server: from Author in Author where (Id == 11657) select Author;

Id: 11657; Name: Carmelita Caruana;

Enter a message to send to the server:
```

Выборка рецептов с учетом фильтров:

```
Enter a message to send to the server: from Recipe in Recipe where (AuthorId == 11657) select Recipe;

AuthorId: 11657; RecipeId: 5316; Name: Peach almond tart; PreparationTime: 0; Description: Authentic Italian Peach and almond tart; RecipeId: 102749; SkillLevel: More effort; CookingTime: 0; ElementId: 5316;

AuthorId: 11657; RecipeId: 4974; Name: Pesce spada stecato a grigliata (Grilled garlic mint stuffed swordfish); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this exciting fish dish - mint and garlic work so well together; RecipeId: 90848; SkillLevel: 6;

AuthorId: 11657; RecipeId: 1094; Name: Pasta con il pesto alla Trapanese (Tomato and almond pesto); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this delicious vegetarian summer pasta; RecipeId: 96042; SkillLevel: Easy; CookingTime: 0; ElementId: 1094;

AuthorId: 11657; RecipeId: 8491; Name: Pasta 'maciata (Aubergine wrapped pasta dish); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this impressive pasta dish - superb for entertaining; RecipeId: 93932; SkillLevel: More effort; CookingTime: 0; ElementId: 8491;

AuthorId: 11657; RecipeId: 1640; Name: Tagliatelle with a lemon pork ragù; PreparationTime: 0; Description: Authentic Italian tagliatelle with a lemon pork ragù; RecipeId: 90913; SkillLevel: Easy; CookingTime: 0; ElementId: 1640;

AuthorId: 11657; RecipeId: 57; Name: Prawn risotto; PreparationTime: 0; Description: Try this authentic seafood risotto, typical of Venetian cooking; RecipeId: 90994; SkillLevel: Easy; CookingTime: 4200; ElementId: 57;

AuthorId: 11657; RecipeId: 5744; Name: Roast chicken with savoury sauce; PreparationTime: 0; Description: Pleasant and delicious Italian style roast chicken ; RecipeId: 180576; SkillLevel: More effort; CookingTime: 6300; ElementId: 5744;

AuthorId: 11657; RecipeId: 5524; Name: Potate in agrodolce (Sweet sour warm potato salad); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this simple potato dish; RecipeId: 182332; SkillLevel: Easy; CookingTime: 0; ElementId: 5524;

AuthorId: 11657; RecipeId: 10547; Name: Venetian-style scallops; PreparationTime: 0; Description: A quick but impressive starter of plump scallops scattered with toasted breadcrumbs; RecipeId: 90461; SkillLevel: More effort; CookingTime: 2100; ElementId: 10547;

Enter a message to send to the server:
```

Выборка тех же рецептов, только с join-ом:

```
Enter a message to send to the server: from Recipe in Recipe join Author in Author on Recipe.AuthorId equals Author.Id where (AuthorId == 11657) select Recipe;

AuthorId: 11657; RecipeId: 5316; Name: Peach almond tart; PreparationTime: 0; Description: Authentic Italian Peach and almond tart; RecipeId: 102749; SkillLevel: More effort; CookingTime: 0; ElementId: 5316; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 4974; Name: Pesce spada stecato a grigliata (Grilled garlic mint stuffed swordfish); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this exciting fish dish - mint and garlic work so well together; RecipeId: 90848; SkillLevel: 6;

AuthorId: 11657; RecipeId: 1094; Name: Pasta con il pesto alla Trapanese (Tomato and almond pesto); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this delicious vegetarian summer pasta; RecipeId: 96042; SkillLevel: Easy; CookingTime: 0; ElementId: 1094; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 8491; Name: Pasta 'maciata (Aubergine wrapped pasta dish); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this impressive pasta dish - superb for entertaining; RecipeId: 93932; SkillLevel: More effort; CookingTime: 0; ElementId: 8491; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 1640; Name: Tagliatelle with a lemon pork ragù; PreparationTime: 0; Description: Authentic Italian tagliatelle with a lemon pork ragù; RecipeId: 90913; SkillLevel: Easy; CookingTime: 0; ElementId: 1640; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 57; Name: Prawn risotto; PreparationTime: 0; Description: Try this authentic seafood risotto, typical of Venetian cooking; RecipeId: 90994; SkillLevel: Easy; CookingTime: 4200; ElementId: 57; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 5744; Name: Roast chicken with savoury sauce; PreparationTime: 0; Description: Pleasant and delicious Italian style roast chicken ; RecipeId: 180576; SkillLevel: More effort; CookingTime: 6300; ElementId: 5744; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 5524; Name: Potate in agrodolce (Sweet sour warm potato salad); PreparationTime: 0; Description: Carmelita Caruana celebrates the tastes of Sicily with this simple potato dish; RecipeId: 182332; SkillLevel: Easy; CookingTime: 0; ElementId: 5524; Id: 11657; Name: Carmelita Caruana;

AuthorId: 11657; RecipeId: 10547; Name: Venetian-style scallops; PreparationTime: 0; Description: A quick but impressive starter of plump scallops scattered with toasted breadcrumbs; RecipeId: 90461; SkillLevel: More effort; CookingTime: 2100; ElementId: 10547; Id: 11657; Name: Carmelita Caruana;

Enter a message to send to the server:
```

Выборка без фильтров с join-ом:

Recipe ID	Name	Preparation Time	Description	Difficulty	Level	Effort	Cost
Author: 11916; RecipeID: 9791; Name: Coffee cake; PreparationTime: 1200; Description: Whip up a classic coffee cake with ease. You can freeze the scones, without the buttercream icing, if you want to save time later on. Ideal for a bake sale. RecipeID: 7587831; SkillLevel: Easy; CookingTime: 18							
Author: 11917; RecipeID: 9801; Name: Syzzy; PreparationTime: 1500; Description: Learn to make these authentic Japanese root or vegetable mousses, known as syzy, with our step-by-step guide. RecipeID: 5125441; SkillLevel: A challenge; CookingTime: 600; ElementID: 9070; Is: 11917; Name: Yu							
Author: 11918; RecipeID: 9495; Name: Tempura squid; PreparationTime: 3000; Description: Say Brett's crunchy tempura squid makes a great sharing dish. You can prep your squid in advance and keep it in the fridge until you're ready. RecipeID: 6299231; SkillLevel: More effort; CookingTime: 600; Etc							
Author: 11919; RecipeID: 9072; Name: Anzane bar; PreparationTime: 1500; Description: Meet ideas for a coffee morning or afternoon tea. These bars from Canada are so mouth-watering, with a crunchy kiwifruit base, custard layer and chocolate topping. RecipeID: 6088161; SkillLevel: Easy; CookingTime: 300;							
Author: 11920; RecipeID: 10112; Name: Pan-fried raw asparagus with avocado herb butter; PreparationTime: 900; Description: Let the fresh English asparagus shine in this simple spring dish. A combination of contrasting textures and textures gives added mouth, served with a hint of wild rice or							
Author: 11921; RecipeID: 10219; Name: Burst submerge veggie chili; PreparationTime: 1500; Description: This warming submerge chili is low fat and 4 of your 5 a day. Serve up this smoky spiced soup with brown rice and all your favourite trimmings. RecipeID: 6299151; SkillLevel: Easy; Cookin							
Author: 11922; RecipeID: 10220; Name: Chocolate almond nougat bomb cake; PreparationTime: 1200; Description: Great British Bake Off's Ruby Tandoh creates a two-tone decorative sponge riss. Pich a patterned bomb cake tin for an extra-special finish. RecipeID: 7392381; SkillLevel: Easy; Cookin							
Author: 11923; RecipeID: 10230; Name: Rainbow yolk pop; PreparationTime: 2700; Description: Play the rainbow play with this oh-so-pretty Swiss roll, a vibrant party place that will get everyone talking. Celebrate different in the sweetest way. RecipeID: 6263631; SkillLevel: More effort; CookingTime							
Author: 11924; RecipeID: 10377; Name: Amazing chocolate cupcakes; PreparationTime: 900; Description: Whip up a batch of simple chocolate cupcakes for a party, coffee morning or afternoon tea. Top with buttercream, or perhaps a cream cheese frosting. RecipeID: 7041171; SkillLevel: Easy; CookingTime							
Author: 11925; RecipeID: 10450; Name: Swiss roll; PreparationTime: 900; Description: The ultimate nostalgic treat, save your own Swiss roll with fruity strawberry jam filling. This easy dessert is great for entertaining. RecipeID: 7395757; SkillLevel: Easy; CookingTime: 600; ElementID: 10450; Dis							
Author: 11926; RecipeID: 10480; Name: Chelsea chowls chili; PreparationTime: 1400; Description: This low fat, vibrant chili from Johannesburg is based on canned beans, tomatoes, onions and chillies. You can add whatever vegetables you have to hand. RecipeID: 6306031; SkillLevel: Easy; Cookin							
Author: 11927; RecipeID: 10570; Name: Easy rocky road; PreparationTime: 900; Description: Great for a bake sale, a gift, or simply an afternoon treat to enjoy with a cuppa, this rocky road is quick to make and uses mainly store cupboard ingredients. RecipeID: 7587821; SkillLevel: Easy; CookingTime							
Author: 11928; RecipeID: 10610; Name: Beef chorizo empanadas; PreparationTime: 2400; Description: These beef and chorizo empanadas as a kind of South American party. They can be baked or fried and are tasty served with guacamole and sour cream. RecipeID: 7507260; SkillLevel: Easy; Cook							
Author: 11929; RecipeID: 10617; Name: Pistachio on taste (Pistachio custard tart); PreparationTime: 1600; Description: A classic Portuguese custard tart with buttery pastry and a hint of cinnamon and lemon in the filling. Who could resist these delicious tiny treats? RecipeID: 925848; Skilllev							
Author: 11930; RecipeID: 10625; Name: Spiced sweet potato burgers with avocado basil salsa; PreparationTime: 1200; Description: These hearty veggie burgers are bursting with flavour, with sunflower seeds and sweet potato crisps for added crunch. RecipeID: 5171316; SkillLevel: More effort; Cookin							
Author: 11931; RecipeID: 10640; Name: Vanilla cupcakes; PreparationTime: 1400; Description: Decorate these easy vanilla cupcakes with buttercream and the topping of your choice. They're super-simple to make and taste divine - great as a tea-time treat. RecipeID: 7407551; SkillLevel: Easy; Cooking							
Author: 11932; RecipeID: 10694; Name: Coconut cones; PreparationTime: 1200; Description: This easy coconut leaf cake will give you a taste of the tropics. Bake with a handful of store cupboard ingredients and enjoy with a cuppa. RecipeID: 7593551; SkillLevel: Easy; CookingTime: 3000; ElementID: 1							
Author: 11933; RecipeID: 10894; Name: Chocolate sponge with hot chocolate custard; PreparationTime: 1800; Description: This rich, sticky chocolate cake is given a low-fat 'bassener', then served with chocolate custard for a perfect retro, weekend pud. RecipeID: 6088401; SkillLevel: Easy; Cookin							
Author: 11934; RecipeID: 10930; Name: Vietnamese chicken noodle soup; PreparationTime: 1200; Description: Name our comforting bowl of chicken noodle soup with warming Vietnamese spices. This easy 'pho' recipe makes an ideal low-fat supper for a crowd. RecipeID: 7595940; SkillLevel: Easy; C							
Author: 11935; RecipeID: 10980; Name: Gingerbread men; PreparationTime: 1800; Description: Use this versatile guide to bake gingerbread men and other shapes. Let the kids decide what they want for a perfect festive day activity. RecipeID: 8602310; SkillLevel: Easy; CookingTime: 600; ElementID							
Author: 11936; RecipeID: 11047; Name: Sweet-rice chili cornbread; PreparationTime: 1200; Description: Bitten potato and frozen sweetcorn are a deliciously different alternative to your regular 'las'. Best eaten fresh from the oven. RecipeID: 5064756; SkillLevel: More effort; CookingTime: 1800							
Author: 11937; RecipeID: 11057; Name: Cheese, leek potato pie; PreparationTime: 4500; Description: Pies, often called Turkish pizzas, are boat-shaped tarts. Chef Selin Klarin gives us her own take on these pastries, filling them with leek, potato and cheese . RecipeID: 5076431; SkillLevel: A							

Выводы:

В ходе выполнения данной работы были соединены предыдущие работы воедино при помощи протокола xml. Это помогло во-первых осознать мотивацию того, что происходило, а именно понять как должна взаимодействовать система на разных уровнях и что от чего зависит, какие подводные камни могут встретиться из-за мелких недочетов на предыдущих шагах. В целом получился более менее готовый продукт, который уже представляет собой нормальное работающее приложение.