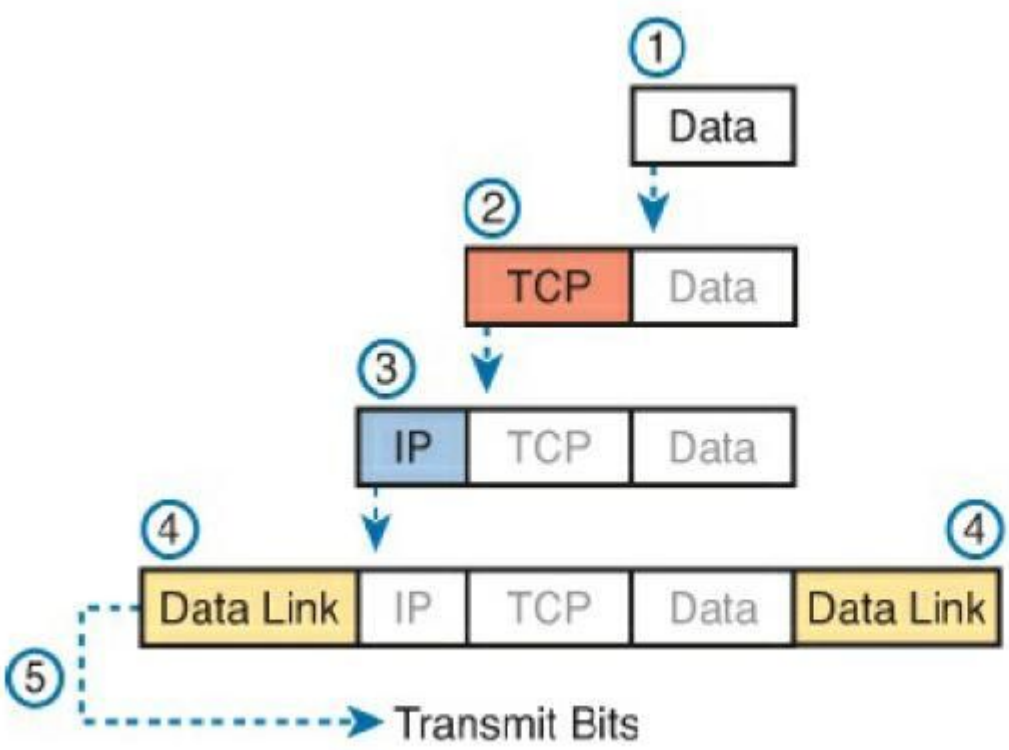
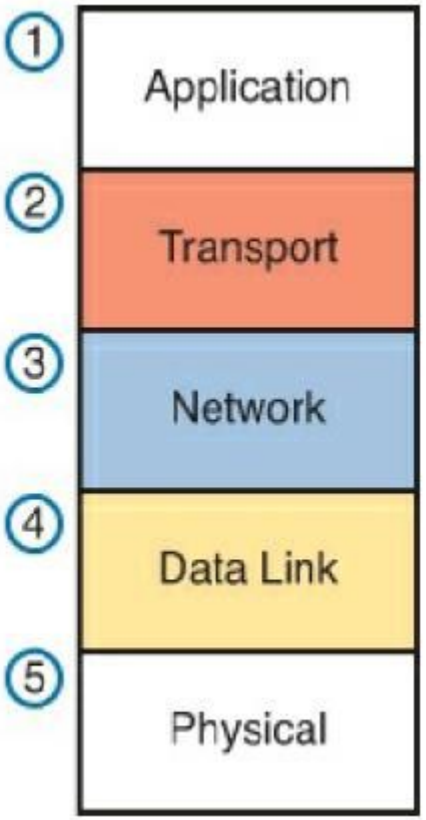
SOCKETS

Инкапсуляция





https://4anm.wordpress.com/2013/07/05/the-tcpip-and-osi-networking-models/

SOCKET

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

int socket(int domain, int type, int protocol);

domain: AF_UNIX, AF_LOCAL, AF_INET, AF_INET6,...
type:  SOCK_STREAM, SOCK_DGRAM,...
protocol - 0, IPPROTO_TCP, IPPROTO_UDP,...

Типичное использование:
int fd = socket(AF_INET, SOCK_STREAM,0); //IPv4 сокет
int fd = socket(AF_UNIX, SOCK_STREAM,0); //UNIX сокет
int fd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP); //UDP
сокет
```

Настройка сокета

int setsockopt(int sockfd, int level, int optname,
const void *optval, socklen_t
optlen);

- позволяет задать режим работы сокета
- level - режим какого уровня требуется изменить
  (SOL_SOCKET - режим самого сокета, IPPROTE_TCP - режим протокола,...)
- optname - имя изменяемого параметра
- optval - значение

Пример:

```
int optval = 1;
setsockopt(fd, SOL_SOCKET, SO_REUSEADDR, &val, sizeof(val));
setsockopt(fd, SOL_SOCKET, SO_REUSEPORT, &val, sizeof(val));
```

позволяет использовать адрес сразу после завершения процесса

BIND

```
int bind(int sockfd, const struct sockaddr *addr,
                     socklen_t addrlen);
```

- связывание сокета с адресом

UNIX сокет

```
struct sockaddr_un {
    sa_family_t sun_family; //AF_UNIX
    char sun_path[108];
};
```

IPv4 сокет

```
struct sockaddr_in {
sa_family_t sin_family; //AF_INET
in_port_t   sin_port; // htons(port)
            struct in_addr sin_addr;
};

struct in_addr {
    uint32_t    s_addr; //htonl()
};
```

LISTEN & ACCEPT

int listen(int sockfd, int backlog);
- перевод сокета в режим прослушивания

backlog - максимальный размер очереди ожидающих соединений

int accept(int sockfd, struct sockaddr *addr,
socklen_t *addrlen);
- ожидание соединения и создание нового сокета

sockfd - сокет в режиме прослушивания
addr - адрес входящего соединения

возвращает дескриптор на новый созданный сокет
sockfd - не меняется и остается в слушающем состоянии

CONNECT

```
int connect(int sockfd, const struct sockaddr *addr,
                socklen_t addrlen);
```

- устанавливает соединение с сокетом указ. в *addr
- возвращает 0 если соединение установлено,
  при ошибке -1, errno

RECV & SEND

ssize_t recv(int sockfd, void *buf, size_t len, int flags);

- чтение данных из сокета в буфер

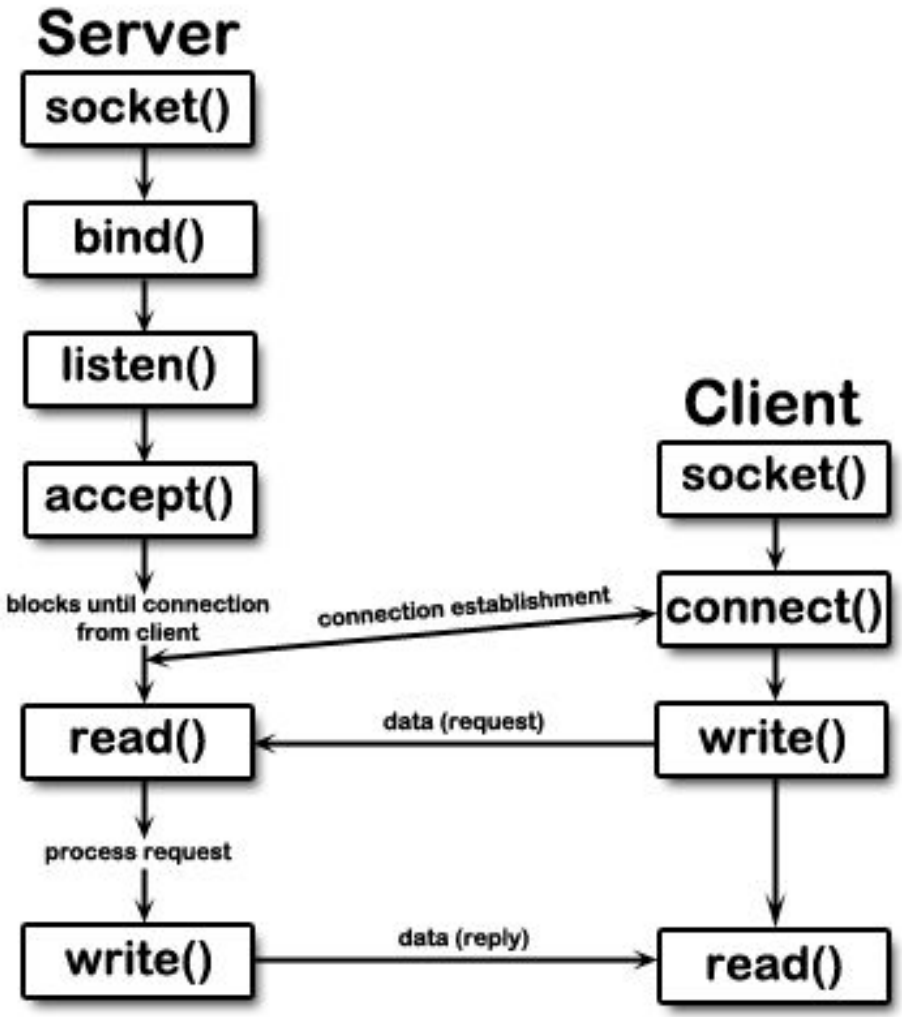
ssize_t send(int sockfd, const void *buf,
size_t len, int flags);

- запись данных из буфера в сокет

- flags: MSG_NOSIGNAL, ...

read(sockfd, buf, size) ⇒ recv(sockfd, buf, size, 0)
write(sockfd, buf, size) ⇒ send(sockfd, buf, size, 0)

Соединение по TCP



Utils

ncat

- эхо сервер: (ncat **-k -l** 12345 --exec ‘/bin/cat’)
- клиент: ncat <ip> <port>

netstat

-a - все подключения, в том числе (TCP estabilished connection)

-t|--tcp - только TCP

-n - не обращаться к DNS, выводить IP

-p - показать программу с которой связан сокет

lstat **-p** <pid>

- список открытых файлов относящихся к процессу <pid>

strace **-p**

ltrace **-p**

gdb **-p**

Согласование целых чисел при пересылке

Сеть - big-endian, i386 - little-endian

htonl()
- 32 bit: local --- network

ntonl()
- 32 bit: network --- local

htons()
- 16 bit: local --- network

ntons()
- 16 bit: network --- local