

Сеть

7 апреля 2018 г.

- `socket(2)`, `bind(2)`, `listen(2)`,
`connect(2)`, `accept(2)`, ...

Сокеты

- `socket(2)`, `bind(2)`, `listen(2)`,
`connect(2)`, `accept(2)`, ...
- «Сервер»
 - `socket` → `[bind]` → `listen` → `accept` →
`(read/write)*` → `close`
- «Клиент»
 - `socket` → `connect` → `(read/write)*` → `close`

socket

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

socket

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

- domain
 - AF_UNIX, AF_INET, AF_INET6, ...
 - AF_ — address family

socket

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

- domain
 - AF_UNIX, AF_INET, AF_INET6, ...
 - AF_ — address family
- type
 - SOCK_STREAM, SOCK_DGRAM, SOCK_SEQPACKET,
...

socket

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

- domain
 - AF_UNIX, AF_INET, AF_INET6, ...
 - AF_ — address family
- type
 - SOCK_STREAM, SOCK_DGRAM, SOCK_SEQPACKET, ...
- protocol
 - Почти всегда 0

bind

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


bind

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

- Наследование, полиморфизм

bind

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

- Наследование, полиморфизм

- `struct sockaddr` {
 `sa_family_t` sa_family;
 `char` sa_data[14];
}
- `struct sockaddr_un` {
 `sa_family_t` sun_family;
 `char` sun_path[108];
};
- `struct sockaddr_in` {
 `sa_family_t` sin_family;
 `in_port_t` sin_port;
 `struct in_addr` sin_addr;
};

listen, accept

- «Сервер»

```
int listen(int sockfd, int backlog);  
int accept(int sockfd,  
           struct sockaddr *addr, socklen_t *addrlen);
```

- «Клиент»

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

UNIX-сокеты

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

# UNIX-сокеты

- AF\_UNIX
- ```
struct sockaddr_un {  
    sa_family_t sun_family;  
    char        sun_path[108];  
};
```
- ```
socketpair(2)

int socketpair(int domain, int type,
 int protocol, int sv[2]);
```

# UNIX-сокеты

- AF\_UNIX
- ```
struct sockaddr_un {  
    sa_family_t sun_family;  
    char        sun_path[108];  
};
```
- ```
socketpair(2)
int socketpair(int domain, int type,
 int protocol, int sv[2]);
```
- Abstract namespace

# Datagram sockets

```
ssize_t sendto(int sockfd, const void *buf,
 size_t len, int flags,
 const struct sockaddr *dest_addr,
 socklen_t addrlen);
```

```
ssize_t recvfrom(int sockfd, void *buf,
 size_t len, int flags,
 struct sockaddr *src_addr,
 socklen_t *addrlen);
```

# Datagram sockets

```
ssize_t sendto(int sockfd, const void *buf,
 size_t len, int flags,
 const struct sockaddr *dest_addr,
 socklen_t addrlen);
```

```
ssize_t recvfrom(int sockfd, void *buf,
 size_t len, int flags,
 struct sockaddr *src_addr,
 socklen_t *addrlen);
```

- connect





- Модель OSI

|               |                                  |
|---------------|----------------------------------|
| Прикладной    | HTTP, DNS, очень много           |
| Представления | SSL, TLS, мало                   |
| Сеансовый     | RPC, мало                        |
| Транспортный  | TCP, UDP, SCTP, ...              |
| Сетевой       | IPv4, IPv6, ICMP, IGMP, ...      |
| Канальный     | ARP, PPP, Ethernet, ...          |
| Физический    | USB, Bluetooth, стандарты Eth, . |

- Модель OSI

|               |                                  |
|---------------|----------------------------------|
| Прикладной    | HTTP, DNS, очень много           |
| Представления | SSL, TLS, мало                   |
| Сеансовый     | RPC, мало                        |
| Транспортный  | TCP, UDP, SCTP, ...              |
| Сетевой       | IPv4, IPv6, ICMP, IGMP, ...      |
| Канальный     | ARP, PPP, Ethernet, ...          |
| Физический    | USB, Bluetooth, стандарты Eth, . |

- Долго стандартизовали, TCP/IP победил

# TCP/IP

Прикладной  
Представления  
Сеансовый  
Транспортный  
Сетевой  
Канальный  
Физический

# Коммутация

- Канальная

# Коммутация

- Канальная
- Пакетная
  - MTU

# Стек протоколов

Стек протоколов TCP/IP



## Стек протоколов TCP/IP

Тут могла бы быть картинка, но не сегодня

- $\geq 20$  байт
- Примечательные поля:
  - Версия
  - Длина
  - Сдвиг фрагмента
  - TTL
  - Протокол
  - Контрольная сумма
  - Адрес отправителя
  - Адрес получателя

- +8 байт
  - Порт отправителя
  - Порт получателя
  - Длина
  - Контрольная сумма

- +20 байт
  - Порт отправителя
  - Порт получателя
  - Seq no
  - Ack no
  - Флаги
    - ACK, SYN, FIN, RST, ...
  - Контрольная сумма
  - ...

# ICMP

- echo request/reply
  - ping

# ICMP

- echo request/reply
  - ping
- TTL expired
  - tracepath
- . . .

# DNS

- ifmo.ru → 77.234.212.58



# DNS

- ifmo.ru → 77.234.212.58
- Рекурсивный/нерекурсивный

# DNS

- ifmo.ru → 77.234.212.58
- Рекурсивный/нерекурсивный
- ```
int getaddrinfo(const char *node, const char *service,  
               const struct addrinfo *hints,  
               struct addrinfo **res);
```

```
struct addrinfo {  
    int            ai_flags;  
    int            ai_family;  
    int            ai_socktype;  
    int            ai_protocol;  
    socklen_t      ai_addrlen;  
    struct sockaddr *ai_addr;  
    char           *ai_canonname;  
    struct addrinfo *ai_next;  
};
```

Сырые сокеты

- `raw(7)`

-

```
socket(AF_INET, SOCK_RAW, int protocol);
```

Если вдруг осталось время

- NAT
- маршрутизация