

Семестр 4 (2019), Зачет

Общая постановка задачи

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 в цикле

- печатает в стандартный поток вывода приглашение `>` ;
- считывает из стандартного потока ввода введенную пользователем строку;
- обрабатывает введенную строку;
- печатает в стандартный поток вывода результат обработки.

Обработка введенной пользователем строки состоит в следующем.

Если введенная строка представляет собой разделенную пробелами последовательность целых чисел (значений типа `int`), то эта последовательность в виде массива передается на сервер. Внимание, отсутствует предварительная отправка размера массива на сервер.

В ответ сервер отправляет клиенту также массив целых чисел (значений типа `int`). При этом, сначала отправляется длина массива в виде значения типа `int`, а за тем отправляются элементы массива. Полученный массив клиент печатает в следующем формате. Печатается длина массива, печатается символ `:` и пробел, через пробел печатаются все элементы массива.

Если введенная строка не представляет собой разделенную пробелами последовательность целых чисел, то печатается слово `Error`.

Пример работы клиента (задача 1).

```
$ ./client
> 1 2 3 4 5 -6 7 8<Enter>
6: 1 2 3 4 5 -6
> 9 10 -1 -2<Enter>
5: 7 8 9 10 -1
> abcd<Enter>
Error
...
> Ctrl-D
$
```

При запуске сервера ему в качестве аргумента командной строки передается положительное целое число N .

Пример.

```
$ ./server 100
```

В дальнейшем, при считывании данных от клиента сервер должен пытаться считывать за один раз (один системный вызов `read`) массив значений типа `int` длины N .

Обработчик каждого сетевого соединения на сервере обладает состоянием. Состояние представляет собой последовательность целых чисел. При создании соединения состояние инициализируется пустой последовательностью.

Функционирование обработчика сетевого соединения сводится к многократному выполнению следующей последовательности шагов.

1. Пусть текущее состояние имеет вид

$$x_1, \dots, x_m.$$

2. Предположим, что от клиента была получена последовательность чисел

$$x_{m+1}, \dots, x_{m+n} \quad (n \leq N).$$

3. Новым текущим состоянием становится последовательность чисел

$$x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n}.$$

4. Предположим, что текущее состояние может быть представлено в виде

$$x_1, \dots, x_j, x_{j+1}, \dots, x_{m+n}, \quad (1)$$

где последовательность x_1, \dots, x_j является допустимой, а последовательности вида x_1, \dots, x_s ($s < j$) не являются допустимыми. Определение допустимой последовательности вводится индивидуально для каждой отдельной задачи.

В этом случае, клиенту отправляется последовательность чисел

$$x_1, \dots, x_j,$$

а текущим состоянием становится последовательность чисел

$$x_{j+1}, \dots, x_{m+n}.$$

5. Если текущее состояние не может быть представлено в виде (1), то клиенту отправляется число 0 как значение типа `int`. Это интерпретируется как отправка пустой последовательности.

Критерии допустимости

Определение допустимой последовательности целых чисел вида

$$x_1, x_2, \dots, x_j \quad (j > 1)$$

для каждой отдельной задачи.

Задача 1. Число $x_j < 0$.

Задача 2. Число $|x_j|$ – простое число.

Задача 3. Числа $|x_{j-1}|$ и $|x_j|$ – взаимнопростые.

Задача 4. Число $|x_j|$ является степенью чис-

ла 2.

Задача 5. Число $|x_j|$ является степенью числа 3.

Задача 6. Наибольший общий делитель чисел $|x_1|, |x_2|, \dots, |x_j|$ равен 1.

Задача 7. Сумма всех элементов последовательности равна нулю.

Задача 8. Сумма всех элементов последовательности является простым числом.

Задача 9. Наименьшее общее кратное всех положительных элементов последовательности больше 100.