

# Системное программирование

## Лабораторная работа № 2

### Процессы, потоки и средства межпроцессного взаимодействия

Написать на языке C две программы для POSIX-совместимой ОС:

- сервер, поддерживающий заданный вариантом тип многозадачности (Табл. 2), транспортный протокол (Табл. 3) и прикладной протокол (Табл. 4);
- клиент, поддерживающий заданный вариантом протокол и предназначенный для тестирования сервера.

Программы должны представлять собой консольные утилиты, настройка работы которых осуществляется путем передачи аргументов в строке запуска и/или с помощью переменных окружения:

```
lab2server [ опции ]  
lab2client [ опции ]
```

Тип многозадачности серверной программы определяется согласно варианту из Табл. 2 путем подсчета количества букв в фамилии студента, выполняющего лабораторную работу. Транспортный протокол, поддерживаемый серверной и клиентской программами, определяется согласно варианту из Табл. 3 путем подсчета количества букв в имени студента, выполняющего лабораторную работу. Прикладной протокол, поддерживаемый серверной и клиентской программами, определяется согласно номеру варианта из Табл. 4. Серверная и клиентская программы должны поддерживать опции командной строки и переменные среды, перечисленные в Табл. 1 (расширение списка опций не запрещается).

Серверная программа должна обрабатывать поступление сигналов SIGINT, SIGTERM и SIGQUIT и корректно завершать работу при их поступлении. Также должен обрабатываться сигнал SIGUSR1, при поступлении которого серверная программа должна выводить в лог и в стандартный поток ошибок накопленную статистику — время работы, количество успешно обслуженных запросов и количество запросов, выполнение которых привело к ошибкам.

Серверная программа должна поддерживать запись отладочных сообщений и сообщений об ошибках в лог-файл, расположение которого определяется с помощью опции командной строки -l, соответствующей переменной окружения или умалчиваемым значением (см. Табл. 1). Сообщения, добавляемые в лог-файл, должны содержать временные метки в формате «ДД.ММ.ГГ чч:мм:сс». В лог также должны выводиться сообщения о поступлении нового запроса и завершении обслуживания очередного запроса.

Серверная программа должна поддерживать возможность запуска в режиме демона. Чтобы запустить её в этом режиме необходимо указать опцию -d (см. Табл. 1).

Адрес (IPv4) и порт, на котором сервер должен ожидать поступления запросов, передаются через опции -a и -p, если же они не заданы при запуске, тогда проверяются значения переменных окружения. Если же и переменные окружения не заданы, программа должна использовать значения по умолчанию (см. Табл. 1).

Поступивший запрос серверная программа должна передавать для обработки в рабочую задачу (процесс или поток в зависимости от варианта, см. Табл. 2) и не блокироваться на время обработки запросов.

Клиентская программа, с помощью которой выполняется проверка взаимодействия с сервером, должна позволять задать IP-адрес и номер порта и ввести запрос к серверу (в произвольной, но приемлемой форме; наиболее предпочтителен вариант ввода запроса через параметры командной строки).

Проект (исходные коды, заголовочные файлы, Makefile и прочие файлы, необходимые для сборки) должен содержаться в отдельном каталоге и собираться с помощью стандартной утилиты make.

Демонстрация работы программ должна выполняться в терминале с оболочкой `bash/csh/*sh` в POSIX-совместимой ОС (Linux, Unix, macOS, \*nix, ...). Программы должны быть скомпилированы с помощью `gcc` или `clang`. Для демонстрации работы программ рекомендуется открыть несколько окон терминалов, в одном из которых следует запустить программу-сервер, в другом — открыть лог-файл сервера (например, с помощью `tail -f имя_файла`), в третьем — запускать программу-клиент.

Отчет должен содержать:

- правильно оформленную титульную страницу;
- задание;
- Make-файл;
- текст программы с комментариями;
- примеры работы программ (скриншоты).

**Замечание 1.** При выполнении лабораторной работы следует использовать функции стандартной библиотеки C и системные вызовы операционной системы. Использовать ввод-вывод в стиле C++ (классы `ifstream/ofstream/...`) запрещено. Использовать контейнеры и алгоритмы STL (`<string>`, `<vector>`, `<map>`, ...) запрещено. Использовать сторонние библиотеки (кроме `pthread`) запрещено.

**Замечание 2.** В программах должна присутствовать обработка ошибок: в случаях, если пользователь задал неверную комбинацию опций, указал файлы, которые невозможно открыть, и т.д. программа должна выдавать диагностическое сообщение на консоль (в стандартный поток ошибок и/или лог-файл), прежде чем завершиться. Получение неверно сформированного запроса не должно приводить к завершению программы-сервера.

**Замечание 3.** Категорически запрещается использовать статические массивы (с размерами, заданными на этапе компиляции) для входных данных. Для хранения входных данных необходимо использовать динамическую память и определять объем необходимой памяти в зависимости от объема входных данных.

**Замечание 4.** В вариантах, использующих в качестве транспортного протокола TCP, клиент должен закрывать соединение в сторону сервера после выдачи запроса, а сервер — после выполнения запроса клиента.

**Замечание 5.** Серверная и клиентская программы должны быть написаны таким образом, чтобы допускать одновременный запуск множества экземпляров (на разных интерфейсах и/или портах).

**Замечание 6.** В вариантах, предусматривающих запись чисел в десятичной или шестнадцатеричной системах счисления, признаком использования шестнадцатеричной системы счисления служит префикс «0x».

**Замечание 7.** Пробельными символами в лабораторной работе считаются собственно пробел `SP` (`\x20`), горизонтальная табуляция `HT` (`\x09` или `\t`), вертикальная табуляция `VT` (`\x0B`), прогон страницы `FF` (`\x0C`) и возврат каретки `CR` (`\x0D` или `\r`). Символ перевода строки `LF` (`\x0A` или `\n`) также относится к пробельным, но в большинстве вариантов имеет специальный смысл (маркер конца строки) и поэтому не может использоваться для разделения слов в строке.

**Замечание 8.** В случаях, когда задана и переменная среды, и опция командной строки, более приоритетной считается опция командной строки.

**Замечание 9.** Подсчет букв в имени и фамилии выполняется в русской транскрипции.

**Замечание 10.** Для обработки сигналов не следует использовать функцию `signal()`. Вся обработка сигналов должна выполняться только с помощью функций интерфейса POSIX `sig*set()/sigprocmask()/sigaction()` и т.д.

Опция	Переменная среды	Назначение	Значение по умолчанию	Поддерживается
-w N	L2WAIT	Имитировать работу путем приостановки обслуживающего запрос процесса/потока на N секунд. Если опция не задана, обслуживать запрос без задержки.	0	Сервером
-d		Работа в режиме демона.		Сервером
-l /path/to/log	L2LOGFILE	Путь к лог-файлу.	/tmp/lab2.log	Сервером
-a ip	L2ADDR	Адрес, на котором слушает сервер и к которому подключается клиент.		Сервером и клиентом
-p port	L2PORT	Порт, на котором слушает сервер и к которому подключается клиент.		Сервером и клиентом
-v		Вывод версии программы.		Сервером и клиентом
-h		Вывод справки по опциям.		Сервером и клиентом

Таблица 1. Переменные среды и опции командной строки, поддерживаемые программой-сервером и программой-клиентом

Вариант (условие)	Тип многозадачности
Количество букв в фамилии студента, выполняющего работу, <b>нечетное</b> .	Многопроцессность (создание рабочих задач с помощью вызова fork).
Количество букв в в фамилии студента, выполняющего работу, <b>четное</b> .	Многопоточность (создание рабочих потоков средствами библиотеки libpthread).

Таблица 2. Тип многозадачности программы-сервера

Вариант (условие)	Транспортный протокол
Количество букв в имени студента, выполняющего работу, <b>нечетное</b> .	UDP
Количество букв в в имени студента, выполняющего работу, <b>четное</b> .	TCP

Таблица 3. Транспортный протокол, поддерживаемый программой-сервером и программой-клиентом

Вариант (номер)	Описание прикладного протокола
1	<p><i>Запрос:</i> строка, содержащая список целых чисел в десятичной или шестнадцатеричной системах счисления, разделенных пробельными символами (возможно, несколькими подряд), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая те числа из списка, которые являются простыми (с сохранением порядка следования в запросе, разделитель — пробел), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где</p>

	N — код ошибки.
2	<p><i>Запрос:</i> 2 строки, каждая из которых завершается символом LF. Первая строка содержит список вещественных чисел в обычном или научном формате, разделенных пробелами или символами табуляции (возможно, несколькими подряд). Вторая строка содержит способ округления (в сторону минус бесконечности, в сторону плюс бесконечности, в сторону нуля).</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая округленные по заданному способу вещественные числа (с сохранением порядка следования в запросе, разделитель — пробел), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
3	<p><i>Запрос:</i> 7 байтов, первые 6 байтов содержат некоторый MAC-адрес, последний байт кодирует тип запроса (добавить, удалить, проверить).</p> <p><i>Ответ, если ошибок не было:</i> для операций добавления и удаления — 1 байт со значением 0; для операции проверки — 1 байт со значением 0, если указанный MAC-адрес не добавлен, и со значением 1 — если добавлен.</p> <p><i>Ответ, если были ошибки:</i> 1 байт со значением, отличным от 0 и 1, кодирующим ошибку.</p> <p><i>Дополнительные требования:</i> добавленные MAC-адреса должны сохраняться сервером между перезапусками.</p>
4	<p><i>Запрос:</i> 24 байта, содержащих 3 пары вещественных чисел одинарной точности (float), задающих координаты вершин некоторого треугольника на плоскости.</p> <p><i>Ответ, если ошибок не было:</i> 5 байтов, первый из которых — 0, а остальные четыре содержат вещественное число одинарной точности, представляющее собой периметр треугольника, заданного координатами в запросе.</p> <p><i>Ответ, если были ошибки:</i> 1 байт со значением, отличным от 0, кодирующим ошибку.</p>
5	<p><i>Запрос:</i> строка, содержащая список целых чисел в десятичной или шестнадцатеричной системах счисления, разделенных пробелами или символами табуляции (возможно, несколькими подряд), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая отсортированный список чисел (разделитель — пробел, критерий сортировки — по убыванию количества разрядов в десятичном представлении числа), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
6	<p><i>Запрос:</i> 2 строки, каждая из которых завершается символом LF. Первая строка содержит команду (добавить, удалить, получить). Вторая строка содержит список вещественных чисел в обычном или научном формате, разделенных пробелами или символами табуляции (возможно, несколькими подряд) в случае команд добавления или удаления, и целое число N — в случае команды получения.</p> <p><i>Ответ, если ошибок не было:</i> в случае команд добавления и удаления — строка «OK», завершающаяся символом LF; в случае команды получения — строка, содержащая N случайно выбранных из множества добавленных на момент поступления запроса чисел, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> множество добавленных чисел должно сохраняться сервером между перезапусками.</p>
7	<p><i>Запрос:</i> строка, содержащая целое число N в десятичной записи и два вещественных числа F и T в обычном или научном формате, разделенных пробельными символами, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая N случайных, не повторяющихся с момента первого запуска вещественных чисел из диапазона F .. T, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> история сгенерированных чисел должна сохраняться между перезапусками.</p>

8	<p><i>Запрос:</i> строка, содержащая слова, разделенные пробельными символами, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая анаграммы слов, переданных в запросе, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
9	<p><i>Запрос:</i> строка, содержащая целые числа в десятичной или шестнадцатеричной записи, разделенные пробельными символами, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка «ALL», если все числа являются какими-либо членами последовательности Фибоначчи (не обязательно следующими подряд); строка «SOME», если некоторые из чисел являются членами последовательности Фибоначчи; «NONE», если членов последовательности Фибоначчи среди чисел, переданных в запросе, не было.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
10	<p><i>Запрос:</i> строка, завершающаяся символом LF, содержащая либо «GET имя_переменной», либо «SET имя_переменной=значение», либо «DELETE имя_переменной», причем в имени переменной не может содержаться знака равенства.</p> <p><i>Ответ, если ошибок не было:</i> на запрос «GET имя_переменной» возвращается строка вида «FOUND имя_переменной=значение» или строка вида «NOT FOUND»; на запрос «SET имя_переменной=значение» возвращается строка «OK»; на запрос «DELETE имя_переменной» возвращается строка «OK»; в каждом случае строки завершаются символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> набор переменных и их значения должны сохраняться сервером между перезапусками.</p>
11	<p><i>Запрос:</i> 5 байтов, первые 4 байта содержат некоторый IP-адрес (IPv4), последний байт кодирует тип запроса (добавить, удалить, проверить).</p> <p><i>Ответ, если ошибок не было:</i> для операций добавления и удаления — 1 байт со значением 0; для операции проверки — 1 байт со значением 0, если указанный IP-адрес не добавлен, и со значением 1 — если добавлен.</p> <p><i>Ответ, если были ошибки:</i> 1 байт со значением, отличным от 0 и 1, кодирующим ошибку.</p> <p><i>Дополнительные требования:</i> добавленные IP-адреса должны сохраняться сервером между перезапусками.</p>
12	<p><i>Запрос:</i> 2 строки, завершающихся символом LF. Первая содержит двухбуквенный селектор языка («EN» либо «RU»), вторая — целое число в десятичной системе счисления.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая число, переданное в запросе, прописью (обеспечить поддержку как минимум до дециллиона по короткой шкале включительно), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
13	<p><i>Запрос:</i> строка, содержащая целые числа в десятичной или шестнадцатеричной записи, разделенные пробельными символами, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка «ALL», если все числа являются центральными многоугольными числами; строка «SOME», если некоторые из чисел являются центральными многоугольными числами; «NONE», если среди чисел, переданных в запросе, центральных многоугольных чисел не было.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
14	<p><i>Запрос:</i> строка, содержащая список из трех пар вещественных чисел в обычном или научном формате, разделенных пробелами или символами табуляции (возможно, несколькими подряд), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая вещественное число, представляющее собой площадь треугольника, заданного координатами на</p>

	<p>плоскости (парами чисел в запросе), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
15	<p><i>Запрос:</i> строка, содержащая список целых чисел в десятичной или шестнадцатеричной системах счисления, разделенных пробелами или символами табуляции (возможно, несколькими подряд), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая отсортированный список чисел (разделитель — пробел, критерий сортировки — по возрастанию количества единичных разрядов в двоичном представлении числа), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
16	<p><i>Запрос:</i> от 2 до 256 байтов, первый из которых содержит количество байтов данных.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая двоичное представление каждого байта данных (байты отделяются друг от друга пробелами), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
17	<p><i>Запрос:</i> строка, содержащая арифметическое выражение с целыми и/или вещественными числами и операциями сложения, вычитания, деления, умножения и скобками, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая результат вычисления выражения, переданного в запросе, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Указание:</i> для вычисления арифметических выражений можно воспользоваться стандартными утилитами dc или bc.</p>
18	<p><i>Запрос:</i> от 3 до 65536 байтов, первые два из которых кодируют количество байтов данных, остальные представляют собой байты данных.</p> <p><i>Ответ, если ошибок не было:</i> 3 байта, первый из которых нулевой, а другие два содержат контрольную сумму байтов данных, переданных в запросе, подсчитанную по алгоритму CRC-16-ANSI.</p> <p><i>Ответ, если были ошибки:</i> 3 байта, первый из которых ненулевой и содержит код ошибки, два других — нулевые.</p>
19	<p><i>Запрос:</i> строка, содержащая имя хоста или домена, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая список IP-адресов, в которые разрешается переданное в запросе имя, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
20	<p><i>Запрос:</i> строка, содержащая 16 цифр номера банковской карты (допускаются дефисы или пробельные символы между группами четырех цифр), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> «OK», если номер корректный согласно алгоритму проверки ISIN, «FAILED», если номер некорректный.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
21	<p><i>Запрос:</i> строка «PUSH число», где число — вещественное или целое число, либо строка «POP». Строки завершаются символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка «OK» в случае запроса «PUSH», помещающего число в стек, или строка, содержащая извлеченное из стека число в случае запроса «POP». Строки завершаются символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> по умолчанию для каждого клиента сервер должен поддерживать свой стек (клиенты идентифицируются по IP-адресу отправителя запроса). Необходимо реализовать возможность запустить сервер в режиме общего для всех клиентов стека.</p>

22	<p><i>Запрос:</i> строка, содержащая 11 цифр номера СНИЛС (допускаются дефисы и пробельные символы между группами цифр), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> «OK», если номер корректный согласно алгоритму проверки контрольного числа СНИЛС, «FAILED», если номер некорректный.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
23	<p><i>Запрос:</i> строка, содержащая арифметическое выражение в инфиксной записи, включающее целые и вещественные числа, знаки арифметических операций и скобки, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая выражение из запроса, преобразованное в постфиксную (обратную польскую) запись, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
24	<p><i>Запрос:</i> строка, содержащая выражение вида <i>OP X1 ... XN</i>, завершающаяся символом LF, где <i>OP</i> — одна из поразрядных операций AND, OR, XOR, NAND или NOR, <i>X1 ... XN</i> — целочисленные операнды, записанные в десятичной или шестнадцатеричной форме, разделенные пробельными символами.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая результат операции из запроса, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
25	<p><i>Запрос:</i> строка, содержащая выражение вида «RGB R G B» или «HSL H S L», завершающаяся символом LF, где R, G, B, H, S, L — целые числа, компоненты соответствующих цветовых пространств, записанные в десятичной или шестнадцатеричной форме, разделенные пробельными символами.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая результат преобразования из одного цветового пространства в другое в таком же формате, что и строка запроса, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
26	<p><i>Запрос:</i> строка, содержащая разделенные пробельными символами числа, записанные арабскими цифрами (в десятичной системе счисления) и римскими цифрами, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая разделенные пробелами переводы чисел (из десятичной системы счисления в римскую и наоборот), завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
27	<p><i>Запрос:</i> от 3 до 65536 байтов, первые два из которых кодируют количество байтов данных, остальные представляют собой байты данных.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая представление байтов данных, переданных в запросе, в кодировке base64, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
28	<p><i>Запрос:</i> строка вида «CIDR1 in CIDR2», завершающаяся символом LF, где CIDR1 и CIDR2 — IPv4-сети, записанные в формате CIDR.</p> <p><i>Ответ, если ошибок не было:</i> завершающаяся символом LF строка, содержащая «YES», если CIDR1 является подсетью CIDR2, или «NO», если не является.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
29	<p><i>Запрос:</i> строка, содержащая 17 знаков номера VIN, завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> «OK», если номер корректный согласно алгоритму проверки, утвержденному стандартом ISO 3779-1983, «FAILED», если номер некорректный.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
30	<p><i>Запрос:</i> строка «PUT число», где число — вещественное или целое число, либо</p>

	<p>строка «GET». Строки завершаются символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка «OK» в случае запроса «PUT», помещающего число в очередь, или строка, содержащая извлеченное из очереди число в случае запроса «GET». Строки завершаются символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> по умолчанию сервер поддерживает общую для всех клиентов очередь. Необходимо реализовать возможность запустить сервер в режиме отдельной очереди для каждого клиента (клиенты идентифицируются по IP-адресу отправителя запроса).</p>
31	<p><i>Запрос:</i> строка, содержащая список пар вещественных чисел в обычном или научном формате, разделенных пробелами или символами табуляции (возможно, несколькими подряд), завершающаяся символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка, содержащая вещественное число, представляющее собой площадь под ломаной линией, заданной координатами на плоскости — отсортированными по величине абсциссы парами чисел в запросе, завершающаяся символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p>
32	<p><i>Запрос:</i> строка «ADD число», где число — вещественное или целое число, либо строка «GET». Строки завершаются символом LF.</p> <p><i>Ответ, если ошибок не было:</i> строка «OK» в случае запроса «ADD», помещающего число в множество, или строка, содержащая случайное извлеченное из множества число в случае запроса «GET». Строки завершаются символом LF.</p> <p><i>Ответ, если были ошибки:</i> строка «ERROR N», завершающаяся символом LF, где N — код ошибки.</p> <p><i>Дополнительные требования:</i> по умолчанию сервер поддерживает общее для всех клиентов множество. Необходимо реализовать возможность запустить сервер в режиме отдельного множества для каждого клиента (клиенты идентифицируются по IP-адресу отправителя запроса).</p>

**Таблица 4. Прикладной протокол, поддерживаемый программой-сервером и программой-клиентом**