Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3 по дисциплине

«**Низкоуровневое программирование**»

Выполнил:
Миху Вадим Дмитриевич
Факультет “ПИиКТ”
Группа: P33301

Преподаватель:
Кореньков Юрий Дмитриевич
Вариант 5

Санкт-Петербург, 2023 г.

Оглавление

Задание

Вариант №5

Создать 2 консольных приложения, которые обмениваются информацией через XML по своему протоколу. Серверный модуль должен использовать функции, реализованные в первой ЛР и по сути представляет из себя удаленную СУБД. Клиентский модуль должен парсить запросы на языке AQL и пересылать их серверу.

- Изучить доступные средства для парсинга и валидации XML.
- Спроектировать структуры данных для представления запроса.
- Реализовать публичный интерфейс для приведенных выше операций
- Реализовать тестовую программу для демонстрации работоспособности решения

Выполнение

Ссрверу в параметры передается адрес, порт и название файла базы данных, а клиенту только адрес и порт.

Создание тестовой таблицы

```
Server address: 127.0.0.1
Port: 6788
CREATE_TABLE TEST {i: INT, s: STRING}

<?xml version="1.0"?>
<sqlQuery><requestType>CREATE_TABLE</requestType><tableName>TEST</tableName><fields><field><name>i<

Message: Table successfully created!
```

Вставка данных

```
Server address: 127.0.0.1
Port: 6788
INSERT {i: 13, s: "test"} IN TEST
"test"
<?xml version="1.0"?>
<sqlQuery><requestType>INSERT</requestType><tableName>TEST</tableName><insertValues><field><name>i<
/type></field></insertValues></sqlQuery>

Message: Successfully inserted
```

Выборка данных

```
Server address: 127.0.0.1
Port: 6788
FOR u IN TEST
        RETURN u


<?xml version="1.0"?>
<sqlQuery><requestType>SELECT</requestType><tableName>TEST</tableName><filters/></sqlQuery>

i: 13
s: test
Message: End of table
```

Описание примененных решений

Для разработки использовалась библиотека LIBXML2 — одна из самых популярных библиотек под XML на C. На клиенте и на сервере были реализованы модули net и xml, которые позволяют передавать сообщения и парсить их. Сообщение передается по обычным сокетам, парсинг вида команды происходит с помощью специального тега.

Сервер имеет структуру реквестов, которые может принять и обработать. На клиенте запрос формируется в дерево, после чего конвертируется в xml, отправляется на сервер. Сервер формирует структуру запроса и выполняет его в соответствии с типом.

```c
enum requestType {
    CREATE_TABLE,
    DROP_TABLE,
    INSERT,
    SELECT,
    DELETE
};

struct createTableRequest {
    char *tableName;
    int32_t columnNum;
    enum DataType *types;
    const char **names;
};

struct dropTableRequest {
    char *tableName;
};

struct insertRequest {
    char *tableName;
    int32_t dataCount;
    void **data;
};
```

```
struct selectRequest {
    char *tableName;
    struct Condition *conditions;
};


struct deleteRequest {
    char *tableName;
    int32_t conditionCount;
    struct Condition *conditions;
};

struct request {
    enum requestType type;
    union {
        struct createTableRequest createTableRequest;
        struct dropTableRequest dropTableRequest;
        struct insertRequest insertRequest;
        struct selectRequest selectRequest;
        struct deleteRequest deleteRequest;
    };
};
```

Все запросы выполняются в отдельном потоке, это позволяет не блокировать основное ожидание сокета и переиспользовать его.

```
    while (1) {
        printf("ACCEPTING CONNECTIONS\n");
        int clientSocket = acceptConnection(fd);
        if (clientSocket < 0) {
            printf("Unable to accept connection");
        }
        int *clientSocketPtr = malloc(sizeof(int));
        *clientSocketPtr = clientSocket;
        pthread_t tid;
        if (pthread_create(&tid, NULL, handleClient, (void *) clientSocketPtr) != 0) {
            perror("Thread creation failed");
            exit(EXIT_FAILURE);
        }
        pthread_detach(tid);
    }
```

Разработанный протокол формата xml в виде xsd файла

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="sqlQuery">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="requestType">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="SELECT"/>
                            <xs:enumeration value="INSERT"/>
                            <xs:enumeration value="DELETE"/>
                            <xs:enumeration value="UPDATE"/>
                            <xs:enumeration value="CREATE_TABLE"/>
                            <xs:enumeration value="DROP_TABLE"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="tableName" type="xs:string"/>
                <xs:choice>
                    <xs:sequence>
                        <!-- If requestType is DELETE or SELECT -->
                        <xs:element name="filters" 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="&lt;"/>
                                                            <xs:enumeration
value="&gt;"/>
                                                        </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>
                        <xs:element name="jo  in">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="joinConditions"
type="JoinConditionType"/>
                                    <xs:element name="filterConditions"
type="FilterConditionType" minOccurs="0"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <!-- 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="name" type="xs:string"/>
                                                <xs:element name="value" type="xs:string"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <!-- If requestType is UPDATE -->
                        <xs:element name="updateValues" minOccurs="0">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="field" maxOccurs="unbounded">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="name" type="xs:string"/>
                                                <xs:element name="type">
                                                    <xs:simpleType>
                                                        <xs:restriction base="xs:string">
                                                            <xs:enumeration value="string"/>
                                                            <xs:enumeration value="int"/>
                                                            <xs:enumeration value="double"/>
                                                            <xs:enumeration value="boolean"/>
                                                        </xs:restriction>
                                                    </xs:simpleType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <!-- If requestType is CREATE_TABLE -->
```

```
                    <xs:element name="fields" minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="field" maxOccurs="unbounded"
minOccurs="0">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="name" type="xs:string"/>
                                            <xs:element name="type" type="xs:string"/>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:choice>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="JoinConditionType">
        <xs:sequence>
            <xs:element name="leftTable" type="xs:string"/>
            <xs:element name="leftColumn" type="xs:string"/>
            <xs:element name="operator" type="xs:string"/>
            <xs:element name="rightTable" type="xs:string"/>
            <xs:element name="rightColumn" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="FilterConditionType">
        <xs:sequence>
            <xs:element name="table" type="xs:string"/>
            <xs:element name="column" type="xs:string"/>
            <xs:element name="operator" type="xs:string"/>
            <xs:element name="value" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Он предоставляет в основном только структуру запроса, не ограничивая типы, валидация типов происходит впоследствии в коде.

Выводы по работе

В ходе выполнения лабораторной работы были соединены результаты 1 и 2 лр. Реализовано клиент-серверное взаимодействие с помощью протокола на xml

Дополнительное задание

Создадим несколько таблиц с реальными данными и проверим работоспособность соединений на них. Созданная база данных поддерживает только внутренние соединения методом вложенного цикла, поэтому тип соединения дополнительно не указывается.

Для примера использовалась база данных ucheb, хранимая на университетском сервере

```
C:\Windows\system32\wsl.exe --distribution Ubuntu --exec /bin/bash -c "export TERM=xterm-256color && cd /mnt/c/Users/Administrator/CLionProjects/llp3/cmake-build-debu
LionProjects/llp3/cmake-build-debug/client 127.0.0.1 6788"
Server address: 127.0.0.1
Port: 6788
CREATE_TABLE PEOPLE {id: INT, surname: STRING, name: STRING, second_name: STRING, pin: STRING, inn: INT, birthdate: INT, sex: STRING, birthplace: STRING, for: STRING,
, who_changed: STRING, when_changed: INT, death_date: INT, fio: STRING}

<?xml version="1.0"?>
<sqlQuery><requestType>CREATE_TABLE</requestType><tableName>PEOPLE</tableName><fields><field><name>id</name><type>INT</type></field><field><name>surname</name><type>S
name</name><type>STRING</type></field><field><name>second_name</name><type>STRING</type></field><field><name>pin</name><type>STRING</type></field><field><name>inn</na
><name>birthdate</name><type>INT</type></field><field><name>sex</name><type>STRING</type></field><field><name>birthplace</name><type>STRING</type></field><field><name
ield><field><name>who_made</name><type>STRING</type></field><field><name>when_made</name><type>INT</type></field><field><name>who_changed</name><type>STRING</type></f
```

```
C:\Windows\system32\wsl.exe --distribution Ubuntu --exec /bin/bash -c "export TERM=xterm-256color && cd /mnt/c/Users/Administrator/CLionProjects/ll
LionProjects/llp3/cmake-build-debug/client 127.0.0.1 6788"
Server address: 127.0.0.1
Port: 6788
CREATE_TABLE EDUCATIONS {nzk: STRING, chlvk_id: INT, vid_obuch_id: INT, who_made: STRING, when_made: INT, who_changed: STRING, when_changed: INT}

<?xml version="1.0"?>
<sqlQuery><requestType>CREATE_TABLE</requestType><tableName>EDUCATIONS</tableName><fields><field><name>nzk</name><type>STRING</type></field><field>
<name>vid_obuch_id</name><type>INT</type></field><field><name>who_made</name><type>STRING</type></field><field><name>when_made</name><type>INT</typ
G</type></field><field><name>when_changed</name><type>INT</type></field></fields></sqlQuery>

Message: Table successfully created!
```

Сгенерированы скрипты для вставки

```
select concat('INSERT {id: ', "Н_ЛЮДИ"."ИД", ', surname: ', "Н_ЛЮДИ"."ФАМИЛИЯ", ', name: ', "Н_
select concat('INSERT {chlvk_id: ', "Н_СЕССИЯ"."ЧЛВК_ИД", ', uch_god: ', "Н_СЕССИЯ"."УЧГОД", '}
```

Output | concat('INSERT {chlv... IN EDUCATIONS'):text

42 rows

	concat
1	INSERT {chlvk_id: 110136, uch_god: 2002/2003} IN EDUCATIONS
2	INSERT {chlvk_id: 110136, uch_god: 2002/2003} IN EDUCATIONS
3	INSERT {chlvk_id: 110136, uch_god: 2004/2005} IN EDUCATIONS
4	INSERT {chlvk_id: 110136, uch_god: 2004/2005} IN EDUCATIONS
5	INSERT {chlvk_id: 110136, uch_god: 2004/2005} IN EDUCATIONS
6	INSERT {chlvk_id: 110136, uch_god: 2004/2005} IN EDUCATIONS

Ожидаемый результат соединения

```
select "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ЛЮДИ"."ИД", "Н_СЕССИЯ"."УЧГОД"
from "Н_ЛЮДИ"
       join "Н_СЕССИЯ" on "Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
  and "Н_СЕССИЯ"."ЧЛВК_ИД" = 110136; --100012
```

Output | -- 1

42 rows

	"ФАМИЛИЯ"	"ИД"	"УЧГОД"
1	Оголюк	110136	2002/2003
2	Оголюк	110136	2002/2003
3	Оголюк	110136	2004/2005
4	Оголюк	110136	2004/2005
5	Оголюк	110136	2004/2005
6	Оголюк	110136	2004/2005
7	Оголюк	110136	2004/2005
8	Оголюк	110136	2003/2004
9	Оголюк	110136	2008/2009
10	Оголюк	110136	2008/2009
11	Оголюк	110136	2008/2009

Результат на практике

```
Server address: 127.0.0.1
Port: 6794
FOR u IN PEOPLE
      FOR j IN EDUCATIONS
            FILTER u.id == j.chlvk_id
            FILTER u.id == 110136
            RETURN {surname: u.surname, id: u.id, uch_god: j.uch_god}


<?xml version="1.0"?>
<sqlQuery><requestType>JOIN</requestType><tableName>JOIN</tableName><join><aliases><alias
><table>PEOPLE</table><al>u</al></alias><alias><table>EDUCATIONS</table><al>j</al></alias
><alias><table></tabl
e><al></al></alias></aliases><joinConditions><leftAlias>u</leftAlias><rightAlias>j</right
Alias><operator>==</operator><leftColumn>id</leftColumn><rightColumn>chlvk_id</rightColum
n><tableAlias>u</tableAlias><operator>==</operator><column>id</column><value>110136</valu
e></joinConditions><filterConditions/><return/></join></sqlQuery>

surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2009/2010
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2006/2007
surname: Оголюк
        id: 110136
uch_god: 2006/2007
surname: Оголюк
        id: 110136
uch_god: 2005/2006
surname: Оголюк
```

```
        id: 110136
uch_god: 2006/2007
surname: Оголюк
        id: 110136
uch_god: 2006/2007
surname: Оголюк
        id: 110136
uch_god: 2006/2007
surname: Оголюк
        id: 110136
uch_god: 2005/2006
surname: Оголюк
        id: 110136
uch_god: 2005/2006
surname: Оголюк
        id: 110136
uch_god: 2005/2006
surname: Оголюк
        id: 110136
uch_god: 2007/2008
surname: Оголюк
        id: 110136
uch_god: 2007/2008
surname: Оголюк
        id: 110136
uch_god: 2007/2008
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2010/2011
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2008/2009
surname: Оголюк
        id: 110136
uch_god: 2003/2004
surname: Оголюк
```

```
        id: 110136
uch_god: 2004/2005
surname: Оголюк
        id: 110136
uch_god: 2004/2005
surname: Оголюк
        id: 110136
uch_god: 2004/2005
surname: Оголюк
        id: 110136
uch_god: 2004/2005
surname: Оголюк
        id: 110136
uch_god: 2004/2005
surname: Оголюк
        id: 110136
uch_god: 2002/2003
surname: Оголюк
        id: 110136
uch_god: 2002/2003
Message: End of table

        Process finished with exit code 0
```