

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №3**  
по дисциплине  
«Низкоуровневое программирование»  
Вариант №3

**Студент:**

Лунева Арина Алексеевна

Группа Р33312

**Преподаватель:**

Кореньков Юрий Дмитриевич

Санкт-Петербург, 2023

**Цель работы:** 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 и изучить библиотеку
2. На основе существующей библиотеки реализовать модуль, обеспечивающий взаимодействие
3. Реализовать серверную часть в виде консольного приложения
  - В качестве аргументов командной строки приложение принимает адрес локальной конечной точки для прослушивания входящих соединений и имя файла данных, который необходимо открыть, если он существует, иначе создать
  - Работает с файлом данных посредством модуля из задания 1
  - Принимает входящие соединения и взаимодействует с клиентами посредством реализованного модуля
  - 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
  - В качестве аргументов командной строки приложение принимает адрес конечной точки для подключения
  - Подключается к серверу и взаимодействует с ним посредством модуля из п2
  - Читает со стандартного ввода текст команд и анализирует их посредством реализованного модуля из задания 2
  - Преобразует результат разбора команды к структурам данных модуля из п2, передаёт их для обработки на сервер, возвращаемые результаты выводит в стандартный поток вывода

### **Описание программы:**

*client* – модуль клиента, производящий синтаксический анализ запроса

*server* – серверный модуль, отвечающий за взаимодействие с файлом данных

*protos* – модуль, реализующий сериализацию передаваемых данных

### **Аспекты реализации:**

- Для сборки модуля синтаксического анализа использованы утилиты flex и bison
- В качестве библиотеки для сериализации выбрана библиотека nanopb
- Передача по сети реализована при помощи сетевых сокетов API ОС

### Дополнительное задание:

Для выполнения задания был разработан модуль на языке Python с использованием библиотеки winreg, выполняющий рекурсивный обход директорий реестра и записывающий полученные данные в файл.

Код модуля можно найти в файле [main.py](#)

```
db.find({parent: "CanonicalData"})
result:
tuple 6:
parent          CanonicalData
name             Catalogs
type             key
value_type       REG_SZ
value            None
```

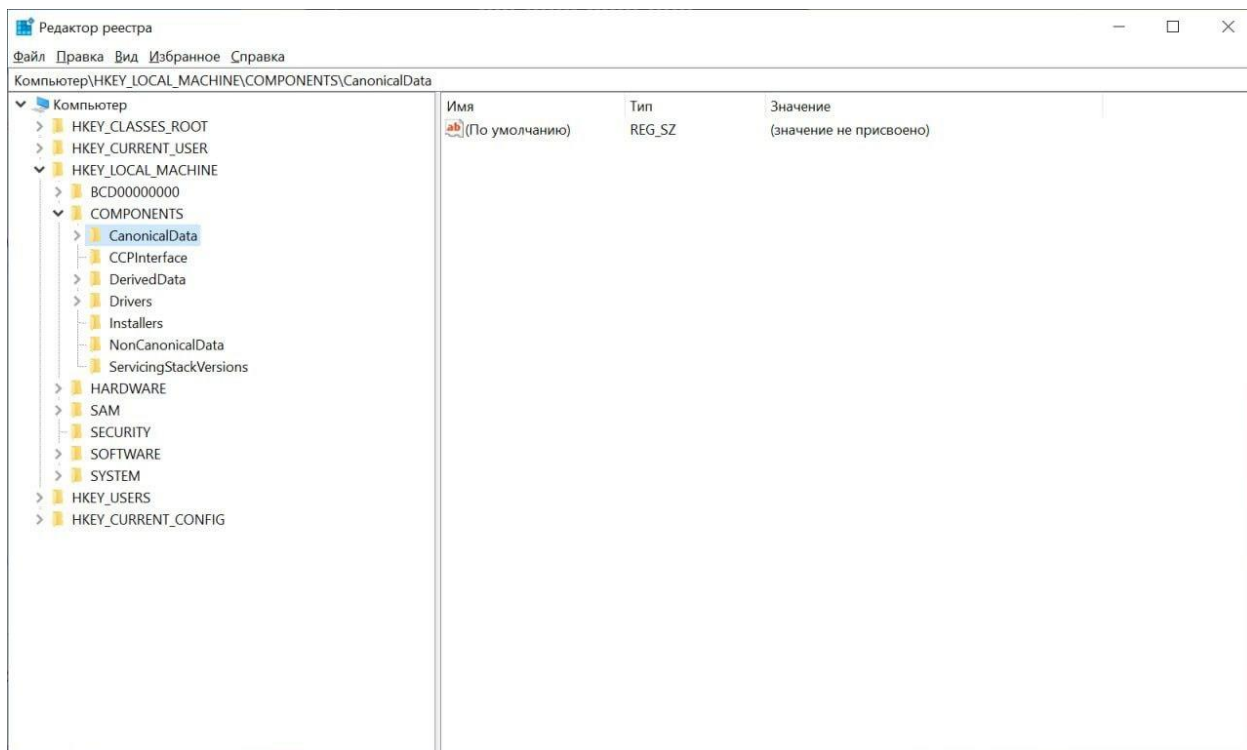
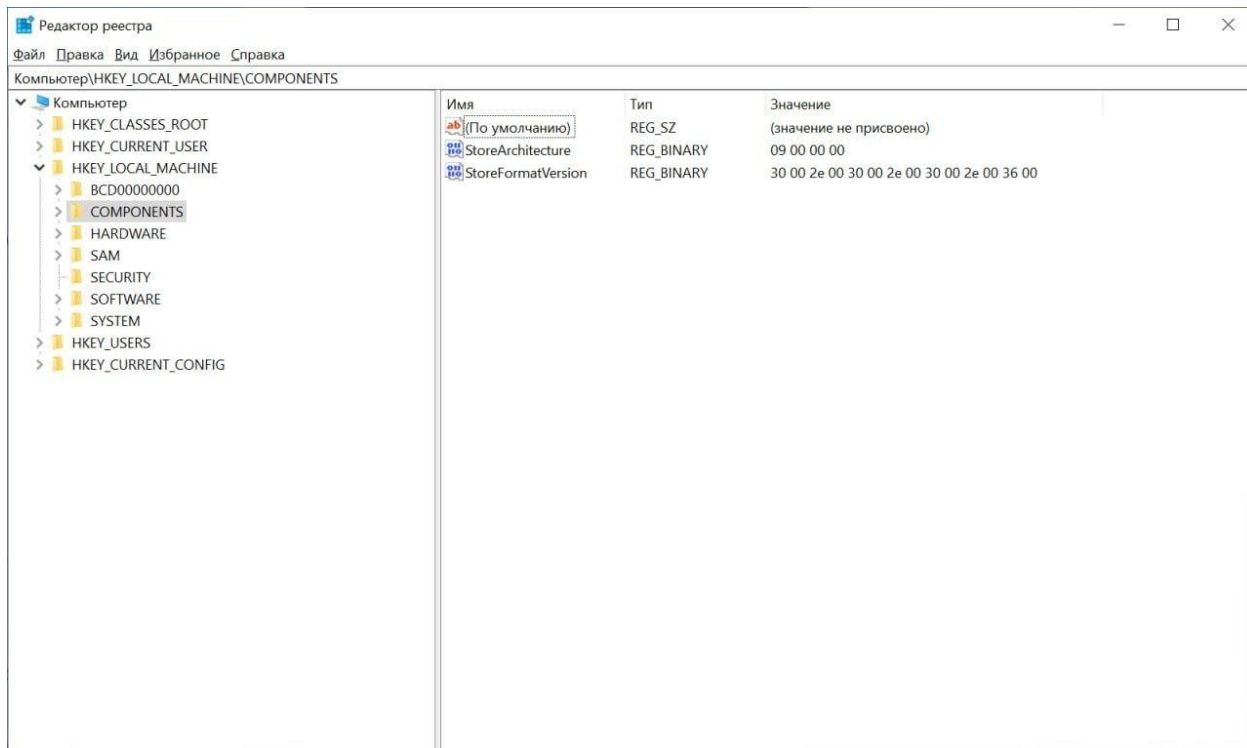
```
db.find({parent: "COMPONENTS"})
result:
tuple 5:
parent          COMPONENTS
name             CanonicalData
type             key
value_type       REG_SZ
value            None
tuple 4:
parent          COMPONENTS
name             StoreArchitecture
type             value
value_type       REG_BINARY
value            09000000
tuple 3:
parent          COMPONENTS
name             StoreFormatVersion
type             value
value_type       REG_BINARY
value            30002e0030002e0030002e003600
```

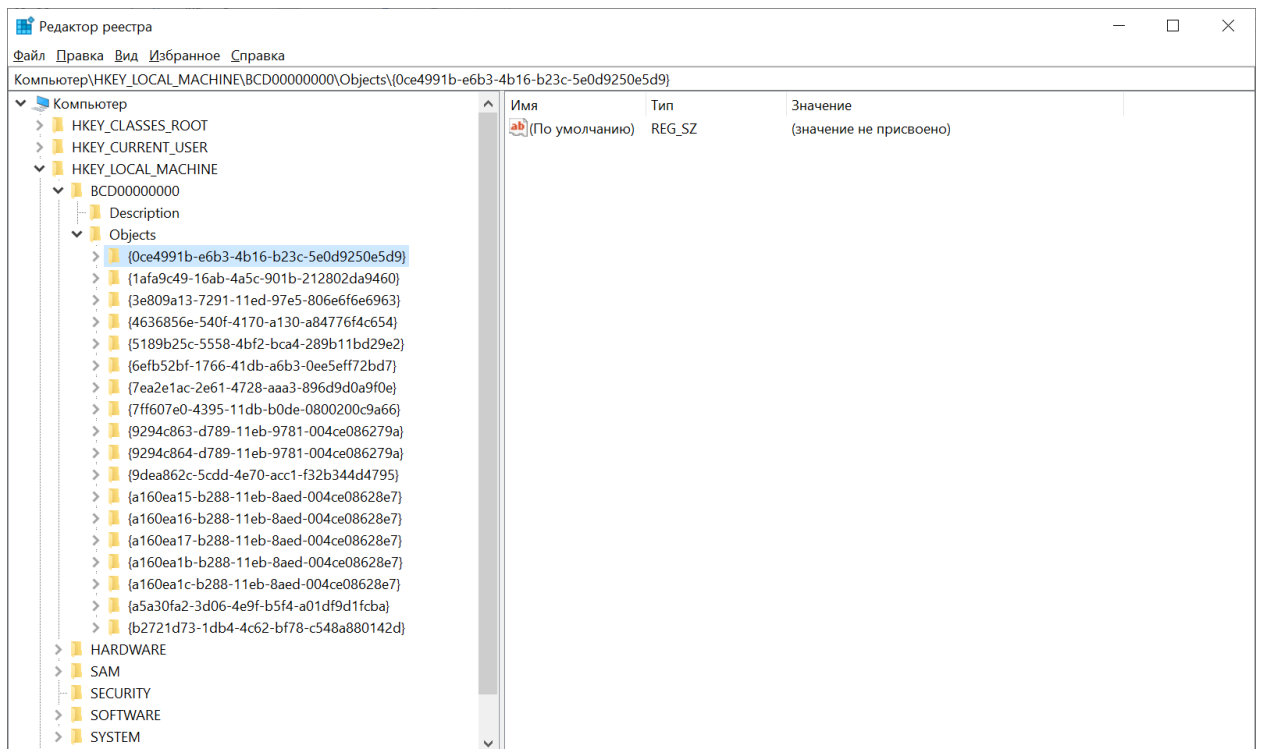
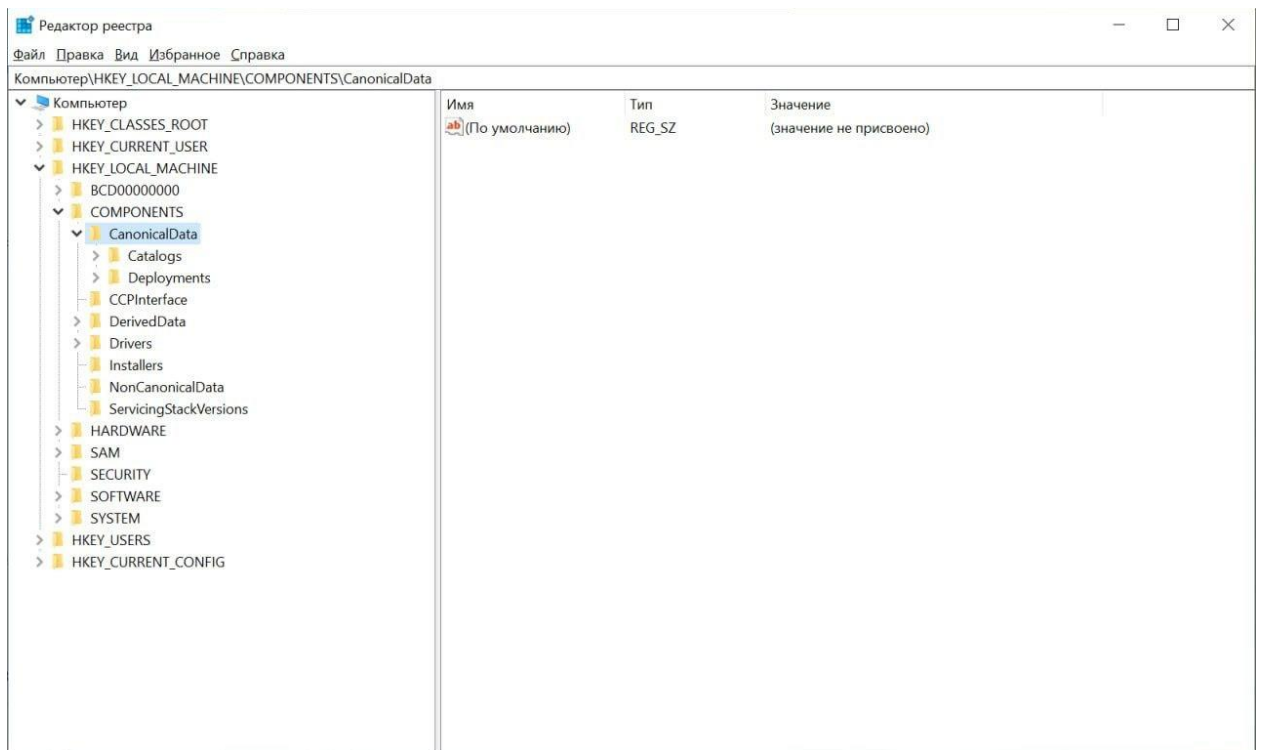
```

db.find({parent: "COMPONENTS", $or: [name: "StoreArchitecture", value_type: "REG_SZ"]})
result:
tuple 5:
parent      COMPONENTS
name        CanonicalData
type        key
value_type   REG_SZ
value       None
tuple 4:
parent      COMPONENTS
name        StoreArchitecture
type        value
value_type   REG_BINARY
value       09000000
■

db.find({parent: "Ari", value: "1", value_type: "REG_DWORD_LITTLE_ENDIAN"})
result:
tuple 499:
parent      Ari
name        SettingsVersion
type        value
value_type   REG_DWORD_LITTLE_ENDIAN
value       1

```





Во время выполнения задания был обнаружен недостаток лексера, реализованного во второй лабораторной работе. Как видно на скриншоте ниже, названия элементов могут содержать в себе не только буквы, цифры, нижние подчеркивания, но и любые другие символы. Лексер поддерживает работу только с ограниченным набором символов (строка проверяется регулярным выражением), вследствие чего при работе с элементами, имя которых содержит и другие символы, программа выдает синтаксическую ошибку. Для ее исправления необходимо включить в регулярное выражение больший диапазон символов, тогда она будет работать корректно.

Также был обнаружен еще один недостаток, связанный со значениями данных, хранимых в реестре. Значения могут представлять из себя очень длинную последовательность двоичных данных, вследствие чего при обработке некоторых значений может возникать ошибка.

## tcpdump

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
01:28:31.548571 IP localhost.34536 > localhost.3939: Flags [S], seq 3021582130, win 65495, options [mss 65495,sackOK,TS val 1360303485 ecr 0,nop,wscale 7], length 0
01:28:31.548581 IP localhost.3939 > localhost.34536: Flags [S.], seq 3472966119, ack 3021582131, win 65483, options [mss 65495,sackOK,TS val 1360303485 ecr 1360303485,nop,wscale 7], length 0
01:28:31.548590 IP localhost.34536 > localhost.3939: Flags [P.], seq 1:2, ack 1, win 512, options [nop,nop,TS val 1360316990 ecr 1360303485], length 1
01:28:45.053668 IP localhost.34536 > localhost.3939: Flags [P.], seq 1:2, ack 1, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 1
01:28:45.053674 IP localhost.3939 > localhost.34536: Flags [P.], seq 2:3, ack 1, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 1
01:28:45.053680 IP localhost.34536 > localhost.3939: Flags [P.], seq 2:3, ack 1, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 1
01:28:45.053798 IP localhost.3939 > localhost.34536: Flags [P.], seq 23, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 0
01:28:45.053801 IP localhost.34536 > localhost.3939: Flags [P.], seq 23:50, ack 1, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 27
01:28:45.053808 IP localhost.3939 > localhost.34536: Flags [P.], seq 50, win 512, options [nop,nop,TS val 1360316990 ecr 1360316990], length 0
01:29:04.641777 IP localhost.34536 > localhost.3939: Flags [P.], seq 50:51, ack 171, win 512, options [nop,nop,TS val 1360336578 ecr 1360316991], length 1
01:29:04.641783 IP localhost.3939 > localhost.34536: Flags [P.], seq 51, win 512, options [nop,nop,TS val 1360336578 ecr 1360336578], length 0
01:29:04.641789 IP localhost.34536 > localhost.3939: Flags [P.], seq 51:52, ack 171, win 512, options [nop,nop,TS val 1360336578 ecr 1360336578], length 1
01:29:04.641931 IP localhost.3939 > localhost.34536: Flags [P.], seq 72, win 512, options [nop,nop,TS val 1360336578 ecr 1360336578], length 0
01:29:04.641934 IP localhost.34536 > localhost.3939: Flags [P.], seq 72:96, ack 171, win 512, options [nop,nop,TS val 1360336578 ecr 1360336578], length 24
01:29:04.641941 IP localhost.3939 > localhost.34536: Flags [P.], seq 96, win 512, options [nop,nop,TS val 1360336578 ecr 1360336578], length 0
01:29:57.782387 IP localhost.34536 > localhost.3939: Flags [P.], seq 96:98, ack 706, win 512, options [nop,nop,TS val 1360389719 ecr 1360336579], length 2
01:29:57.782396 IP localhost.3939 > localhost.34536: Flags [P.], seq 98, win 512, options [nop,nop,TS val 1360389719 ecr 1360389719], length 0
01:29:57.782405 IP localhost.34536 > localhost.3939: Flags [P.], seq 98:99, ack 706, win 512, options [nop,nop,TS val 1360389719 ecr 1360389719], length 1
01:29:57.782626 IP localhost.3939 > localhost.34536: Flags [P.], seq 119, win 512, options [nop,nop,TS val 1360389719 ecr 1360389719], length 0
01:29:57.782631 IP localhost.34536 > localhost.3939: Flags [P.], seq 119:234, ack 706, win 512, options [nop,nop,TS val 1360389719 ecr 1360389719], length 115
01:29:57.782672 IP localhost.3939 > localhost.34536: Flags [P.], seq 234, win 512, options [nop,nop,TS val 1360389719 ecr 1360389719], length 0
01:29:57.783995 IP localhost.34536 > localhost.3939: Flags [P.], seq 708:709, ack 234, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 1
01:29:57.784000 IP localhost.3939 > localhost.34536: Flags [P.], seq 709, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 0
01:29:57.784024 IP localhost.34536 > localhost.3939: Flags [P.], seq 709:711, ack 234, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 2
01:29:57.784029 IP localhost.3939 > localhost.34536: Flags [P.], seq 711, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 0
01:29:57.784052 IP localhost.34536 > localhost.3939: Flags [P.], seq 711:890, ack 234, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 179
01:29:57.784056 IP localhost.3939 > localhost.34536: Flags [P.], seq 890, win 511, options [nop,nop,TS val 1360389720 ecr 1360389720], length 0
01:29:57.784079 IP localhost.34536 > localhost.3939: Flags [P.], seq 890:891, ack 234, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 1
01:29:57.784083 IP localhost.3939 > localhost.34536: Flags [P.], seq 891, win 511, options [nop,nop,TS val 1360389720 ecr 1360389720], length 0
01:29:57.784106 IP localhost.34536 > localhost.3939: Flags [P.], seq 891:892, ack 234, win 512, options [nop,nop,TS val 1360389720 ecr 1360389720], length 1
01:29:57.784110 IP localhost.3939 > localhost.34536: Flags [P.], seq 892, win 511, options [nop,nop,TS val 1360389720 ecr 1360389720], length 0
```

**Вывод:** в результате выполнения лабораторной работы было реализовано два консольных приложения, обеспечивающих взаимодействие клиента и сервера и передачу данных с помощью protobuf. Кроме того, был разработан модуль на Python с использованием библиотеки winreg, который достает данные из реестра. Использование информации из реестра в качестве входных данных продемонстрировало недостатки созданного лексера.