



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

имени Н.Э. БАУМАНА

Учебное пособие

Курс лекций

«Основы телекоммуникационных технологий»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

имени Н.Э. БАУМАНА

Курс лекций

«Основы телекоммуникационных технологий»

Москва
МГТУ имени Н.Э. Баумана

2012

УДК 681.3.06(075.8)

ББК 32.973-018

И201

Курс лекций «Основы телекоммуникационных технологий» / Коллектив авторов
– М.: МГТУ им. Н.Э. Баумана, 2012. – 73 с.: ил.

В курсе лекций были рассмотрены основные этапы курса «Основы телекоммуникационных технологий».

Ил. 39. Табл. 5. Библиогр. 7 назв.

УДК 681.3.06(075.8)

АННОТАЦИЯ

В курсе лекций рассмотрены основные темы курса «Основы телекоммуникационных технологий» такие как: сети, основы их построения и передачи данных.

ANNOTATION

The course of lectures addressed the main themes of the course "Basis of telecommunication technology" such as networks and basis of their building and transferring information via networks.



Вводная лекция

Основы телекоммуникационных технологий
Курс лекций

План лекции

ИУ4 – «сети»

- 1 Основные термины и определения
- 2 Виды и топологии сетей
- 3 Сетевое взаимодействие
- 4 Дополнительная информация

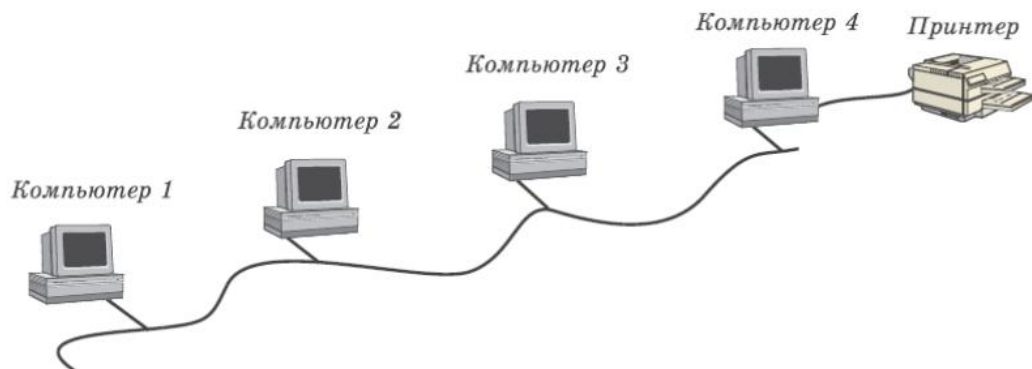
Основные термины и определения

ИУ4 – «сети»

- **Сеть (Network)** — группа компьютеров и/или других устройств, каким-либо способом соединенных для обмена информацией и совместного использования ресурсов.
 - Компьютерная сеть
 - Распределенная система
- **Ресурсы** — программы, файлы данных, а также принтеры и другие совместно используемые периферийные устройства в сети.

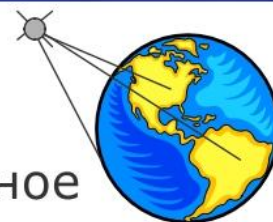
Основные термины и определения

ИУ4 – «сети»



Основные термины и определения

ИУ4 – «сети»



- **Internet** — вообще говоря не является сетью, это собирательное название разных сетей, использующих определенные общие протоколы и предоставляющие определенные сервисы.
- **ARPANET** – сеть министерства обороны США (начало 70х)
- **NSFNET** – открытая университетская сеть (конец 70х)

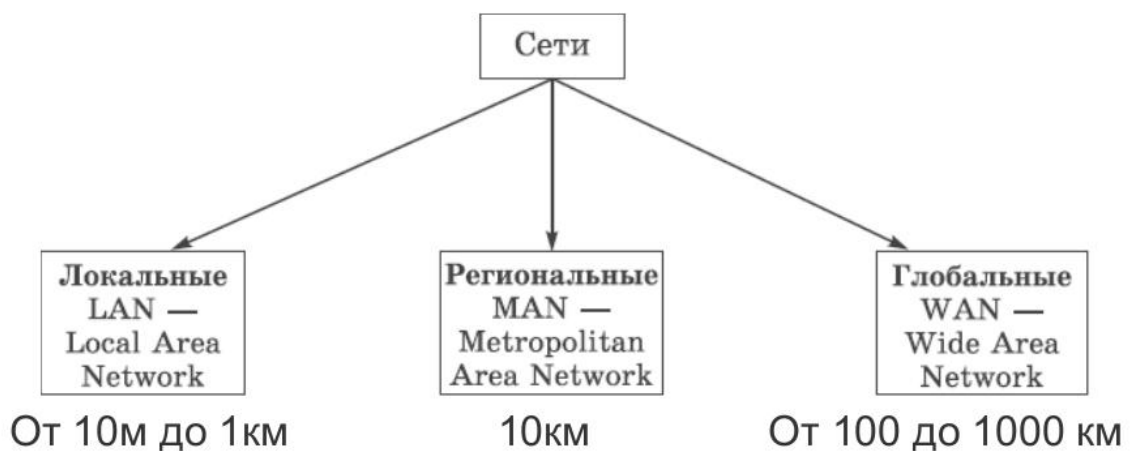
Основные термины и определения

ИУ4 – «сети»

- **Сервер** — специально выделенный высокопроизводительный компьютер, оснащенный соответствующим программным обеспечением, централизованно управляющий работой сети и/или предоставляющий другим компьютерам сети свои ресурсы.
- **Клиентский компьютер (клиент, рабочая станция)** — компьютер рядового пользователя сети, получающий доступ к ресурсам сервера (серверов).
- **Службы (services)** — работающие на серверах программы, выполняющие какие-либо действия по запросу клиента.

- **Протокол** — набор правил и процедур, регулирующих порядок взаимодействия компьютеров в сети.

- **Классификация сетей по расстоянию между узлами**



Виды и топологии сетей

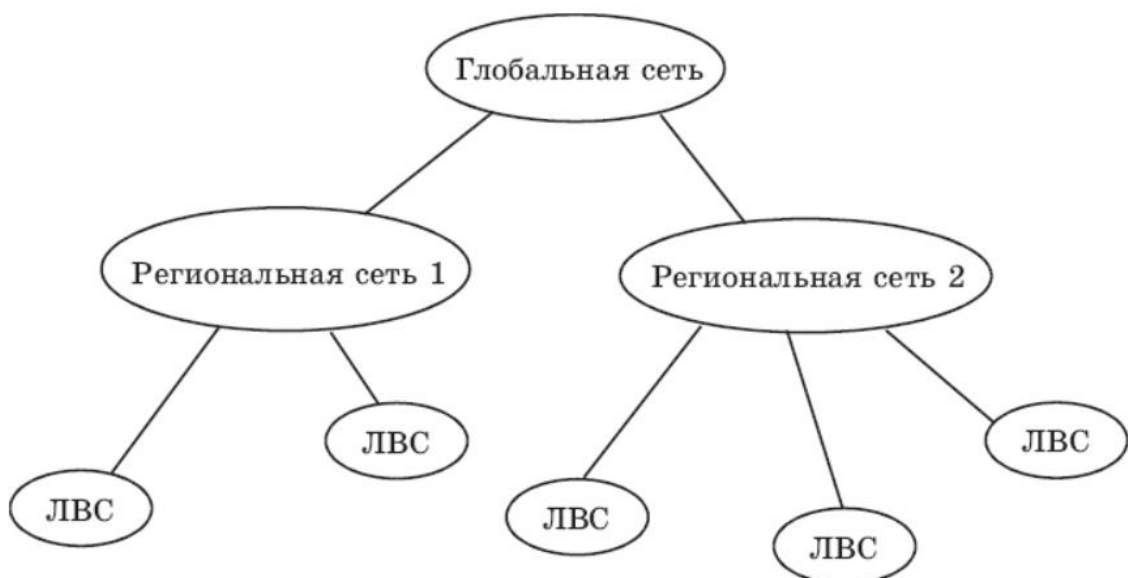
ИУ4 – «сети»

- **Локальная вычислительная сеть (ЛВС) (Local Area Network - LAN)** - небольшая группа компьютеров, связанных друг с другом и расположенных обычно в пределах одного здания или организации.
- **Региональная сеть (Metropolitan Area Network - MAN)** - сеть, соединяющая множество локальных сетей в рамках одного района, города или региона.
- **Глобальной сетью (Wide Area Networks - WAN)** - сеть, объединяющая компьютеры разных городов, регионов и государств.

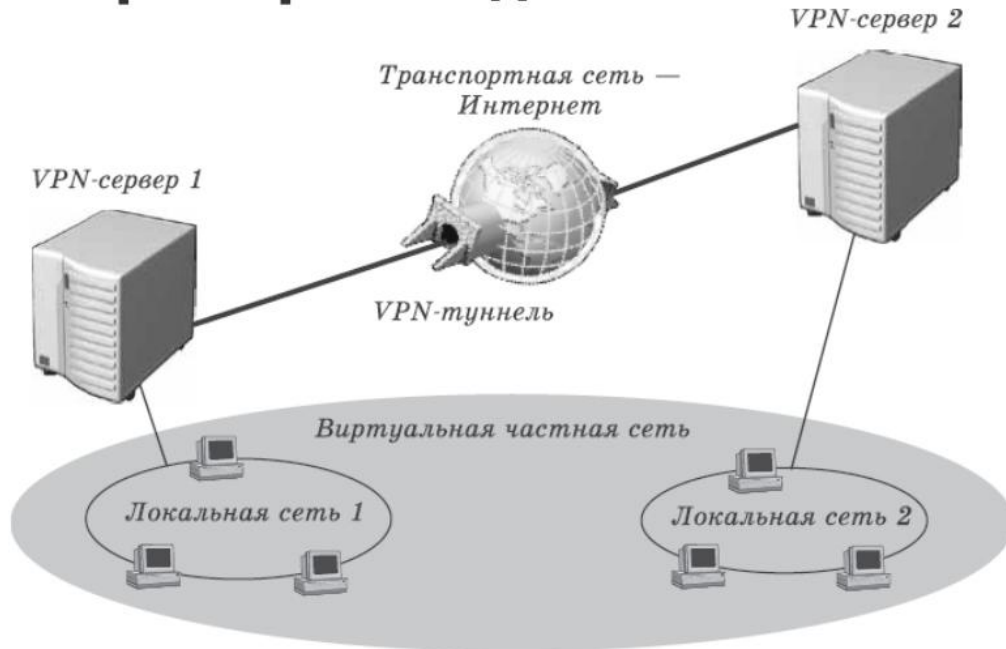
Виды и топологии сетей

ИУ4 – «сети»

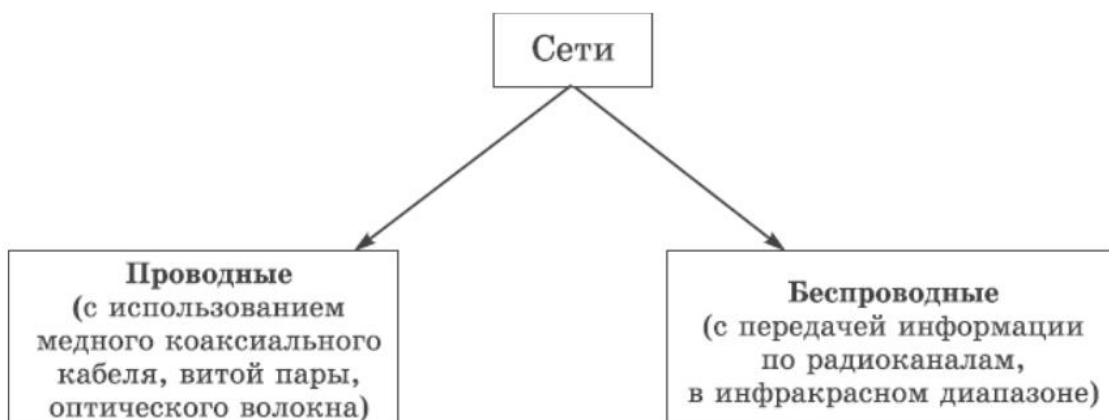
Пример объединения сетей



Пример объединения сетей



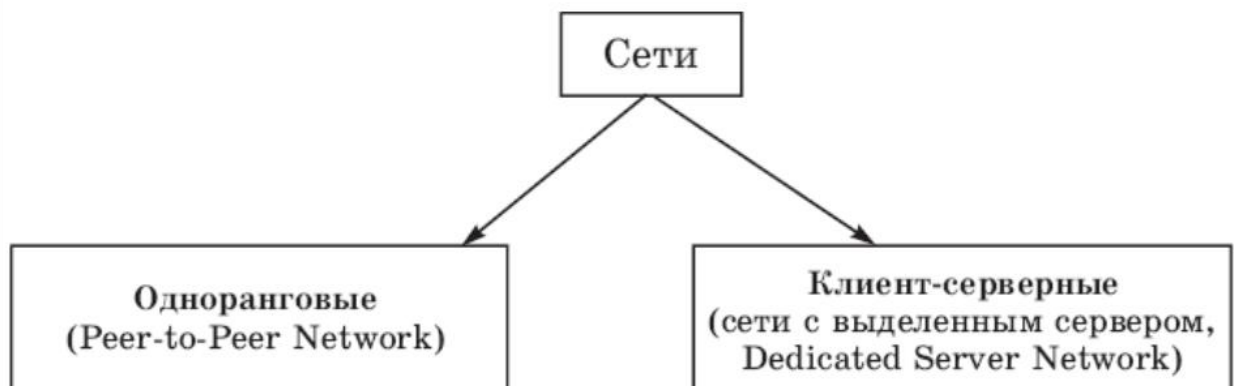
- **Классификация сетей по типу среды передачи**



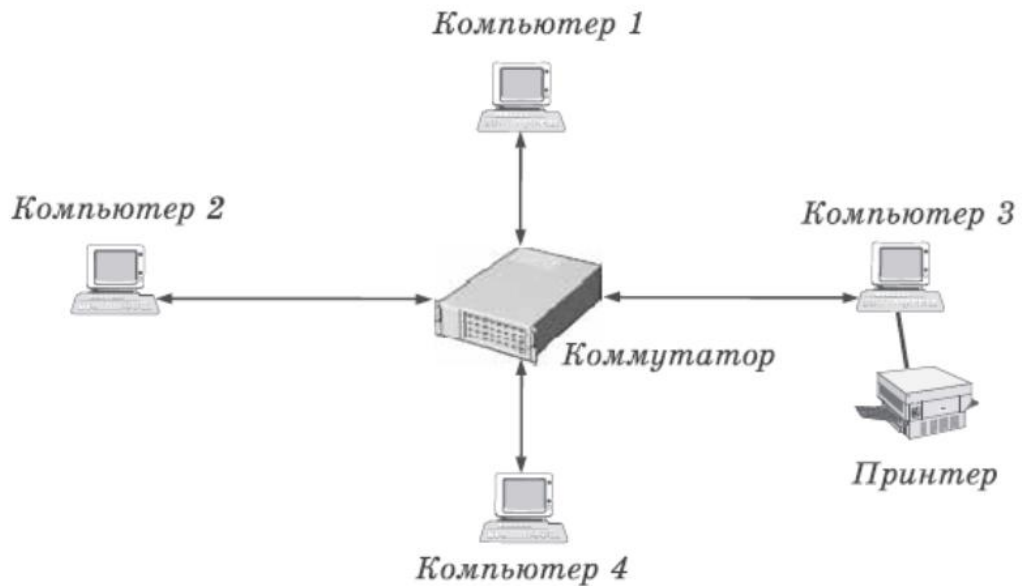
- **Классификация сетей по скорости передачи**



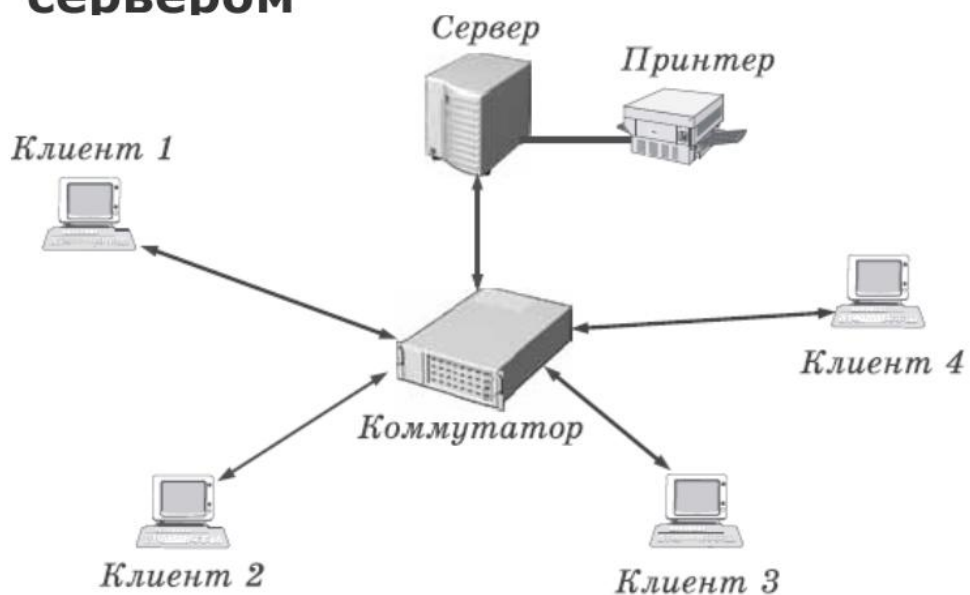
- **Классификация сетей по распределению ролей компьютеров**



● Пример одноранговой сети



● Пример сети с выделенным сервером



ПОНЯТИЕ КОНВЕРГЕНЦИИ СЕТЕЙ

Понятие сети:

- Различно для различных моделей.
- Цель: обеспечивает совместное использование ресурсов.
- Современные сети сложны и часто являются объединением множества различных типов сетей.

Глобальная цифровая
телекоммуникационная
сеть

- Сети передачи данных
- Телефонные сети
- Сети вещания радио и TV

Сетевые архитектуры

Локальные сети

- Ethernet,
- Token Ring,
- FDDI,
- ARCnet



Глобальные сети

- SLIP, PPP,
- X.25, Frame Relay, ATM



Виды и топологии сетей

ИУ4 – «сети»

Устройства физического уровня

- СКС: Провода, разъемы, терминаторы, фильтры...
- Каналообразующее оборудование: Концентраторы, коммутаторы, маршрутизаторы, мосты, репитеры, конверторы, модемы...

Виды и топологии сетей

ИУ4 – «сети»

Топологии Ethernet



Геометрическое расположение узлов и связей

Шина

Звезда

Кольцо

Большие системы состоят из смеси из двух топологий – шины и звезды

Необходимость **решений взаимосвязи**

Увеличение расстояний,

количества устройств,

разделение трафика,

объединение различных

сегментов в единую сеть

Виды и топологии сетей

ИУ4 – «сети»

Топология «звезда»

Наименование	Схематическое изображение типа топологии	Характеристики	Надежность
Звезда			

Виды и топологии сетей

ИУ4 – «сети»

Топология «звезда»

Устройства связаны с концентратором или коммутатором в центре
Сообщение передаётся в центр и рассылается другим узлам
Вышедшие из строя узлы не затрагивают другие узлы в сети
Концентраторы могут быть объединены для создания сетей большего масштаба

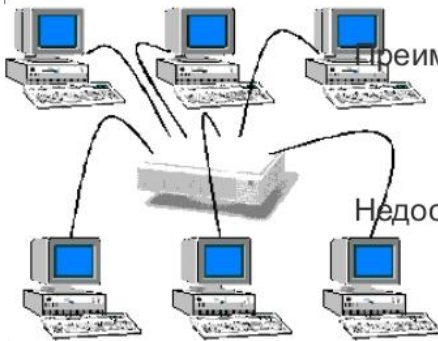
Вышедшие из строя части сети могут быть как отдельными устройствами, так и целыми сегментами

Преимущества топологии «Звезда»:

Недорого, быстрая установка, упрощённый механизм связи, простое расширение в случае необходимости

Недостатки топологии «Звезда»:

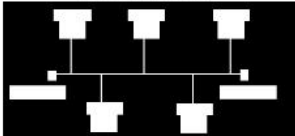
100 метров – максимальное расстояние между узлом и концентратором



Виды и топологии сетей

ИУ4 – «сети»

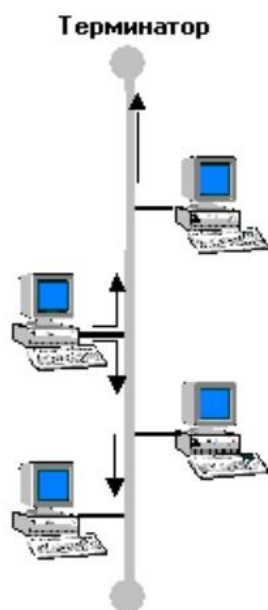
Топология «шина»

Наименование	Схематическое изображение типа топологии	Характеристики	Надежность
Шина			

Виды и топологии сетей

ИУ4 – «сети»

Топология «шина»



Все узлы подключены параллельно на один провод коаксиала

Сигналы проходят в оба направления от передающего узла

Базовый блок создания сети Ethernet

Если кабель нарушен в любой точке, то система терпит неудачу

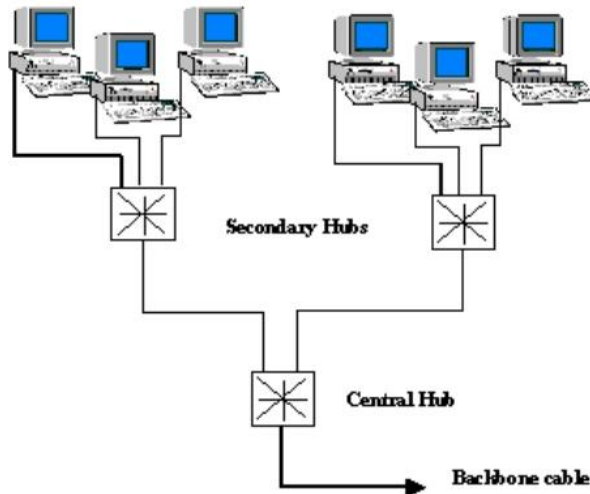
Недостаток необходимого использования терминаторов для согласования сети на концевых узлах, подавления шума в шине и пр.

Used primarily a "backbone" to interconnect other LAN sections

Виды и топологии сетей

ИУ4 – «сети»

Топология «дерево»



Объединяет характеристики шинной топологии и звезды.
Состоит из групп АРМ, объединённых по топологии «звезда». Группы связаны друг с другом шиной.
Топология «Дерево» учитывает расширение существующей сети
Преимущество - Больше количество устройств подключено к центральному центру
Сеть изолирует и располагает по приоритетам коммуникации от различных ПК

Виды и топологии сетей

ИУ4 – «сети»

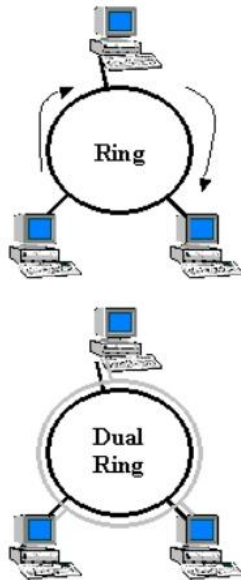
Топология «кольцо»

Наименование	Схематическое изображение типа топологии	Характеристики	Надежность
Кольцо		<ol style="list-style-type: none">1. Сложность монтажа – средняя,2. Стоимость монтажа – низкая, не требует наличия дополнительного оборудования3. Максимальная длина сегмента – 185м4. Расстояние между узлами – 2.5м.5. Возможности расширения – низкие.	Низкая, неполадки на одной станции выводят из строя всю сеть.

Виды и топологии сетей

ИУ4 – «сети»

Топология «кольцо»



Кольцо - устройства связаны друг с другом по сети, образующей кольцо

Сигналы проходят только в одном направлении

Каждое устройство действует как повторитель и усиливает сигнал перед передачей его на следующее устройство в кольце

Если любое устройство терпит неудачу, кольцо нарушается

Двойное Кольцо - два оптических волокна с взаимно противоположными каналами передачи данных

Виды и топологии сетей

ИУ4 – «сети»

Что же такое Ethernet?

Устойчивый стереотип:
Ethernet – один кабель,
проходящий через предприятие



- Различные типы носителей
- Сеть, разделяемая различными устройствами
- Различные топологии сети



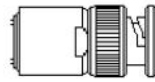
Виды и топологии сетей

ИУ4 – «сети»

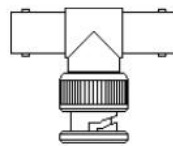
СКС – тонкий коаксиал (Ethernet)

Позиция, наименование, внешний вид

I – терминатор



II – T-коннектор



III – I-коннектор

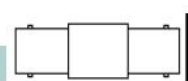
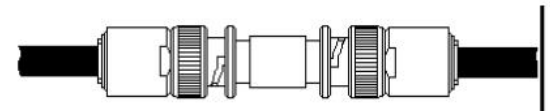
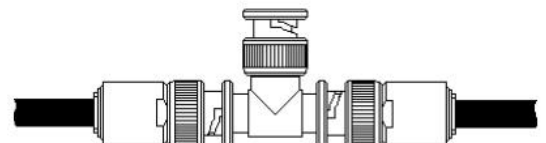
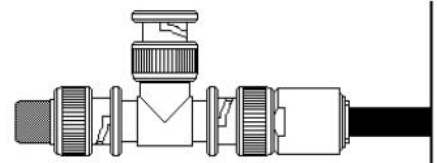


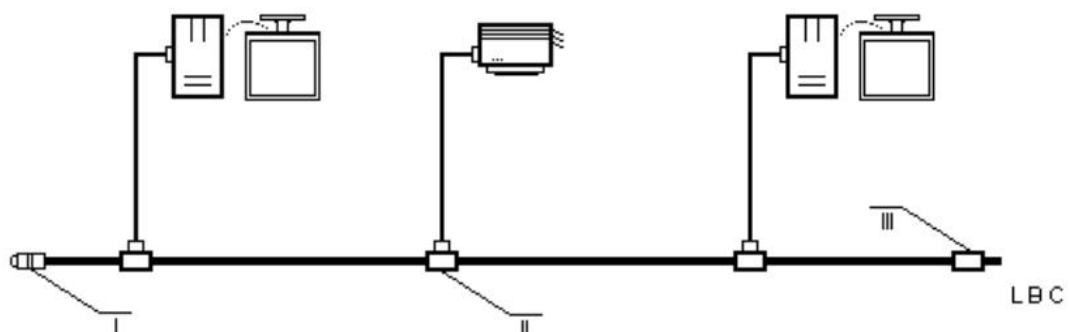
Схема соединения



Виды и топологии сетей

ИУ4 – «сети»

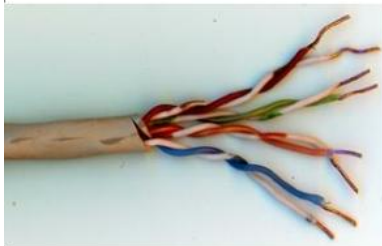
СКС – тонкий коаксиал (Ethernet)



Виды и топологии сетей

ИУ4 – «сети»

СКС витая пара



Наименование операции

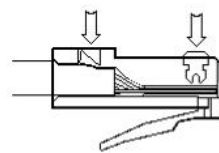
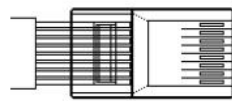
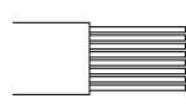
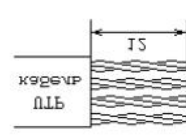
1. Удаление внешней изолирующей оболочки

2. Раскрутка пар, распрямление проводов

3. Укладка проводов в гнезда разъема RJ-45

4. Обжимание разъема

Эскиз операции



Описание операции

Производится удаление внешней изолирующей оболочки кабеля УТР на расстояние 10..12 мм при помощи бокорезов

Раскручиваются пары, распрямляются проводники и производится их укладка в стандартной последовательности (для прямого кабеля): б/оранжевый, оранжевый, б/зеленый, синий, б/синий, зеленый, б/коричневый, коричневый (на эскизе сверху вниз).

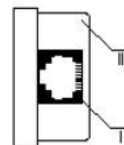
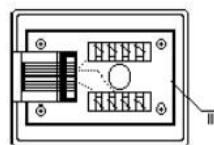
Укладка проводников в соответствующие гнезда разъема RJ-45 и вталкивание их до упора. Контроль правильности расположения проводников

Обжимание разъема при помощи специального обжимного инструмента. При этом фиксатор (слева) обеспечивает фиксацию проводника, а контакты (справа) утапливаются внутрь корпуса и прорезают изоляцию, что обеспечивает контакт.

Виды и топологии сетей

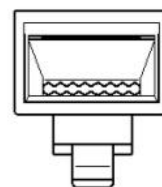
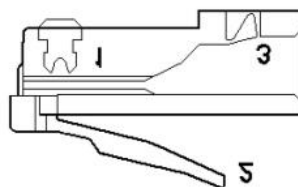
ИУ4 – «сети»

I – розетка



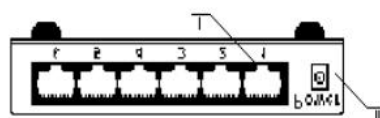
1 – порт
2 – корпус
3 – коммутационная плата

II – разъемы RJ-45



1 – контакты
2 – фиксатор разъема
3 – фиксатор провода

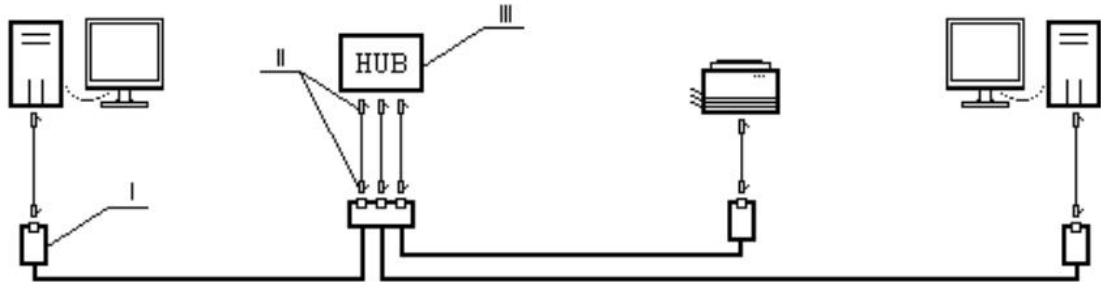
III – хаб



1 – порт
2 – корпус

Виды и топологии сетей

ИУ4 – «сети»



Виды и топологии сетей

ИУ4 – «сети»

СКС - инструменты

Внешний вид
инструмента/прибора

Наименован
ие

Назначение



Обжимные
клевши
(crimping).

Обжимные клещи (crimping) предназначены для монтажа разъема RJ-45 на кабель типа «витая пара». Проводники располагаются в порядке, соответствующем выбранной схеме, после чего вставляются в разъем и обжимаются клещами. При этом фиксатор прижимает кабель, а контакты утапливаются в разъем и прорезают изоляцию проводников, обеспечивая электрический контакт.

Тоновый
генератор,
индукцион
ный
трассировщи
к
-
усилитель.

Индукционный трассировщик – усилитель применяется для поиска электрических проводов в капитальных конструкциях. В случае наличия в зоне работы прибора электромагнитных полей он издает характерный звук. В комбинации с тоновым генератором применяется для обнаружения обрыва в проводниках. В этом случае с одного конца проводника подключается генератор импульсов и производится поиск места обрыва, руководствуясь наличием или отсутствием характерного сигнала.

Нож
монтажный.

Нож монтажный применяется для монтажа портов розеток на кабель типа «витая пара». Проводники распутываются и располагаются в соответствии с выбранной схемой монтажа, после чего при помощи ножа вдавливаются в соответствующие гнезда порта. При этом происходит одновременно зачистка изоляции проводника для обеспечения электрического контакта и подрезка его выступающей части.

Тестер
многофунк
цио
нальный,
заглушка.

Тестер многофункциональный предназначен для проверки правильности монтажа и наличия электрического контакта в проводниках. Позволяет определить общую длину отрезка кабеля и в случае наличия ошибок монтажа обнаружить их. При работе с прибором на противоположный конец проводника подключается заглушка.

Виды и топологии сетей

ИУ4 – «сети»

СКС - инструменты

Внешний вид инструмента/прибора



Наименование

Обжимные клещи (crimper).

Тоновый генератор, индукционный трассировщик усилитель.

Нож монтажный.

Тестер многофункциональный, заглушка.

Назначение

Обжимные клещи (crimper) предназначены для монтажа разъемов RJ-45 на кабель типа «витая пара». Проводники располагаются в порядке, соответствующем выбранной схеме, после чего вставляются в разъем и обжимаются клещами. При этом фиксатор прижимает кабель, а контакты утапливаются в разъем и прорезают изоляцию проводников, обеспечивая электрический контакт.

Индукционный трассировщик – усилитель применяется для поиска электрических проводов в капитальных конструкциях. В случае наличия в зоне работы прибора электромагнитных полей он издает характерный звук. В комбинации с тоновым генератором применяется для обнаружения обрыва в проводниках. В этом случае с одного конца проводника подключается генератор импульсов и производится поиск места обрыва, руководствуясь наличием или отсутствием характерного сигнала.

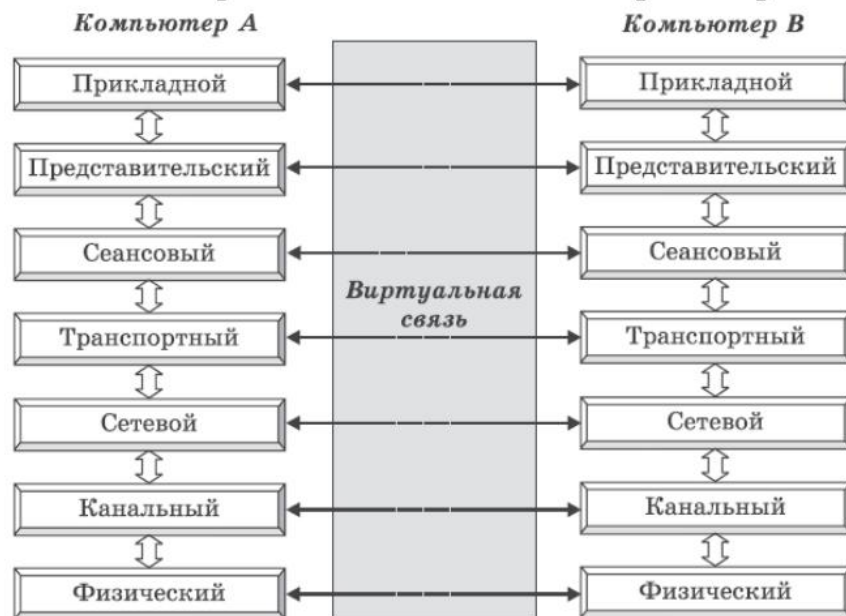
Нож монтажный применяется для монтажа портов розеток на кабель типа «витая пара». Проводники распутываются и располагаются в соответствии с выбранной схемой монтажа, после чего при помощи ножа вдавливаются в соответствующие гнезда порта. При этом происходит одновременно зачистка изоляции проводника для обеспечения электрического контакта и подрезка его выступающей части.

Тестер многофункциональный предназначен для проверки правильности монтажа и наличия электрического контакта в проводниках. Позволяет определить общую длину отрезка кабеля и в случае наличия ошибок монтажа обнаружить их. При работе с прибором на противоположный конец проводника подключается заглушка.

Сетевое взаимодействие

ИУ4 – «сети»

Эталонная модель взаимодействия открытых систем (OSI)



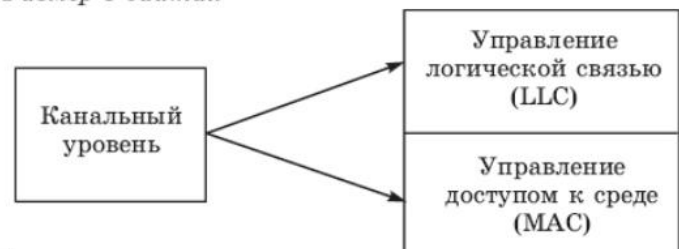
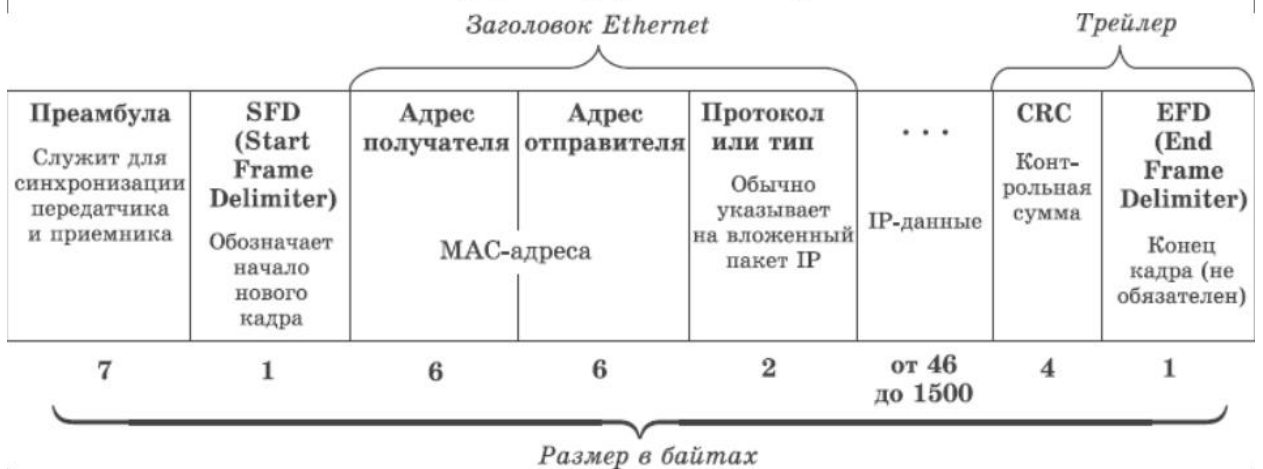
Уровни модели OSI

- **Уровень 0** — На этом уровне ничего не описывается, уровень 0 предоставляет физическому уровню 1 только *среду передачи* (кабели различных типов, радио-, ИК-сигналы и т. д.).
- **Уровень 1 — Физический (Physical)**.
Здесь осуществляется передача неструктурированного потока битов, полученных от вышележащего канального уровня 2, по физической среде - например, в виде электрических или световых сигналов. Физический уровень отвечает за поддержание связи (*link*) и детально описывает электрические, оптические, механические и функциональные интерфейсы со средой передачи.

Уровни модели OSI

- **Уровень 2 — Канальный (Data Link)**.
Обеспечивает *безошибочную передачу данных*, полученных от вышележащего сетевого уровня 3, через физический уровень 1, который сам по себе отсутствия ошибок не гарантирует и может исказить данные. Информация на этом уровне помещается в *кадры (frames)*, где в начале (*заголовке кадра*) содержатся адреса получателя и отправителя, а также управляющая информация, а в конце — *контрольная сумма*, позволяющая выявить возникающие при передаче ошибки.

Структура кадра



Уровни модели OSI

- Уровень 3 – Сетевой (Network).**
 Отвечает за обеспечение связи между любыми, даже находящимися в разных концах земного шара, точками в сети. Этот уровень осуществляет проводку сообщений по сети, которая может состоять из множества отдельных сетей, соединенных множеством линий связи. Такая доставка требует маршрутизации, т. е. определения пути доставки сообщения, а также решения задач управления потоками данных и обработки ошибок передачи.

Уровни модели OSI

- **Уровень 4 – Транспортный (Transport).**

Гарантирует доставку информации от одного компьютера другому. На этом уровне компьютера-отправителя большие блоки данных разбиваются на более мелкие пакеты, которые доставляются компьютеру-получателю *в нужной последовательности, без потерь и дублирования*. На транспортном уровне компьютера-получателя пакеты вновь собираются в исходные блоки данных. Таким образом, транспортный уровень *завершает процесс передачи данных*, скрывая от более высоких уровней все детали и проблемы, связанные с доставкой информации *любого объема* между *любыми точками* во всей сети.

Уровни модели OSI

- **Уровень 5 – Сеансовый (Session).**

Позволяет двум *сетевым приложениям* на разных компьютерах устанавливать, поддерживать и завершать соединение, называемое *сетевым сеансом*. Этот уровень также отвечает за восстановление аварийно прерванных сеансов связи. Кроме того, на пятом уровне выполняется преобразование удобных для людей имен компьютеров в сетевые адреса (*распознавание имен*), а также реализуются функции *защиты сеанса*.

Уровни модели OSI

- **Уровень 6 – Представительский, или Уровень представления данных (Presentation).**
Определяет *форматы* передаваемой между компьютерами информации. Здесь решаются такие задачи, как *перекодировка* (перевод информации в вид, понятный для всех участвующих в обмене компьютеров), сжатие и распаковка данных, шифрование и дешифровка, поддержка сетевых файловых систем и т. д.

Уровни модели OSI

- **Уровень 7 – Прикладной (Application), или Уровень Приложений.** Обеспечивает интерфейс взаимодействия программ, работающих на компьютерах в сети. Именно с помощью этих программ пользователь получает доступ к таким сетевым услугам, как обмен файлами, передача электронной почты, удаленный терминальный доступ и т.д.

модель OSI



модель OSI

- Каждый уровень модели OSI имеет свою систему адресации (адресное пространство)
- Примеры адресных пространств:
 - MAC-адреса (канальный уровень)
 - IP-адреса (сетевой уровень)
 - номера портов (транспортный уровень)
- По утвержденному стандарту порядок передачи байт - старший байт передается первым.

Понятие маршрутизации

- Смысл - выбор пути доставки пакетов (дейтаграмм).
- Осуществляется с помощью доставки на сетевом уровне (IP).
- Производится шлюзами.
- Контроль осуществляется специальным протоколом.

Понятие маршрутизации

Данные

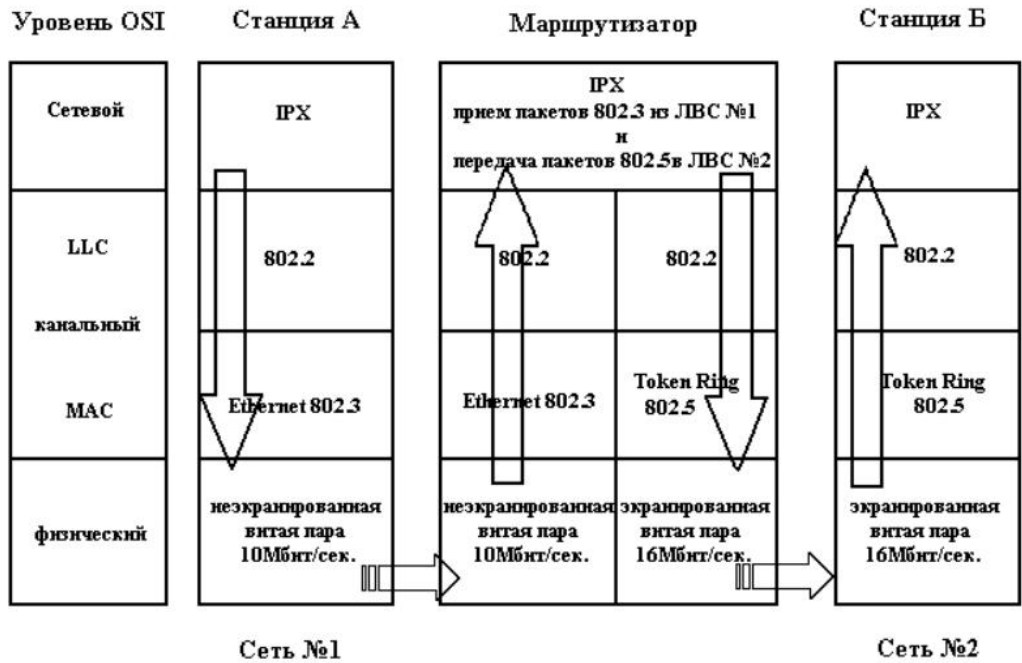
Данные



Сетевое взаимодействие

ИУ4 – «сети»

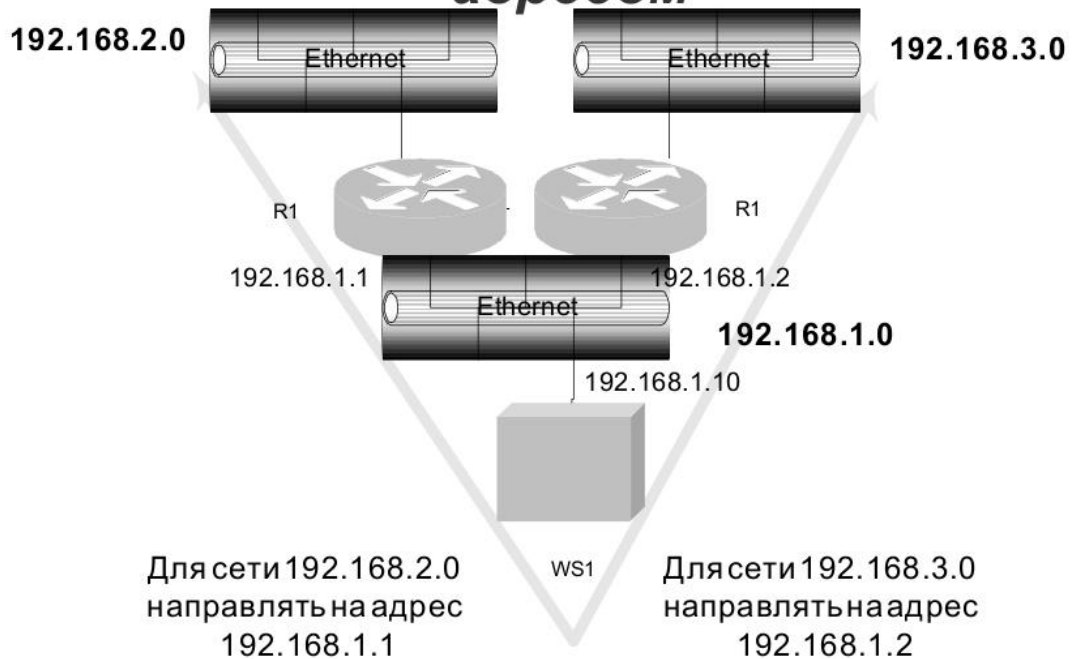
Понятие маршрутизации



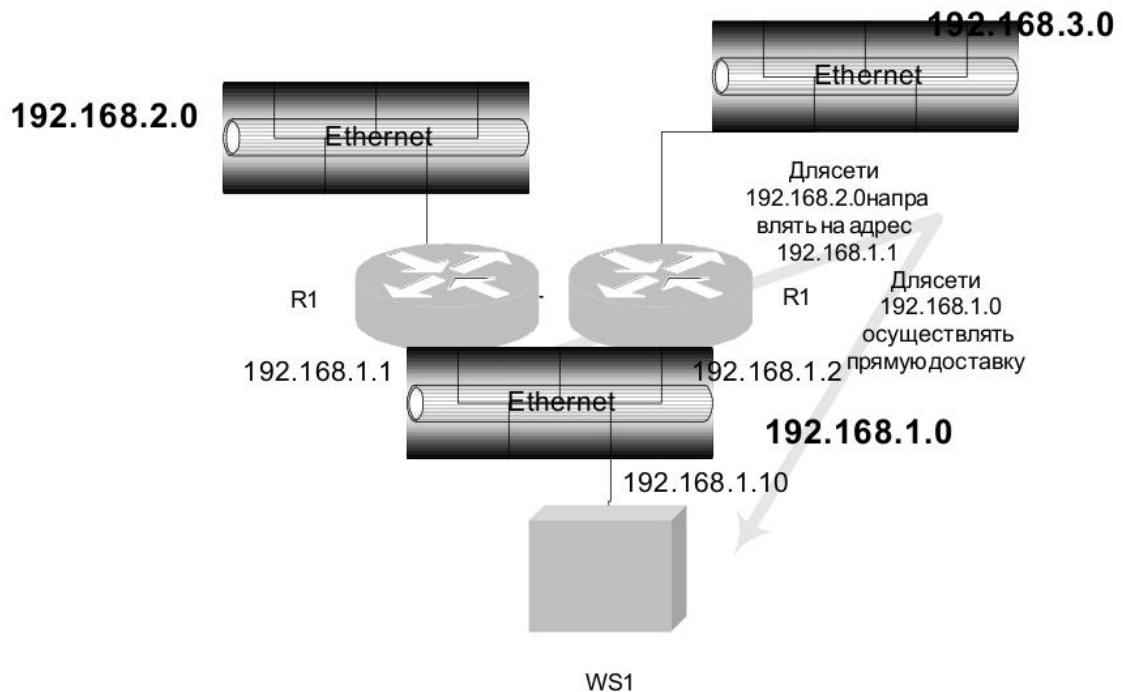
Сетевое взаимодействие

ИУ4 – «сети»

Маршрутизация для узла с одним адресом



Маршрутизация шлюзами



Статическая и динамическая маршрутизация

- Стратегия маршрутизации.
- Маленький блок имеет приоритет.
- Таблицы маршрутизации.
- Протоколы маршрутизации.
- Маршруты по умолчанию.
- Различные способы оптимизации выбора шлюза в локальной сети.

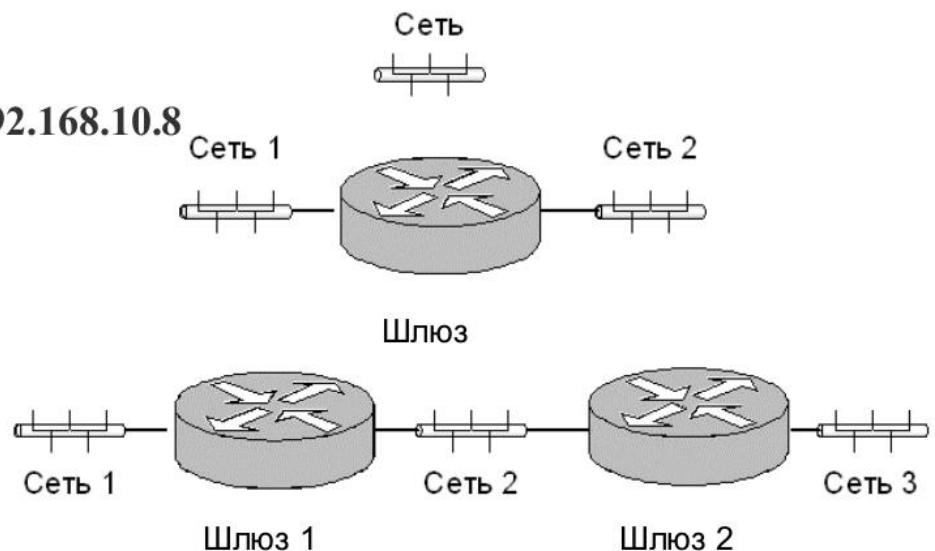
Сетевые сервисы

- **Telnet** - 1972 г., **FTP** - 1973 г., TCP - 1974 г., IP - 1981 г., TCP/IP Suite - 1982 г., DNS - 1984 г.
- Другие протоколы: SMTP, SNMP, X-Windows, HTTP.

Дополнительная информация

Команды анализа сетевой инфраструктуры

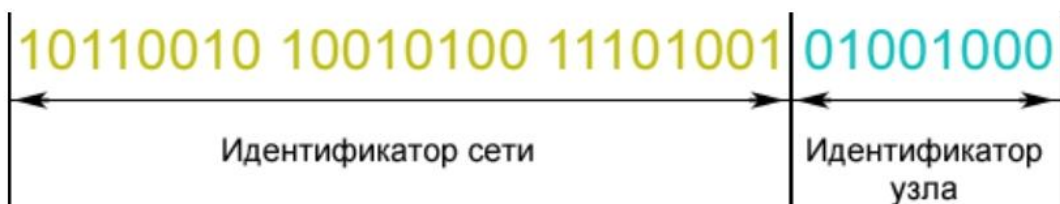
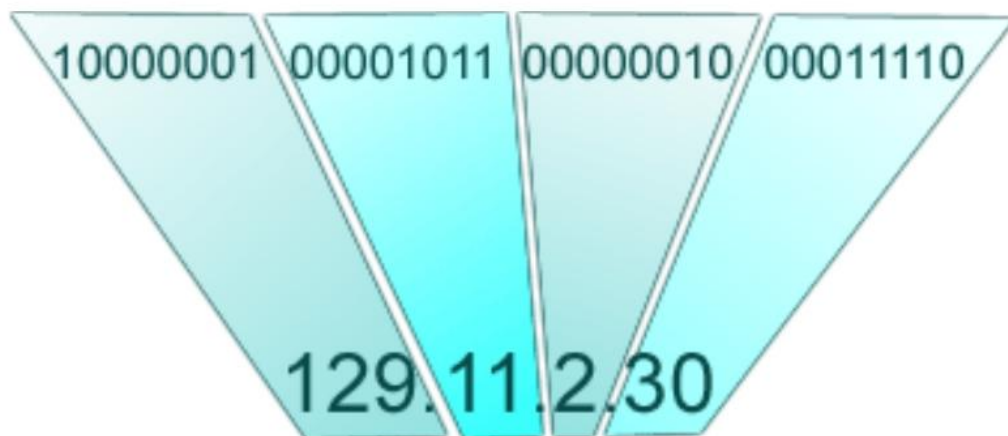
ARP
IPCONFIG
NET
PING
TRACERT 192.168.10.8



Адресация в IP-сетях

- Разрабатывалась для больших сетей с ненадежной доставкой.
- Адрес - уникальный 32-разрядный номер.
- В адресе закодирована сетевая и узловая части.
- Адрес описывает соединение, а не оборудование сети.
- Некоторые адреса зарезервированы.
- Требуется механизм сопоставления с адресами канального уровня.

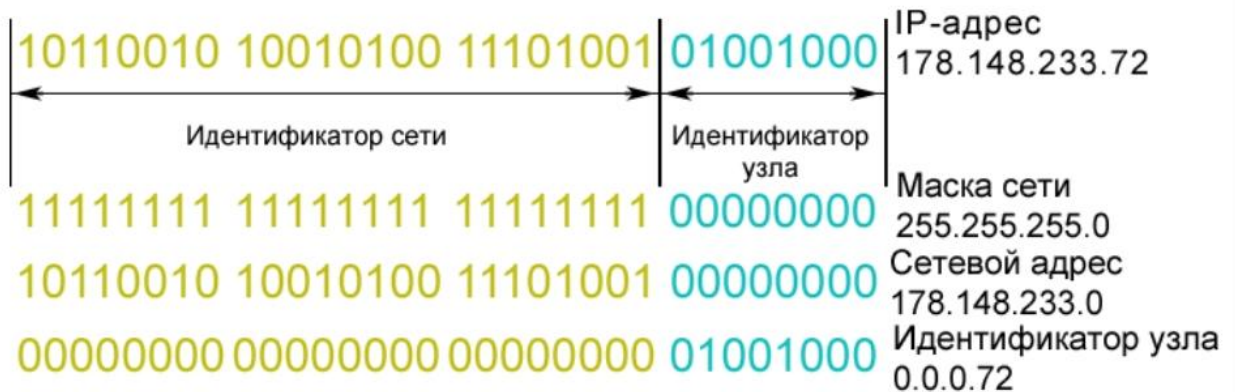
Десятичная нотация IP-адреса



Дополнительная информация

ИУ4 – «сети»

Сетевая маска



Дополнительная информация

ИУ4 – «сети»

Классовая модель IP адресации

0XXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	Класс А
10XXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	Класс В
110XXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	Класс С
1110XXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	Класс D
11110XXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	Класс E

- XX Идентификатор сети
- XX Идентификатор узла
- 10 Определяющие биты
- XX Групповой адрес
- XX Резервировано

Классовая модель IP адресации

- 10.0.0.0/8 — сеть класса А. Занимает пространство адресов 10.0.0.0 — 10.255.255.255.
- 172.16.0.0/16 — 172.31.0.0/16 — 16 сетей класса В. Занимают пространство адресов 172.16.0.0 — 172.31.255.255.
- 192.168.0.0/24 — 192.168.255.0/24 — 256 сетей класса С. Занимают 192.168.0.0 — 192.168.255.255.

Бесклассовая модель IP адресации

Класс В **10110010 10010100** 11101001 01001000

IP-адрес: 178.148.233.72

Маска в соответствии с классом: 255.255.0.0 (/16)

Сетевой адрес в соответствии с классом: 178.148.0.0

Подсеть **10110010 1001000** 11101001 01001000

IP-адрес: 178.148.233.72

Маска подсети: 255.255.128.0 (/17)

Сетевой адрес подсети: 178.148.128.0

Бесклассовая модель IP адресации

Класс В 10110010 10010100 11101001 01001000

IP-адрес: 178.148.233.72

Маска в соответствии с классом: 255.255.0.0 (/16)

Сетевой адрес в соответствии с классом: 178.148.0.0

Supernet

10110010 10010100 11101001 01001000

IP-адрес: 178.148.233.72

Маска supernet: 255.248.0.0 (/13)

Сетевой адрес supernet: 178.144.0.0

Специальные типы IP адресов

Сетевой адрес. (Маршруты, определение сетей)

«Моя сеть». Destination – NO!

«Я». Destination – NO!

«Я в моей сети». Destination – NO!

Широковещательный адрес. Source – NO!

Ограниченный широковещательный адрес.

Source – NO!

Групповые адреса. (Class D)

Адрес loopback.

Зарезервированные адреса. (Class E)

Дополнительная информация

ИУ4 – «сети»

Протоколы: ARP

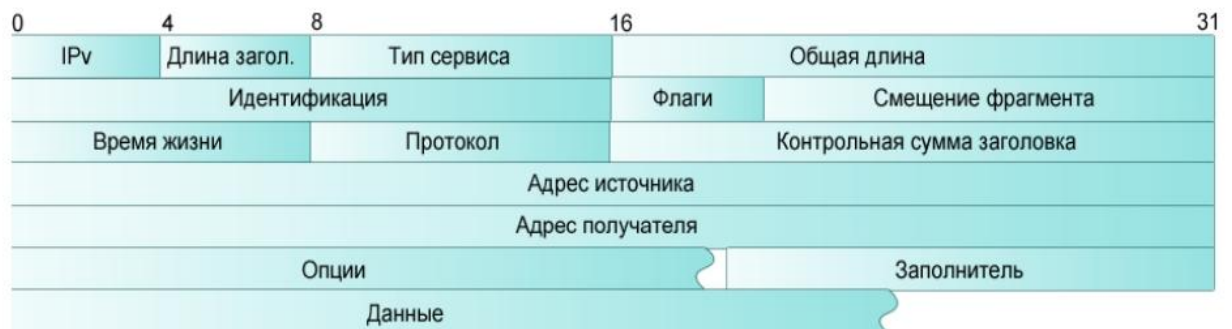
- ARP - поиск аппаратного адреса по IP адресу.
- ARP - реализация кэширования в Ethernet-сетях.
- Reverse ARP - поиск IP адреса по аппаратному.
- ARP-представитель.
- Утилита arp.



Дополнительная информация

ИУ4 – «сети»

Протоколы: IP

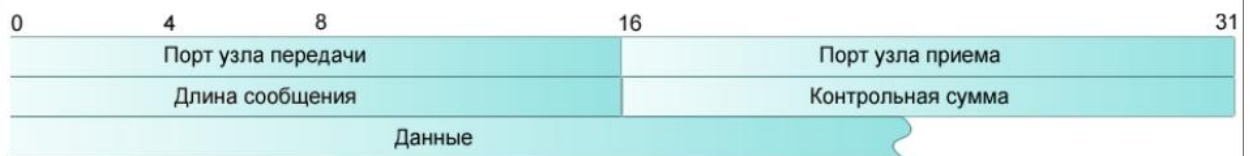


Дополнительная информация

ИУ4 – «сети»

Протоколы: UDP

- Только IP-протокол в качестве транспортного.
- Нет контроля.
- Необязательный контроль целостности.
- Поддержка многопоточковых и многозадачных сред.
- Простота.

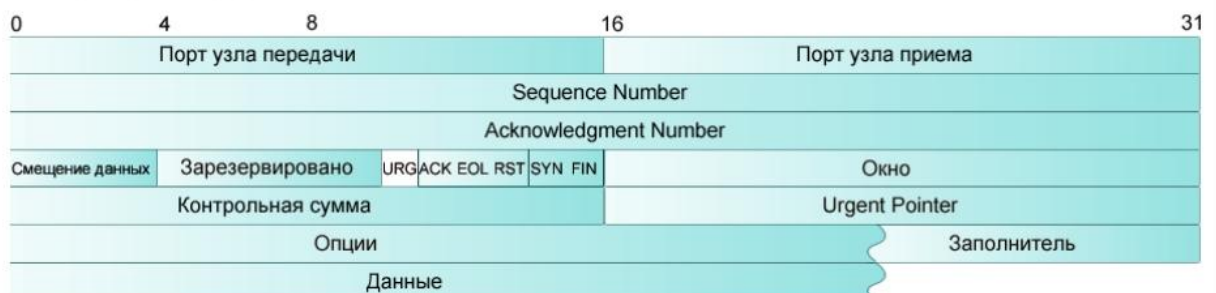


Дополнительная информация

ИУ4 – «сети»

Протоколы: TCP

- Любой сетевой уровень.
- Контроль доставки («сессии») и управление потоком («окно»).
- Контроль целостности.
- Поддержка многопоточковых и многозадачных сред.
- Сложность.





skycluster@gmail.com



Сети своими руками

Основы телекоммуникационных технологий
Курс лекций

План лекции

ИУ4 – «сети»

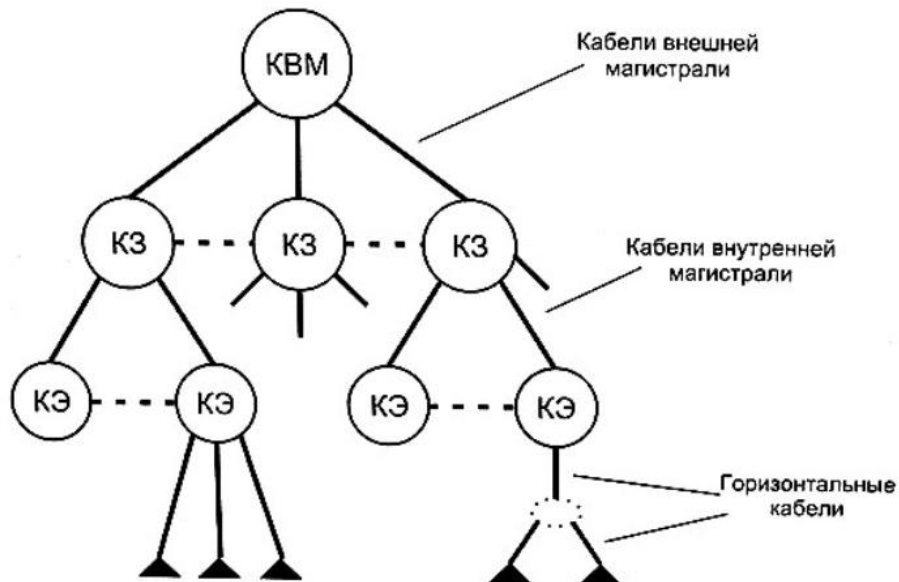
- 1 Основные термины и определения
- 2 Сетевые технологии
- 3 ??
- 4 Дополнительная информация

Основные термины и определения

ИУ4 – «сети»

- **СКС** – Структурированная Кабельная Система
- **КЗ** – Кроссовая здания
- **КЭ** – Кроссовая этажа
- **КВМ** – Кроссовая Внешних Магистралей
- **АТС** – Автоматическая Телефонная Станция
- **ISO** – International Organization for Standartization
- **ЛВС** – локальная вычислительная сеть
- **MUTO** – Multiuser Telecommunication Outlet

Топология СКС



Топология СКС

Основой для применения именно иерархической звездообразной топологии является возможность ее использования для поддержки работы всех основных сетевых приложений.

Протокол	Логическая топология	Физическая топология
Token Ring	Кольцо	Кольцо, звезда
High Speed Token Ring	Кольцо	Кольцо, звезда
FDDI	Кольцо	Кольцо, звезда
Ethernet	Шина	Шина, звезда
Fast Ethernet	Шина	Звезда
Gigabit Ethernet	Шина	Звезда
ATM	Виртуальный канал	Кольцо, звезда

Технические помещения

Для построения СКС и информационной системы предприятия в целом необходимы технические помещения двух видов:

- аппаратные
- кроссовые

Аппаратной в дальнейшем называется техническое помещение, в котором располагается сетевое оборудование коллективного пользования (АТС, серверы, концентраторы).

Технические помещения

Кроссовая представляет собой помещение, в котором размещается коммутационное оборудование СКС, сетевое и другое вспомогательное оборудование.

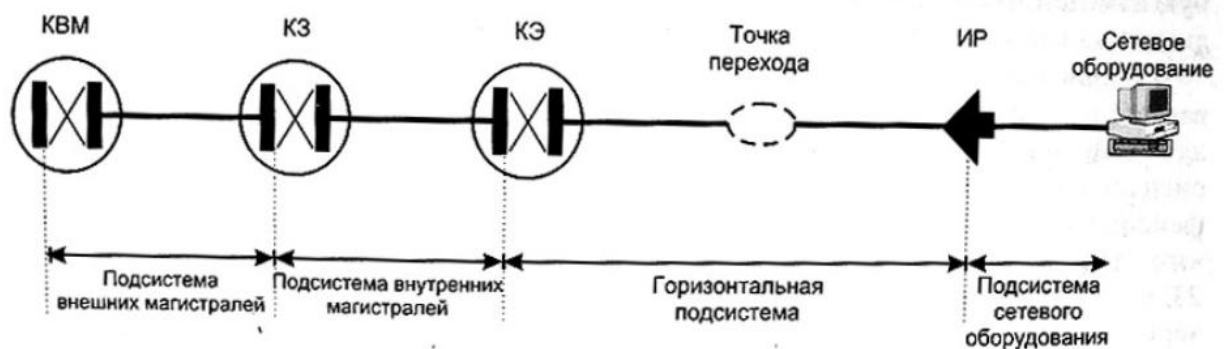
Кроссовые на практике достаточно часто называют просто (этажными) техническими помещениями, а иногда — хабовыми.

Подсистемы СКС

Аппаратная может быть совмещена с кроссовой здания (КЗ).

В кроссовую внешних магистралей (КВМ) сходятся кабели внешней магистрали, подключающие к ней КЗ.

В КЗ заводятся внутренние магистральные кабели, подключающие к ним кроссовые этажей (КЭ).



Подсистемы СКС

В самом общем случае СКС включает в себя три подсистемы:

- **подсистема внешних магистралей**, или, по терминологии некоторых СКС европейских производителей первичная подсистема, состоит из внешних магистральных кабелей между КВМ и КЗ, коммутационного оборудования в КВМ и КЗ, к которому подключаются внешние магистральные кабели, и коммутационные шнуры и/или перемычки в КВМ.

физическая кольцевая топология

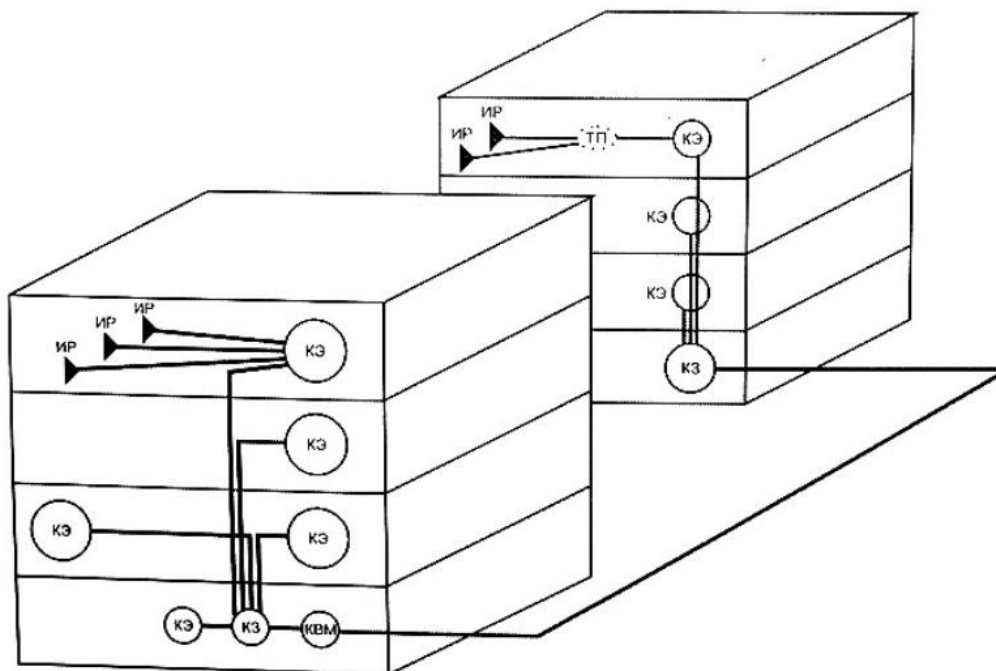
Подсистемы СКС

- **подсистема внутренних магистралей**, называемая в некоторых СКС вертикальной, или вторичной, подсистемой, содержит проложенные между КЗ и КЭ внутренние магистральные кабели, подключенное к ним коммутационное оборудование в КЗ и КЭ, а также коммутационные шнуры и/или пере-мычки в КЗ.

Подсистемы СКС

- **горизонтальная, или третичная, подсистема** образована внутренними горизонтальными кабелями между КЭ и информационными розетками рабочих мест, самими информационными розетками, коммутационным оборудованием в КЭ, к которому подключаются горизонтальные кабели, и коммутационными шнурами и/или перемычками в КЭ.

Подсистемы СКС



Подсистемы СКС

В подавляющем большинстве случаев подключение к СКС сетевого оборудования производится с помощью коммутационного шнура. В некоторых ситуациях кроме шнура может понадобиться адаптер, обеспечивающий согласование сигнальных и механических параметров оптических или электрических интерфейсов (разъемов) СКС и сетевого оборудования. Например, адаптеры применяются для подключения к СКС сетевого оборудования с интерфейсами V.24 (RS-232), устройств кабельного телевидения, систем IBM AS/400 с терминалами 5250, терминальных контроллеров IBM 3274 и терминалов 3270, а также других приложений, которые разрабатывались для других кабельных систем.

Коммутация в СКС

Принципиальной особенностью любой СКС является то, что коммутация в ней, в отличие от электронных АТС и сетевого компьютерного оборудования, всегда производится вручную коммутационными шнурами и/или переключателями.

Наиболее важным следствием такого подхода является то, что функционирование СКС принципиально не зависит от состояния электропитающей сети.

Коммутация в СКС

Известны лишь отдельные доведенные до серийного производства разработки, направленные на внедрение активных компонентов в некоторые подсистемы СКС.

- Обратной стороной отказа от применения штатного источника электропитания является:
 - необходимость использования коммутационных шнуров - ухудшают массогабаритные показатели, требуют применения специальных мер администрирования;
 - невозможность введения в состав СКС штатных коммутаторов, контроллеров, датчиков и пр. - снижает удобство эксплуатации, увеличивает время поиска неисправности, затрудняет текущую диагностику и т.д.

Принципы администрирования СКС

Принципы администрирования, или управления, СКС полностью определяются ее структурой.

- **Администрирование:**
 - Одноточечное - применяется в тех ситуациях, когда требуется максимально упростить управление кабельной системой.
 - Многоточечное - управление СКС, которая построена по классической архитектуре иерархической звезды.

Принципы администрирования СКС

Принципы администрирования, или управления, СКС полностью определяются ее структурой.

- **Администрирование:**
 - Одноточечное - применяется в тех ситуациях, когда требуется максимально упростить управление кабельной системой.
 - Многоточечное - управление СКС, которая построена по классической архитектуре иерархической звезды.

Основные термины и определения

ИУ4 – «сети»

Длины кабелей

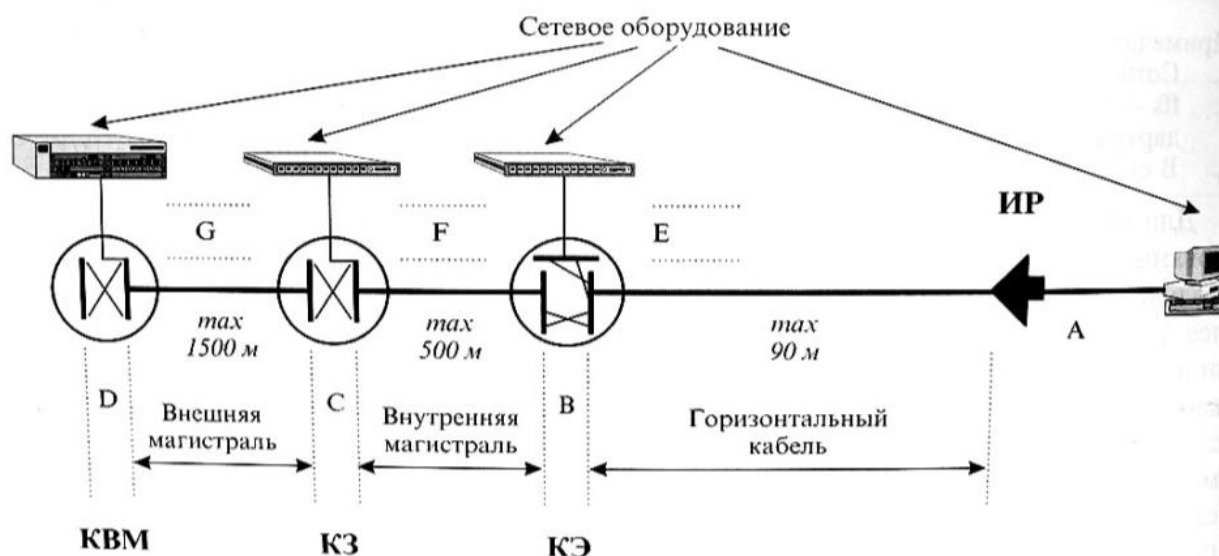
Класс линии и приложения	Определение
A	Телефонные каналы и низкочастотный обмен данными. Максимальная частота сигнала — 100 кГц
B	Приложения со средней скоростью обмена. Максимальная частота сигнала — 1 МГц
C	Приложения с высокой скоростью обмена. Максимальная частота сигнала — 16 МГц
D	Приложения с очень высокой скоростью обмена. Максимальная частота сигнала — 100 МГц
Оптический	Приложения, использующие в качестве среды передачи сигнала оптический кабель

Класс приложений	A	B	C	D	Оптики
Среда передачи сигнала					
Симметричный кабель категории 3	2 км	200 м	100 м ¹⁾		
Симметричный кабель категории 4	3 км	260 м	150 м		
Симметричный кабель категории 5	3 км	260 м	160 м	100 м	
Симметричный кабель 150 Ом	3 км	400 м	250 м	150 м	
Многомодовый оптический кабель	-	-	-	-	2 км
Одномодовый оптический кабель	-	-	-	-	3 км ²⁾

Виды сетей

ИУ4 – «сети»

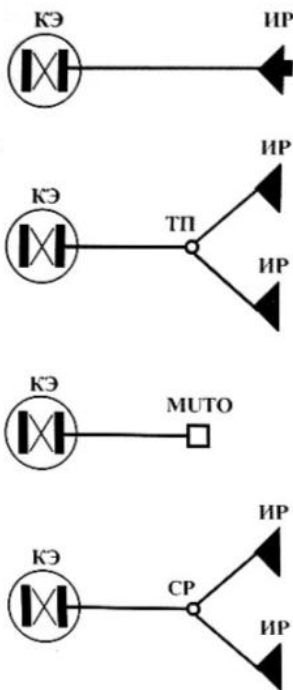
Длины кабелей



- $A+B+E \leq 10$ м — суммарная длина всех шнуров и перемычек горизонтальной подсистемы;
- C и $D \leq 20$ м — длина коммутационных шнуров (перемычек) в КЗ и КВМ;
- F и $G \leq 30$ м — длина оконечных шнуров в КЗ и КВМ.

Виды сетей

ИУ4 – «сети»



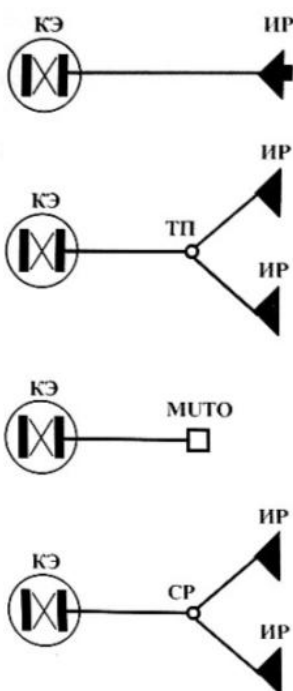
Варианты построения горизонтальной подсистемы СКС

Горизонтальная подсистема СКС при ее реализации на кабелях из витых пар может быть построена по четырем различным схемам

многопарный + четырехпарный и круглый + плоский с одинаковым количеством пар

Виды сетей

ИУ4 – «сети»



Варианты построения горизонтальной подсистемы СКС

Последние два варианта построения горизонтальной подсистемы СКС широко применяются в так называемых открытых офисах (open offices), то есть в рабочих помещениях большой площади,

- **Расчет оконечного шнура MUTO**

$$W = (102 - H) / 1.2 - 7 \text{ м}, W \leq 20 \text{ м}$$

H – длина горизонтального кабеля

1.2 – коэффициент учитывает затухание

7 – максимальная длина ком. шнуров в кроссовой

Топологии с централизованным администрированием

относятся к случаю построения разводки внутри одного здания полностью на оптическом кабеле

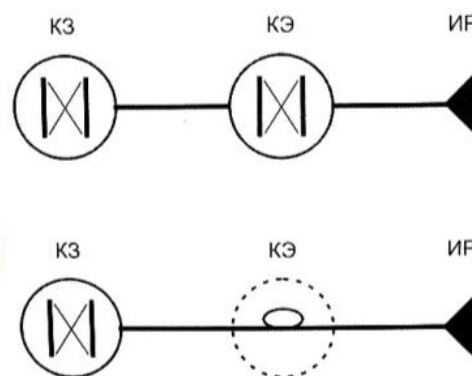
нет 90-метрового ограничения витой пары
переход за счет этого от двухуровневой звездообразной топологии к простой одноуровневой

Топологии с централизованным администрированием

Возможны два варианта: с межсоединением и без него.

Вариант с одним межсоединением позволяет сохранить прежнюю телекоммуникационную инфраструктуру здания

Максимальная длина канала с межсоединением выбрана равной 300 м из соображений получения на кабеле с волокном типа 62,5/125 пропускной способности канала связи 1 Гбит/с

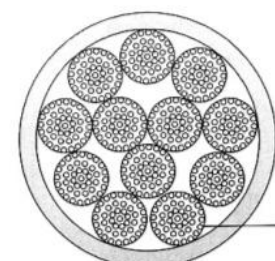
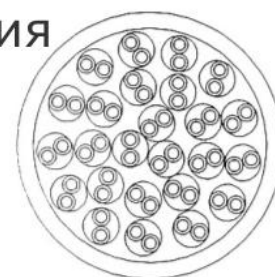


Магистральный кабель

Предназначен для использования в магистральных подсистемах СКС для связи между собой кроссовых.

Подсистема внутренних магистралей – вертикально

Внешних - горизонтально

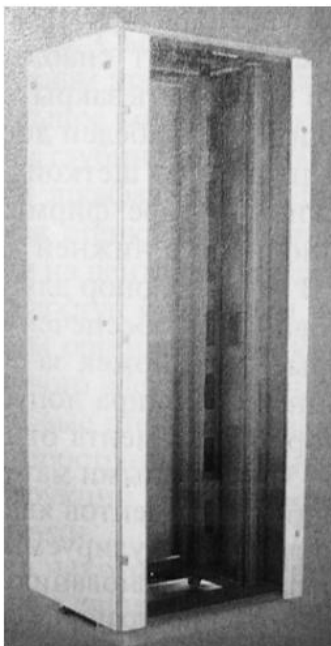


Категория кабеля	Количество пар
3	25, 50, 75, 100, 200, 300, 600, 900, 1800
5	25, 50, 100

Монтажное оборудование



Монтажное оборудование



Напольный шкаф

Высота рабочей зоны в условных единицах – юнитах (U) – 44,45 мм

Высота от 21 до 48 U

Ширина от 600 до 800 мм

Глубина от 600 до 800 мм

Состав:

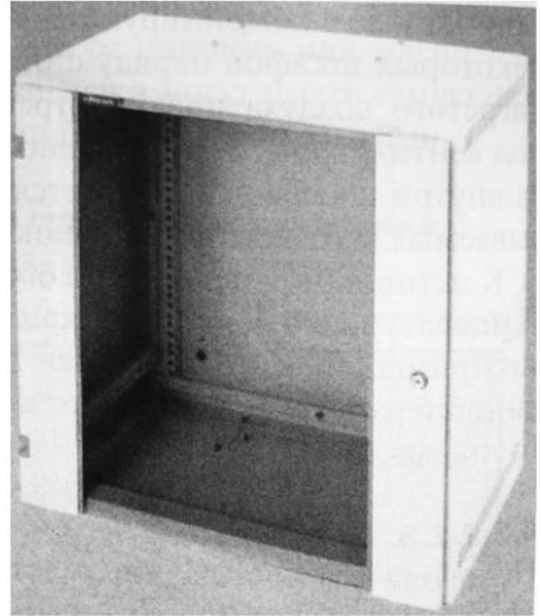
- Монтажные направляющие
- Каркас с основанием
- Верхняя крышка
- Боковые стенки
- Передняя, задняя двери

Основные термины и определения

ИУ4 – «сети»

Монтажное оборудование

Высота от 3 до 15 U
Ширина от 600 мм
Глубина от 240 до 450 мм

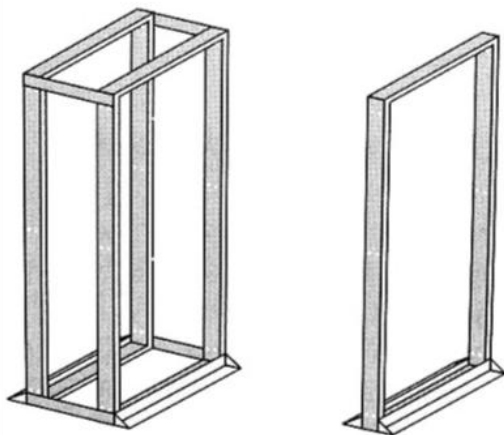


Настенный шкаф

Основные термины и определения

ИУ4 – «сети»

Монтажное оборудование



Открытая стойка

- Не ставятся требования по ограничению доступа к оборудованию
- Требуется очень частый доступ к оборудованию
- Необходимо более эффективное охлаждение

Аналог напольного шкафа

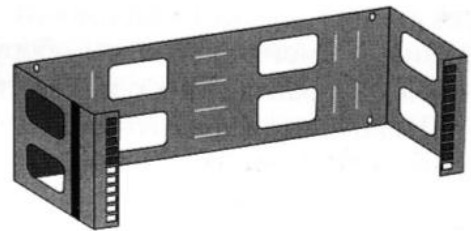
Монтажное оборудование

- Не ставятся требования по ограничению доступа к оборудованию
- Требуется очень частый доступ к оборудованию
- Необходимо более эффективное охлаждение

Высота обычно не превышает 10 U

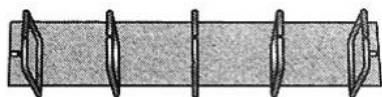
Глубина 250мм

Аналог настенного шкафа



Монтажная рама

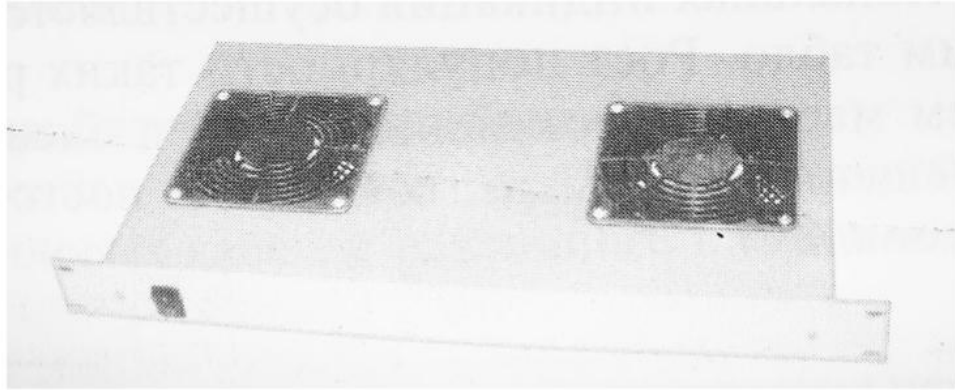
Монтажное оборудование



Организаторы кроссовых шнуров

Монтажное оборудование

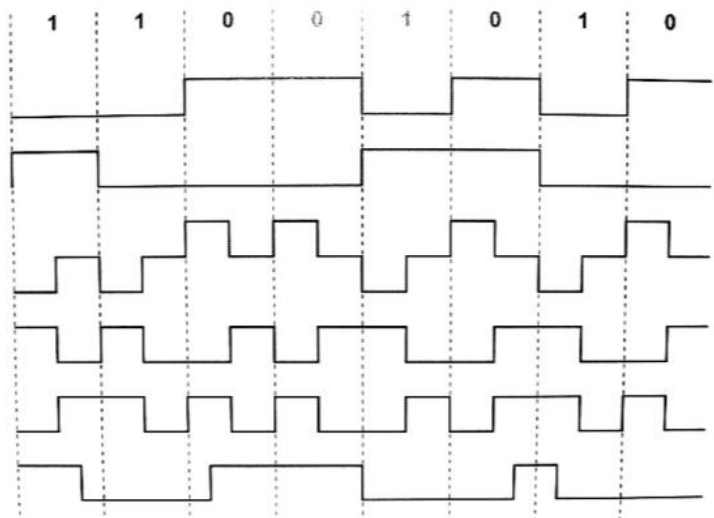
Высота 1 или 2 U



Вентиляторная полка

Схемы синхронизации

NRZ-кодирование
NRZI – кодирование
RZ – кодирование
Манчестерский
Дифференциальный манчестерский
Блочный код 4B5B+NRZI
PNRZ
8B/6T
MLT-3
Enhanced TX/T2
PAM-5
8B10B





skycluster@gmail.com



Сети своими руками

Основы телекоммуникационных технологий
Курс лекций

Основные термины и определения

ИУ4 – «сети»

1	2	3
NRZ-кодирование Манчестерский код Блочный код 4B5B+NRZI PAM-5	NRZI – кодирование Дифференциальный манчестерский код 8B10B Enhanced TX/T2	PNRZ - кодирование RZ – кодирование 8B/6T MLT-3

Указать:

- Расшифровку названия
- Суть кодирования – как происходит процесс кодирования
- Временную диаграмму для последовательности битов 11001010
- Где применяется? Стандарт Ethernet, 10,100, 1000 ...

План лекции

ИУ4 – «сети»

- 1 Основные принципы и определения
- 2 Синхронизация
- 3 Коды передачи
- 4 Дополнительная информация

Основные принципы и определения

ИУ4 – «сети»

Типы сигналов:

Линейный сигнал

Информационный сигнал

Полоса пропускания канала - диапазон частот, в пределах которого амплитудно-частотная характеристика (АЧХ) акустического, радиотехнического или оптического устройства достаточно равномерна для того, чтобы обеспечить передачу сигнала без существенного искажения его формы.

Модуляция - процесс изменения одного или нескольких параметров высокочастотного модулируемого колебания под воздействием относительно низкочастотного управляющего модулирующего сигнала. Передаваемая информация заложена в управляющем сигнале.

Синхронизация

ИУ4 – «сети»

Способы обеспечения тактовой синхронизации в цифровой аппаратуре

При передаче сигнала возникает необходимость тактовой синхронизации передатчика и приемника.

Почему необходимо:

приемник должен знать в какой момент времени стробировать входной сигнал и декодировать информацию

Тактовая синхронизация:

- синхронная;
- асинхронная.

Синхронизация

ИУ4 – «сети»

Методы обеспечения тактовой синхронизации

Метод	Старт-стопный	По выделенному каналу	Предопределенные синхро-последовательности	Самосинхронизации
Вид	Асинхронный	Синхронный	Синхронный	Синхронный
Эффективность использования полосы пропускания канала связи	Низкая	Низкая (требует выделенного канала)	Высокая	Средняя
Сложность схемотехнических решений	Низкая	Низкая	Высокая	Средняя – низкая*
Скорость вхождения в синхронизм	Высокая	Высокая	Средняя	Высокая – средняя*
Помехоустойчивость	Высокая	Высокая	Средняя	Высокая

Синхронизация

ИУ4 – «сети»

В асинхронных системах тактовые генераторы приемника и передатчика работают примерно на одинаковых частотах, однако функционируют полностью независимо друг от друга, а синхронизация обеспечивается за счет введения дополнительных импульсов.

Алгоритм: старт(передатчик) – небольшая посылка (передатчик) – стоп (передатчик) – прием старта (приемник) – декодирование посылки (приемник) – завершение по стопу

Синхронизация

ИУ4 – «сети»

Очередная последовательность данных может быть передана после паузы большей или меньшей длительности (защитного интервала) и также должна сопровождаться стартовым и стоповым сигналами.

Подобный принцип используется в работе интерфейса RS-232 в асинхронном режиме. Ранее такой метод передачи широко использовался также в низкоскоростной телеграфии, где был известен под названием старт-стопного метода.

Синхронизация

ИУ4 – «сети»

В группу синхронных методов тактовой синхронизации входят:

- - синхронизация по выделенному каналу;
- - predetermined синхропоследовательности;
- - самосинхронизация.

Синхронизация

ИУ4 – «сети»

Синхронизация по выделенному каналу

заключается в использовании передатчиком и приемником одного общего тактового генератора с передачей его синхро-сигналов по выделенному параллельному каналу.

Способ применяется в каналах связи небольшой протяженности, например в различных вариантах синхронного интерфейса RS-232.

Однако для большинства высокоскоростных систем связи способ оказывается экономически нецелесообразным. Кроме того, на длинных линиях могут возникнуть дополнительные технические трудности, связанные с ненулевой разностью задержек прохождения кабельного тракта информационным и синхронизирующим сигналами (skew).

Синхронизация

ИУ4 – «сети»

Предопределенные синхропоследовательности

(синхрокомбинации) периодически вставляются в сигнал на передающем конце между посылками данных. Устройство выделения тактовой частоты приемника принимает синхропоследовательность и обрабатывает для постоянной подстройки своего внутреннего тактового генератора.

Метод чрезвычайно эффективен в смысле использования полосы пропускания канала связи, но требует достаточно сложных схем выделения синхрокомбинаций из непрерывного потока данных с их последующей обработкой и не обеспечивает высокой скорости вхождения в синхронизм, очень важной для большинства типов сетевой аппаратуры ЛВС.

широко применяется в цифровых системах телефонной связи и другого оборудования, которое устанавливается в сетях связи общего пользования.

Синхронизация

ИУ4 – «сети»

Метод самосинхронизации также основан на применении двух разных тактовых генераторов в передатчике и в приемнике. Идея реализации в том, чтобы придать такой вид самому линейному сигналу, который позволяет приемнику постоянно подстраивать частоту своего тактового генератора под частоту сигнала, компенсируя тем самым взаимный дрейф частот синхросигналов.

Алгоритм формирования линейного кода выбирается таким, чтобы сигнал в линии имел частые достаточно регулярные смены состояния между логическими 0 и 1. Переходы являются тем управляющим воздействием, которое осуществляет подстройку частоты тактового генератора приемника.

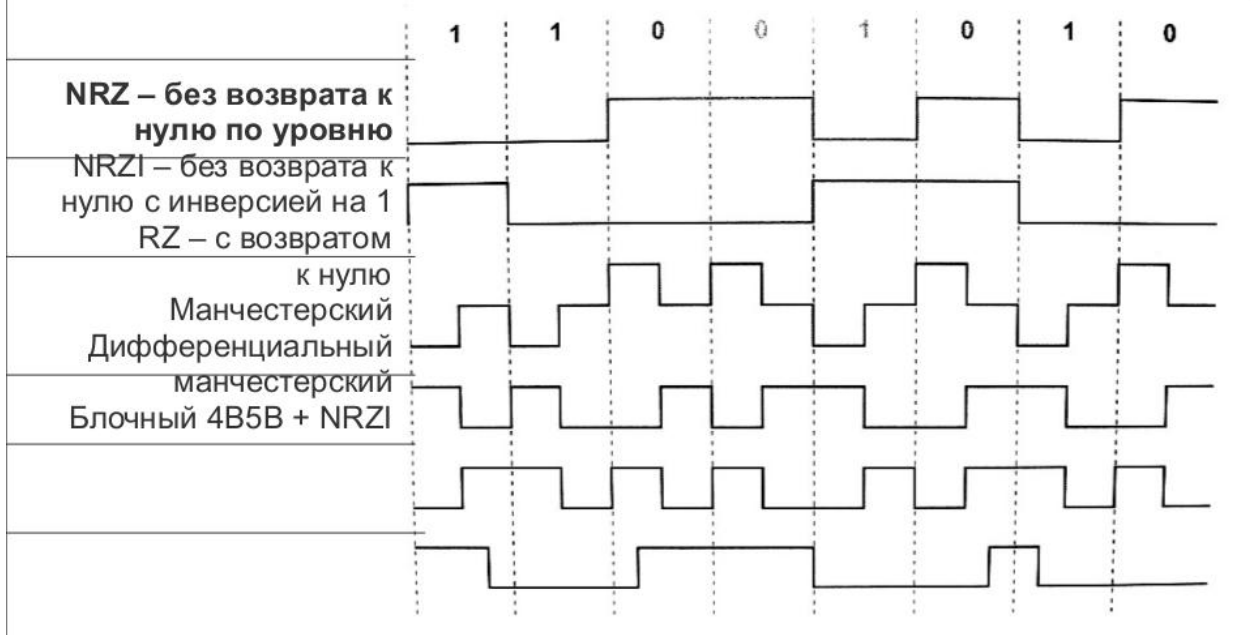
В отличие от старт-стопного механизма; такой способ обеспечения синхронизации не требует наличия защитного интервала между отдельными блоками данных, что позволяет более эффективно использовать пропускную способность канала связи.

От метода predetermined синхроследовательностей рассматриваемый метод выгодно отличается высокой скоростью вхождения в синхронизм приемника и передатчика.

Синхронизация

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды без возврата к нулю

Код NRZ - последовательная без каких-либо предварительных преобразований передача отдельных битов исходного сообщения в канал связи,

Является примером простейшей реализации модуляции уровнем сигнала.

Основное преимущество — простота реализации и минимальная потребность в ширине полосы пропускания канала.

Главный недостаток - сложность синхронизации. Это обусловлено тем, что на длинных последовательностях следующих подряд друг за другом нулей и единиц подстройка тактовых генераторов приемника невозможна.

Коды без возврата к нулю

Код NRZ обычно используется в цифровых микросхемах сетевого оборудования. Синхронизация устройств обеспечивается подключением к общей шине синхронизации сетевого интерфейса.

В каналах связи код NRZ обычно применяется там, где организация выделенных линий синхронизации не вызывает проблем, например для реализации интерфейсов V.24 (RS-232).

Код NRZ имеет постоянную составляющую, то есть для его передачи линия связи должна пропускать сигналы с нулевой частотой.

Коды без возврата к нулю

Одним из условий реализации балансной передачи по витым парам является применение в приемопередатчиках сетевого оборудования развязывающих согласующих трансформаторов, передача постоянной составляющей сигнала через которые невозможна.

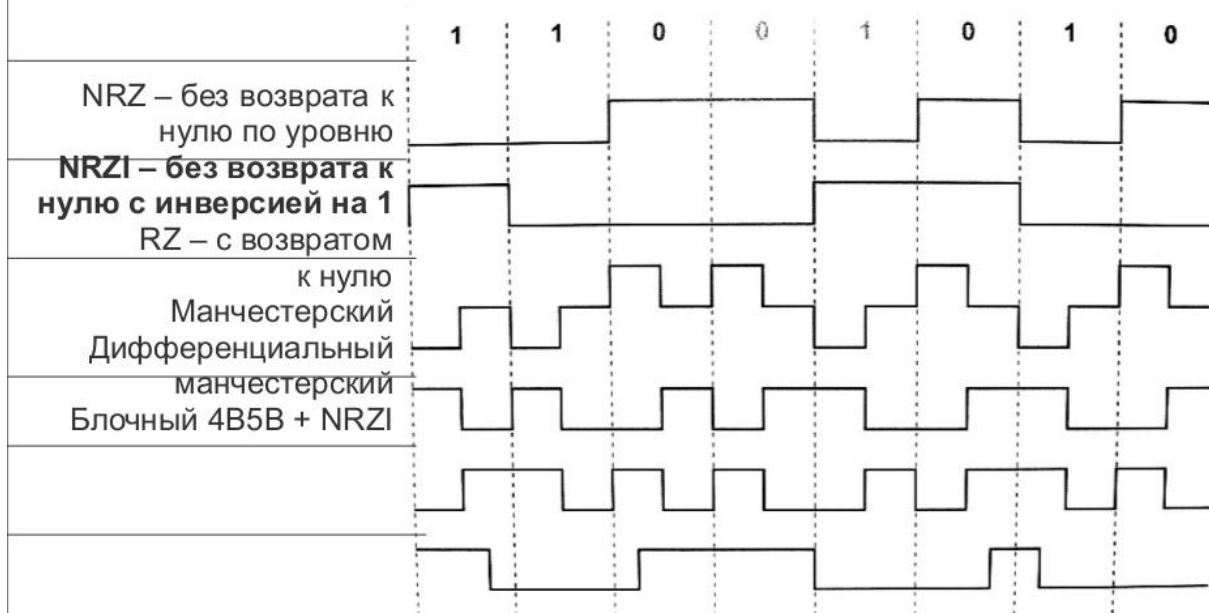
Коды без возврата к нулю

Для устранения этого недостатка в симметричных цепях используется модификация кода NRZ, получившая название полярного кода без возврата к нулю (**PNRZ - Polar Non Return to Zero**).

Логическая 1 кодируется отрицательным напряжением, а логический 0 положительным.

Однако и в этом случае из-за невозможности передачи постоянной составляющей сигнал в коде PNRZ на длинных последовательностях нулей и единиц стремится к нулю, то есть испытывает сильные искажения и правильно передаваться не будет.

Коды низкоскоростных электрических систем



Коды без возврата к нулю

Код без возврата к нулю с инверсией на единицах — NRZ-I (Non Return to Zero, Invert on ones). Код NRZ-I является простейшей реализацией принципа кодирования сменой уровня сигнала или дифференциального кодирования.

Сигнал на протяжении всего тактового интервала не меняется, при этом логические 0 и 1 кодируются отсутствием и наличием смены уровня сигнала в начале интервала соответственно.

Применение дифференциальных кодов дает преимущества при организации канала связи в сложной помеховой обстановке.

В этой ситуации определение смены уровня сигнала оказывается более надежным, чем сравнение его абсолютного уровня с пороговым значением, выполняемое компаратором.

Коды без возврата к нулю

Использование кода NRZ-I наиболее эффективно в тех случаях, когда вероятность P появления одного логического символа в последовательности существенно превышает вероятность появления другого, например $P(1) \gg P(0)$. Потеря синхронизации может произойти только на длинных последовательностях нулей.

Для устранения этого недостатка используется так называемое скремблирование, основанное на прерывании таких последовательностей избыточной логической единицей. На практике применяются как программные, так и аппаратные реализации скремблера.

Коды без возврата к нулю

Программная реализация скремблера обычно возлагается на протокол более высокого уровня. Примером может служить протокол кодирования SDLC/HDLC кодом NRZ-I. Аппаратный скремблер, используемый в сетях CDDI (TP-PMD) и 100Base-TX.

Коды передачи

ИУ4 – «сети»

Коды с возвратом к нулю

Для устранения недостатков кодов NRZ и PNRZ используются принципы кодирования, которые основаны на добавлении новых импульсов к исходной последовательности, а также возможной инверсии элементов этой последовательности по специальному алгоритму.

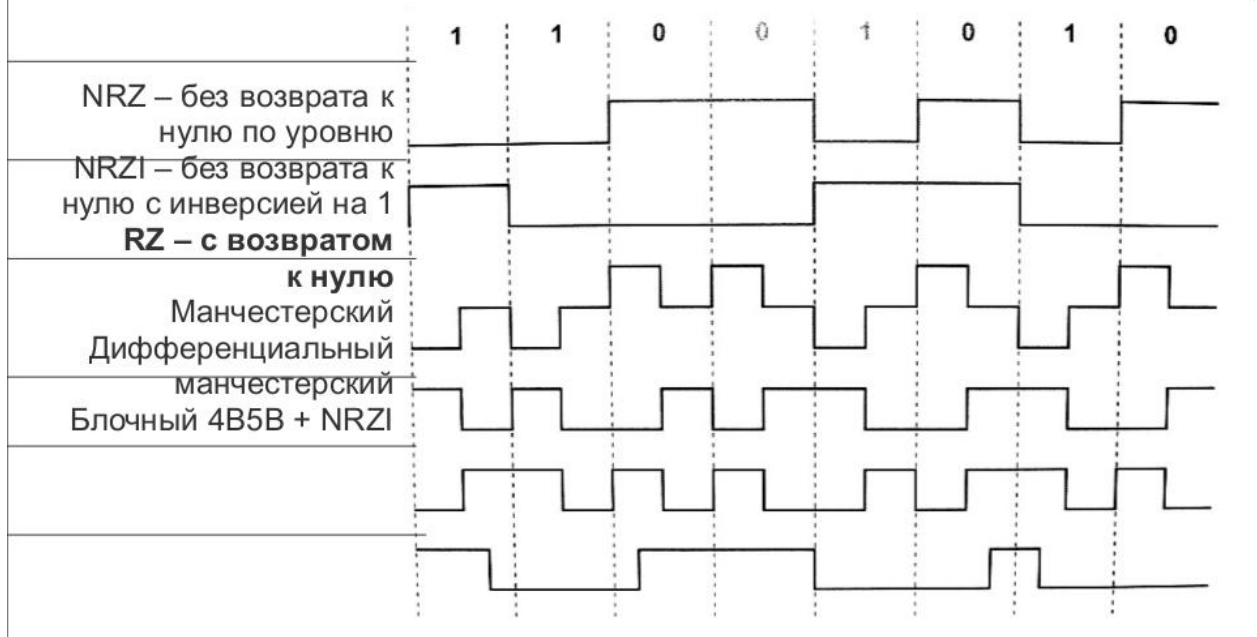
Оптимальное согласование свойств сигнала с параметрами канала связи по тому или иному критерию сопровождается: увеличением помехоустойчивости, обеспечением надежной работы цепей тактовой синхронизации, позволяет более полно использовать потенциальную пропускную способность канала связи и увеличить за счет этого быстродействие сетевой аппаратуры.

Основой положительного эффекта является введение в линейный сигнал определенной избыточности. Внешне она выражается в том, что тактовая частота фл линейного сигнала возрастает по сравнению с информационной, то есть $f_l > f_i$.

Коды передачи

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды передачи

ИУ4 – «сети»

Коды с возвратом к нулю

Код с возвратом к нулю — RZ (Return to Zero). Код RZ использует импульсы разной полярности для представления логических 0 и 1 с возвратом к среднему уровню в середине тактового интервала.

Относится к классу самосинхронизирующихся, так как обеспечивает гарантированную смену уровня сигнала на каждом тактовом интервале. Другим его преимуществом является отсутствие постоянной составляющей сигнала, что делает возможным его применение в системах связи с балансной передачей.

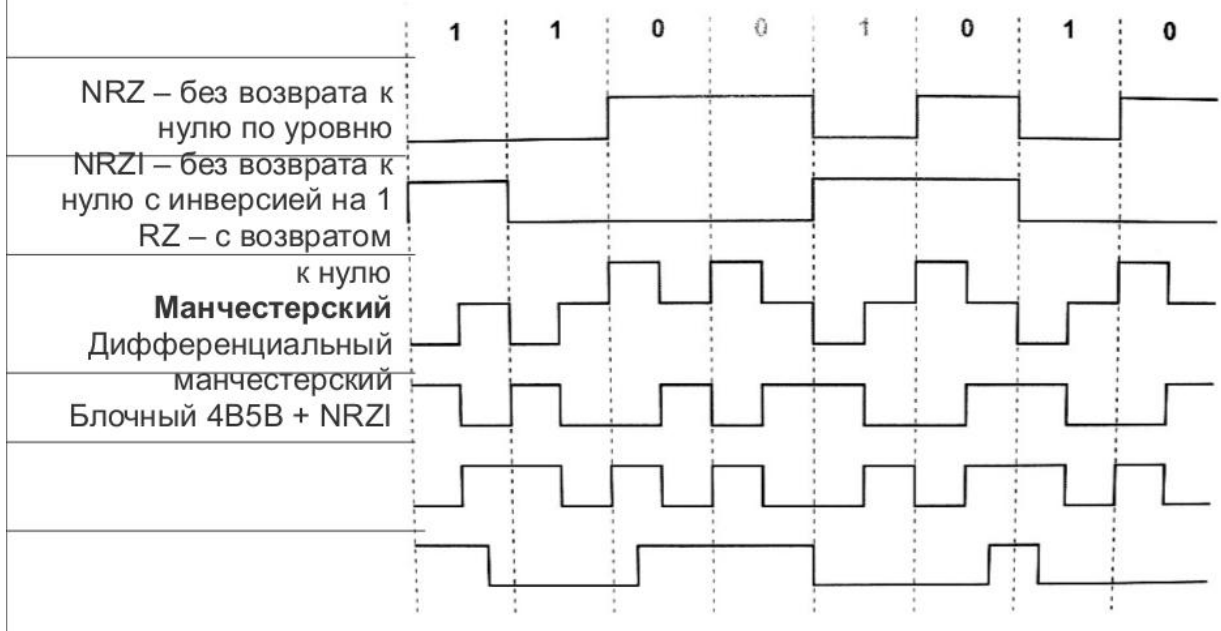
Однако при передаче каждого бита сигнал дважды меняет свой уровень, что приводит к удвоению тактовой частоты и, соответственно, удвоению необходимой ширины полосы частот канала по сравнению с NRZ-кодом.

Требует формирования и обработки трехуровневых сигналов, что усложняет схемотехнические решения приемопередатчиков электрических систем и практически не позволяет применять его в технике оптической связи.

Коды передачи

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды передачи

ИУ4 – «сети»

Коды с возвратом к нулю

Манчестерский (Manchester) код. Самосинхронизирующийся код без постоянной составляющей формирует сигнал только с двумя уровнями и определяет значение бита направлением смены уровня сигнала на середине тактового интервала.

Логический 0 кодируется сменой уровня сигнала от низкого к высокому, а логическая 1 — от высокого к низкому (то есть на тактовом интервале T всегда присутствует импульс длительностью $T/2$). При переходе от нуля к единице или обратно изменения уровня не происходит.

Отсутствие изменения уровня сигнала в середине тактового интервала используется в качестве признака срабатывания решающего устройства декодера в некоторых типах приемников. Код рассматриваемого вида за счет наличия гарантированного перепада в середине тактового интервала существенно упрощает схемы синхронизации приемника, и его сигнал не имеет постоянной составляющей.

Применение приводит к удвоению тактовой частоты относительно NRZ-кодов.

Коды передачи

ИУ4 – «сети»

Коды с возвратом к нулю

Простота схемотехнической реализации приемников и передатчиков с манчестерским кодированием привела к очень широкому их распространению на практике.

Манчестерское кодирование используется во всех разновидностях интерфейсов сети Ethernet со скоростью передачи данных 10 Мбит/с (10Base-5, 10Base-2, 10Base-T и FOIRL). Для них тактовая частота линейного сигнала составляет 20 МГц.

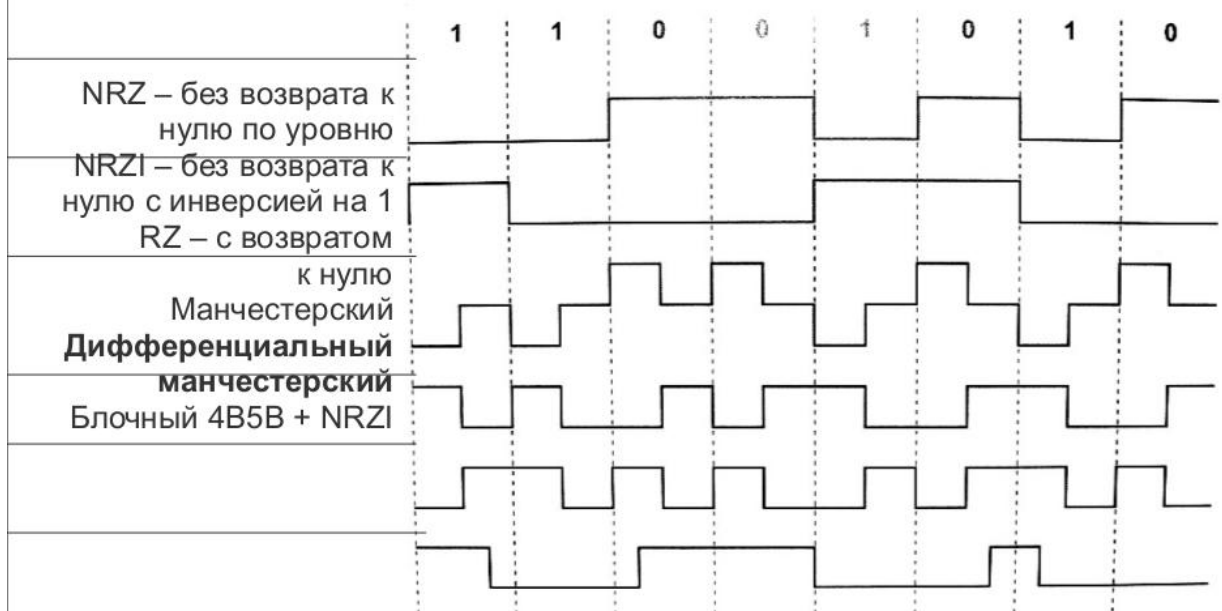
Для восстановления сигнала на приемной стороне верхняя граничная частота канала связи должна составлять не менее половины тактовой частоты цифрового линейного сигнала.

Каналы связи категории 3, на которых работает сеть Ethernet 10Base-T, имеют верхнюю граничную частоту 16 МГц, что в 1,6 раза превышает теоретический минимум. Все частотные составляющие свыше 16 МГц рассматриваются как помехи и подавляются низкочастотным фильтром на входе приемника.

Коды передачи

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды передачи

ИУ4 – «сети»

Коды с возвратом к нулю

Дифференциальный манчестерский (Differential Manchester) код, является разновидностью манчестерского кода. Середину тактового интервала линейного сигнала он использует только для синхронизации, и на ней всегда происходит смена уровня сигнала.

Логические 0 и 1 передаются наличием или отсутствием смены уровня сигнала в начале тактового интервала, соответственно.

Имеет те же самые преимущества и недостатки, что и манчестерский.

Примером системы передачи данных с дифференциальным манчестерским кодированием является сеть Token Ring 4 и 16 Мбит/с. 16-мегабитный вариант этой сети работает по каналам связи на основе витых пар категории 4 с верхней граничной частотой 20 МГц, что всего в 1,25 раза превышает теоретический минимум. Столь малый запас требует достаточно сложных схем обработки сигнала, что не в последнюю очередь определяет высокую стоимость сетевого оборудования этого стандарта.

Коды передачи

ИУ4 – «сети»

Особенности линейных кодов для оптических каналов связи

- линейный сигнал может принимать только нулевое или положительное значение («отрицательный» свет не существует), то есть в линии всегда будет присутствовать постоянная составляющая;
- линии оптической связи используются для передачи высокоскоростных сигналов на большие расстояния, то есть требования минимального расширения полосы частот исходного сообщения имеют более важное значение по сравнению с электрическими системами;
- особенности элементной базы, используемой для построения оптических каналов связи, в частности заметная временная и температурная нестабильность мощности выходного сигнала оптических излучателей (особенно полупроводникового лазера), не позволяют широко применять многоуровневые схемы кодирования;

Коды передачи

ИУ4 – «сети»

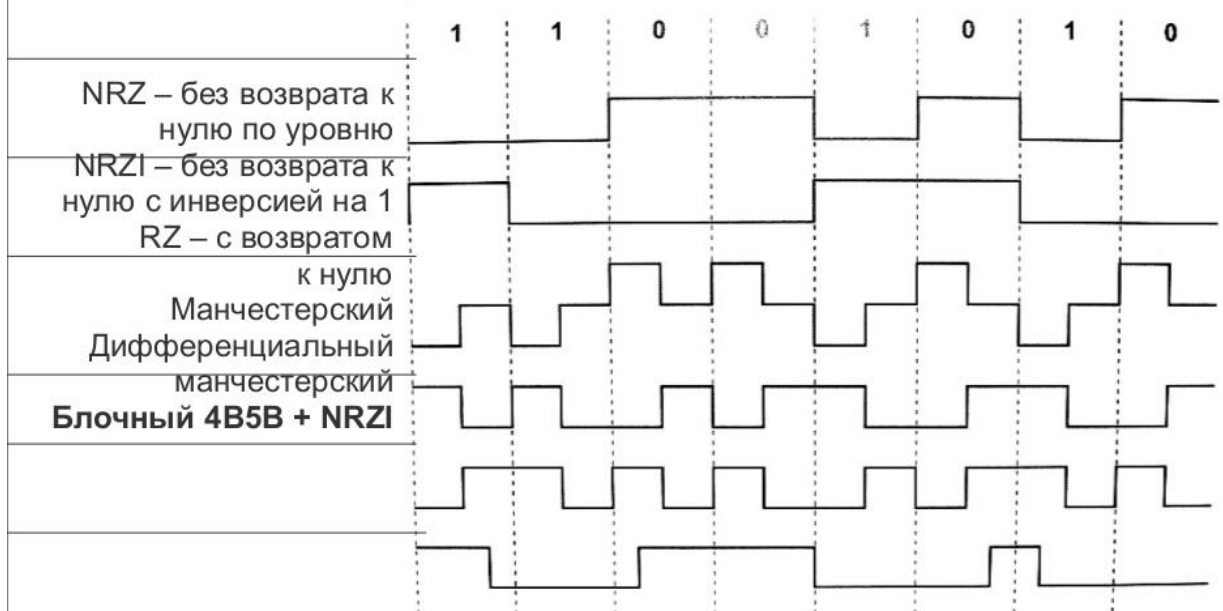
Особенности линейных кодов для оптических каналов связи

- современные полупроводниковые излучатели не могут генерировать чисто монохроматическое излучение - в подавляющем большинстве линий оптической связи используется модуляция интенсивности (мощности) излучения и применение дискретной фазовой и частотной модуляции излучения является невозможным;
- значительно более высокая стоимость световода по сравнению с витой парой (примерно 30 центов волокна 62,5/125 против 6 центов витой пары категории 5) делает экономически нецелесообразным использование широко применяемого в электрических системах принципа распараллеливания информационных потоков и их передачу по отдельным подканалам с меньшей скоростью;
- оптическая сетевая аппаратура из-за наличия так называемого квантового шума обладает существенно меньшим энергетическим потенциалом.

Коды передачи

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды передачи

ИУ4 – «сети»

Блочные коды

Код 4B5B является примером блочного самосинхронизирующего кодирования, который используется в системах FDDI и Fast Ethernet 100Base-FX. Согласно алгоритму его реализации, каждые четыре входных информационных бита кодируются пятью линейными.

Правила кодирования задаются с помощью так называемой кодовой таблицы. Вид кодовой таблицы может быть самым различным в зависимости от поставленной задачи и требуемых свойств кода.

В системе FDDI для обеспечения устойчивости тактовой синхронизации и минимизации флуктуации средней оптической мощности линейный сигнал при передаче данных всегда имеет не менее двух изменений в каждом блоке. Схема блочного кодирования системы FDDI приведена в табл.

Коды передачи

ИУ4 – «сети»

Символ блока данных	Двоичный код	Представление сигнального блока	
		десятичное	двоичное
0	0000	30	11110
1	0001	09	01001
2	0010	20	10100
3	0011	21	10101
4	0100	10	01010
5	0101	11	01011
6	0110	14	01110
7	0111	15	01111
8	1000	18	10010
9	1001	19	10011
A	1010	22	10110
B	1011	23	10111
C	1100	26	11010
D	1101	27	11011
E	1110	28	11100
F	1111	29	11101

Линейный сигнал при блочном кодировании обладает избыточностью по сравнению с информационным. Избыточность используется для увеличения помехоустойчивости и обнаружения ошибок, так как часть кодовых комбинаций при этом оказывается запрещенной. При их обнаружении выдается команда VIOLATION. В сравнении с RZ и манчестерским кодами кодирование **4B5B** обеспечивает тактовую частоту не в два, а только в 1,25 раза превышающую тактовую частоту информационного сигнала. Это позволяет намного более эффективно использовать полосу пропускания линии связи.

Коды передачи

ИУ4 – «сети»

Блочные коды

Код 8B10B изначально использовался в аппаратуре Fibre Channel и отсюда был заимствован для применения в сетях Gigabit Ethernet 1000Base-SX (многомодовая оптика) и 1000Base-LX (многомодовая или одномодовая оптика).

В коде 8B10B для представления 8 битов данных используется 10 сигнальных битов. Незакодированная информация состоит из восьми информационных битов A, B, C, D, E, F, G, H и контрольного бита Z. Эти биты кодируются с помощью таблицы в биты a, b, c, d, e, i, f, g, h, j десятибитового символа.

Контрольный бит принимает значение D* для символов, представляющих исходные данные, и K — для специальных символов. Каждой входной последовательности из 8 информационных и одного контрольного бита ставится в соответствие название, составленное по формуле Zxx.y, где Z - контрольный бит, xx - десятичное число, составленное из трех первых битов A, B, и C.

Например, специальный (типа K) шестнадцатеричный символ «BC» называется K28.5. Приемник декодирует полученную информацию блоками по 10 бит, после чего символы типа D* преобразуются в одну из 256 восьмибитовых комбинаций, а символы типа K используются для управления протоколом. Символы, не относящиеся к типу D* или K, рассматриваются как ошибки протокола.

Коды передачи

ИУ4 – «сети»

Схема кодирования сети 100Base-T4

100Base-T4 разрабатывался с учетом возможностей использования существующей кабельной проводки категории 3 для передачи 100-мегабитных сигналов в полудуплексном режиме. Использование трех витых пар одновременно позволяет передавать по каждой из них информационный поток со скоростью $100/3 = 33,33$ Мбит/с. Для уменьшения тактовой частоты применяется формирование линейного сигнала с трехуровневым **кодом 8В/6Т**.

Алгоритм реализации: в кодере передатчика любой поступающий на него 8-битовый символ в соответствии со специальной кодовой таблицей преобразуется в шесть трехуровневых символов, которые попарно передаются по трем витым парам. В результате этого тактовая частота линейного сигнала снижается до значения $(100/3) \times 6/8 = 25$ МГц. Минимальная ширина канала связи для передачи цифрового сигнала с такой частотой составляет 12,5 МГц, что в 1,28 раза меньше ширины полосы пропускания стандартного кабельного тракта категории 3 (16 МГц).

Коды передачи

ИУ4 – «сети»

Схема кодирования сети 100Base-T4

В оборудовании рассматриваемого стандарта для передачи информации в каждый конкретный момент задействованы одновременно три пары. Пары 1 и 2 применяются для однонаправленной передачи сигналов, что обеспечивает возможность функционирования механизма обнаружения коллизий, тогда как пары 3 и 4 могут работать в режиме двенаправленной передачи.

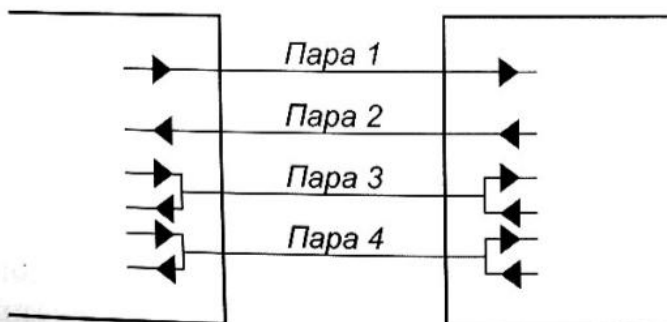
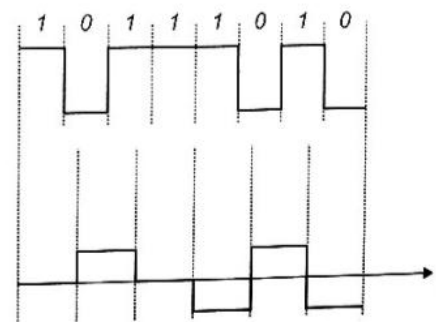


Схема взаимодействия



Кодирование 8В/6Т

Коды передачи

ИУ4 – «сети»

Схема кодирования TP-PMD 100Base-TX

Схемы кодирования линейных сигналов сетевых интерфейсов TP-PMD и 100Base-TX полностью идентичны друг другу и используют трехуровневый сигнал типа MLT-3. При этом передача выполняется только по двум витым парам категории 5 и может вестись в полнодуплексном режиме.

Особенность кодеров - разработанный сначала сетевой интерфейс TP-PMD должен был обеспечить совместимость с уровнем РНУ спецификации FDDI, на котором используется блочное кодирование типа **4B5B**. Таким образом, на вход линейного кодера MLT-3 в этих интерфейсах всегда поступает сигнал с тактовой частотой 125 МГц.

Коды передачи

ИУ4 – «сети»

Схема кодирования TP-PMD 100Base-TX

Код MLT-3 (Multi Level Transmission) реализуется аналогично коду NRZ-I и обозначается иногда MLT-3 + NRZ-I. Изменение уровня линейного сигнала происходит только в том случае, если на вход кодера поступает единица, однако, в отличие от кода NRZ-I, алгоритм формирования выбран таким образом, чтобы два соседних изменения всегда имели противоположные направления.

Формально код MLT-3 является двоичным, однако алгоритм построения позволяет получить сигнал с тремя состояниями и постоянной составляющей, мало отличающейся от нуля даже в самых неблагоприятных ситуациях.

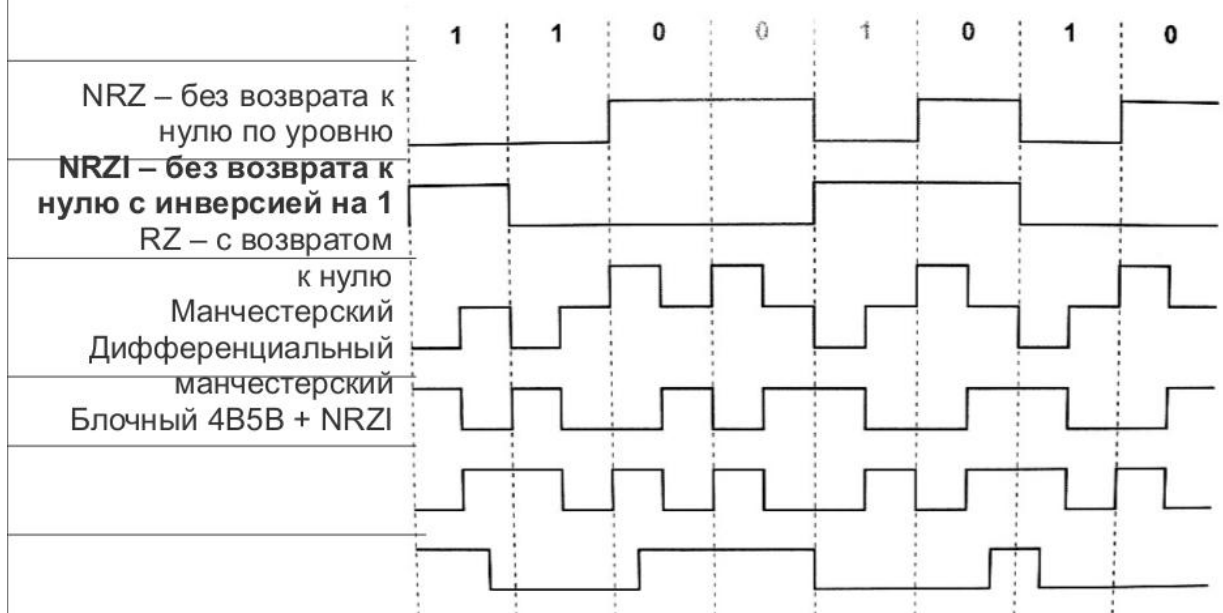
Чтобы подчеркнуть эту особенность, коды вида MLT-3 иногда выделяют в отдельный подкласс под названием квазитроичных.

Код NRZ-I не позволяет обеспечить устойчивую синхронизацию без применения специальных мер. В данном конкретном случае эти меры сводятся к принудительному введению во входной сигнал кодера сигналов логической единицы при обнаружении длинных последовательностей нулей.

Коды передачи

ИУ4 – «сети»

Коды низкоскоростных электрических систем



Коды передачи

ИУ4 – «сети»

Схема кодирования TP-PMD 100Base-TX

В интерфейсах TP-PMD и 100Base-TX данная функция реализуется с использованием аппаратного скремблера. Технически это устройство представляет собой сдвиговый регистр и генерирует квазислучайную последовательность в соответствии с порождающим полиномом вида $x^{11} + x^9$.

Входным сигналом регистра является сама информационная последовательность. Скремблер устанавливается на входе кодера MLT-3.

В приемнике на выходе декодера включается дескремблер, который восстанавливает исходный вид информационного сигнала. Синхронизация их работы обеспечивается с помощью сигналов состояния линии. Скремблер устраняет выбросы на энергетическом спектре линейного сигнала что дает возможность получения приемлемого уровня переходного затухания без использования экранированной витой пары. Еще одно следствие использования этого устройства — минимизация отклонения от нуля постоянной составляющей линейного сигнала.

Коды передачи

ИУ4 – «сети»

Схема кодирования TP-PMD 100Base-TX

Применение трехуровневого кодирования MLT-3 позволяет вдвое уменьшить тактовую частоту линейного сигнала, которая составляет $125/2 = 62,5$ МГц. Значительный запас при построении с помощью элементов категории 5.

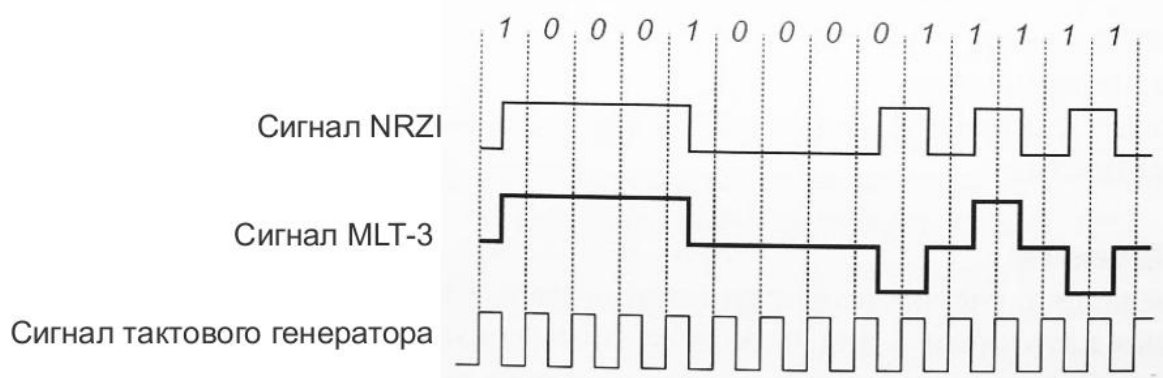
Сравнение сетевых интерфейсов 100Base-T4 и 100Base-TX показывает, что при одинаковых скоростях передачи (100 Мбит/с) переход 100Base-TX на работу по более качественному тракту категории 5 обеспечивает:

- возможность реализации полнодуплексного режима работы сетевого интерфейса 100Base-TX, что фактически эквивалентно увеличению пропускной способности канала связи вдвое;
- существенное упрощение схем электронной обработки сигналов за счет отказа от установки двунаправленных усилителей, цепей мультиплексирования и демультимплексирования и т.д.; по некоторым оценкам, сложность реализации за счет этого уменьшается на 80-90%;
- отсутствие потерь помехоустойчивости, вызываемых отличным от нуля значением параметра skew.

Коды передачи

ИУ4 – «сети»

Схема кодирования TP-PMD 100Base-TX



Коды передачи

ИУ4 – «сети»

Технические решения 1000BASE-T

- Проект стандарта сети со скоростью передачи данных в 1 Гбит/с по четырем неэкранированным витым парам - Gigabit Ethernet 1000Base-T — разрабатывался подкомитетом IEEE 802.3ab. Сеть должна работать в полнодуплексном режиме по каналу длиной 100м с вероятностью ошибки не более 10^{-10} .
- В настоящее время имеется ряд предложений по удовлетворению этих требований, наибольшее распространение из которых получили технические решения, основанные на схеме кодирования PAM-5.

Коды передачи

ИУ4 – «сети»

Технические решения 1000BASE-T

Для обеспечения возможности передачи информационного потока со скоростью 1000 Мбит/с по электрическим трактам СКС категорий 5 и 6 при разработке сетевых интерфейсов 1000Base-T использован следующий комплекс мероприятий:

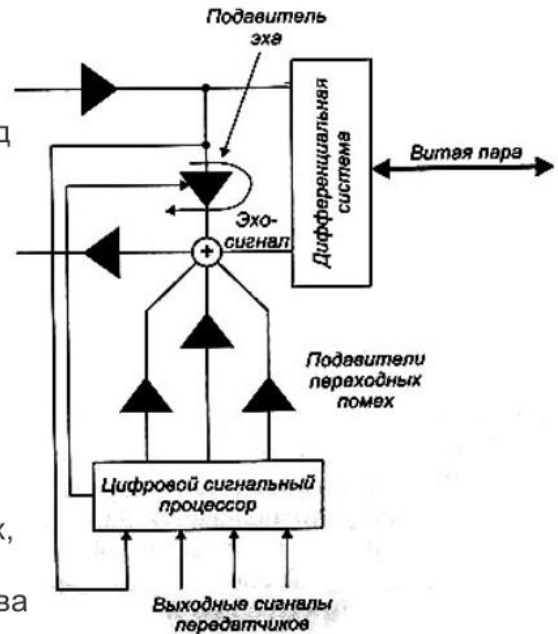
- для передачи задействованы все четыре пары одновременно, причем передачи по каждой паре ведется сразу в двух направлениях;
- в состав приемопередатчиков введены дополнительные узлы минимизации определенных видов помеховых составляющих;
- использован специальный алгоритм синхронизации сетевых интерфейсов;
- применено пятиуровневое кодирование PAM-5.

Коды передачи

ИУ4 – «сети»

Технические решения 1000BASE-T

Наличие высоких скоростей передачи информации диктует очень жесткие требования к синхронизации интерфейсов 1000Base-T. Для ее обеспечения они всегда функционируют в режиме Master-Slave. Перед началом работы с использованием расширенного механизма Autonegotiation определяется соотношение приоритетов связываемых устройств. По результатам сравнения интерфейс с более высоким приоритетом (как правило, в его роли выступает коммутирующий концентратор) берет на себя функции мастер-устройства системы синхронизации, подстраивая под частоту своего тактового генератора работу передатчика на дальнем конце. В тех случаях, когда приоритеты связываемых устройств одинаковы вопрос о выборе мастер-устройства решается жребием.

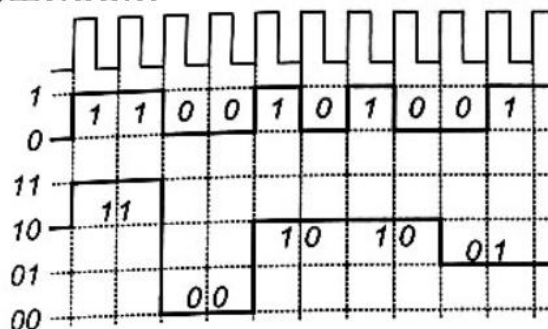


Коды передачи

ИУ4 – «сети»

Технические решения 1000BASE-T

Для уменьшения тактовой частоты до величин, позволяющих передавать информацию по витым парам категории 5 и 6, данные в линии представляются в так называемом **Enhanced TX/T2-коде**. В нем передаваемый сигнал имеет набор из пяти фиксированных уровней $\{-2, -1, 0, +1, +2\}$. Четыре из них используются для кодирования информационных битов, а пятый предназначен для коррекции ошибок.



Коды передачи

ИУ4 – «сети»

Технические решения 1000BASE-T

На наборе из четырех фиксированных уровней одной посылкой можно закодировать сразу два информационных бита, поскольку число возможных комбинаций из четырех по два равно четырем - 00, 11, 01 и 10. Таким образом, сигнал в коде Enhanced TX/T2 имеет тактовую частоту 125 МГц, что в 2 раза ниже тактовой частоты отдельных составляющих информационной последовательности.

Передаваемые одновременно по всем парам кабеля четверки пятеричных знаков представляют собой неделимую единицу информации размером в один байт. При тактовой частоте $f_T = 125$ МГц по всем четырем парам кабеля передается информационный поток $125 \text{ МГц} \cdot 2 \text{ бита/пара} \cdot 4 \text{ пары} = 1000 \text{ Мбит/с}$.

Пятый избыточный уровень Enhanced TX/T2 кода используется для построения механизма коррекции ошибок. Он реализуется кодером Треллиса и декодером Витерби. Применение механизма коррекции ошибок позволяет увеличить помехоустойчивость приемника в 6 дБ.





Передача данных в сетях

Основы телекоммуникационных технологий
Курс лекций

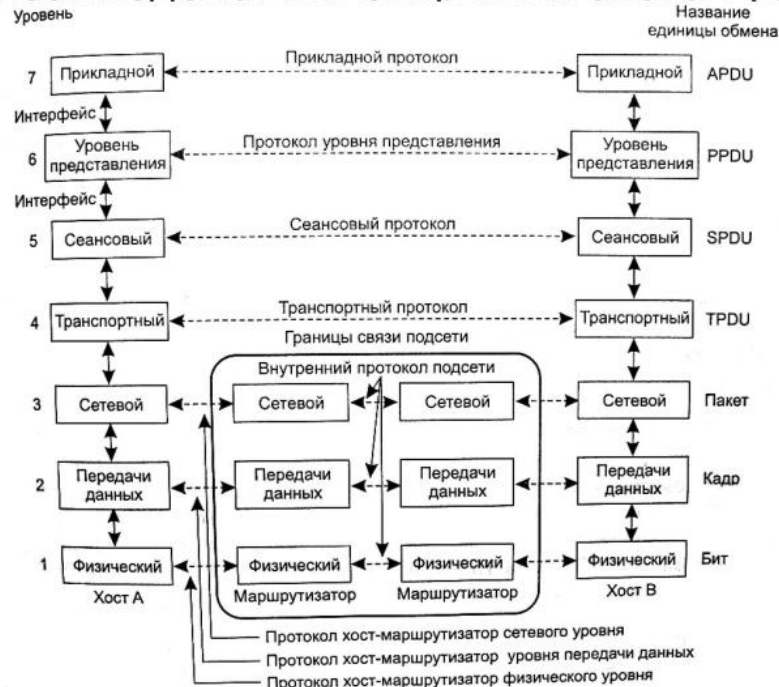
План лекции

ИУ4 – «сети»

- 1 Основные принципы и определения
- 2 Модели сетевого взаимодействия
- 3 Протоколы
- 4 Дополнительная информация

Эталонная модель

Взаимодействия открытых систем (OSI)



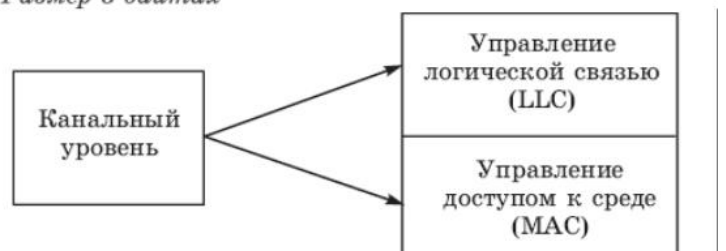
Уровни модели OSI

- Уровень 0** — На этом уровне ничего не описывается, уровень 0 предоставляет физическому уровню 1 только *среду передачи* (кабели различных типов, радио-, ИК-сигналы и т. д.).
- Уровень 1 — Физический (Physical).** Осуществляется передача неструктурированного потока битов, полученных от вышележащего канального уровня 2, по физической среде - например, в виде электрических или световых сигналов. Физический уровень отвечает за поддержание связи (link) и детально описывает электрические, оптические, механические и функциональные интерфейсы со средой передачи.

Уровни модели OSI

- Уровень 2 – Канальный (Data Link).**
 Обеспечивает безошибочную передачу данных, полученных от вышележащего сетевого уровня 3, через физический уровень 1, который сам по себе отсутствия ошибок не гарантирует и может исказить данные. Информация на этом уровне помещается в кадры (*frames*), где в начале (*заголовке кадра*) содержатся адреса получателя и отправителя, а также управляющая информация, а в конце — *контрольная сумма*, позволяющая выявить возникающие при передаче ошибки.

Структура кадра



Уровни модели OSI

- **Уровень 3 – Сетевой (Network).** Отвечает за *обеспечение связи* между любыми, даже находящимися в разных концах земного шара, точками в сети. Этот уровень осуществляет *проводку сообщений по сети*, которая может состоять из множества отдельных сетей, соединенных множеством линий связи. Такая доставка требует *маршрутизации*, т. е. *определения пути доставки сообщения*, а также решения задач управления потоками данных и обработки ошибок передачи.

Уровни модели OSI

- **Уровень 4 – Транспортный (Transport).** Гарантирует доставку информации от одного компьютера другому. На этом уровне компьютера-отправителя большие блоки данных разбиваются на более мелкие пакеты, которые доставляются компьютеру-получателю *в нужной последовательности, без потерь и дублирования*. На транспортном уровне компьютера-получателя пакеты вновь собираются в исходные блоки данных. Таким образом, транспортный уровень *завершает процесс передачи данных*, скрывая от более высоких уровней все детали и проблемы, связанные с доставкой информации *любого объема* между любыми точками во всей сети.

Уровни модели OSI

- **Уровень 5 – Сеансовый (Session).**

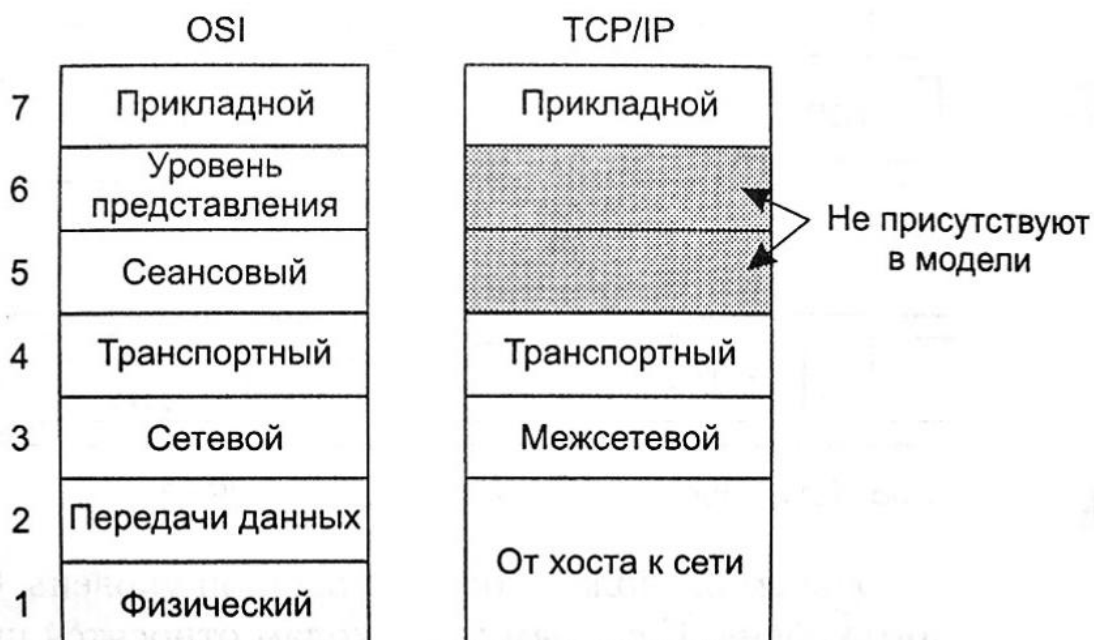
Позволяет двум сетевым приложениям на разных компьютерах устанавливать, поддерживать и завершать соединение, называемое *сетевым сеансом*. Этот уровень также отвечает за восстановление аварийно прерванных сеансов связи. Кроме того, на пятом уровне выполняется преобразование удобных для людей имен компьютеров в сетевые адреса (*распознавание имен*), а также реализуются функции *защиты сеанса*.

Уровни модели OSI

- **Уровень 6 – Уровень представления данных (Presentation).** Определяет *форматы* передаваемой между компьютерами информации. Здесь решаются такие задачи, как *перекодировка* (перевод информации в вид, понятный для всех участвующих в обмене компьютеров), сжатие и распаковка данных, шифрование и дешифровка, поддержка сетевых файловых систем и т. д.
- **Уровень 7 – Прикладной (Application), или Уровень Приложений.** Обеспечивает интерфейс взаимодействия программ, работающих на компьютерах в сети. Именно с помощью этих программ пользователь получает доступ к таким сетевым услугам, как обмен файлами, передача электронной почты, удаленный терминальный доступ и т.д.

Эталонная модель TCP/IP

• Первое описание 1974 год



Сравнение OSI и TCP/IP

Обе модели основаны на стеке независимых протоколов. Функциональность уровней схожа.

Транспортная служба независимая от сети.

Центральные концепции OSI

1. Сервисы – что делает уровень, но не как он это делает
2. Интерфейсы – способ доступа для процессов выше
3. Протоколы – определяют реализацию сервиса

Результат: относительная легкость замены протоколов при смене технологии OSI разрабатывалась до протоколов и не была настроена на конкретный стек протоколов.

TCP/IP наоборот описывала стек протоколов TCP/IP.

Связь на основе соединений: сетевой и транспортный уровни.

Протоколы Интернет

ИУ4 – «сети»

● На канальном уровне

- RS-232
- RS-422
- RS-423
- RS-449
- RS-485
- xDSL
- ISDN (T1, E1)
- Ethernet (10BASE-T, 10BASE2, 10BASE5)
- Fast Ethernet (100BASE-T, 100BASE-TX, 100BASE-T4, 100BASE-FX)
- Gigabit Ethernet (1000BASE-T, 1000BASE-TX, 1000BASE-SX)

Протоколы Интернет

ИУ4 – «сети»

- **RS-232** — это стандартный электрический интерфейс для последовательной двунаправленной передачи данных, поддерживающий асинхронную связь.
- Этот стандарт соединения оборудования был разработан в **1962** году рядом крупных промышленных корпораций и опубликован Ассоциацией электронной промышленности США (Electronic Industries Association — **EIA**). Международный союз электросвязи **ITU-T** использует аналогичные рекомендации под названием V.24 и V.28. В **России** подобный стандарт описан в ГОСТ 18145-81.



Протоколы Интернет

ИУ4 – «сети»

- **xDSL** — семейство технологий, позволяющих значительно расширить пропускную способность абонентской линии местной **телефонной сети** путём использования эффективных линейных кодов и адаптивных методов коррекции искажений линии на основе современных достижений микроэлектроники и методов **цифровой обработки сигнала**.
- В аббревиатуре xDSL символ «x» используется для обозначения первого символа в названии конкретной технологии.
- Технологии xDSL позволяют передавать данные со скоростями, значительно превышающими те скорости, которые доступны даже самым лучшим аналоговым и цифровым **модемам**.

Протоколы Интернет

ИУ4 – «сети»

- Основное назначение ISDN — передача **данных** со скоростью до 64 кбит/с по 4-килогерцовой проводной линии и обеспечение интегрированных телекоммуникационных услуг (**телефон, факс**, и пр.). Использование для этой цели телефонных проводов имеет два преимущества: они уже существуют и могут использоваться для подачи питания на терминальное оборудование.

- На канальном уровне

- Ethernet
- FDDI
- ATM
- Frame relay
- HDLC
- PPP
- SLIP

- High-Level Data Link Control (HDLC) — бит-ориентированный кодопрозрачный сетевой протокол управления каналом передачи данных канального уровня сетевой модели OSI, разработанный ISO.

HDLC может быть использован в соединениях точка-многоточка, но в настоящее время в основном используется в соединениях точка-точка с использованием асинхронного сбалансированного режима (ABM).

Его несильно измененные дочерние протоколы — LAPB, LAPM, LAPF, LAPD были встроены ITU соответственно в стеки протоколов X.25, V.42, Frame Relay, ISDN. Также HDLC был базой при разработке кадровых механизмов в протоколе PPP, широко используемом в Интернете.

Протоколы Интернет

ИУ4 – «сети»

PPP ([англ. Point-to-Point Protocol](#)) — [протокол точка-точка](#).

- PPP — это механизм для создания и запуска [IP](#) (Internet Protocol) и других [сетевых протоколов](#) на последовательных линиях связи — будь это прямая последовательная связь (по нуль-модемному кабелю), связь поверх [Ethernet](#), модемная связь по [телефонным линиям](#), мобильная связь по технологиям [CSD](#), [GPRS](#) или [EDGE](#).
- Используя PPP, можно подключить [компьютер](#) к PPP-серверу и получить доступ к ресурсам сети, к которой подключён сервер (почти) так, как будто вы подключены непосредственно к этой сети.
- PPP включает [IP](#), [IPX](#) и [NetBEUI](#) пакеты внутри PPP [кадров](#).

Протоколы Интернет

ИУ4 – «сети»

SLIP (Serial Line Internet Protocol) — устаревший [сетевой протокол канального уровня](#) для доступа к сетям [стека TCP/IP](#) через низкоскоростные линии связи путем простой [инкапсуляции IP-пакетов](#). Используются [коммутируемые соединения](#) через [последовательные порты](#) для соединений клиент-сервер типа [точка-точка](#). В настоящее время вместо него используют более совершенный протокол [PPP](#).

- На сетевом уровне

- BGP
- ICMP
- IGMP
- IP
- OSPF
- RIP
- EIGRP

- BGP (*Border Gateway Protocol*, протокол граничного шлюза) — основной протокол динамической маршрутизации в Интернете.
- BGP, в отличие от других протоколов динамической маршрутизации, предназначен для обмена информацией о маршрутах не между отдельными маршрутизаторами, а между целыми автономными системами. BGP не использует технические метрики, а осуществляет выбор наилучшего маршрута исходя из правил, принятых в сети.
- BGP поддерживает бесклассовую адресацию и использует суммирование маршрутов для уменьшения таблиц маршрутизации. С 1994 года действует четвертая версия протокола.
- BGP, наряду с DNS, является одним из главных механизмов, обеспечивающих функционирование Internet.

Протоколы Интернет

ИУ4 – «сети»

- **ICMP** (Internet Control Message Protocol — межсетевой протокол управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных. Также на ICMP возлагаются некоторые сервисные функции. Хотя формально ICMP использует IP (ICMP пакеты инкапсулируются в IP пакеты), он является неотъемлемой частью IP и обязателен при реализации стека TCP/IP.
- Утилита ping, служащая для проверки возможности доставки IP пакетов использует ICMP сообщения с типом 8 (эхо-запрос) и 0 (эхо-ответ).



Протоколы Интернет

ИУ4 – «сети»

- **IGMP** (*Internet Group Management Protocol* — протокол управления группами Интернета) сетевой протокол, используется узлами в сети, основанной на протоколе IPv4, для сообщения принадлежности к IP-группе сетевым маршрутизаторам, а также выполнения других функций управления групповой маршрутизацией.
- IGMP версии 3 определён в RFC 3376

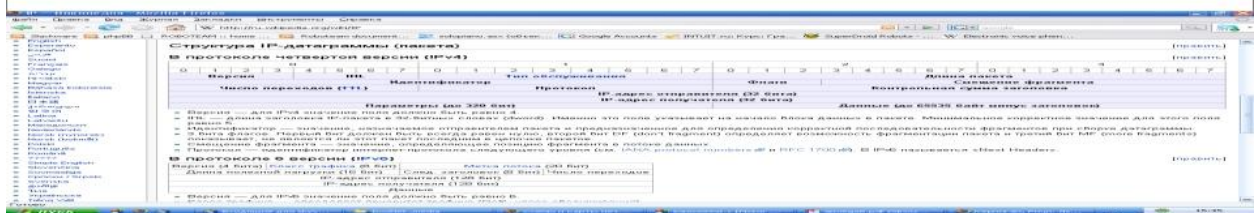
Протоколы Интернет

ИУ4 – «сети»

- **IP** (Internet Protocol — межсетевой протокол) — маршрутизируемый сетевой протокол, основа стека протоколов TCP/IP.
- Протокол IP (RFC 791) используется для ненадёжной доставки данных (разделяемых на так называемые *пакеты*) от одного узла сети к другому. Это означает, что на уровне этого протокола не даётся гарантий надёжной доставки пакета до адресата. В частности, пакеты могут прийти не в том порядке, в котором были отправлены, оказаться повреждёнными или не прибыть вовсе. Гарантии безошибочной доставки пакетов дают протоколы более высокого уровня.
- В современной сети Интернет используется IP четвёртой версии, также известный как IPv4.

Протоколы Интернет

ИУ4 – «сети»



Версия — для IPv6 значение поля должно быть равно 6.

Класс трафика — определяет приоритет трафика (QoS, класс обслуживания).

Метка потока — уникальное число, одинаковое для однородного потока пакетов.

Длина полезной нагрузки — длина данных (заголовок IP-пакета не учитывается).

Следующий заголовок — Определяет следующий инкапсулированный протокол.

Число переходов — максимальное число роутеров, которые может пройти пакет.

При прохождении роутера это значение уменьшается на единицу и по достижению нуля пакет отбрасывается.

Протоколы Интернет

ИУ4 – «сети»

- **OSPF** (*Open Shortest Path First*) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала (link-state technology) и использующий для нахождения кратчайшего пути Алгоритм Дейкстры (Dijkstra's algorithm).
- Протокол **OSPF** был разработан IETF в 1988 году. Последняя версия протокола представлена в RFC 2328. Протокол OSPF представляет собой протокол внутреннего шлюза (Interior Gateway Protocol — IGP). Протокол OSPF распространяет информацию о доступных маршрутах между маршрутизаторами одной автономной системы.
 - OSPF предлагает решение следующих задач:
 - Увеличение скорости сходимости;
 - Поддержка сетевых масок переменной длины (VLSM);
 - Достижимость сети;
 - Использование пропускной способности;
 - Метод выбора пути.

Протоколы Интернет

ИУ4 – «сети»

- **Протокол RIP** (*Routing Information Protocol*) — один из наиболее распространенных протоколов маршрутизации в небольших компьютерных сетях, который позволяет маршрутизаторам динамически обновлять маршрутную информацию (направление и дальность в хопах), получая ее от соседних маршрутизаторов.
- RIP — так называемый протокол *вектор-расстояния*, который оперирует *хопами* в качестве метрики маршрутизации. Максимальное количество хопов, разрешенное в RIP — 15 (метрика 16 означает "бесконечно большую метрику"). Каждый RIP-маршрутизатор по умолчанию вещает в сеть свою полную таблицу маршрутизации раз в 30 секунд, генерируя довольно много трафика на низкоскоростных линиях связи. RIP работает на сетевом уровне стека TCP/IP, используя UDP порт 520.
- В современных сетевых средах RIP — не самое лучшее решение для выбора в качестве протокола маршрутизации, так как его возможности уступают более современным протоколам, таким как EIGRP, OSPF. Ограничение на 15 хопов не дает применять его в больших сетях. Единственный плюс этого протокола — простота конфигурирования.

Протоколы Интернет

ИУ4 – «сети»

- **EIGRP** (*Enhanced Interior Gateway Routing Protocol*) — [протокол](#) маршрутизации, разработанный фирмой [Cisco](#) на основе протокола [IGRP](#) той же фирмы. EIGRP использует механизм DUAL для выбора наиболее короткого маршрута.
- Более ранний и практически не используемый ныне протокол IGRP был создан как альтернатива протоколу [RIP](#) (до того, как был разработан [OSPF](#)). После появления OSPF, Cisco представила EIGRP — переработанный и улучшенный вариант IGRP, свободный от основного недостатка дистанционно-векторных протоколов — особых ситуаций с зацикливанием [маршрутов](#) — благодаря специальному [алгоритму](#) распространения информации об изменениях в [топологии сети](#). Несмотря на то, что в общем случае протоколы состояния связей (OSPF) обрабатывают изменения в топологии сети быстрее, чем EIGRP, а также OSPF имеет ряд дополнительных возможностей, EIGRP более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора.

Протоколы Интернет

ИУ4 – «сети»

- На [транспортном уровне](#)
 - [TCP](#)
 - [UDP](#)

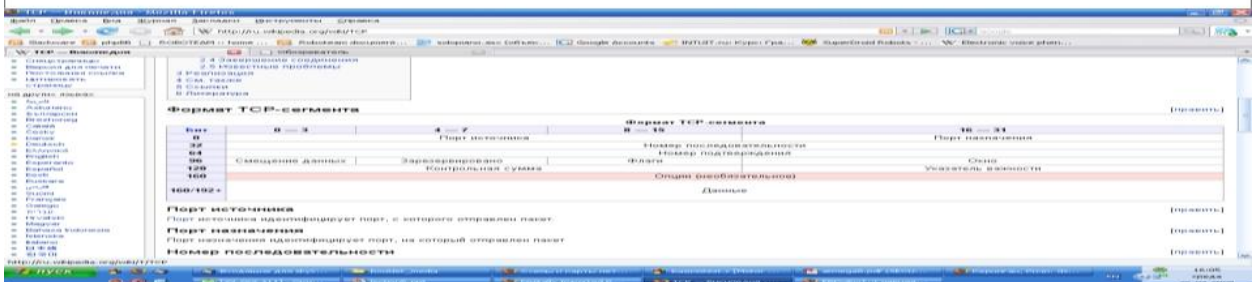
Протоколы Интернет

ИУ4 – «сети»

- **TCP** (*Transmission Control Protocol* — протокол управления передачей) — один из основных сетевых протоколов Internet, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.
- IP-идентификатор — 6.
- TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в безошибочности получаемых данных, осуществляет повторный запрос данных в случае потери пакетов и устраняет дублирование при получении двух копий одного пакета. В отличие от UDP, TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.

Протоколы Интернет

ИУ4 – «сети»



Порт источника идентифицирует порт, с которого отправлен пакет.

Порт назначения идентифицирует порт, на который отправлен пакет.

Номер последовательности выполняет две задачи:

- Если установлен флаг SYN, то это начальное значение номера последовательности и первый байт данных — это номер последовательности плюс 1.
- В противном случае, если SYN не установлен, первый байт данных — номер последовательности

Если установлен флаг ACK, то это поле содержит номер последовательности, ожидаемый отправителем в следующий раз. Помечает этот пакет как подтверждение получения.

Протоколы Интернет

ИУ4 – «сети»

The screenshot shows a web browser window displaying a page titled "Механизм действия протокола" (Protocol Action Mechanism). The page content includes:

- Указатель важности:** 16-битовое значение положительного смещения от порядкового номера в данном сегменте. Это поле указывает порядковый номер октета, с которого начинаются важные (urgent) данные. Поле принимается во внимание только для пакетов с установленным флагом U.
- Механизм действия протокола**
- Состояния сеанса TCP**

Состояние	Описание
CLOSED	Начальное состояние узла. Фактически фактивное.
LISTEN	Сервер ожидает запросов установления соединения от клиента.
SYN_SENT	Клиент отправил запрос серверу на установление соединения и ожидает ответа.
SYN_RECEIVED	Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения.
ESTABLISHED	Соединение установлено, идет передача данных.
FIN_WAIT_1	Одна из сторон сокета (на узле 1) прекращает соединение, отправил сегмент FIN.
CLOSE_WAIT	Другая сторона (узел 2) переходит в это состояние, отправил сегмент ACK и продолжает одностороннюю передачу.
FIN_WAIT_2	Узел 1 получает ACK, продолжает слушать и ждет получения сегмента FIN.
LAST_ACK	Узел 2 заканчивает передачу и отправляет сегмент FIN.
TIME_WAIT	Узел 1 получил сегмент FIN, отправил ACK и ждет 2*MSL секунд, перед окончательным разрушением канала.
CLOSING	Состояние закрытия соединения (фактивное?)
- Установки соединения:** Процесс начала сеанса TCP называется "трехным рукопожатием".
- Передача данных**
- Завершение соединения**

Протоколы Интернет

ИУ4 – «сети»

- **UDP (User Datagram Protocol** — протокол пользовательских датаграмм) — это **транспортный протокол** для передачи данных в сетях **IP**. Он является одним из самых простых протоколов **транспортного уровня**. Его **IP-идентификатор** — 17.
- В отличие от **TCP**, UDP не гарантирует доставку пакета, поэтому аббревиатуру иногда расшифровывают как «*Unreliable Datagram Protocol*» (протокол ненадежных датаграмм). Это позволяет ему гораздо быстрее и эффективнее доставлять данные для приложений, которым требуется большая пропускная способность линий связи, либо требуется малое время доставки данных.

Протоколы Интернет

ИУ4 – «сети»



65527 байт max

Для взаимодействия сетевых приложений протокол UDP использует 16-ти битные порты, которые могут принимать значения от 0 до 65535. Порт 0 является зарезервированным, но может использоваться как порт источника, если приложение не ожидает ответных данных.

Порты с 1 по 1023 являются системными и фиксированными, во многих ОС привязка к ним требует повышенных привилегий приложения.

Порты с 1024 по 49151 — зарегистрированные.

Порты с 49152 по 65535 — свободно используемые и временные. Используются клиентскими приложениями для связи с серверами.

Протоколы Интернет

ИУ4 – «сети»

● На сеансовом уровне/уровне представления

- SSL
- TLS

Протоколы Интернет

ИУ4 – «сети»

- **SSL** (*Secure Sockets Layer* — уровень защищённых гнёзд) — криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создаётся защищённое соединение между клиентом и сервером.
- Использует шифрование с открытым ключом для подтверждения подлинности передатчика и получателя. Поддерживает надёжность передачи данных за счёт использования корректирующих кодов и безопасных хэш-функций.
- Для доступа к страницам, защищённым протоколом SSL, в URL вместо обычного префикса http, как правило, применяется префикс https (порт 443), указывающий на то, что будет использоваться SSL-соединение.

Протоколы Интернет

ИУ4 – «сети»

- **TLS** (*Transport Layer Security*) — криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети Интернет.
- TLS-протокол основан на Netscape SSL-протоколе версии 3.0 и состоит из двух частей — TLS Record Protocol и TLS Handshake Protocol. Различия между SSL 3.0 и TLS 1.0 незначительные.
- В текущей версии протокола доступны следующие алгоритмы:
 - Для обмена ключами и проверки их подлинности применяются комбинации алгоритмов: RSA (асимметричный шифр), Diffie-Hellman (безопасный обмен ключами), DSA (алгоритм цифровой подписи) и алгоритмы технологии Fortezza.
 - Для симметричного шифрования: RC2, RC4, IDEA, DES, Triple DES или AES;
 - Для хэш-функций: MD5 или SHA.

Протоколы Интернет

ИУ4 – «сети»

● На прикладном уровне:

- [DNS](#)
- [FTP](#)
- [HTTP](#)
- [HTTPS](#)
- [IMAP](#)
- [LDAP](#)
- [POP3](#)
- [SMTP](#)
- [SSH](#)
- [Telnet](#)
- [XMPP \(Jabber\)](#)
- [SNMP](#)

Протоколы Интернет

ИУ4 – «сети»

- DNS (*Domain Name System* — система доменных имён) — распределённая система преобразования имени [хоста](#) в IP адрес. DNS работает в сетях [TCP/IP](#). Как частный случай, DNS может хранить и обрабатывать обратные запросы, определения имени хоста по его IP ([PTR-записи](#)).
- DNS обладает следующими характеристиками:
 - **распределённость хранения информации.** Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.
 - **Кеширование информации.** Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
 - **Иерархическая структура,** в которой все узлы объединены в [дерево](#), и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.
 - **Резервирование** За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Протоколы Интернет

ИУ4 – «сети»

Наиболее важные типы DNS-записей:

- **Запись А** (*address record*) или **запись адреса** связывает имя хоста с адресом IP.
- **Запись CNAME** (*canonical name record*) или **каноническая запись имени** (псевдоним) используется для перенаправления на другое имя
- **Запись MX** (*mail exchange*) или **почтовый обменник** указывает сервер(ы) обмена почтой для данного домена.
- **Запись PTR** (*pointer*) или **запись указателя** связывает IP хоста с его каноническим именем. Запрос в домене in-addr.arpa на IP хоста в reverse форме вернёт имя данного хоста.
- **Запись NS** (*name server*) указывает на DNS-серверы для данного домена.
- **Запись SOA** (*Start of Authority*) или **начальная запись зоны** указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, тайминги кеширования зонной информации и взаимодействия первичных и вторичных DNS-серверов.

Протоколы Интернет

ИУ4 – «сети»

- **FTP** (*File Transfer Protocol* — протокол передачи файлов) — протокол, предназначенный для передачи файлов в компьютерных сетях. FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами. TCP-порт 21.
- FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, в 1971 году. Он и сегодня широко используется для распространения ПО и доступа к удалённым хостам.
- Протокол не шифруется, при аутентификации передаёт логин и пароль открытым текстом. Если злоумышленник находится в одном сегменте сети с пользователем FTP, то, используя сниффер, он может перехватить логин и пароль пользователя, или, при наличии специального ПО, получать передаваемые по FTP файлы без авторизации. Чтобы предотвратить перехват трафика, необходимо использовать протокол шифрования данных SSL, который поддерживается многими современными FTP-серверами и некоторыми FTP-клиентами.

Протоколы Интернет

ИУ4 – «сети»

- **HTTP** (*HyperText Transfer Protocol* — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных в первую очередь в виде текстовых сообщений. Основой HTTP является технология «клиент-сервер».
- Основным объектом манипуляции в HTTP является *ресурс*, на который указывает **URI** (*Uniform Resource Identifier*) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т.д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Протоколы Интернет

ИУ4 – «сети»

Строка запроса выглядит так:

- **Метод URI HTTP/Версия** — для остальных версий.
- Здесь:
 - **URI** определяет путь к запрашиваемому документу.
 - **Метод** (*Method*) — английское слово в верхнем регистре, указывающее тип запроса. В версии HTTP 0.9 использовался только метод GET.
 - **Версия** (*Version*) — пара разделённых точкой арабских цифр. Например: 1.0.

GET /wiki/Http HTTP/1.0

Стартовая строка ответа сервера имеет следующий формат:

- **HTTP/Версия КодСостояния Пояснение**
- Здесь:
 - **Версия** — пара разделённых точкой арабских цифр как в запросе.
 - **КодСостояния** (*Status Code*) — три арабские цифры. По коду статуса определяется дальнейшее содержимое сообщение и поведение клиента.
 - **Пояснение** (*Reason Phrase*) — текстовое короткое пояснение к коду ответа для пользователя. Никак не влияет на сообщение и является необязательным.

HTTP/1.0 200 Ok

Протоколы Интернет

ИУ4 – «сети»

- **IMAP** (*Internet Message Access Protocol*) — [протокол](#) прикладного уровня для доступа к [электронной почте](#).
- IMAP предоставляет пользователю богатые возможности для работы с почтовыми ящиками, находящимися на центральном [сервере](#). Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с [клиента](#) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем. Для отправки писем используется протокол [SMTP](#).
- IMAP был разработан для замены более простого протокола [POP3](#).

Протоколы Интернет

ИУ4 – «сети»

- **SMTP** (*Simple Mail Transfer Protocol* — простой протокол передачи почты) — это [сетевой протокол](#), предназначенный для передачи [электронной почты](#) в [сетях TCP/IP](#).
- **ESMTP** (*Extended SMTP*) — масштабируемое расширение протокола SMTP. В настоящее время под «протоколом SMTP», как правило, подразумевают ESMTP и его расширения.
- SMTP используется для отправки почты от пользователей к [серверам](#) и между серверами для дальнейшей пересылки к получателю. Для приёма почты почтовый клиент должен использовать протоколы [POP3](#) или [IMAP](#).
- Чтобы доставить сообщение до адресата, необходимо переслать его почтовому серверу [домена](#), в котором находится адресат. Для этого обычно используется запись типа [MX](#) (*Mail eXchange* — обмен почтой) системы [DNS](#).

Протоколы Интернет

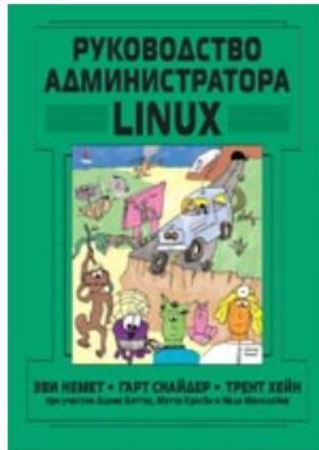
ИУ4 – «сети»

- Есть ещё целый ряд протоколов, ещё не стандартизированных, но уже очень популярных в Интернет. Эти протоколы в большинстве своём нужны для обмена файлами и текстовыми сообщениями, на некоторых из них построены целые файлообменные сети. Вот эти протоколы:
 - OSCAR
 - CDDB
 - eDonkey2000 (название сети; сам протокол называется MFTP)
 - BitTorrent
 - Gnutella
 - Skype

Протоколы Интернет

ИУ4 – «сети»

- **Gnutella** — полностью децентрализованная файлообменная сеть в рамках Интернета, потомок Napster, отличается принципиальным отсутствием центрального сервера. Сеть формируется, когда один пользователь Gnutella соединяется с другим пользователем, после чего они могут обмениваться доступной информацией. Обмениваться можно файлами любых форматов, есть множество удобных функций.
- **OSCAR** — открытый сетевой протокол, обеспечивающий обмен мгновенными и оффлайновыми текстовыми сообщениями. В данный момент используется для двух систем компании AOL (сейчас Time Warner): ICQ и AIM.
- **CDDB** (*Compact Disc Database*) — централизованная база данных по музыкальным компакт-дискам доступная через Интернет. Клиентская программа вычисляет идентификатор компакт-диска и делает запрос к этой базе данных. Вероятность совпадения идентификаторов для разных дисков очень мала.



Руководство администратора Linux

Эви Немет, Гарт Снайдер, Трент Хейн

Linux Administration Handbook First Edition

Evi Nemeth, Garth Snyder, Trent R. Hein

В книге рассмотрены три основных дистрибутива Linux: Red Hat 7.2, SuSE 7.3 и Debian 3.0. Эти дистрибутивы выбраны потому, что они наиболее популярны и позволяют продемонстрировать весь спектр подходов к вопросу администрирования Linux-систем. В то же время большая часть материала книги применима и к другим дистрибутивам общего назначения. Это одна из немногих книг, предназначенных не для широкого круга пользователей, а для системных администраторов, работающих в среде Linux. Изложенный материал будет полезен как профессионалам, так и новичкам, еще только постигающим тонкости этой увлекательной и трудной работы.





Передача данных в сетях

Основы телекоммуникационных технологий
Курс лекций

Рубежный контроль

ИУ4 – «сети»

Общие вопросы:

- Модель сетевого взаимодействия OSI, указать уровни, что на этом уровне происходит.
- 2. Модель сетевого взаимодействия TCP/IP. Указать уровни в модели, чем отличается и схожа с моделью OSI.

Вопросы по вариантам:

Протоколы Интернета, указать на каком уровне OSI применяется, каковы возможности протокола, цель его применения, где возможно, укажите формат пакета/кадра.

1
RS-232
FDDI
IP
PPP
SSL

2
xDSL
ATM
BGP
TCP
TLS

3
ISDN
HDLC
ICMP
RIP
HTTP

4
Ethernet
SLIP
EIGRP
UDP
FTP

План лекции

ИУ4 – «сети»

- 1 Основные принципы и определения
- 2 Распред. система доменных имен
- 3 Администрирование
- 4 Дополнительная информация

Основные принципы и определения

ИУ4 – «сети»

DNS (*Domain Name System* — система доменных имён) — распределённая система преобразования имени хоста (компьютера или другого сетевого устройства) в IP адрес. DNS работает в сетях TCP/IP. Как частный случай, DNS может хранить и обрабатывать и обратные запросы, определения имени хоста по его IP (PTR-записи).

Ключевые понятия DNS

Зона — логический узел в дереве имён. Право администрировать зону может быть передано третьим лицам, за счёт чего обеспечивается распределённость базы данных. При этом персоне, передавшая право на управление в своей базе данных хранит информацию только о существовании зоны (но не подзон!), информацию о персоне (организации), управляющей зоной и адрес серверов, которые отвечают за зону. Вся дальнейшая информация хранится уже на серверах, ответственных за зону.

Ключевые понятия DNS

Домен — название зоны в системе доменных имён Интернета, выделенной какой-либо стране, организации или для иных целей. Структура доменного имени отражает порядок следования зон в иерархическом виде; доменное имя читается справа налево (в порядке убывания значимости), корневым доменом всей системы является точка ('.'), следом идут домены первого уровня (географические или тематические), затем - домены второго уровня, третьего и т.д. (например, для адреса ru.wikipedia.org домен первого уровня — org, второго wikipedia, третьего ru). На практике точку в конце имени часто опускают, но она бывает важна в случаях разделения между относительными доменами и **FQDN** (*Fully Qualified Domain Name*, полностью определённое имя домена).

Ключевые понятия DNS

Поддомен — имя подчинённой зоны.

(например, wikipedia.org — поддомен домена org, а ru.wikipedia.org — домена wikipedia.org). Теоретически такое деление может достигать глубины 127 уровней, а каждая метка может содержать до 63 символов, пока общая длина вместе с точками не достигнет 254 символов. Но на практике [регистраторы доменных имён](#) используют более строгие ограничения.

Ключевые понятия DNS

DNS-сервер — специализированное ПО для обслуживания DNS. DNS-сервер может быть ответственным за некоторые зоны и/или может перенаправлять запросы вышестоящим серверам.

DNS-клиент — специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

Основные принципы и определения

ИУ4 – «сети»

Ключевые понятия DNS

ответственность (*authoritative*) — признак размещения зоны на DNS-сервере. Ответы DNS-сервера могут быть двух типов:

ответственные (когда сервер заявляет, что сам отвечает за зону) и

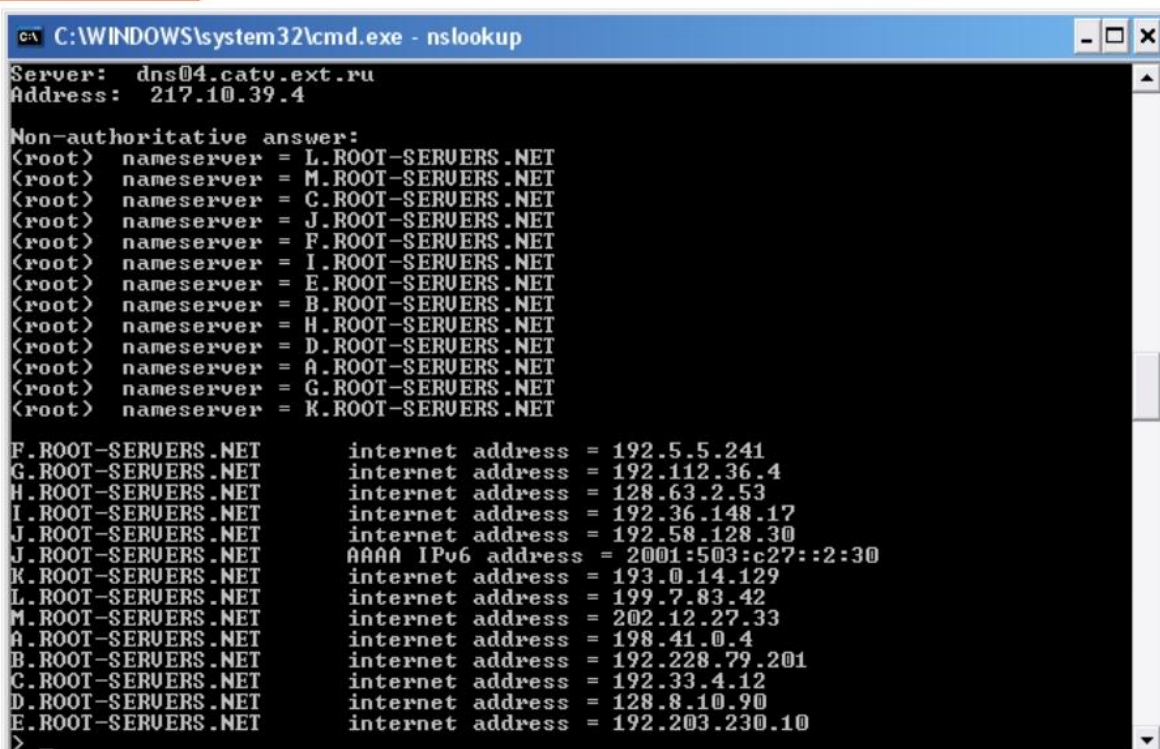
не ответственные (*Non-authoritative*), когда сервер обрабатывает запрос, и возвращает ответ других серверов. В некоторых случаях вместо передачи запроса дальше DNS-сервер может вернуть уже известное ему (по запросам ранее) значение (режим кеширования)



```
C:\WINDOWS\system32\cmd.exe - nslookup
> google - ru
Server: dns@4.catv.ext.ru
Address: 217.10.39.4
DNS request timed out.
  Timeout was 2 seconds.
Non-authoritative answer:
Name: google.ru
Address: 66.249.93.104, 216.239.59.104, 72.14.221.104
>
```

Основные принципы и определения

ИУ4 – «сети»



```
C:\WINDOWS\system32\cmd.exe - nslookup
Server: dns@4.catv.ext.ru
Address: 217.10.39.4
Non-authoritative answer:
(root) nameserver = L.ROOT-SERVERS.NET
(root) nameserver = M.ROOT-SERVERS.NET
(root) nameserver = C.ROOT-SERVERS.NET
(root) nameserver = J.ROOT-SERVERS.NET
(root) nameserver = F.ROOT-SERVERS.NET
(root) nameserver = I.ROOT-SERVERS.NET
(root) nameserver = E.ROOT-SERVERS.NET
(root) nameserver = B.ROOT-SERVERS.NET
(root) nameserver = H.ROOT-SERVERS.NET
(root) nameserver = D.ROOT-SERVERS.NET
(root) nameserver = A.ROOT-SERVERS.NET
(root) nameserver = G.ROOT-SERVERS.NET
(root) nameserver = K.ROOT-SERVERS.NET
F.ROOT-SERVERS.NET internet address = 192.5.5.241
G.ROOT-SERVERS.NET internet address = 192.112.36.4
H.ROOT-SERVERS.NET internet address = 128.63.2.53
I.ROOT-SERVERS.NET internet address = 192.36.148.17
J.ROOT-SERVERS.NET internet address = 192.58.128.30
J.ROOT-SERVERS.NET AAAA IPv6 address = 2001:503:c27::2:30
K.ROOT-SERVERS.NET internet address = 193.0.14.129
L.ROOT-SERVERS.NET internet address = 199.7.83.42
M.ROOT-SERVERS.NET internet address = 202.12.27.33
A.ROOT-SERVERS.NET internet address = 198.41.0.4
B.ROOT-SERVERS.NET internet address = 192.228.79.201
C.ROOT-SERVERS.NET internet address = 192.33.4.12
D.ROOT-SERVERS.NET internet address = 128.8.10.90
E.ROOT-SERVERS.NET internet address = 192.203.230.10
>
```

Основные принципы и определения

ИУ4 – «сети»

Ключевые понятия DNS

DNS-запрос *DNS query* — запрос от клиента серверу. Запрос может быть *рекурсивным* или *нерекурсивным*.

Нерекурсивный запрос либо возвращает данные о зоне, которая находится в зоне ответственности DNS-сервера (который получил запрос) или возвращает адреса корневых серверов (точнее, адрес любого сервера, который обладает большим объёмом информации о запрошенной зоне, чем отвечающий сервер).

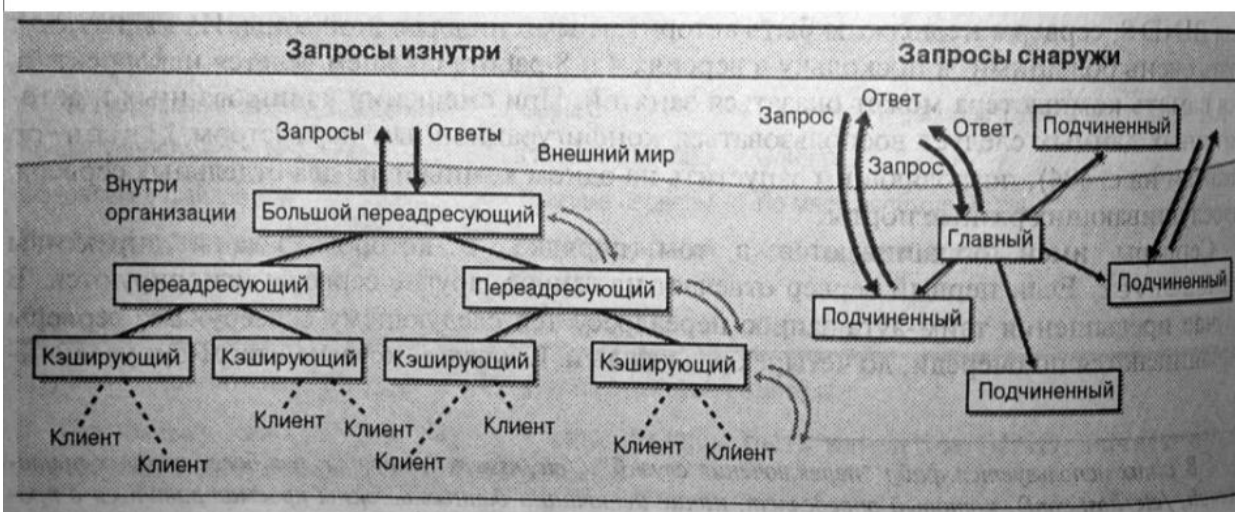
В случае **рекурсивного запроса** сервер опрашивает сервера (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует. На практике поиск начинается с наиболее близких к искомому DNS-серверов, если информация о них есть в кеше и не устарела, сервер может не запрашивать DNS-сервера).

Основные принципы и определения

ИУ4 – «сети»

Ключевые понятия DNS

Схема функционирования серверов DNS

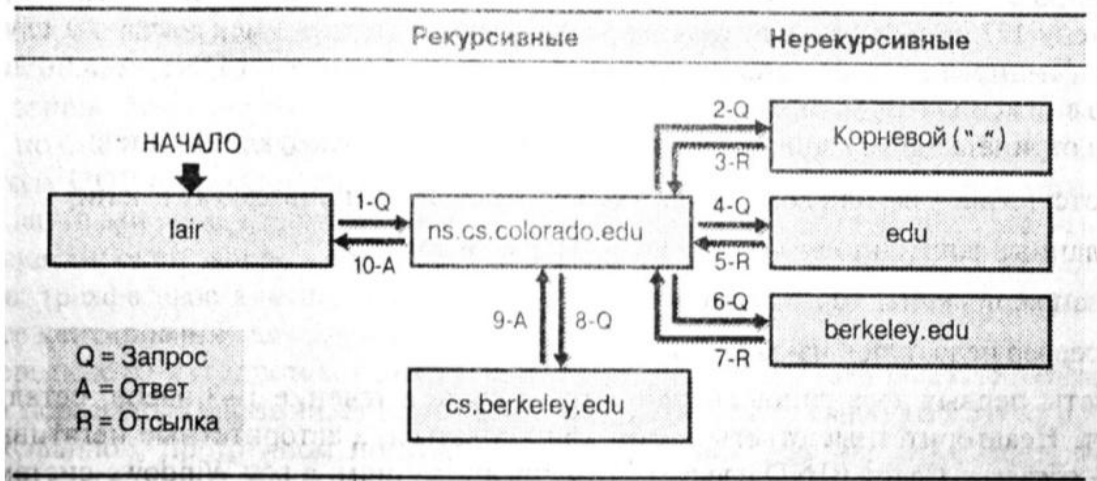


Основные принципы и определения

ИУ4 – «сети»

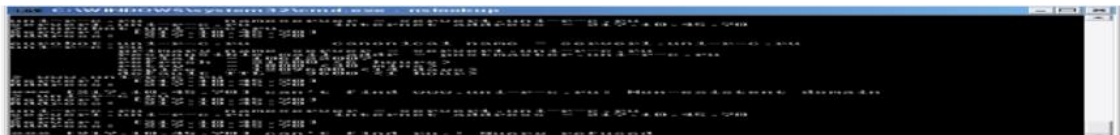
Ключевые понятия DNS

Обработка запроса в DNS



Основные принципы и определения

ИУ4 – «сети»



```
C:\WINDOWS\system32\cmd.exe - nslookup
> eurobot.uni-r-c.ru
Server: [217.10.45.70]
Address: 217.10.45.70

eurobot.uni-r-c.ru    canonical name = server1.uni-r-c.ru
uni-r-c.ru
primary name server = server1.uni-r-c.ru
responsible mail addr = postmaster.uni-r-c.ru
serial = 106122007
refresh = 21600 (6 hours)
retry = 1800 (30 mins)
expire = 1209600 (14 days)
default TTL = 3600 (1 hour)
```

Основные принципы и определения

ИУ4 – «сети»

Тип сервера	Описание
Авторитетный (authoritative)	Официальный представитель зоны
главный (master)	Основное хранилище данных зоны; берет информацию из дискового файла
подчиненный (slave)	Копирует данные с главного сервера
усеченный (stub)	Напоминает подчиненный сервер, но копирует данные только о серверах имен (записи NS)
внутренний (distribution)	Сервер, доступный только в пределах домена* (также называется невидимым сервером)
Неавторитетный** (nonauthoritative)	Отвечает на запросы, пользуясь данными из кэша; не "знает", являются ли эти данные корректными
кэширующий (caching)	Кэширует данные, полученные от предыдущих запросов; обычно не имеет локальных зон
переадресующий (forwarder)	Выполняет запросы от имени многих клиентов; формирует большой кэш
Рекурсивный (recursive)	Осуществляет запросы от имени клиента до тех пор, пока не будет найден ответ
Нерекурсивный (nonrecursive)	Отсылает клиента к другому серверу, если не может получить ответ на запрос

Основные принципы и определения

ИУ4 – «сети»

Простейший поиск соответствия ip-адрес – имя узла

- /etc/hosts - примитивная текстовая база данных, напоминающая старый Интернет. Она работает совместно с DNS и NIS, обеспечивая соответствие доменного имени IP адресу. Отдельные компьютеры, соединённые с помощью локальной сети могут быть записаны тут вместо named(8) сервера с целью упрощения. Кроме того, /etc/hosts используется для записи IP адресов и соответствующих им доменов, избавляя от внешнего траффика, используемого для запросов к DNS серверам.
- Формат /etc/hosts:
- [IP адрес в Интернете] [имя компьютера] [alias1] [alias2]
...Например:
 - 10.0.0.1 myRealHostname.foobar.com myRealHostname foobar1 foobar2

За дополнительной информацией обращайтесь к hosts(5).

Основные принципы и определения

ИУ4 – «сети»

/etc/resolv.conf

- /etc/resolv.conf определяет, как система получает доступ к DNS
- Основные записи resolv.conf:
 - nameserver IP адрес сервера имён. Сервера опрашиваются в порядке описания. Максимальное кол-во адресов - 3.
 - Search Список доменов для поиска с помощью hostname lookup. Обычно определяется доменом, в котором находится компьютер.
 - domain Домен, в котором находится компьютер.
- Типичный вид resolv.conf:
 - search foobar.com
 - nameserver 147.11.1.11
 - nameserver 147.11.100.30
- Если вы используете DHCP, dhclient(8) обычно перезаписывает resolv.conf информацией, полученной от серверов DHCP.

Основные принципы и определения

ИУ4 – «сети»

Название:	Domain Name System
Уровень (по модели OSI):	Прикладной
Семейство:	TCP/IP
Порт/ID:	53/TCP , 53/UDP
Назначение протокола:	Разрешение доменных имён
Спецификация:	RFC 1034 , RFC 1035 / STD 13
Основные реализации (клиенты):	DNS-модуль Microsoft Windows и Linux
Основные реализации (серверы):	BIND

Основные принципы и определения

ИУ4 – «сети»

- DNS обладает следующими характеристиками:
 - **распределённость хранения информации.** Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.
 - **Кеширование информации.** Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
 - **Иерархическая структура,** в которой все узлы объединены в [дерево](#), и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.
 - **Резервирование** За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Основные принципы и определения

ИУ4 – «сети»

- **BIND** (*Berkeley Internet Name Domain*, до этого: *Berkeley Internet Name Daemon*) — это открытая и наиболее распространённая реализация [DNS-сервера](#), обеспечивающая выполнение преобразование DNS-имени в IP-адрес и наоборот.
- BIND поддерживается организацией [Internet Systems Consortium](#). BIND был создан студентами и впервые был выпущен в [BSD 4.3](#).
- В [Unix](#) этот сервер является стандартом де-факто, но имеются и альтернативы:
 - **PowerDNS** - www.powerdns.com
 - **MyDNS** - DNS-сервер, использующий в качестве БД [MySQL](#)
 - **Microsoft DNS Server** - входит в состав Серверных версий [Windows](#).
- На данный момент имеется версия BIND 9.4.2.

Инсталляция и сопровождение BIND

Задача	Субъект	Как часто
Получение доменного имени	Организация	Один раз
Выбор серверов имен	Организация	Один раз или больше
Получение дистрибутива BIND	Организация	Один раз, но следить за обновлениями
Конфигурирование распознавателя	Клиент	Для каждой подсети и распространить
Конфигурирование файла "переключения служб"	Клиент	Для каждой платформы и распространить
Запуск демона named на этапе начальной загрузки	Сервер	Для каждого сервера имен
Настройка конфигурационного файла демона named	Сервер	Для каждого типа сервера
Конфигурирование файла "подсказок"	Сервер	Один раз* и распространить на серверы имен
Конфигурирование файлов зон	Главные серверы	Один раз
Обновление файлов зон	Главные серверы	По мере необходимости
Просмотр журнальных файлов	Регистрационный сервер	Хотя бы раз в неделю
Обучение пользователей	Все узлы	Постоянно

В пакет BIND включены следующие элементы:
named
DNS resolver library
tools for operating the server
named – демон DNS сервера в UNIX

Инструкция	Назначение
include	Подключает внешний файл (например, файл с ключами шифрования, доступный только демону named)
options	Задаёт глобальные параметры конфигурации сервера имен, а также установки по умолчанию
server	Задаёт параметры сервера
lwres	Конфигурирует сервер имен в качестве упрощенного распознавателя (только в BIND 9)
key	Определяет параметры аутентификации
acl	Формирует списки управления доступом
zone	Определяет зону, для которой демон является авторитетным
trusted-keys	Задаёт заранее установленные ключи шифрования
controls	Определяет, как утилита ndc будет управлять сервером имен
logging	Устанавливает категории журнальных сообщений и каналы их распространения
view	Определяет представление пространства имен (только в BIND 9)

Основные принципы и определения

ИУ4 – «сети»

Наиболее важные типы DNS-записей:

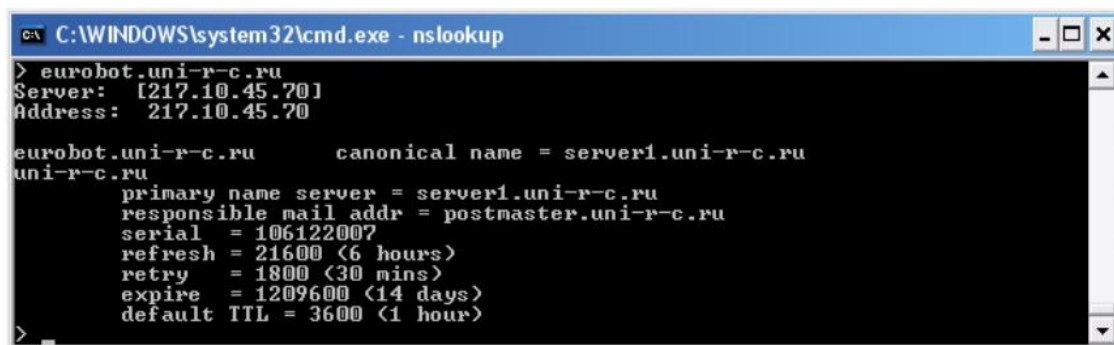
- **Запись A** (*address record*) или **запись адреса** связывает имя хоста с адресом IP.
- **Запись CNAME** (*canonical name record*) или **каноническая запись имени** (псевдоним) используется для перенаправления на другое имя
- **Запись MX** (*mail exchange*) или **почтовый обменник** указывает сервер(ы) обмена почтой для данного домена.
- **Запись PTR** (*pointer*) или **запись указателя** связывает IP хоста с его каноническим именем. Запрос в домене in-addr.arpa на IP хоста в reverse форме вернёт имя данного хоста.
- **Запись NS** (*name server*) указывает на DNS-серверы для данного домена.
- **Запись SOA** (*Start of Authority*) или **начальная запись зоны** указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, тайминги кеширования зонной информации и взаимодействия первичных и вторичных DNS-серверов.

Основные принципы и определения

ИУ4 – «сети»

Наиболее важные типы DNS-записей:

- **Запись SOA** (*Start of Authority*) или **начальная запись зоны** указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, тайминги кеширования зонной информации и взаимодействия первичных и вторичных DNS-серверов.



```
C:\WINDOWS\system32\cmd.exe - nslookup
> eurobot.uni-r-c.ru
Server: [217.10.45.70]
Address: 217.10.45.70

eurobot.uni-r-c.ru      canonical name = server1.uni-r-c.ru
uni-r-c.ru
  primary name server = server1.uni-r-c.ru
  responsible mail addr = postmaster.uni-r-c.ru
  serial = 106122007
  refresh = 21600 (6 hours)
  retry = 1800 (30 mins)
  expire = 1209600 (14 days)
  default TTL = 3600 (1 hour)
>
```

Файл описания зоны

@ - имя текущего домена

; - начало комментария

() – разбивка данных на несколько строк

* - спец. метасимвол

Базовый формат записи:

[имя] [ttl] [класс] тип данные

Класс – тип сети

IN, CH, NS

Файл описания зоны

;начало зоны мифического сервера ns.myth.xyz

\$TTL=18000

@ IN SOA ns.myth.xyz. hostmaster.ns.myth.xyz (

20011111300;порядковый номер

7200;Период обновления (2 часа)

1800;интервал между попытками

604800;интервал устаревания

7200);минимальное время жизни отр. ответов

Файл описания зоны

Запись NS (*name server*) указывает на DNS-серверы для данного домена.

```
зона [ttl] IN NS имя_узла  
myth.xyz. IN NS ns1.myth.xyz.  
myth.xyz. IN NS ns2.myth.xyz.  
либо  
IN NS ns1.myth.xyz.  
IN NS ns2.myth.xyz.
```

Файл описания зоны

Запись A (*address record*) или **запись адреса** связывает имя хоста с адресом IP.

```
имя_узла [ttl] IN A IP-адрес  
www IN A 128.138.243.100
```

Файл описания зоны

Запись PTR (*pointer*) или **запись указателя** связывает IP хоста с его каноническим именем. Запрос в домене in-addr.arpa на IP хоста в reverse форме вернёт имя данного хоста.

адрес [ttl] IN PTR имя_узла

100 IN PTR www.myth.xyz.

Домен по умолчанию 243.138.128.in-addr.arpa

Файл описания зоны

Запись CNAME (*canonical name record*) или **каноническая запись имени** (псевдоним) используется для перенаправления на другое ИМЯ

псевдоним [ttl] IN CNAME имя_узла

www IN A 128.138.243.100

ftp IN CNAME www

Файл описания зоны

Запись MX (*mail exchange*) или **почтовый обменник** указывает сервер(ы) обмена почтой для данного домена.

имя [ttl] IN MX приоритет узел

users IN MX 10 piper

IN MX 20 mailhub

IN MX 30 mail.myth.xyz.

bob@users

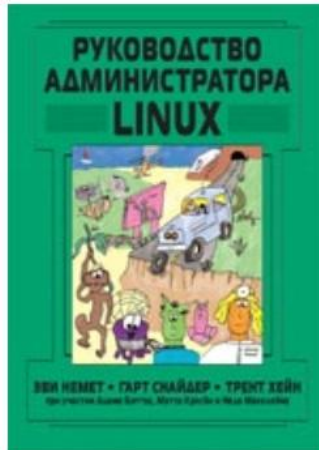
Получение доменного имени

Этапы

- выбор имени
- назначение 2-х или более компьютеров DNS-серверами
- согласование действий с администратором родительского домена

Делегирование зоны происходит только после проверки DNS-серверов нового домена иначе возможен эффект напрасного делегирования

Регистраторы доменных имен



Руководство администратора Linux

Эви Немет, Гарт Снайдер, Трент Хейн

Linux Administration Handbook First Edition

Evi Nemeth, Garth Snyder, Trent R. Hein

В книге рассмотрены три основных дистрибутива Linux: Red Hat 7.2, SuSE 7.3 и Debian 3.0. Эти дистрибутивы выбраны потому, что они наиболее популярны и позволяют продемонстрировать весь спектр подходов к вопросу администрирования Linux-систем. В то же время большая часть материала книги применима и к другим дистрибутивам общего назначения. Это одна из немногих книг, предназначенных не для широкого круга пользователей, а для системных администраторов, работающих в среде Linux. Изложенный материал будет полезен как профессионалам, так и новичкам, еще только постигающим тонкости этой увлекательной и трудной работы.





Сдача долгов

Основы телекоммуникационных технологий
Курс лекций

Рубежный контроль #1

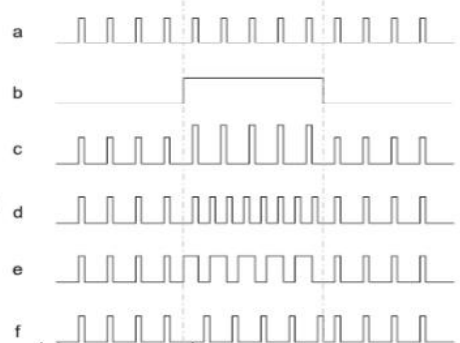
ИУ4 – «сети»

Общие вопросы

Модуляция сигнала:

- Определение модуляции
- Общие классы модуляции
- Основные характеристики модулированного сигнала
- Указать вид модуляции для всех сигналов на изображении

Вопросы по вариантам: указать определение (формулу), где применяется и какому общему классу принадлежит, пример временной диаграммы



1

Амплитудная модуляция

Модуляция с расширением спектра

Импульсно-кодовая модуляция

2

Частотная модуляция

Амплитудно-фазовая манипуляция

Широтно-импульсная модуляция

Рубежный контроль #2

ИУ4 – «сети»

Общие вопросы:

- Модель сетевого взаимодействия OSI, указать уровни, что на этом уровне происходит.
- 2. Модель сетевого взаимодействия TCP/IP. Указать уровни в модели, чем отличается и схожа с моделью OSI.

Вопросы по вариантам:

Протоколы Интернета, указать на каком уровне OSI применяется, каковы возможности протокола, цель его применения, где возможно, укажите формат пакета/кадра.

1

PDH
CDP
IPX
SPX
SOAP

2

SDH
LocalTalk
GRE
9P
NTP

3

SONET
IS-IS
AH
NFS
RADIUS





админ-зона

Shell программирование

@ Contents

- 1 Что такое shell?
- 2 А что в нем можно делать?
- 3 Полезные программы
- 4 Применение на примерах



@ Что такое shell?

Программирование на языке интерпретатора shell приобретает все большую популярность по мере утверждения Linux в качестве удобной в работе и отказоустойчивой операционной системы. Трудно оценить, какое количество пользователей работают с Linux. Эта операционная система распространяется бесплатно, хотя многие компании разрабатывают ее коммерческие варианты. Кроме того, несмотря на сделанные несколько лет назад неутешительные прогнозы специалистов относительно будущего UNIX, данная ОС также не теряет популярности, и число ее приверженцев продолжает расти.



@ Что такое shell?

Основные причины изучения shell-программирования заключаются в следующем:

- язык программирования интерпретатора shell вполне самодостаточен, он содержит все необходимые управляющие конструкции и удобен в применении;**
- открывается возможность быстрого создания сценариев;**
- сценарии позволяют автоматизировать рутинные операции, выполняемые обычно вручную.**

@ Что такое shell?

Интерпретатор bourne shell

- Bourne shell является стандартным интерпретатором команд, который входит в состав всех систем UNIX и совместим с интерпретатором bash в Linux. Учтите, что существуют и другие интерпретаторы, такие как bash, Korn shell и C shell. Если в вашей системе установлен интерпретатор bash, сценарии будут выполняться и в нем, поскольку bash совместим с Bourne shell. Синтаксис интерпретатора Korn shell близок к синтаксису рассматриваемого нами языка.
- Если внимательно изучить сценарии системной инсталляции, то можно обнаружить что более чем на 95 процентов они являются сценариями Bourne shell. Это объясняется тем, что создатели сценариев знали: они будут выполняться в любой системе UNIX и Linux.

@ Что такое shell?

Имя оболочки	Имя программы	Описание
Bourne shell	sh	Исходная оболочка AT&T, используется на машинах UNIX
C shell	csh	Оболочка, являющаяся частью BSD UNIX
Korn shell	ksh	Расширенная версия Bourne Shell, созданная AT&T
Bourne again shell	bash	Оболочка, распространяемая вместе с ОС Linux, включает средства редактирования командной строки и пр.

Отличия:

- Дополнительные возможности (история команд, редактирование командной строки);
- Синтаксис

@ Что такое shell?

Переносимость интерпретатора shell

- Если необходимо, чтобы создаваемый сценарий выполнялся под управлением любой системы, он должен обладать свойством переносимости. Переносимость сценариев определяется двумя основными факторами:
 - синтаксисом языка применяемого интерпретатора shell;
 - используемыми командами интерпретатора.
- Проблемы, связанные с первым фактором, решаются автоматически, если сценарии создаются для интерпретатора Bourne shell.
- Что касается второго фактора, то для большинства shell-сценариев характерна следующая особенность: по крайней мере, 20 процентов (а то и больше) их рабочего времени затрачивается на выполнение таких shell-команд, как `cp`, `mv`, `mkdir` и др. Здесь и заключена проблема переносимости. Дело в том, что поставщики разных операционных систем оснащают эти команды

@ Что такое shell?

Оболочка ОС Linux отвечает за выполнение следующих операций:

- Предоставление интерфейса командной строки;
- Осуществление перенаправления операций ввода-вывода;
- Подстановка имен файлов;
- Подстановка значений переменных;
- Раскрытие скобок;
- Предоставление средств интерпретируемого языка.

Оболочка выполняет следующие действия:

- Ожидает, пока пользователь не введет команду
- Если команда содержит специальные символы – выполняет соответствующую им обработку;
- Ищет исполняемый файл для команды; если такой файл отсутствует, то выводит сообщение об ошибке;
- Создает (`fork`) дочерний процесс, который будет исполнять команду;
- Ожидает, пока выполнение команды не завершится и переходит к первому пункту;

@ А что в нем можно делать?

При входе в систему пользователя автоматически загружается командный интерпретатор, указанный в конце файла `/etc/passwd`.

Задание:

Определите какой интерпретатор запускается при входе в систему пользователя `root`.

Выход из системы:

`exit`

`logout`

`Ctrl-D`

`Ctrl-D` интерпретируется как маркер конца файла. Прочитав его командный интерпретатор завершает свою работу.

Задание:

Запустите различные командные интерпретаторы (`csh`, `ksh`, `sh`) выполните в них команду (`ls -l`) и выйдите из них.

@ А что в нем можно делать?

Ввод и вывод данных в shell

`echo` – отображение на экране текста

`cat` – отображение файла, создание файла

`tee` – одновременное копирование и отображение на экране

Команда:

```
echo hello, Linux command line
```

Интерпретируется как команда `echo` с 4 аргументами:

- `hello,`
- `Linux`
- `command`
- `line`

Вывод этой команды будет следующий:

```
hello, Linux command line
```

Разделительные пробелы удалены

@ А что в нем можно делать?

Специализированные символы:

Символ	Значение
Пустое пространство	(символы табуляции и пробелы) Используются для отделения параметров командной строки
Символ новой строки	Показывает окончание команды
` " \	Символы цитирования. Позволяют изменить способ обработки специализированных символов
&	Используется в конце команды для указания того, что команда должна исполняться в фоновом режиме
< > << `	Перенаправление ввода-вывода
* ? [...] [!...]	Символы подстановки имен файлов
\$	Указывает на переменную оболочки
;	Используется для отделения команд, записанных последовательно в одну строку

@ А что в нем можно делать?

Для запуска подряд нескольких команд используется следующая запись:
`# ls ; cd /etc ; ls -l`

При выполнении задания в экранном режиме происходит «захват терминала» на весь период. Перевод задания в фоновый режим позволяет освободить терминал для других целей.

Для запуска команды в фоновом режиме: **&**

`$ sleep 100 &`

`$ find /etc -name "srm.conf" -print > find.dt 2>&1 &`

`[1] 5017`

команда > выходной файл 2>&1 &

Для возврата в экранный режим: **fg 1**

Для отображения исполняемых заданий: **ps**

@ А что в нем можно делать?

Подстановки в именах файлов

Символ	Описание
*	Соответствует произвольной строке
?	Соответствует любому символу
[...]	Соответствует любому символу из числа заключенных в скобки
[!...]	Соответствует любому символу за исключением тех, которые указаны в скобках

Примеры:

cat *

Соответствует всем файлам текущего каталога

ls a*bc

Соответствует файлам, начинающимся с a, и заканчивающимся bc

ls a?bc

Соответствует файлам, начинающимся с a, заканчивающимся bc, и имеющих РОВНО один символ между ними.

ls [ic]???

ls ![ic]???

Любой файл, содержащий 4 символа и не начинающийся не с i, не с c.

@ А что в нем можно делать?

Символ	Значение
Одинарная кавычка (')	Оболочка игнорирует все специальные символы, помещенные между двумя одинарными кавычками
Двойная кавычка (")	Оболочка игнорирует все специализированные символы, кроме \$ ` \, которые заключены между двумя двойными кавычками
Обратная косая черта (\)	Оболочка игнорирует специализированное значение любого символа, непосредственно следующего за \
Обратные кавычки ` `	Позволяют присвоить переменным данные вывода системных команд.

Примеры:

echo I'm David.

echo 'one two three four'

echo I\'m David.

echo "one two three four"

echo *

echo hello there \

echo `*`

my name is david

echo *

echo files = ; ls

echo one two three four

echo files = \; ls

echo `date`

@ А что в нем можно делать?

Команды создания файла

```
$ touch myfile
```

```
$ > myfile
```

```
$ cat > myfile
```

Создайте файлы со следующими именами, а затем удалите их:

stars*

-top

hello my friend

"goodbye"

@ А что в нем можно делать?

Перенаправление ввода вывода позволяет изменить источник и/или приемник потоков информации, обрабатываемой командами. Этот механизм позволяет комбинировать множество команд.

Для выполнения перенаправления используются *дескрипторы файлов*. При запуске (exec) новой команды оболочка в ней по умолчанию использует три дескриптора.

Имя	Номер	Приемник
standard input (stdin)	0	Стандартный поток ввода (Клавиатура)
standard output (stdout)	1	Стандартный поток вывода (Экран)
standard error (stderr)	2	Стандартный поток ошибок (Экран)

@ А что в нем можно делать?

Символ(ы)	Описание
command < file	В качестве стандартного входного потока получает данные из файла
command > file	Направляет стандартный поток вывода в новый файл
command >> file	Направляет стандартный поток вывода в указанный файл (режим присоединения)
command << разделитель	Получает данные из стандартного потока ввода до тех пор, пока не встретится разделитель
`command` (обратные кавычки)	Выполняет команду и подставляет на ее место данные стандартного выходного потока
command1 command2	Направляет выходной поток команды command1 во входной поток команды command2
command 2> file	Направляет стандартный поток ошибок в указанный файл
command <&m	В качестве стандартного входного потока получает данные из файла с дескриптором m
command >&m	Направляет стандартный поток вывода в файл с дескриптором m.

@ А что в нем можно делать?

Задание:

Выполните следующие команды и объясните результат

```
# ls > the.files
```

```
# ls /fred 2> /dev/null
```

```
# cat the.files | more
```

```
# ls /etc >> the.files
```

```
# echo Количество строк в файле the.files = `wc -l the.files`
```

```
# cat << finished > input
```

```
# cd /etc > output.file
```

```
# ls | cd
```

А ЧТО В НЕМ МОЖНО ДЕЛАТЬ?

Два основных действия:

- Назначение переменной значения
- Использование этого значения по имени переменной

1) Назначение значения переменной

```
# variable_name=value
```

```
# read variable_name
```

Например:

```
# my_variable=hello
```

```
# theNum=5
```

```
# myName="Иван Иванов"
```

Пробелы интерпретируются как особые символы. Не должно быть пробелов вокруг символа равенства (=).

2) Доступ к значению переменной

Для доступа к значению переменной используется специализированный символ \$, за которым указывается имя переменной

Например:

```
# myName="Иван Иванов"
```

А ЧТО В НЕМ МОЖНО ДЕЛАТЬ?

Установка переменной в пустое значение:

```
# myName=
```

Запрет изменения переменной :

```
# readonly my_variable
```

Значение такой переменной нельзя изменять. Для вывода всех таких переменных используйте команду `readonly` без параметров.

Очистка переменной:

```
# unset variable
```

Нельзя использовать эту команду над переменными "read-only", а также над некоторыми predefined переменными: IFS, PATH, PS1, PS2.

@ А что в нем можно делать?

Все переменные оболочки строкового типа

Для выполнения арифметических действий используется команда

expr:

```
# expr 5 + 6
```

```
# expr 10 / 5
```

```
# expr 5 \* 10
```

Использование **expr**

```
# count=`expr $count + 1`
```

Ограничения expr

- Команда работает только с целочисленной арифметикой
- Все аргументы должны разделяться пробелами.
- Для каждой операции необходимо создавать новый процесс.

Альтернатива expr

- Использование **bc** позволяет работать с десятичной точкой:

```
# echo 5.5 + 5 | bc
```

2. Использование арифметических возможностей **bash**:

```
# echo ${ 5 + 5 }
```

@ А что в нем можно делать?

Допустимым именем переменной оболочки является имя, начинающееся с буквы латинского алфавита или символа подчеркивания, за которой может следовать одна или несколько букв, цифр или подчеркиваний

Если необходимо использовать переменную в составе более длинного выражения, то для отделения имени переменной от выражения могут использоваться фигурные скобки - `{ }`.

Пример:

```
# directory=/etc/
```

```
# home=/home/vanja
```

Следующая строка не будет выполнена должным образом:

```
# cp $directorypasswd $home
```

Для выполнения операции используйте фигурные скобки:

```
# cp ${directory}passwd $home
```

@ А ЧТО В НЕМ МОЖНО ДЕЛАТЬ?

Для каждого нового процесса создается независимый блок переменных окружения:

```
# myName=david
# echo $myName
david
# bash
# echo my name is $myName
my name is
# exit
# echo $myName
david
```

Команда `export` позволяет экспортировать переменные оболочки в переменные среды:

```
# myName=David Jones
# bash
# echo my name is $myName
my name is
# logout
# export myName
# bash
# echo my name is $myName
my name is david
```

@ А ЧТО В НЕМ МОЖНО ДЕЛАТЬ?

Локальность переменных среды

В окружении нового процесса создаются локальные копии переменных окружения родительского процесса. Модификации этих переменных в дочернем процессе не приводит к их изменению в родительском и наоборот.

```
$ env
```

```
$ set
```

Пример:

```
# echo my name is $myName
my name is david
# export myName
# bash
# myName=fred
# exit
# echo my name is $myName
my name is david
```

А что в нем можно делать?

Раскрытие скобок – это обычное средство генерирования списка строк, имеющих общий префикс или постфикс.

Для описания этого средства используются фигурные скобки `{}`.
Строки в скобках должны разделяться с помощью запятой.

Например,

```
# echo a{b,c,d}e  
abe ace ade
```

Символы, указанные в скобках и в постфиксной и префиксной частях не интерпретируются и рассматриваются как обычные строки:

```
# echo *{5,6}  
*5 *6
```

А что в нем можно делать?

Интерпретация команды, введенной пользователем, осуществляется по шагам в следующей последовательности:

- 1) Перенаправление ввода-вывода;
- 2) Подстановка переменных
- 3) Подстановка имен файлов

Пример 1.

```
# pipe=\  
# echo $pipe  
|
```

Пример 2.

```
# star=\  
# echo $star  
Mail News README VMSpec.ps.bak acm.bhx acm2.dot
```


А что в нем можно делать?

Интерпретация команды, введенной пользователем, осуществляется по шагам в следующей последовательности:

- 1) Перенаправление ввода-вывода;
- 2) Подстановка переменных
- 3) Подстановка имен файлов

Пример 1.

```
# pipe=\  
# echo $pipe  
|
```

Пример 2.

```
# star=\  
# echo $star  
Mail News README VMSpec.ps.bak acm.bhx acm2.dot
```

А что в нем можно делать?

Команда `eval` является внутренней командой интерпретатора, позволяющая осуществить синтаксический анализ переданных параметров, методом повторяющим стандартный алгоритм оболочки.

Пример.

```
# pipe=\  
# ls $pipe more  
# eval ls $pipe more
```

Осуществляется повторный анализ, на второй итерации которого символ конвейера воспринимается как операция перенаправления.

@ А что в нем можно делать?

Встроенные команды shell

:

.

break

cd

continue

echo

eval

exec

exit

export

pwd

read

readonly

return

set

shift

test

times

trap

type

ulimit

umask

unset

wait

@ Литература

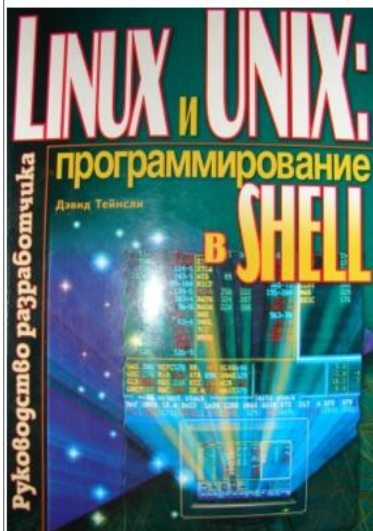
- Дэвид Тейнсли

Linux и UNIX: программирование в shell.

Руководство разработчика: Пер с англ. - К.:

Издательская группа BHV, 2001. — 464 с.

ISBN 966-552-085-7 ISBN 5-7315-0114-9



Данная книга является практическим руководством по программированию интерпретатора Bourne shell — стандартного командного интерпретатора в UNIX, полностью совместимого с интерпретатором BASH shell в Linux. Книга предназначена для начинающих и опытных программистов и содержит множество полезных примеров, советов и подсказок. С ее помощью читатель сможет быстро научиться создавать shell-сценарии для реальных задач и ситуаций, возникающих в большинстве систем UNIX и Linux.

Семинар
ы
ИУ4 - Сети



skycluster@gmail.com

Семинар
ы
ИУ4 - Сети



админ-зона
Shell программирование

- 1 Файлы и права доступа
- 2 Полезные программы
- 3 Немного о ядре linux
- 4 Домашние Задания

Категории доступа

ls -l

- владелец
- группа
- все пользователи

Доступ к файлу

- чтение - r
- запись - w
- исполнение - x

Информация о файле

```
-rwxr-xr-x 1 root root 3579 Oct 14 04:44  
dmesg
```



Файлы и права доступа

Типы файлов

d – каталог

l – символическая ссылка

s – сокет

b – специальный блочный файл

c – специальный символический файл

p – файл именованного канала

<-> – обычный файл

`cd ~`

`touch myfile`

`ls -l`

`chmod`

Информация о файле

`-rw-r--r-- 1 root root 0 Mar 13 08:50 myfile`



Файлы и права доступа

Изменение прав доступа

`chmod`

`chmod [кто] оператор [разрешения] файл`

Кто

u – владелец

g – группа

o – другие пользователи

a – все (владелец, группа, др. пользователи)

Оператор

+ добавление разрешения

- удаление разрешения

= установка заданного разрешения

Разрешения

r, w, x – чтение, запись, выполнение

X – выполнение, если у кого-то уже есть право

s – бит SUID или SGID

t – установка sticky-бита

u, g, o – установка тех же прав, что и у **u, g, o**



Файлы и права доступа

Изменение прав доступа

`chmod режим файл`

Режим

`гwx`

`4 + 2 + 1 = 7`

Для каталога

`г` – вывод содержания каталога

`w` – создание файлов в каталоге

`x` – возможность поиска в каталоге и перехода

`drwxrwxr-x 1 root root 2390 Mar 13 8:45 docs`

`chmod 666 myfile`

`man chmod`

`chmod -R`



Файлы и права доступа

Биты SUID и GUID

могут игнорироваться системой
трансфер привилегий владельца

Нужны, чтобы выполнять определенные административные задачи

Например: доступ в закрытые директории модератором для архивации и резервного копирования БД

`rwsr-xr-x 1 root root 0 Mar 13 8:45 myfile`

`cd /bin`

`ls -l | grep '^...s'`

`cd /sbin`

`ls -l | grep '^...s..s'`

`cd ~`

`chmod 4775 myfile`

`chmod u+s myfile`



Файлы и права доступа

chown и chgrp

/etc/passwd

передача прав другому владельцу

chown *владелец файл*

chgrp *владелец файл*

adduser

useradd

usermod

userdel

groupadd

groupdel

cd ~

groupadd vasya

adduser vasya

touch myfile

chown vasya myfile

chgrp vasya myfile

ls -l

userdel vasya

groupdel vasya



Файлы и права доступа

umask

стандартный режим доступа к файлам

/etc/profile – глобальные настройки

.profile – настройки пользователя

cd ~

cat > .profile

umask 333

ctrl+C

logout

>> login

touch myfile

ls -l

rm myfile

Цифра umask Результат – файл - директория

0	6	7
1	6	6
2	4	5
3	4	4
4	2	3
5	2	2
6	0	1
7	0	0



Файлы и права доступа

Символические ссылки

In `-s исх_файл [имя_ссылки]`

In `исх_файл [имя_ссылки]`

`cd ~`

`touch myfile`

In `-s myfile hisfile`

`ls -l`

In `myfile hardlink`

`cat > hardlink`

`cat myfile`



shell

Регистрация в системе

процесс регистрации

файл `/etc/passwd`

файл `/etc/shadow`

файл `/etc/profile`

- глобальные и локальные переменные среды

- путь к файлам PATH

- параметры терминала

- меры безопасности (`umask`)

- совет дня

файл `$HOME/.profile`

- переопределение глобального файла

`man 5 passwd`

`man 5 shadow`



shell

Параметры командной строки

shift

`./script`

script

#!/bin/bash

сценарий script

echo \$0

echo \$1

\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Наименование
сценария

Did

You

See

The

Full

Moon



shell

Специальные параметры shell

\$# Число аргументов, передаваемых сценарию

\$* **Shell – сценарий**

В отдельной строке отображаются все аргументы, которые передаются сценарию. Здесь может содержаться более девяти параметров, в отличие от позиционных параметров

Зачем?

Структура

\$? Текущий идентификатор PID для выполняющегося сценария

#! Идентификатор PID для последнего процесса, который выполняется в фоновом режиме

\$@ Означает то же самое, что и параметр **\$***, но, если параметр заключен в кавычки, то и каждый аргумент отображается в кавычках

\$- Отображение текущих опций интерпретатора команд; аналогично применению команды `set`

\$? Показывает код завершения последней команды. Значение 0 свидетельствует об отсутствии ошибок, любое другое значение — о их наличии

@ Что такое shell?

Проверка условий

test условие

или

[условие] – пробелы это важно!

- d Каталог
- f Обычный файл
- L Символическая связь
- r Файл для чтения
- s Файл имеет ненулевой размер, он не пуст
- w Файл для записей
- u Файл имеет установленный бит *suid*
- x Исполняемый файл

```
cd ~
```

```
[ -f myfile ]
```

```
echo $?
```

```
[ -f somefile ]
```

```
echo $?
```

@ Что такое shell?

Проверка условий

test условие

или

[условие] – пробелы это важно!

- a Логическое *AND*, возвращает истину, если обе части оператора принимают истинное значение
- o Логическое *OR*, возвращает истину, если какая-либо из частей оператора может принимать истинное значение
- ! Логическое *NOT*, возвращает истину, если условие ложно

```
cd ~
```

```
[ -w myfile -a -w .profile ]
```

```
echo $?
```

@ Что такое shell?

Проверка условий

test условие

или

[условие] – пробелы это важно!

[оператор строка]

[строка оператор строка]

```
cd ~
[ -z $EDITOR]
echo $?
var1="hello"
var2="well"
[ "$var1" = "$var2" ]
echo $?
```

= Две строки равны
! = Две строки не равны
-z Эта строка нулевая
-n Эта строка не является нулевой

@ Что такое shell?

Проверка условий

test условие

или

[условие] – пробелы это важно!

```
cd ~
var1="12"
[ "$var1" -eq 12 ]
echo $?
[ "$var1" -eq 120 ]
echo $?
```

-eq Два числа равны
-ne Два числа не равны
-gt Первое число больше второго числа
-lt Первое число меньше второго числа
-le Первое число меньше или равно второму числу
-ge Первое число больше или равно второму числу

@ Что такое shell?

Управляющие конструкции

- коды завершения
- операторы **if then else**
- циклы **while for until**
- принятие решений в сценариях
- создание меню

Коды завершения

`exit n`

`echo $?`

0 – успешное завершение

1 – неудачное завершение

@ Что такое shell?

Управляющие конструкции

- операторы **if then else**

`if условие1`

`then`

`команды1`

`elif условие2`

`then`

`команды2`

`else`

`команды3`

`fi`

`cd ~`

`if ["10" -lt "12"]; then`

`echo "abcd"; fi`

`if ["$name" = "vasya"];`
`then echo "well"; fi`

@ Что такое shell?

Управляющие конструкции

- оператор case

case *значение* in

шаблон1)

команды1

::

*)

команды2

::

esac

@ Что такое shell?

Управляющие конструкции

- цикл for

for *имя_переменной* in list

do

команды1

done

cd ~

for loop in 1 2 3 4 5

do

echo \$loop

done

@ Что такое shell?

Управляющие конструкции

- цикл until

until *условие*

*команды*1

done

```
cd ~
```

```
if [ "10" -lt "12" ]; then
```

```
    echo "abcd"; fi
```

@ Что такое shell?

Управляющие конструкции

- цикл while

while *команда*

do

*команды*1

done

```
cd ~
```

```
if [ "10" -lt "12" ]; then
```

```
    echo "abcd"; fi
```



Полезные программы

Программы

find
xargs
cron – crontab – at
ps
grep
awk
sed
tr



Задания

у, п

пользователь задает маску и директорию, программа ищет в директории файлы по маске и считает их количество, создать меню вывода, в нем указать имя пользователя который совершил запрос, сам запрос, результат подсчета

.logout

trap "\$HOME/.logout" 0

вывести имя пользователя, время

создать файл лога, если нет

в него записать количество файлов

начинающихся с «.» в директории \$HOME, имя пользователя и дату выхода.

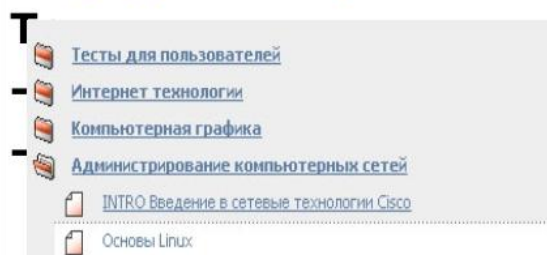


Проверь свои знания

Центр компьютерного обучения и сертификации специалистов

<http://www.specialist.ru>

<http://tests.specialist.ru> – тесты на сертификат



linux

Linux Professional Institute - международная общественная организация, усилия которой направлены на создание всеобъемлющей сертификации (Linux Professional Institute Certification - LPIC) для специалистов по Linux. Сертификация не привязана к какому-либо конкретному дистрибутиву.



Справочная информация

Встроенные команды shell

:

.

break

cd

continue

echo

eval

exec

exit

export

pwd

read

readonly

return

set

shift

test

times

trap

type

ulimit

umask

unset

wait

@ Литература

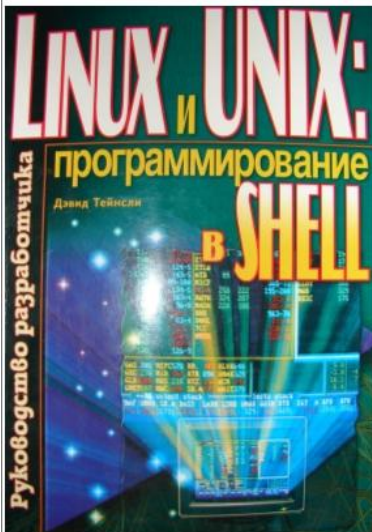
- Дэвид Тейнсли

Linux и UNIX: программирование в shell.

Руководство разработчика: Пер с англ. - К.:

Издательская группа BHV, 2001. — 464 с.

ISBN 966-552-085-7 ISBN 5-7315-0114-9



Данная книга является практическим руководством по программированию интерпретатора Bourne shell — стандартного командного интерпретатора в UNIX, полностью совместимого с интерпретатором BASH shell в Linux. Книга предназначена для начинающих и опытных программистов и содержит множество полезных примеров, советов и подсказок. С ее помощью читатель сможет быстро научиться создавать shell-сценарии для реальных задач и ситуаций, возникающих в большинстве систем UNIX и Linux.

Семинары
ИУ4 - Сети



skycluster@gmail.com