

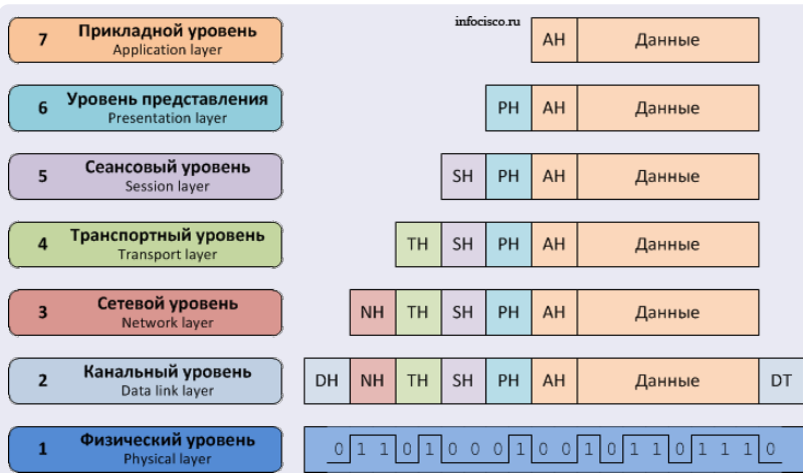
Лекция 5. Сетевое взаимодействие

ИУ8

October 30, 2018

Модель OSI/ISO

	Единица данных	Уровень	Функция	Примеры протоколов
ОС	Поток	Прикладной	Прикладная задача	HTTP, SMTP, DNS, etc.
		Представления	Представление данных, шифрование, etc.	MIME, SSL
		Сеансовый	Взаимодействие хостов (на уровне ОС)	NetBIOS, именов. пайпы
	Сегмент	Транспортный	Соединение конец-в-конец, контроль передачи данных	TCP, UDP
Сеть	Пакет	Сетевой	Логическая адресация и маршрутизация пакетов	IP, ICMP
	Фрейм	Канальный	Физическая адресация	IEEE 802.3, ARP, DHCP
	Бит	Физический	Кодирование и передача данных по физическому каналу	IEEE 802.3



Модель TCP/IP

TCP/IP	OSI Model	Protocols
Application Layer	Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP, POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
	Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
	Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	Transport Layer	TCP, UDP
Internet Layer	Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Link Layer	Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
	Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi

Сетевые протоколы

Отвечают за взаимодействие между узлами сети.

IPv4

Адрес узла сети в IPv4 является 32-битным. Например, 192.168.0.10

127.0.0.1 - локальный IP-адрес

IPv6

Адрес узла сети в IPv6 является 128-битным. Например,
2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d

::1 - локальный IP-адрес

Транспортные протоколы

Отвечают за взаимодействие между отдельными процессами узлов сети

Адресация: 16-битное число - порт.

Пара "IP-адрес/порт" определяют конечную точку сетевого взаимодействия

Порты

0 - 1023 – хорошо известные службы. Например:

21 – SSH

80 – HTTP

443 – HTTPS

1024 - 49151 – зарегистрированные IANA службы

49152 - 65535 – любые цели

UDP

- проверяется только контрольная сумма пакета;
- при избытке пакетов в очереди пакеты отбрасываются;
- получатель никак не управляет входящим потоком данных.

Протокол UDP применяется в системах, где скорость важнее надежности. Например, в передаче голоса или видеопотока.

TCP

- протокол "с установлением соединения";
- согласование параметров соединения;
- проверка корректности получаемых данных;
- повторный запрос данных в случае некорректного получения.

Протокол TCP применяется в системах, где безошибочность данных важнее, даже если приходится жертвовать скоростью.

Сокеты Беркли

Для работы с сетью на низком уровне традиционно используют сокеты. Сокет — это абстракция, которая позволяет работать с сетевыми ресурсами, как с файлами. Мы можем писать и читать данные из сокета почти так же, как из обычного файла.

Псевдокод

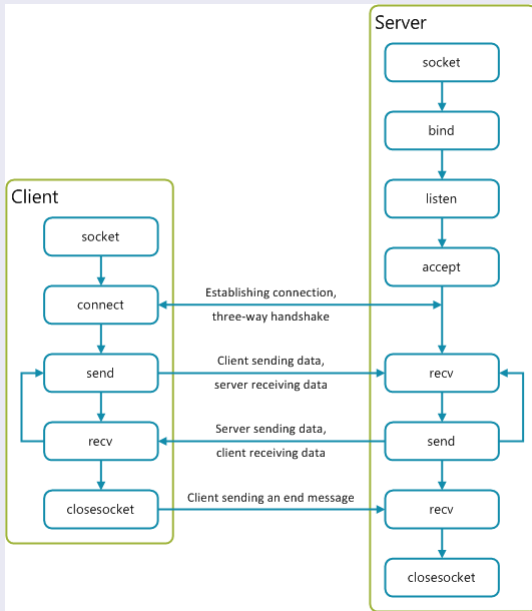
```
1 std::vector<Socket> sockets;  
2  
3 for (auto& socket : sockets) {  
4     socket << "some info from server";  
5     std::string data;  
6     socket >> data;  
7 }
```

Клиент

```
1 Socket srv("192.41.1.1", 50123);  
2 std::string data;  
3 srv >> data;  
4 srv << "send info to server";
```

Основные функции для работы с сокетами

- socket
- bind
- listen
- accept
- connect
- recv,recvfrom,recvmsg
- send,sendto,sendmsg
- close



Прототип

```
1 int socket(int domain, int type, int protocol);  
2  
3 int sock_udp = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);  
4 int sock_tcp = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
```

Описание

Функция `socket` - создает новый сокет.

Документаци

<http://man7.org/linux/man-pages/man2/socket.2.html>

Прототип

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

Описание

Функция `bind` - связывает сокет с конкретным адресом. Если компьютер имеет несколько сетевых интерфейсов с различными IP-адресами, вы можете принимать соединения только с одного из них, передав его адрес функции `bind`. Если же вы готовы соединяться с клиентами через любой интерфейс, задайте в качестве адреса константу `INADDR_ANY`.

Документаци

<http://man7.org/linux/man-pages/man2/bind.2.html>

Прототип

```
1 | int listen(int sockfd, int backlog);
```

Описание

Функция `listen` - переводит сокет в пассивный режим. Функция принимает дескриптор слушающего сокета и размер очереди запросов. Каждый раз, когда очередной клиент пытается соединиться с сервером, его запрос ставится в очередь, так как сервер может быть занят обработкой других запросов. Если очередь заполнена, все последующие запросы будут игнорироваться.

Документация

<http://man7.org/linux/man-pages/man2/listen.2.html>

accept

Прототип

```
1 | int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

Описание

Функция `accept` - ожидание клиентских соединений. В качестве аргумента ей передается дескриптор слушающего сокета. При успешной установке соединения, для него создается новый сокет. Функция `accept` возвращает дескриптор этого сокета. Если произошла ошибка соединения, то возвращается значение `INVALID_SOCKET`.

В структуру, на которую ссылается `addr`, записывается адрес сокета клиента, который установил соединение с сервером. В переменную, адресуемую указателем `addrlen`, изначально записывается размер структуры; функция `accept` записывает туда длину, которая реально была использована. Если вас не интересует адрес клиента, вы можете просто передать `NULL` в качестве второго и третьего параметров.

Функция `accept` является **блокирующей**.

Документация

<http://man7.org/linux/man-pages/man2/accept.2.html>

Прототип

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

Описание

Функция connect соединяет пользовательский сокет с удаленным сокетом. Эта функция используется на клиентской стороне подключения, т.к. именно клиент является инициатором подключения.

Документация

<http://man7.org/linux/man-pages/man2/connect.2.html>

boost::asio

Boost.Asio - кросс-платформенная C++ библиотека для программирования сетевых приложений и других низкоуровневых программ ввода/вывода, которая абстрагирует понятия input и output, что позволяет делать их синхронными или асинхронными.

Подробности работы с библиотекой в примере к лекции и на лабораторных работах.

Самостоятельное изучение

- асинхронное программирование
- boost.asio

Список литературы

- <https://code-live.ru/post/cpp-http-server-over-sockets/>
- <http://rdsn.org/article/unix/sockets.xml>
- <http://www.vr-online.ru/content/s-setevoe-programmirovanie-chast-1-3999>
- Radchuk - Boost.Asio C++ Network Programming Cookbook
- Камер Д., Стивенс Д. - Сети TCP/IP, том 3. Разработка приложений типа клиент/сервер для Linux/POSIX
- <http://habrahabr.ru/post/192284>
- John Torjo - Boost.Asio C++ Network Programming
- Boost.Asio - 4. TCP Socket Programming
- <http://www.eli.sdsu.edu/courses/fall12/cs580/>