

```

server.1
Os Oe Oc Od Ot Op ; обнуление переменных
@+F1(16) @+F2(16) @+F3(16) ; 1 комплекс - для настроек сервера, 2 -
для настроек клиента, 3 - для обмена сообщением
@'0.0.0.0'>F2
2=>d 1=>t 0=>p ; domain=AF_INET type=SOCK_STREAM protocol=auto
*set_sockaddr_bind_domain(F1,2/) ;настройка сервера domain=AF_INET
*set_sockaddr_bind_port(F1,1195/) ;настройка сервера port=1195
*set_sockaddr_bind_ip(F1,F2/) ;настройка сервера IP=любой
*socket(d,t,p/s)
*bind(s,F1,10h/e) ;привязываем сокет s к комплексу F1 длины 16 байт
*listen(s,5/e) ;ожидаем не более 5 соединений на соquete s
*accept(s,F2,10h/c) ;устанавливаем связь на соquete s
*freadf(c,F3,0,5/a) ;считываем 5 символов с клиентского сокета в ком-
плекс F3
*fwritef(c,F3,0,5/a) ;записываем 5 символов в клиентский сокет из ком-
плекса F3
*close(c/) ;закрываем сокет клиента
*close(s/) ;закрываем сокет сервера
**

```

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.

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

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

set_sockaddr_bind_ip(sockaddr, ip) - функция, позволяющая заполнить поле комплекса sockaddr значением, взятым из комплекса ip.

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 |

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 eggr

| Значение | Обозначение ошибки |
|----------|--|
| 1 | операция не разрешена |
| 2 | файл или директория не существует |
| 3 | процесс не существует |
| 4 | прерывание системы |
| 5 | ошибка ввода/вывода |
| 6 | устройство или адрес не существует |
| 7 | аргументов слишком много |
| 8 | ошибка формата исполняемого файла |
| 9 | некорректный номер дескриптора |
| 10 | дочернего процесса не существует |
| 11 | перезапустите процесс |
| 12 | ошибка памяти |
| 13 | доступ запрещен |
| 14 | некорректный адрес |
| 15 | требуется блочное устройство |
| 16 | устройство или ресурс занято |
| 17 | файл существует |
| 18 | нельзя создавать жесткие ссылки между устройствами |
| 19 | устройство не существует |
| 20 | не является директорией |
| 21 | является директорией |
| 22 | неверный аргумент |
| 23 | переполнение таблицы дескрипторов |
| 24 | много открытых файлов |
| 25 | не является пишущим устройством |
| 26 | текстовый файл занят |
| 27 | файл слишком большой |
| 28 | на устройстве не хватает свободного места |
| 29 | запрещено передвижение указателя на файл |
| 30 | доступны только права на чтение |
| 31 | слишком много ссылок |
| 32 | ошибка ввода данных |
| 33 | результат математической операции нельзя представить |