

Лабораторная работа №1

«Простейший протокол прикладного уровня»

Цель работы

Целью работы является знакомство с принципами разработки протоколов прикладного уровня и их реализацией на языке Go.

Исходные данные

В качестве примера предлагается исходный текст на языке Go для сервера и клиента, реализующих протокол вычисления среднего арифметического последовательности дробей:

<https://disk.yandex.ru/d/s3ETiId4CcJE1g>

Сборка примера

Для сборки проекта нужно распаковать архив и запустить скрипт в корне папки «sample»:

```
./build.sh
```

В процессе сборки понадобится соединение с Интернетом для скачивания зависимостей:

1. библиотека чтения из стандартного потока ввода <https://disk.yandex.ru/d/PIGzPr6oocTSfw> или <https://github.com/skorobogatov/input>;
2. библиотека порождения логов <https://github.com/mgutz/logxi/tree/master/v1>.

Запуск примера

Проект состоит из двух частей:

1. сервер, принимающий от клиентов последовательность дробей и вычисляющий среднее арифметическое;
2. клиент, осуществляющий взаимодействие с пользователем и сервером.

Исполняемые файлы для сервера и клиента находятся в каталоге «bin». Их нужно запускать каждый в своём окне терминала. В терминале, предназначенном для сервера, нужно прежде всего установить переменные окружения, контролирующие работу библиотеки логов:

```
export LOGXI=*  
export LOGXI_FORMAT=pretty,happy
```

Затем нужно запустить сервер:

```
./server
```

Клиент не использует библиотеку логов и запускается командой
./client

Командная строка клиента

Клиент взаимодействует с клиентом интерактивно через собственную командную строку. Его приглашение к вводу выглядит как

command=

Пользователю доступны три команды:

- «quit» – завершает работу клиента;
- «add» – запрашивает у пользователя числитель и знаменатель дроби и отправляет дробь на сервер;
- «avg» – отправляет на сервер запрос на вычисление среднего арифметического дробей, а затем получает от сервера результат и выводит его в стандартный поток вывода.

Структура проекта

Проект состоит из трёх пакетов: «server», «client» и «proto». Исходные тексты пакетов находятся в подкаталогах каталога «src». Из пакетов «server» и «client» получаются исполняемые файлы сервера и клиента, а пакет «proto» содержит объявления типов данные, представляющих сообщения протокола.

Документация к используемым технологиям и библиотекам

Клиент и сервер обмениваются тестовыми сообщениями в формате JSON. Документацию по JSON можно найти здесь: <http://www.json.org/json-ru.html>.

Исходные тексты клиента и сервера не работают напрямую с текстовыми строками сообщений в формате JSON. Вместо этого они оперируют структурами языка Go, объявленными в пакете «proto», а преобразования между строками и этими структурами осуществляют функции стандартного пакета «encoding/json»:
<https://golang.org/pkg/encoding/json>.

Клиент и сервер передают друг другу сообщения через протокол TCP. Для этого используется стандартный пакет «net»: <https://golang.org/pkg/net>.

Сервер осуществляет ведение лога через библиотеку logxi:
<https://github.com/mgutiz/logxi>

Задание

Выполнение лабораторной работы состоит из двух частей.

1. Разработать вариант протокола из таблиц 1–3. Протокол должен базироваться на текстовых сообщениях в формате JSON. Результатом разработки протокола должен быть набор типов языка Go, представляющих сообщения, и документация к ним в виде комментариев в исходном тексте.
 2. Написать на языке Go клиент и сервер, взаимодействующие по разработанному протоколу.
- Основные требования к клиенту и серверу:

1. полная проверка данных, получаемых из сети (необходимо учитывать, что сообщения могут приходить в неправильном формате и в неправильном порядке, а также могут содержать неправильные данные);
2. устойчивость к обрыву соединения;
3. возможность одновременного подключения нескольких клиентов к одному серверу;
4. сервер должен вести подробный лог всех ошибок, а также других важных событий (установка и завершение соединения с клиентом, приём и передача сообщений, и т.п.).

Сроки

ИУ9-21Б (староста Катя Дужеева)

- В день получения задания (пятница) на паре минимум 50% работы остальное до 23:59 — 3 балла.
- Понедельник до 23:59 — 2 балла.
- Четверг до 23:59 — 1 балл.

ИУ9-22Б (староста Максим Кежаев)

- В день получения задания (вторник) на паре минимум 50% работы остальное до 23:59 — 3 балла.
- Пятница до 23:59 — 2 балла.
- Понедельник до 23:59 — 1 балл.

Варианты

п/н	Студент	Группа	Условие задачи
1			Протокол передачи текстовых сообщений от одного клиента к другому (каждый клиент выбирает себе уникальное имя, имена используются в качестве адресов получателей сообщений).
2			Протокол многократного поиска подстрок в строке.
3			Протокол многократного вычисления значений полинома в различных точках.
4			Протокол вычисления длины ломаной линии на плоскости, заданной последовательностью точек.
5			Протокол проверки делимости целого числа, заданного последовательностью десятичных цифр, на 3.
6			Протокол подсчёта слов в предложении, заданном последовательностью символов. Слова разделяются одним или несколькими пробелами.

- 7 Протокол вычисления площадей кругов на плоскости, заданных центром и точкой на окружности.
- 8 Протокол поиска подпоследовательности целых чисел с максимальной суммой.
- 9 Протокол редактирования текстовой строки.
- 10 Протокол подсчёта количества уникальных слов в последовательности слов.
- 11 Протокол построения графа и вычисления количества компонент связности.
- 12 Протокол вычисления точки пересечения двух отрезков на плоскости.
- 13 Протокол работы с очередью целых чисел, разделяемой всеми клиентами.
- 14 Протокол широковещательной передачи текстовых сообщений (один клиент отправляет сообщение – все клиенты его получают).
- 15 Протокол вычисления наибольшего общего делителя и наименьшего общего кратного пары целых чисел.
- 16 Протокол сортировки массивов целых чисел.
- 17 Протокол поиска минимума и максимума в последовательности вещественных чисел.
- 18 Протокол многократного вычисления различных членов арифметической прогрессии.
- 19 Протокол вычисления скалярного произведения двух n -мерных векторов.
- 20 Протокол вычисления последовательности простых делителей целого числа.
- 21 Протокол получения имён файлов в указанном каталоге.
- 22 Протокол напоминаний (клиент просит сервер передать ему сообщение через определённое время).
- 23 Протокол вычисления количества вхождений указанной цифры в длинном целом числе, заданном последовательностью десятичных цифр.
- 24 Протокол поиска прямоугольника с максимальной площадью в последовательности прямоугольников, заданных координатами противоположных вершин.
- 25 Протокол вычисления значений различных тригонометрических функций.
- 26 Протокол вычисления наибольшей высоты, на которую поднимется камень, брошенный с земли под углом к горизонту.
- 27 Протокол редактирования массива целых чисел с возможностью вычисления суммы чисел на отрезке.
- 28 Протокол поиска минимального по площади прямоугольника, любая сторона которого параллельна одной из осей координат, и который содержит внутри себя последовательность точек.
- 29 Протокол поиска минимального по площади круга, покрывающего последовательность точек.
- 30 Протокол редактирования ломаной линии на плоскости.

- 31 Протокол работы с размещённой на сервере хеш-таблицей, отображающей строки в целые числа.
- 32 Протокол редактирования предложения с возможностью вставлять и удалять слова.
- 33 Протокол перевода чисел из одной системы счисления в другую.
- 34 Протокол редактирования квадратной вещественной матрицы с возможностью вычисления её определителя.
- 35 Протокол решения квадратных уравнений.
- 36 Протокол работы с разреженным массивом размера 2^{32} с возможностью вычисления максимального элемента на отрезке.
- 37 Протокол редактирования многоугольника на плоскости с возможностью проверки его выпуклости.
- 38 Протокол проверки сбалансированности круглых скобок в последовательности символов.
- 39 Протокол редактирования числа в Фибоначчиевой системе счисления.
- 40 Протокол вычисления количества пиков в последовательности целых чисел.
- 41 Протокол получения размеров изображения в формате JPEG, а также цветов отдельных точек. (Нужно воспользоваться пакетом "image" стандартной библиотеки.)
- 42 Протокол обхода дерева элементов HTML-страницы. (Нужно воспользоваться пакетом "golang.org/x/net/html".)
- 43 Протокол вычисления контрольных сумм текстовых строк. (Нужно воспользоваться пакетом "hash/crc64" стандартной библиотеки.)
- 44 Протокол получения оглавления tar-архива. (Нужно воспользоваться пакетом "archive/tar" стандартной библиотеки.)