

Сетевое взаимодействие на языке Ляпас

Содержание

1	Описание функций	2
2	Примеры использования	5

1 Описание функций

`socket(domain, type, protocol/socket)` - эта функция используется для создания сокета. Первый параметр - домен - накладывает определенные ограничения на формат используемых процессов адресов и их интерпретацию. Второй параметр определяет тип канала связи с сокетом, который должен быть использован. Третий параметр позволяет выбрать нужный протокол для канала связи; если он равен 0, то операционная система выбирает его автоматически. Функция возвращает целое положительное число - номер дескриптора или -1 в случае ошибки.

Таблица 1: Значения domain

domain	значение
AF_UNIX, AF_LOCAL	1
AF_INET	2
AF_INET6	10
AF_IPX	4
AF_NETLINK	16
AF_X25	9
AF_AX25	3
AF_ATMPVC	8
AF_APPLETALK	5
AF_PACKET	17

Таблица 2: Значения type

type	значение
SOCK_STREAM	1
SOCK_DGRAM	2
SOCK_SEQPACKET	5
SOCK_RAW	3
SOCK_RDM	4
SOCK_PACKET	10
SOCK_NONBLOCK	2048
SOCK_CLOEXEC	524288

Таблица 3: Значения protocol

protocol	значения
IPPROTO_IP	0
IPPROTO_HOPOPTS	0
IPPROTO_ICMP	1
IPPROTO_IGMP	2
IPPROTO_IPIP	4
IPPROTO_TCP	6
IPPROTO_EGP	8
IPPROTO_PUP	12
IPPROTO_UDP	17
IPPROTO_IDP	22
IPPROTO_TP	29
IPPROTO_DCCP	33
IPPROTO_IPV6	41
IPPROTO_ROUTING	43
IPPROTO_FRAGMENT	44
IPPROTO_RSVP	46
IPPROTO_GRE	47
IPPROTO_ESP	50
IPPROTO_AH	51
IPPROTO_ICMPV6	58
IPPROTO_NONE	59
IPPROTO_DSTOPTS	60
IPPROTO_MTP	92
IPPROTO_ENCAP	98
IPPROTO_PIM	103
IPPROTO_COMP	108
IPPROTO_SCTP	132
IPPROTO_UDPLITE	136
IPPROTO_RAW	255

`bind(socket,sockaddr, sockaddr_len/error)` - используется сервером для присваивания сокету имени. До выполнения этой функции сокет недоступен программам-клиентам. Первый аргумент - сокет-дескриптор, который данная функция именуется. Второй параметр - указатель на комплекс, описывающий sockaddr. Третий параметр - длина комплекса. В случае успешного выполнения функция возвращает значение равное 0, иначе - код ошибки.

Таблица 4: Комплекс sockaddr

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

`set_sockaddr_domain(sockaddr,domain)` - функция, позволяющая заполнить поле комплекса sockaddr значением domain.

`set_sockaddr_port(sockaddr,port)` - функция, позволяющая заполнить поле комплекса sockaddr значением port.

`set_sockaddr_ip(sockaddr,ip)` - функция, позволяющая заполнить поле комплекса sockaddr значением, взятым из комплекса ip. В случае, если комплекс заполнен нулями, для установления соединения будут использованы все IP-адреса сервера.

listen(socket, backlog/error) - используется сервером, чтобы уведомить операционную систему, что он ожидает запросы на установление соединения на данном сокете. Первый аргумент - сокет для прослушивания, второй аргумент - целое положительное число, определяющее как много запросов может быть принято на сокет. В случае успешного выполнения функция возвращает значение равное 0, иначе - код ошибки.

accept(socket,sockaddr,sockaddr_len/socket) - используется сервером для принятия соединения на сокет. Сокет должен быть уже слушающим в момент вызова функции. Если сервер устанавливает соединение с клиентом, то функция accept возвращает новый сокет-дескриптор, через который и происходит общение клиента с сервером, в противном случае -1. Пока устанавливается соединение клиента с сервером, функция accept блокирует другие запросы соединения с данным сервером, а после установления связи "прослушивание"запросов возобновляется. Первый аргумент функции - сокет-дескриптор для принятия соединений от клиентов. Второй аргумент - комплекс sockaddr для соответствующего домена. Третий аргумент - длина комплекса sockaddr. Второй и третий аргументы заполняются соответствующими значениями в момент установления соединения клиента с сервером и позволяют серверу точно определить, с каким именно клиентом он общается.

connect(socket,sockaddr,sockaddr_len/error) - данная функция используется клиентом для установления соединения с сервером. Первый аргумент - сокет клиента, второй - комплекс sockaddr, соответствующий серверу, третий - длина комплекса sockaddr. Если соединение прошло успешно возвращается значение равное нулю, иначе - код ошибки.

Таблица 5: Значения error

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
20	Аргумент не является директорией
22	Некорректное значение аргумента
23	Переполнение таблицы дескрипторов
24	Слишком много открытых файлов
30	Файловая система доступна только на чтение
36	Комплекс sockaddr слишком длинный
40	Слишком много символических ссылок
71	Ошибка протокола
88	Значение аргумента socket, не является дескриптором
91	Тип сокета не поддерживает протокол,выбранный сервером
93	Данный тип протокола не поддерживается
95	Функция не поддерживает данные типы сокетов
97	Семейство адресов не поддерживается протоколом
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 Oq Oc Oy Or Od Ot Op
  @+F1(16) @+F2(16) @+F3(16) @+F4(10)
  @'0.0.0.0'>F2
  2⇒d 1⇒t 0⇒p
  *set__sockaddr__domain(F1,2/)
  *set__sockaddr__port(F1,1195/)
  *set__sockaddr__ip(F1,F2/)
  *socket(d,t,p/s)
  *bind(s,F1,10h/e)
  *listen(s,100/e)
§1 *accept(s,F3,10h/c)
  *freadf(c,F4,0,5/a)
  *fwritef(c,F4,0,5/a)
  *close(c/)
  →1
  *close(s/)
  **
```

Эхо-клиент

```
main(/)
  Ox Oq Oy Oe Od Ot Op
  @+F1(16) @+F2(16) @+F3(5) @+F4(5)
  @'192.168.203.1'>F2
  @'echo'>F3
  2⇒d 1⇒t 0⇒p
  *set__sockaddr__domain(F1,2/)
  *set__sockaddr__port(F1,1195/)
  *set__sockaddr__ip(F1,F2/)
  *socket(d,t,p/x)
  *connect(x,F1,10h/e)
```

```
*fwritef(x,F3,0,5/a)
*freadf(x,F4,0,5/a)
*fwritef(1,F4,0,5/a)
*close(x/)
**
```