



OS Lab 33

Status	approved
checkbox	
class	OS
due date	@Apr 21, 2021

```
struct hostent {
    char    *h_name;           /* canonical name of host */
    char    **h_aliases;       /* alias list */
    int      h_addrtype;        /* host address type */
    int      h_length;          /* length of address */
    char    **h_addr_list;     /* list of addresses */
};
```

```
...

struct sockaddr_in {
    sa_family_t    sin_family; /* семейство адресов */
    in_port_t      sin_port;   /* номер порта */
    struct in_addr sin_addr;    /* адрес IPv4 */
    unsigned char  sin_zero[8]; /* заполнитель */
};
```

<https://man7.org/linux/man-pages/man2/sigaction.2.html>

```
cc [flag... ]
file...-lnsl [library... ]
#include <netdb.h>

struct hostent *gethostbyname(
const char *name);
```

Solaris 10	Sun SPARC	Прямой (big-endian)
------------	-----------	---------------------

Некоторые типы процессоров допускают возможность изменения порядка байтов, что вносит еще большую путаницу.

Чтобы не возникало путаницы с порядком байтов при обмене данными между разнородными компьютерными системами, сетевые протоколы жестко задают порядок байтов. Набор протоколов TCP/IP использует сетевой (прямой, big-endian) порядок байтов. Порядок байтов приобретает важность, когда приложения начинают обмениваться форматированными данными. При использовании протоколов TCP/IP адреса имеют сетевой порядок байтов, поэтому в приложениях иногда возникает необходимость преобразовать порядок байтов, поддерживаемый аппаратной архитектурой, в сетевой порядок байтов. Такое преобразование обычно производится, например, при выводе адреса в удобочитаемой форме. Преобразования между сетевым и аппаратным порядком байтов производятся с помощью следующих четырех функций.

```
#include <arpa/inet.h>

uint32_t htonl(uint32_t hostint32);
        Возвращает 32-разрядное целое с сетевым порядком байтов

uint16_t htons(uint16_t hostint16);
        Возвращает 16-разрядное целое с сетевым порядком байтов

uint32_t ntohl(uint32_t netint32);
        Возвращает 32-разрядное целое с аппаратным порядком байтов

uint16_t ntohs(uint16_t netint16);
        Возвращает 16-разрядное целое с аппаратным порядком байтов
```

16.3.4. Присваивание адресов сокетам

Адрес, присваиваемый клиентскому сокету, не представляет для нас особого интереса, потому мы можем позволить системе выбирать адрес по умолчанию. Однако для сервера важно присвоить сокету предопределенный адрес, на который клиенты будут присылать запросы. Клиентам необходимо заранее знать требуемый адрес, чтобы войти в контакт с сервером, и самое простое решение заключается в том, чтобы зарезервировать адрес сервера в файле `/etc/services` или в службе имен. Присвоить адрес сокету можно с помощью функции `bind`.

```
#include <sys/socket.h>

int bind(int sockfd, const struct sockaddr *addr, socklen_t len);
        Возвращает 0 в случае успеха, -1 — в случае ошибки
```

- Существует несколько ограничений, касающихся адресов:
- Указываемый адрес должен быть действительным адресом для машины, на которой выполняется процесс, — нельзя задать адрес, который принадлежит другой машине.
 - Формат адреса должен совпадать с форматом, который поддерживается семейством адресов, указанным при создании сокета.
 - Номер порта не может быть меньше 1024, если процесс не имеет соответствующих привилегий (например, привилегий суперпользователя).
 - Обычно каждый конкретный адрес может быть связан только с одним сокетом, хотя некоторые протоколы допускают присвоение одного и того же адреса нескольким сокетам.
- В домене Интернета имеется специальный IP-адрес `INADDR_ANY`, который соответствует адресам всех сетевых интерфейсов в системе. Это означает, что существует возможность принимать пакеты с любого сетевого интерфейса, установленного в системе. В следующем разделе мы увидим, что система сама может присвоить адрес сокету при обращении к функциям `connect` и `listen`.