

Семестр 4 (25.10.2019), Зачет

Общая постановка задачи

get.

Требуется, разработать клиент-серверное приложение (программы `client` и `server`). Клиент взаимодействует с неблокирующим сервером по протоколу TCP. Все сетевые соединения обрабатываются сервером независимо друг от друга.

Клиент

Работа клиента осуществляется в цикле пока не будет закрыт стандартный поток ввода или не будет закрыто сетевое соединение.

На каждой итерации цикла клиент пытается прочитать строку введенную пользователем. Корректно введенная строка содержит текстовое представление запроса к серверу. Запросы бывают двух типов.

Тип 1. Строка имеет следующий формат

```
send <num> ... <num> end.
```

Из подобной строки может быть извлечена (возможно пустая) последовательность чисел.

Например, строка `send end.` содержит пустую последовательность чисел, а строка `send 1.1 1.2 1.3 end.` содержит последовательность 1.1, 1.2, 1.3.

Запрос типа 1 отправляется на сервер следующим образом. Передается один байт со значением 1. Передается длина последовательности (тип `int`). Если это необходимо, в виде массива передаются элементы последовательности (тип `double[]`). В ответ на подобный запрос сервер не посылает сообщений.

Тип 2. Строка имеет следующий формат

Запрос типа 2 отправляется на сервер в виде одного байта со значением 2. После отсылки запроса этого типа клиент пытается получить ответ от сервера, который печатает в стандартный поток вывода.

Сервер

Каждое сетевое соединение имеет состояние, которое представляет собой последовательность чисел. Начальным состоянием является пустая последовательность.

Каждое сетевое соединение обрабатывается в отдельном цикле пока это соединение не будет закрыто.

При получении запроса типа 1 происходит изменение состояния. Если текущим состоянием является последовательность x_1, \dots, x_n , а вместе с запросом была передана последовательность y_1, \dots, y_m , то новым состоянием становится последовательность $x_1, \dots, x_n, y_1, \dots, y_m$.

При получении запроса типа 2 сервер вычисляет некоторую характеристику состояния (тип `int`), которую посылает клиенту.

Характеристики состояния

Задача 1. Количество возрастающих участков последовательности.

Задача 2. Длина возрастающего участка последовательности с наибольшим количеством элементов.

Задача 3. Максимальное расстояние (количество элементов) между двумя соседними локальными минимумами последовательности.