

# Сокеты и сети

АКОС 17

# Способы взаимодействия

- Через файловый ввод-вывод
- Каналы (pipes)
- Именованные каналы (FIFO)
- Разделяемые страницы памяти (mmap)
- POSIX Shared Memory (shm)
- Сокеты - позволяют одному процессу общаться сразу со многими процессами

# Сокет

- Файловый дескриптор для ввода/вывода
- Одно имя позволяет создавать подключения нескольких процессов (клиентов) к одному процессу (серверу)
- Пространство имен двух видов:
  - UNIX-сокеты: файл в локальной файловой системе
  - TCP/IP-сокеты: адрес хоста + номер порта

# Создание сокета

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

Создает новый сокет и возвращает его файловый дескриптор

- **domain** — тип семейства адресов (**AD\_UNIX**, **AD\_INET**, **AF\_IPX**, **AF\_APPLETALK**, **AF\_BLUETOOTH** и др.)
  - **type** — **OSCK\_STREAM** или **SOCK\_DGRAM**
  - **protocol** — **0** — выбирается автоматически; **IPPROTO\_TCP** и др.
- 
- Сокет создан, но ещё не готов к использованию
  - Файловый дескриптор может быть наследован (fork) или клонирован (dup2)
  - Дескриптор должен быть закрыт после использования (close)

# Подготовка сокета к работе

- Для клиента - нужно к кому-нибудь подключиться:
  - *системный вызов* **connect**
- Для сервера - нужно анонсировать имя и ожидать подключений:
  - объявить имя с помощью **bind**
  - создать очередь входящих подключений **listen**
  - принять подключение **accept**

# Объявление имени

```
int bind(int socket,  
          const struct sockaddr *addr,  
          socklen_t address_len)
```

Связывает сокет с некоторым адресом

- **struct sockaddr\_in** – адрес IPv4
- **struct sockaddr\_in6** – адрес IPv6
- **struct sockaddr\_un** – адрес локального UNIX-сокета

# Объявление имени

- Обязательно для приема входящих соединений
- Может потребоваться (но не обязательно) для исходящих соединений

# Переключение сокета в режим прослушивания

**listen**(int sockfd, int backlog)

- backlog - размер очереди сообщений
- если подключается много ( $>\text{backlog}$ ) процессов, то у них возникает ошибка подключения
- константа **SOMAXCONN** (128 в Linux) определяет максимальный размер очереди



# Создание соединения

- **connect** - подключить сокет к другому процессу (в том числе на другом компьютере)
- **accept** - дожидаться подключения, возвращает файловый дескриптор подключенного сокета для ввода-вывода

# Ввод-вывод через сокеты

**ssize\_t** **recv**(**int** socket, **void\*** buffer, **size\_t** buf\_size, **int** flags)

Читает данные из сокета в указанный буфер. Флаги:

- **MSG\_PEEK** – отметить данные как непрочитанные
- **MSG\_OOB** – получить данные повышенного приоритета (Out-of-Band; в случае TCP – Urgent Data)
- **MSG\_WAITALL** – дождаться полного получения данных и выдать их за один вызов

**ssize\_t** **send**(**int** socket, **const void** \*buffer, **size\_t** size, **int** flags)

Записывает данные из буфера в сокет. Флаги:

- **MSG\_OOB** – передать данные повышенного приоритета (Out-of-Band; в случае TCP – Urgent Data)
- **MSG\_NOSIGNAL** – не отправлять процессу сигнал SIGPIPE, если сокет закрыт

**read**(Socket, Buffer, Size) → **recv**(Socket, Buffer, Size, 0).  
**write**(Socket, Buffer, Size) → **send**(Socket, Buffer, Size, 0).

# Обработка входящих подключений

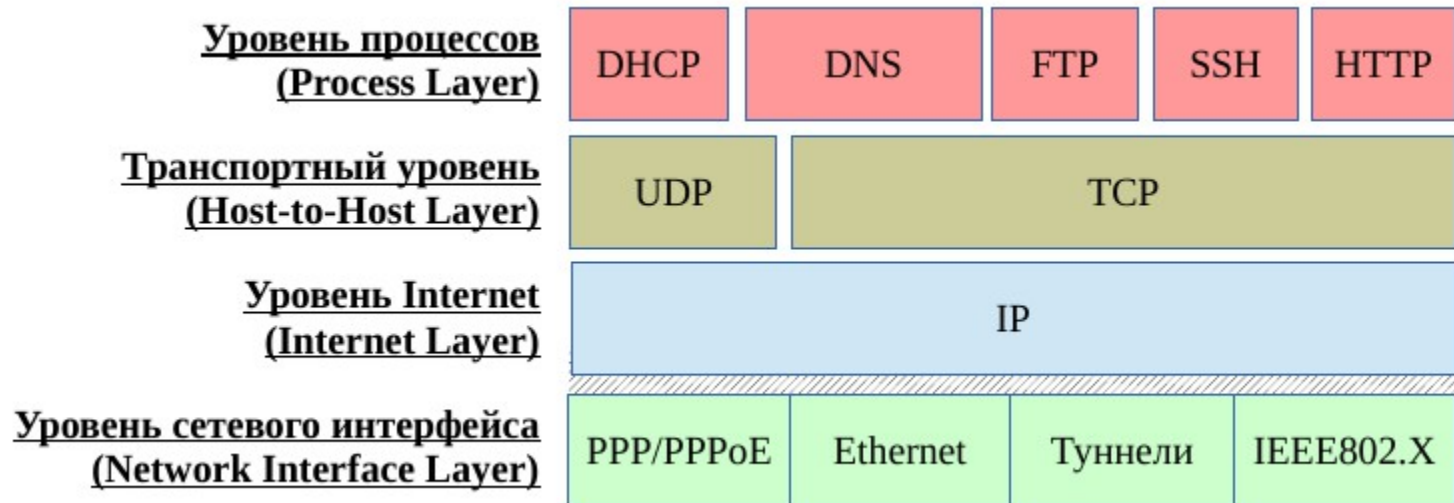
1. Создать сокет [socket]
2. Связать с именем [bind]
3. Переключить в режим прослушивания и создать очередь подключений [listen]
4. Принимать очередное подключение [accept] в виде сокета
5. Ввод-вывод [read/write] или [recv/send]
6. Закрыть сокет, принятый через [accept]
7. goto шаг 4

# Межпроцессное взаимодействие (UNIX-сокеты)

- Для домохозяек
  - X-сервер
  - DBUS-сервер
  - PulseAudio-сервер
- Для серьёзных применений
  - Отдельные веб-приложения
  - SQL-сервер
- Стандартное размещение сокетов:
  - `/var/run`
  - `/tmp/*`

# **СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ**

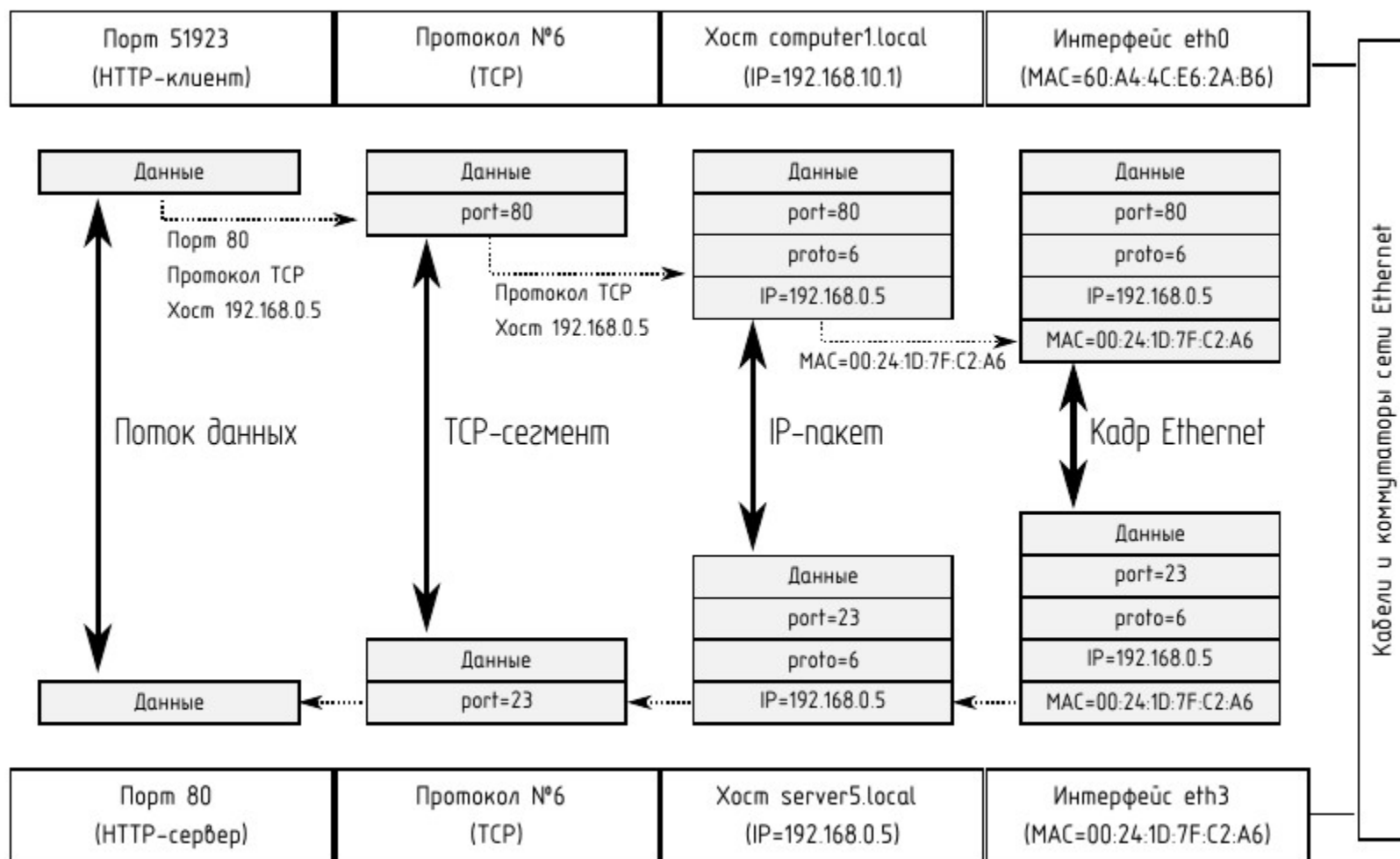
# Стек TCP/IP



# Модель OSI

Стек TCP/IP		Модель OSI (Open Systems Interconnections)	Примеры
Уровень процессов		Уровень приложений (Application)	HTTP, FTP, SSH, Telnet
		Уровень представления (Presentation)	ASCII, GZIP, binary
		Уровень сеанса (Session)	NetBIOS, SSL
Транспортный уровень		Уровень транспорта (Transport)	TCP, UDP
Уровень Internet		Уровень сети (Network)	IPv4, IPv6, IPX, AppleTalk
Уровень сетевого интерфейса		Уровень канала (Data Link)	PPP, IEEE 802.2 (Ethernet)
		Физический уровень (Physical)	USB, IEEE 802.11 IEEE 802.3 (Ethernet)

# Передача данных в TCP/IP





# Ethernet

MAC-адрес получателя	MAC-адрес отправителя	Дополнительные опции	Длина	Данные	Контрольная сумма
6 байт	6 байт	4 байта	2 байта	от 46 до 1500 байт (параметр MTU)	4 байта

# IPv4

Байты	0	1	2	3
0...3	Версия и размер заголовка	Тип службы	Размер IP-пакета	
4...7	ID группы пакетов		Флаги и смещение	
8...11	TTL	Номер протокола	Контрольная сумма заголовка	
12...15	Адрес отправителя			
16...19	Адрес получателя			
20...24	Дополнительные опции			

# Имя хоста

- IP-адрес - это 32-битное (IPv4) или 128-битное (IPv6) число
- Символическая нотация - определяется через службу DNS
- Зарезервированы адрес 127.0.0.1 и имя localhost - текущий компьютер
- Каждому DNS-имени может соответствовать несколько IP-адресов
- С каждым IP-адресом может быть связано несколько DNS-имен

# UDP

Байты	0	1	2	3
0..4	Порт отправителя		Порт назначения	
5..8	Длина пакета		Контрольная сумма	

# TCP

Байты	0	1	2	3								
0...3	Порт отправителя		Порт получателя									
4...7	Порядковый номер пакета											
8...11	Порядковый номер подтверждаемого пакета (с флагом ACK)											
12...15	Размер заголовка в 32-битных словах	000	N S	C W R	E C R E	U R G	A C K	P S H	R S T	S Y N	F I N	Размер окна (буфера для приема данных, ожидаемых при ответе)
16...19	Контрольная сумма заголовка и данных								Указатель на порядковый номер пакета, в котором заканчивается приоритетных блок данных			
20.....	Дополнительные опции											

# Номера портов

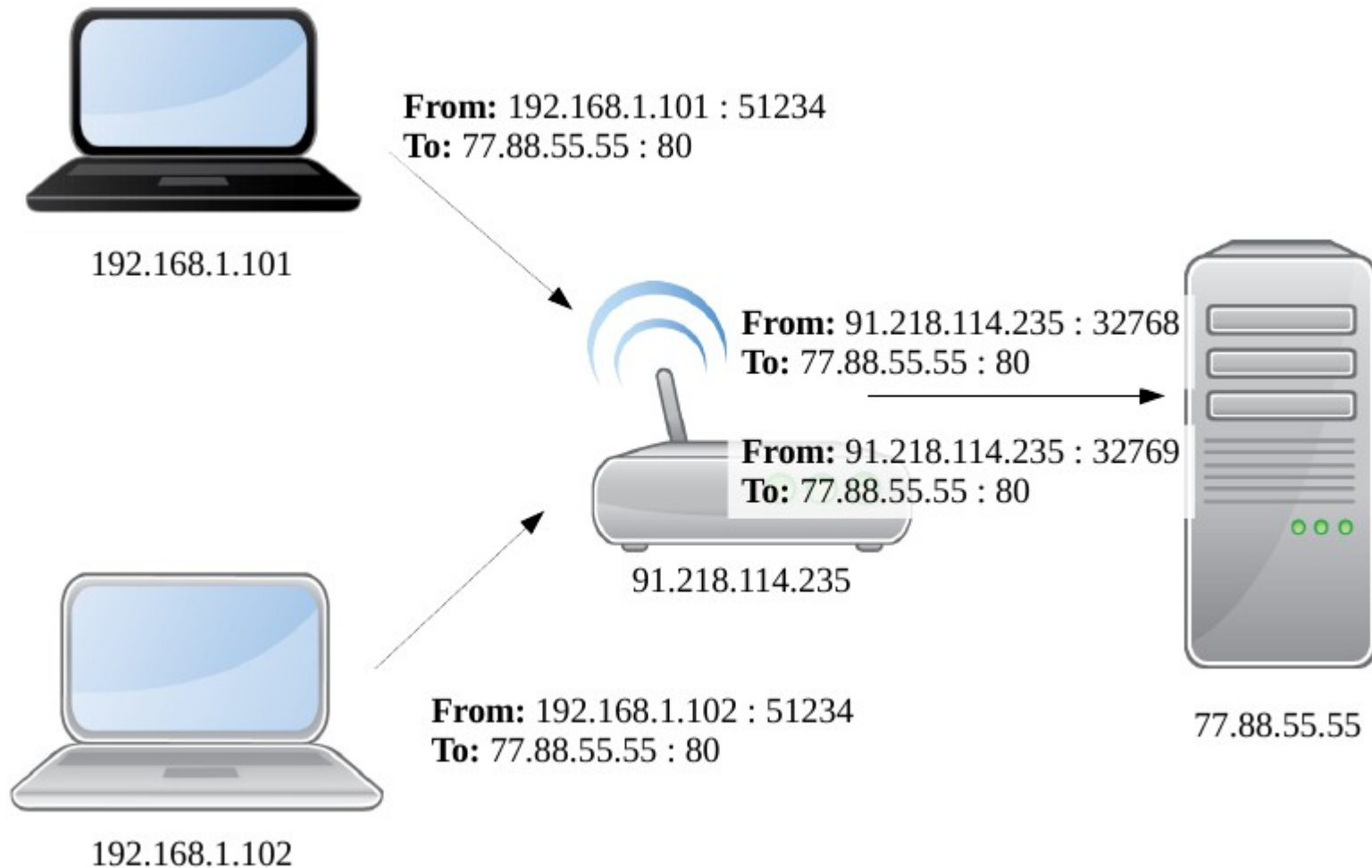
0	- не используется
20, 21	- FTP
22	- SSH
25	- SMTP
80	- HTTP
137, 138, 139	- NetBIOS
143	- IMAP
443	- HTTPS
465	- SMTPS
631	- CUPS
993	- IMAPS

**1024...65535** – порты для клиентских соединений  
(старые UNIX-системы и Windows XP)

**32768...65535** – порты для клиентских соединений (Linux)

**49152...65535** – порты для клиентских соединений (BSD,  
Windows Vista+)

# Переименование портов



# Порты с номерами <1024

- Зарезервированы под стандартные службы
- Открыть может только root
- Запускать сервера под правами root -  
**опасно!**



# Порты с номерами <1024

## Решение проблемы:

- Создать сокет
- Создать дочерний процесс, который наследует fd сокета для выполнения настройки
- Дочерний процесс повышает свои привилегии до root и вызывает bind

*Пакет authbind в  
Debian/Ubuntu*

