

Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Информатика и вычислительная техника
Дисциплина «Низкоуровневое программирование»

Отчет По лабораторной работе №1 Вариант 1 (XML)

Выполнил:
Агеев Дмитрий Сергеевич
Преподаватель:
Кореньков Ю. Д.

Санкт-Петербург, 2023 г.

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:

1. Выбрать библиотеку для реализации
2. Разработать в виде консольного приложения две программы: клиентскую и серверную части.
3. В серверной части получать по сети запросы и операции описанного формата и последовательно выполнять 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.
4. В клиентской части в цикле получать на стандартный ввод текст команд, извлекать из него информацию о запрашиваемой операции с помощью модуля из второго задания и пересылать её на сервер с помощью модуля для обмена информацией, получать ответ и выводить его в человеко-понятном виде в стандартный вывод.

Описание работы:

Программа представляет из себя синтез артефактов полученных в 1 и 2 лабораторных работах (`database_module`), (`parser_module`), в частности объединение их в две взаимодействующих программы: клиент и сервер. Makefile'ы лабораторных работ были переделаны в файлы `smake` и включены в новый модуль в качестве подключаемых библиотек. Для сериализации данных использовался XML, реализуемый библиотекой XSD Code Synthesis. Сборка конечных артефактов (`client`, `server`) осуществляется при помощи `smake`. В качестве аргументов серверу передается адрес локальной конечной точки подключения и адрес подключаемого файла, а также имя файла, с которым будет взаимодействовать сервер. В качестве аргументов клиенту передается адрес локальной конечной точки для подключения.

Аспекты реализации:

Для работы с XML необходимо сформировать два файла (form.xsd, request.xsd). Для этого необходимо написать файла с расширением .xsd в качестве описания схемы xsd.

Form.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="view_t">
    <xs:sequence>
      <xs:element name="operation" type="xs:string"/>
      <xs:element name="level" type="level" maxOccurs="10"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="level">
    <xs:sequence>
      <xs:element name="level_id" type="xs:int"/>
      <xs:element name="relation" type="xs:string"/>
      <xs:element name="is_negative_lvl" type="xs:int"/>
      <xs:element name="id" type="xs:string"/>
      <xs:element name="any_id" type="xs:int"/>
      <xs:element name="filter" type="filter_t" maxOccurs="10"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="filter_t">
    <xs:sequence>
      <xs:element name="filter_id" type="xs:int"/>
      <xs:element name="is_negative_filter" type="xs:int"/>
      <xs:element name="comparator" type="comparator_t"
maxOccurs="25"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="comparator_t">
    <xs:sequence>
      <xs:element name="comparator_id" type="xs:int"/>
      <xs:element name="is_negative_comparator" type="xs:int"/>
      <xs:element name="operator1" type="operator_t"/>
      <xs:element name="operation" type="xs:string"/>
      <xs:element name="operator2" type="operator_t"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="operator_t">
    <xs:sequence>
      <xs:element name="operator" type="xs:string"/>
      <xs:element name="is_field" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="response" type="view_t"/>
</xs:schema>
```

В качестве результата при передаче результата второй лабораторной работы, сформируется сообщение в xml формате для передачи его на сервер.

Request.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="header_t">
    <xs:sequence>
      <xs:element name="col_amount" type="xs:int"/>
      <xs:element name="column" type="xs:string" maxOccurs="50"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="body_t">
    <xs:sequence>
      <xs:element name="nodes" type="node_t" maxOccurs="10"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="node_t">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="attr_name" type="xs:string" maxOccurs="5"
minOccurs="0"/>
      <xs:element name="attr_value" type="xs:string" maxOccurs="5"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="response_t">
    <xs:sequence>
      <xs:element name="status" type="xs:int"/>
      <xs:element name="message" type="xs:string"/>
      <xs:element name="finished" type="xs:boolean"/>
      <xs:choice minOccurs="0">
        <xs:element name="header" type="header_t"/>
        <xs:element name="body" type="body_t"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="resp" type="response_t"/>

</xs:schema>
```

Результатом является структура данных, с которой идет взаимодействия для получения информации о запросе. После по этой информации, выполняются запросы базы данных первой лабораторной.

Результаты:

В клиенте в зависимости от запроса разный вывод:

Прим 1:

Connected to the server!

+1/2[name=dima][age=20]

Awaiting server response...

Insert done!

Прим 2:

Awaiting server response...

select done

Node name:2

---Attribute 0 name:id ----value:1

---Attribute 1 name:age ----value:20

---Attribute 2 name:name ----value:dima

Прим 3:

-/1/2/3

Awaiting server response...

delete done

Пример 4:

?/1/2[age>10]

Awaiting server response...

select done

Node name:2

---Attribute 0 name:id ----value:1

---Attribute 1 name:age ----value:20

---Attribute 2 name:name ----value:dima

Пример 5:

Join

?/1/2/*

Awaiting server response...

select done

Node name:4

---Attribute 0 name:id ----value:2

---Attribute 1 name:age ----value:25

---Attribute 2 name:name ----value:Ira

Node name:2

---Attribute 0 name:parent ----value:true

---Attribute 1 name:id ----value:1

---Attribute 2 name:age ----value:20

---Attribute 3 name:name ----value:dima

Выводы:

- Изучил, неизвестную мне до этого, библиотеку XSD code Synthesis и способ передачи данных по сети — XML.
- Научился интегрировать модули stake, применяя библиотеки и несколько целей сборки (гораздо удобнее, чем Makefile, как мне показалось).
- Был реализован протокол передачи данных при помощи XML, а также передача этих данных по сети.