1. Перечислить все ***сетевые утилиты***.

Ping

Tracert

Route

Netstat

Arp

Nslookup

Hostname

Ipconfig

Nbtstat

Net

1. Перечислить все ***уровни модели OSI/ISO***. Описать назначение каждого уровня.

**Прикладной** – определение способа взаимодействия пользователей с системой (определение интерфейса).

**Представительский** – представление данных в едином (общем) формате и виде; кодирование, шифрование, сжатие.

**Сеансовый** – установка, поддержание и разрыв сеанса.

**Транспортный** – контроль передачи данных, разбиение данных на блоки, индексация блоков данных.

**Сетевой** – адресация и маршрутизация.

**Канальный** – обеспечение доступа к среде передачи, обеспечение достоверности принимаемых данных (на основе алгоритмов контрольных сумм).

**Физический** – определяет свойства среды передачи данных и способы её соединения с сетевыми адаптерами.

1. Поясните понятие сетевой ***протокол***.

**Сетевой протокол** — набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.

Протокол – набор правил и соглашений, описывающих процедуры взаимодействия каждого уровня модели с соседними уровнями.

1. Указать где в OSI/ISO проходит ***граница между аппаратным и программным обеспечением***.

Между подуровнем управлением логической связью (LLC) и подуровнем управления доступом к среде передачи данных (MAC).

1. Определить понятие ***CSMA/CD***.

**CSMA**/**CD** (Carrier Sense Multiple Access with Collision Detection) — множественный доступ с прослушиванием несущей и обнаружением коллизий.

Разрешение коллизий???

Поскольку коллизия разрушает все передаваемые в момент ее образования кадры, то и нет смысла станциям продолжать дальнейшую передачу своих кадров, коль скоро они (станции) обнаружили коллизии. В противном случае, значительной была бы потеря времени при передаче длинных кадров. Поэтому для своевременного обнаружения коллизии станция прослушивает среду на всем протяжении собственной передачи.

1. Как называется ***программное обеспечение реализующий подуровень LLC канального уровня***.

**Драйвер.**

1. Что определяет ***спецификация NDIS***?

**NDIS** (аббр. от англ. *Network Driver Interface Specification*) — спецификация интерфейса сетевого драйвера - для сопряжения драйверов сетевых адаптеров с операционной системой.

1. Свойства ***ненадежных протоколов***. Примеры ненадежных и ***надежных*** протоколов.

1) отсутствие механизмов обеспечения надежности;

2) отсутствие гарантий доставки;

3) отсутствие обработки соединений;

4) отсутствие буферизации;

5) отсутствие фрагментации;

6) отсутствие соединения.

TCP. UDP.

1. Перечислить все уровни ***модели TCP/IP***. Описать назначение каждого уровня. Привести примеры протоколов каждого уровня.



Прикладной: HTTP, FTP, SMTP, DNS, DHCP, POP3

Транспортный: TCP, UDP

Межсетевой: IP, ICMP, ARP, RARP

Доступа к сети: Ethernet, SLIP, PPP

1. Поясните понятия ***хост***, ***адрес хоста***, ***имя хоста***.

Хост – устройство, имеющее IP-адрес.

Адрес хоста – ip адрес хоста.

Имя хоста - это доменное **имя.**

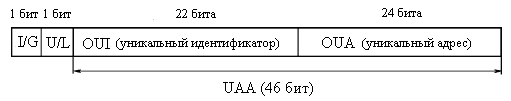
1. Какая ***организация поддерживает сетевые*** протоколы Internet. Как называются ***документы, описывающие эти протоколы***.

IETF (Internet Engineering Task Force).

RFC (Request for Comments).

1. Что такое ***МАС-адрес***? Структура ***Ethernet МАС-адреса***.

**Уникальный идентификатор**, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet. 48-битный адрес.



Младшие 24 разряда кода адреса называются OUA (Organizationally Unique Address) – *уникальный адрес*. Именно их присваивает каждый из зарегистрированных производителей сетевых адаптеров. Всего возможно свыше 16 миллионов комбинаций, то есть каждый изготовитель может выпустить 16 миллионов сетевых адаптеров.

Следующие 22 разряда кода называются OUI (Organizationally Unique Identifier) – *уникальный идентификатор*. IEEE присваивает один или несколько OUI каждому производителю сетевых адаптеров. Это позволяет исключить совпадения адресов адаптеров от разных производителей. Всего возможно свыше 4 миллионов разных OUI, это означает, что теоретически может быть зарегистрировано 4 миллиона производителей. Вместе OUA и OUI называются UAA (Universally Administered Address) – универсально управляемый адрес или IEEE-адрес.

Два старших разряда адреса управляющие, они определяют тип адреса, способ интерпретации остальных 46 разрядов. Старший бит I/G (Individual/Group) указывает на тип адреса. Если он установлен в 0, то индивидуальный, если в 1, то групповой (многопунктовый или функциональный). Пакеты с групповым адресом получат все имеющие этот групповой адрес сетевые адаптеры. Причем групповой адрес определяется 46 младшими разрядами. Второй управляющий бит U/L (Universal/Local) называется флажком универсального/местного управления и определяет, как был присвоен адрес данному сетевому адаптеру. Обычно он установлен в 0. Установка бита U/L в 1 означает, что адрес задан не производителем сетевого адаптера, а организацией, использующей данную сеть. Это случается довольно редко.

1. Как ***посмотреть MAC-адрес сетевой карты*** на компьютере?

Физический **адрес в ipconfig /all.**

1. Основное назначение ***межсетевого уровня***.

Предназначается для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

1. Структура ***IP-адреса***.

Адрес IP представляет собой 32-разрядное двоичное число, разделенное на группы по 8 бит, называемых ***октетами***. Например, 00010001 11101111 00101111 01011110.

IP-адрес состоит из двух логических частей – *номера подсети* (IDподсети) и *номера узла* (ID хоста) в этой подсети. При передаче пакета из одной подсети в другую используется ID подсети. Когда пакет попал в подсеть назначения, ID хоста указывает на конкретный узел в рамках этой подсети.

128 бит по 8 в каждом через двоеточие.

1. Типы ***IP-адресации***. ***Классы адресов Internet***.

Классовая и бесклассовая.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Диапазон адресов** | **Диапазон частных адресов** |
| А | 0.0.0.0 – 127.255.255.255 | 10.0.0.0 – 10.255.255.255 |
| B | 128.0.0.0 – 191.255.255.255 | 172.16.0.0 – 172.31.255.255 |
| C | 192.0.0.0 – 223.255.255.255 | 192.168.0.0 – 192.168.255.255 |
| D | 224.0.0.0 – 239.255.255.255 | не предусмотрен |
| E | 240.0.0.0 – 247.255.255.255 | не предусмотрен |



1. Поясните понятия ***публичный IP-адрес*** и ***частный IP-адрес***.

Публичный – един в сети. Частный – доступ только из текущей сети до выхода в глобальную сеть.

1. Как посмотреть ***IP-адрес компьютера***.

Ipconfig /all ip v4.

1. Как протестировать ***IP-соединение в локальной сети?***

Ping address.

1. Как получить ***перечень сетевых узлов*** между двумя хостами?

Tracert.

1. Перечислите ***параметры настройки TCP/IP***.

Маска сети, адрес локального шлюза, адреса серверов DNS, ip

1. Поясните понятие ***маска подсети***.

Маска – 32-битная последовательность, в которой идёт непрерывная последовательность единиц, за которой следует непрерывная последовательность нулей. Двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети.

1. Основные ***отличия между IPv4 и IPv6***.

Размерность 32 и 128. Формат записи. IPv6: Автоматическая конфигурация, встроенная безопасность.

1. Поясните понятие ***сетевой порт***. На каком уровне модели TCP/IP это понятие определено.

Идентификационный номер процесса, получающего или отправляющего данные с помощью транспортного уровня.

1. Как ***классифицируются сетевые порты***.

|  |  |
| --- | --- |
| Хорошо известные номера портов | 0 – 1023 |
| Зарегистрированные номера портов | 1024 – 49151 |
| Динамические (эфемерные) номера портов | 49152– 65535 |

1. Как посмотреть какими программами заняты сетевые порты на компьютере?

Netstat.

1. Поясните понятие архитектура ***клиент/сервер***.

***Архитектура клиент–сервер*** (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов

1. Что такое ***сетевая служба***. Приведите примеры сетевых служб.

Чаще всего фоновая программа, ожидающая действий юзера и отвечающая на них.

***DHCP (Dynamic Host Configuration Protocol)*** – это сетевая служба (и протокол) Прикладного уровня TCP/IP, обеспечивающая выделение и доставку IP-адресов и сопутствующей конфигурационной информации (маска сети, адрес локального шлюза, адреса серверов DNS и т.п.) хостам. Применение DHCP дает возможность отказаться от фиксированных IP-адресов в зоне действия сервера DHCP. Описание протокола DHCP содержится в документах: RFC 1534, 2131, 2132, 2141.

1. Поясните понятие ***интерфейс внутренней петли***.

***Интерфейс внутренней петли (loopback interface)*** – интерфейс, который позволяет двум прикладным процессам, находящимся на одном хосте, обмениваться данными посредством протокола TCP/IP. При этом, как обычно, формируются дейтаграммы, но они не покидают пределы одного хоста. Для интерфейса внутренней петли, зарезервирована сеть 127.0.0.0. В соответствии с общепринятыми соглашениями, большинство операционных систем назначают для интерфейса внутренней петли адрес 127.0.0.1 и присваивают символическое имя ***localhost***.

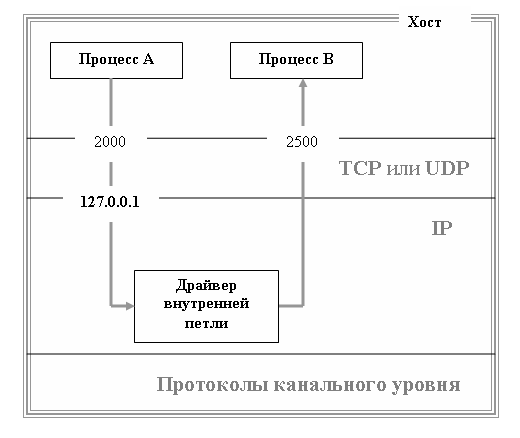


Рисунок 2.6.1. Схема работы интерфейса внутренней петли

1. Назначение сетевых служб ***DNS*** и ***DHCP***.

**DNS**: Службу DNS можно рассматривать, как распределенную иерархическую базу данных, основное назначение которой отвечать на два вида запросов: выдать IP-адрес по символическому имени хоста и наоборот – выдать символическое имя хоста по его

**DHCP** (англ. Dynamic Host Configuration Protocol — протокол динамической настройки узла) — сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

1. Организация, ведающая распределением ***IP-адресов***, поддержкой ***сетевых доменов Internet верхнего уровня***, ***регистрацией портов***.

**ICANN** (Internet Corporation for Assigned Names and Numbers).

1. Поясните понятие ***сетевой сокет***.

**Сокеты** (англ. *socket* — разъём) — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью.

Сокет — абстрактный объект, представляющий конечную точку соединения.

Совокупность адрес + порт.

1. Назначение ***стандарта POSIX***.

Стандарт POSIX (Portable Operating System Interface for Unix) – это набор документов, описывающих интерфейсы между прикладной программой и операционной системой. Стандарт создан для обеспечения совместимости различных Unix-подобных операционных систем и переносимости исходных программ на уровне исходного кода.

Прикладной софт + ОС.

1. ***Структура TCP/IP TCP-сервера***. Все функции и все параметры функций.

int WSAGetLastError(void); // Получить диагностирующий код ошибки > код

int WSAStartup( // инициализировать библиотеку WS2\_32.DLL > 0 : !0

WORD ver, //[in] версия Windows Sockets

lpWSAData wsd //[out] указатель на WSADATA

);

int WSACleanup(void); // завершить работу с библиотекой WS2\_32.DLL > 0 : SOCKET\_ERROR

SOCKET socket( // создать сокет > дескриптор сокета : INVALID\_SOCKET

int af, //[in] формат адреса (AF\_INET для стека TCP/IP)

int type, //[in] тип сокета (SOCK\_DGRAM (UDP), SOCK\_STREEM (TCP))

)

int prot //[in] протокол (NULL для TCP/IP)

);

SOCKET closesocket( // закрыть существующий сокет > нуль : SOCKET\_ERROR

SOCKET s, //[in] дескриптор сокета

);

int bind( // связать сокет с параметрами > нуль : SOCKET\_ERROR

SOCKET s, // [in] сокет

cost struct sockaddr\_in\* a, //[in] указатель на SOCKADDR\_IN

int la // [in] длина SOCKADDR\_IN в байтах

)

u\_short htons ( // преобразовать u\_short в формат TCP/IP > 16 битов в формате TCP/IP

u\_short hp // [in] 16 битов данных

);

int listen( // переключить сокет в режим прослушивания > нуль : SOCKET\_ERROR

SOCKET s, // [in] дескриптор связанного сокета

int mcq, // [in] максимальная длина очереди

);

SOKET accept( // разрешить подключение к сокету > декср-р нового сокета : INVALID\_SOCKET

SOCKET s, // [in] дескриптор связанного сокета

struct sockaddr\_in\* a, // [out] указатель на SOCKADDR\_IN

int\* la // [out] указатель на длину SOCKADDR\_IN

);

int connect ( // установить соединение с сокетом > нуль : SOCKET\_ERROR

SOCKET s, // [in] дескриптор связанного сокета

struct sockaddr\_in\* a, // [in] указатель на SOCKADDR\_IN

int la // [in] длина SOCKADDR\_IN в байтах

);

int send ( // отправить данные по установленному каналу > кол-во байт : SOCKET\_ERROR

SOCKET s, // [in] дескриптор сокета (канал для передачи)

const char\* buf, // [in] указатель буфер данных

int lbuf, // [in] количество байт данных в буфере

int flags // [in] индикатор особого режима маршрутизации

);

int recv ( // принять данные по установленному каналу > кол-во байт : SOCKET\_ERROR

SOCKET s, // [in] дескриптор сокета (канал для приема)

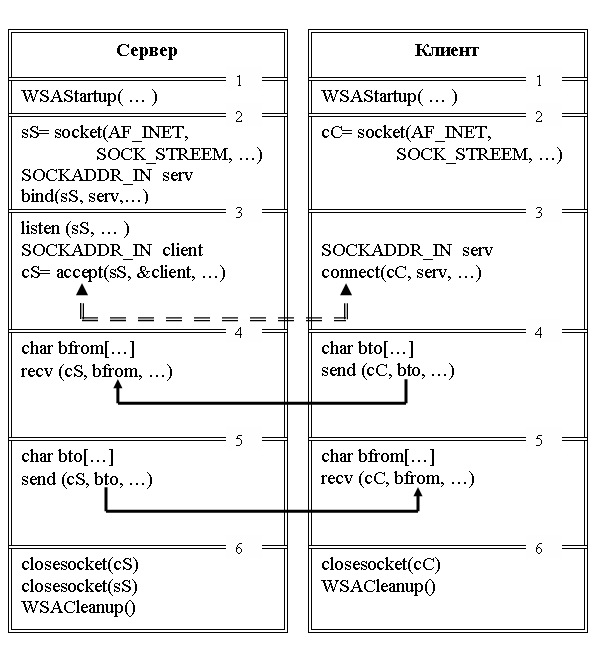
const char\* buf, // [in] указатель буфер данных

int lbuf, // [in] количество байт данных в буфере

int flags // [in] индикатор

);

1. ***Структура TCP/IP TCP-клиента***. Все функции и все параметры функций.

****