

Системное программное обеспечение

Лабораторная работа 1.5

Вариант 9

Студент: Иван Чепрасов

Группа: Р33022

## Задание

Разработать способ организации данных в файле, позволяющий хранить, выбирать и гранулярно обновлять наборы записей общим объёмом от 10GB соответствующего варианту вида. Реализовать модуль или библиотеку для работы с ним в режиме курсора. Используя данный способ сериализации, воспользоваться существующей библиотекой для описания схемы и реализации модуля, обеспечивающего функционирование протокола обмена запросами создания, выборки, модификации и удаления данных, и результатами их выполнения. Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора некоторого подмножества языка запросов по выбору в соответствии с вариантом формы данных. Должна быть обеспечена возможность описания команд создания, выборки, модификации и удаления данных. Используя созданные модули разработать в виде консольного приложения две программы: клиентскую и серверную части. Серверная часть – получающая по сети запросы и операции описанного формата и выполняющая их над файлом, организованным в соответствии с разработанным способом. Имя фала данных для работы получать с аргументами командной строки, создавать новый в случае его отсутствия. Клиентская часть –получающая от пользователя команду, пересылающая её на сервер, получающая ответ и выводящая его в человекопонятном виде.

### Вариант 9:

Граф узлов с атрибутами; Xml

### Ссылка на исходный код лабораторной работы

<https://github.com/ivancheprasov/spoLab15>

### Вывод:

Во время выполнения лабораторной познакомился с языком запросов CYPHER и различными библиотеками C (синтаксическим анализатором Cypher Lint, библиотекой для работы с xml документами LibXml2), утилитой truncate и монитором производительности ubuntu. Также значительно улучшил навыки отладки, устраняя возникшие утечки памяти.

Столкнулся со следующими трудностями:

- Сложность продумывания всех нюансов файловой системы для борьбы с фрагментацией
- Появляющиеся требования и задачи в процессе сдачи заставили множество раз переработать части кода, заняться оптимизацией.