

# Сетевое взаимодействие на языке Ляпас

## Содержание

<b>1</b>	<b>Описание функций</b>	<b>2</b>
<b>2</b>	<b>Примеры использования</b>	<b>4</b>

## 1 Описание функций

`socket(t/s)` - создания сокета. Исходные данные: `t` - тип логического соединения. Результат - номер дескриптора или  $2^{32} - 1$  в случае отказа

`bind(s,F1/e)` - назначение сокету локального адреса. Исходные данные: `s` - сокет, `F1` - комплекс, задающий структуру сокета адреса. Результат - в случае успешного выполнения подпрограмма возвращает значение равное 0, иначе - код ошибки.

`set_sockaddr_port(F1,p)` - установление значения порта в сокетах адреса. Исходные данные: `F1` - комплекс длиной 16 символов, задающий структуру сокета адреса, `p` - номер порта.

`set_sockaddr_ip(F1,F2)` - установление значения сетевого адреса в сокетах адреса. Исходные данные: `F1` - комплекс длиной 16 символов, задающий структуру сокета адреса, `F2` - комплекс, задающий сетевой адрес, записанный в десятичном виде с разделительным символом "." между байтами.

`listen(s,b/e)` - установка сокета в режим "прослушивания". Исходные данные: `s` - сокет для прослушивания, `b` - целое положительное число, определяющее как много запросов может быть принято на сокет. Результат - в случае успешного выполнения подпрограмма возвращает значение равное 0, иначе - код ошибки.

`accept(s,F1/s)` - принятие соединения на сокет. Исходные данные: `s` - слушающий сокет, `F1` - комплекс, описывающий сокета адрес клиента. Результат - номер дескриптора или  $2^{32} - 1$  в случае отказа.

`get_sockaddr_ip(F1,F2)` - взятие значения сетевого адреса из сокета адреса. Исходные данные: `F1` - комплекс, задающий сокета адрес, `F2` - комплекс, задающий сетевой адрес. Результат - в комплексе `F2` будет записан сетевой адрес с использованием разделительного символа "." между байтами сетевого адреса.

`connect(s,F1/e)` - установление соединения с сервером. Исходные данные: `s` - сокет, `F1` - комплекс, задающий сокета адрес сервера. Результат - значение 0 или код ошибки.

Таблица 1: Структура сокета адреса

номер байта	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
значение	2	0	port		IP address				0	0	0	0	0	0	0	0

Таблица 2: Значения `t`

type	значение
с установление логического соединения	1
без установления логического соединения	2

Таблица 3: Значения e

Значение	Описание ошибки
1	Функция заблокирована межсетевым экраном
2	Сокета с заданным номером не существует
4	Системный вызов был прерван сигналом
9	Некорректный номер дескриптора
11	В данный момент функция не может быть выполнена, попробуйте позже
12	Недостаточно свободного места для создания сокета
13	Запрещено создание сокета с заданными параметрами
14	Некорректный логический адрес параметра
22	Некорректное значение аргумента
23	Переполнение таблицы дескрипторов
24	Слишком много открытых файлов
30	Файловая система доступна только на чтение
36	Комплекс F1 слишком длинный
71	Ошибка протокола
88	Значение аргумента s, не является дескриптором
91	Тип сокета не поддерживает протокол, выбранный сервером
93	Данный тип протокола не поддерживается
95	Функция не поддерживает данные типы сокетов
98	Сетевой адрес уже используется
99	Требуемый сетевой адрес не может быть использован
101	Сеть недоступна
103	Соединение разорвано
105	Недостаточно свободного места для создания сокета
106	Соединение на сокете уже произошло
110	Истечение времени ожидания
111	Не обнаружен слушающий порт сервера
114	Сокет является неблокирующим, а предыдущая попытка установить соединение ещё не завершена
115	Сокет является неблокирующим, а соединение не может быть установлено в данный момент

## 2 Примеры использования

### Эхо-сервер

```
main(/)
  Os Oe Oc Oa
  @+F1(16) @+F2(16) @+F3(16) @+F4(10)
  @'0.0.0.0'>F2
  *set__sockaddr__port(F1,1195/)
  *set__sockaddr__ip(F1,F2/)
  *socket(1/s)
  *bind(s,F1/e)
  *listen(s,100/e)
§1 *accept(s,F3/c)
  *freadf(c,F4,0,5/a)
  *fwritef(c,F4,0,5/a)
  *close(c/)
  →1
  *close(s/)
  **
```

### Эхо-клиент

```
main(/)
  Os Oe Oa
  @+F1(16) @+F2(16) @+F3(5) @+F4(5)
  @'192.168.203.1'>F2
  @'echo'>F3
  *set__sockaddr__port(F1,1195/)
  *set__sockaddr__ip(F1,F2/)
  *socket(1/s)
  *connect(s,F1/e)
  *fwritef(s,F3,0,5/a)
  *freadf(s,F4,0,5/a)
  *fwritef(1,F4,0,5/a)
  *close(s/)
  **
```