

[Каталог документации](#) / [Раздел "Документация для Linux"](#) / [Оглавление документа](#)



**Next:** [Введение в сети](#) **Up:** [Примеры](#) **Previous:** [Разделяемые сегменты памяти](#) [Contents](#) [Index](#)

## Работа с сокетами

Пример-оболочка программы "Клиент"

```
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <stdio.h>
#define ADDRESS "mysocket" /* адрес для связи */
void main ()
{
    char c;
    int i, s, len;
    FILE *fp;
    struct sockaddr_un sa;
    /* получаем свой сокет-дескриптор: */
    if ((s = socket (AF_UNIX, SOCK_STREAM, 0)) < 0)
    {
        perror ("client: socket"); exit(1);
    }
    /* создаем адрес, по которому будем связываться с сервером: */
    sa.sun_family = AF_UNIX;
    strcpy (sa.sun_path, ADDRESS);
    /* пытаемся связаться с сервером: */
    len = sizeof ( sa.sun_family) + strlen ( sa.sun_path);
    if ( connect ( s, &sa, len) < 0 ){
```

```

        perror ("client: connect"); exit (1);
    }

/*----- */

/* читаем сообщения сервера */
    fp = fdopen (s, "r");
    c = fgetc (fp);
/* обрабатываем информацию от сервера
    .....
*/

/* посылаем ответ серверу */
    send (s, "client", 7, 0);

/* продолжаем диалог с сервером, пока в этом есть
    необходимость
    .....
*/

/* завершаем сеанс работы */
    close (s);
    exit (0);
}

```

Пример-оболочка программы "Сервер"

```

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <stdio.h>
#define ADDRESS "mysocket" /* адрес для связи */
void main ()
{
    char c;
    int i, d, d1, len, ca_len;
    FILE *fp;

```

```
    struct sockaddr_un sa, ca;
/* получаем свой сокет-дескриптор: */
    if((d = socket (AF_UNIX, SOCK_STREAM, 0)) < 0)
    {
        perror ("client: socket"); exit (1);
    }
/* создаем адрес, с которым будут связываться клиенты */
    sa.sun_family = AF_UNIX;
    strcpy (sa.sun_path, ADDRESS);
/* связываем адрес с сокетом;
   уничтожаем файл с именем ADDRESS, если он существует,
   для того, чтобы вызов bind завершился успешно
*/
    unlink (ADDRESS);
    len = sizeof ( sa.sun_family) + strlen (sa.sun_path);
    if ( bind ( d, &sa, len) < 0 ) {
        perror ("server: bind"); exit (1);
    }
/* слушаем запросы на сокет */
    if ( listen ( d, 5) < 0 ) {
        perror ("server: listen"); exit (1);
    }
/* связываемся с клиентом через неименованный сокет с дескриптором d1:
*/
    if (( d1 = accept ( d, &ca, &ca_len)) < 0 ) {
        perror ("server: accept"); exit (1);
    }

/* ----- */

/* пишем клиенту: */
    send (d1, "server", 7, 0);

/* читаем запрос клиента */
    fp = fdopen (d1, "r");
    c = fgetc (fp);
/* ..... */
```

```
/* обрабатываем запрос клиента,  
   посылаем ответ и т.д.  
   .....  
*/  
  
/* завершаем сеанс работы */  
   close (d1);  
   exit (0);  
}
```

---

*Alex Otwagin 2002-12-16*

Спонсоры:



При поддержке  
**inferno solutions\***

Хостинг:



[Закладки на сайте](#)  
[Проследить за страницей](#)

Created 1996-2022 by [Maxim Chirkov](#)  
[Добавить](#), [Поддержать](#), [Вебмастеру](#)